Rendszermérnöki ismeretek és módszerek

Ötlettől a működő rendszerig a V-modell mentén

Abonyi János



Pannon Egyetem
University of Pannonia

Veszprém, 2026



Pannon Egyetem
University of Pannonia

Abonyi János: Rendszermérnöki ismeretek és módszerek
Ötlettől a működő rendszerig a V-modell mentén

Pannon Egyetem, Veszprém, 2026

Változat: 0.3.0, 2026. november

Online: complexsystemsintelligence.github.io/rendszermernokseg

Forrás: github.com/ComplexSystemsIntelligence/rendszermernokseg

DOI: 10.5281/zenodo.24414135

© 2026 Abonyi János. A mű a Creative Commons Nevezd meg! – Ne add el! – Így add tovább! 4.0 Nemzetközi (CC BY-NC-SA 4.0) licenc feltételeivel szabadon másolható, terjeszthető és átdolgozható.

A Pannon Egyetem neve, címere és logója az egyetem arculati eleme; a licenc ezekre nem terjed ki.

Szedés: LuaLaTeX, IBM Plex és Raleway betűkkel; ábrák: TikZ.

Tartalom

Előszó 1

I. rész – Alapok 5

1. Rendszerek és rendszerszemlélet 7

2. A rendszerfejlesztés életciklusa és a V-modell 23

3. Modellek és modellalapú rendszermérnökség 37

II. rész – A V bal ága: a problémától a tervig 51

4. A probléma meghatározása és a működési koncepció 53

5. Érdekelt felek elemzése 65

6. Követelmények meghatározása 83

7. A vevő hangja a tervezésben: QFD és a minőségház 97

8. Funkció, forma, koncepció: a rendszer architektúrája 107

9. Koncepciógenerálás és kreatív problémamegoldás 125

10. Koncepció-kiválasztás és döntéstámogatás 139

11. Emberközpontú tervezés és emberi rendszerintegráció (HSI) 157

12. Funkciómegosztás és automatizálás 167

13. Emberi teljesítmény és emberi hiba 177

14. Kockázatkezelés 185

15. Hibamód- és hatáselemzés (FMEA) 195

16. Megbízhatóság és rendelkezésre állás 203

17. Biztonságkritikus rendszerek 213

18. Robusztus tervezés és tűrések 221

III. rész – A V csúcsa: megvalósítás és folyamattervezés 229

19. Folyamatelemzés és hatékonyság 231

20. Folyamatmodellezés BPMN-nel 243

21. Ütemezés, készlet és sorbanállás 253

IV. rész – A V jobb ága: integrációtól az üzemeltetésig 263

22. Integráció és interfészmenedzsment 265

23. Verifikáció 271

24. Validáció 281

25. Átadás, üzemeltetés és folyamatos fejlesztés 289

V. rész – Átívelő technikai menedzsment 301

26. Technikai tervezés és irányítás 303

Irodalom 313

Függelékek 327

A. A futó esettanulmányok 327

B. Magyar–angol szószedet 331

Előszó

Ez a könyv arról szól, hogyan lesz egy homályos igényből működő, biztonságos, használható és gazdaságos rendszer. A rendszer lehet mechatronikai berendezés, gyártócella, informatikai alkalmazás vagy egy szervezet működési folyamata. A rendszermérnökség az a gondolkodásmód és módszertan, amely ezt az utat végigköveti. Az érdekelt felek igényeinek feltárásától a követelményeken, a koncepciókon és a terveken át a megvalósításig, az ellenőrzésig, az üzemeltetésig, sőt a kivonásig tart.

Kinek szól a könyv?

Elsősorban mérnöki alapszakos hallgatóknak: műszaki menedzsereknek, mechatronikai mérnököknek és mindazoknak, akik informatikai fejlesztésben vesznek részt. A könyv önálló tanulásra készült. A fogalmakat lépésről lépésre vezeti be, minden módszert kidolgozott példán mutat be, és a fejezetek végén önellenőrző kérdések és projektfeladatok segítik a gyakorlást. A gyakorló mérnökök kézikönyvként is használhatják, egy-egy módszer gyors felidézésére.

A könyv felépítése: a V-modell mentén

A könyv a rendszerfejlesztés V-modelljét követi (2.4. ábra). Az I. rész az alapokat teszi le: a rendszerszemléletet, az életciklust és a V-modellt, valamint a modellek szerepét. A II. rész a V bal ágán halad lefelé, a problémától és az érdekelt felektől a követelményeken, a koncepciókon és a döntéseken át a részletes tervezésig. Ide tartoznak az emberközpontú tervezés, a kockázat-, a megbízhatósági és a biztonsági elemzések is. A III. rész a V csúcsán a megvalósítással és a gyártó- és szolgáltatófolyamatok tervezésével foglalkozik. A IV. rész a V jobb ágán halad felfelé az integrációtól a verifikáción és a validáción át az üzemeltetésig és a folyamatos fejlesztésig. Az V. rész a teljes V-n átívelő technikai menedzsmentet foglalja össze.

Minden fejezet egy kis V-térképpel kezdődik, amely megmutatja, a fejlesztés melyik lépésénél tartunk.

Mit fogsz tudni a könyv végére?

A könyv tanulási eredményei a Magyar Képesítési Keretrendszer (MKKR) négy dimenziójában fogalmazódnak meg: tudás, képességek, attitűd, valamint autonómia és felelősség (Magyarország Kormánya 2012). A szint az alapképzés végzettségi szintjének felel meg, amely az Európai Képesítési Keretrendszer 6. szintjével egyenértékű (Az Európai Unió

Tanácsa 2017). A 1. táblázat a könyv egészére vonatkozó eredményeket foglalja össze; az egyes fejezetek elején ezek részletes, a fejezethez kötött változata áll.

1. táblázat. A könyv egészére vonatkozó tanulási eredmények az MKKR dimenziói szerint.

Dimenzió	A könyv feldolgozása után…
Tudás	ismered a rendszer, az életciklus és a V-modell fogalmát, a rendszermérnöki folyamatokat az ISO/IEC/IEEE 15288 szerint, valamint a követelmények, az architektúra, a döntéstámogatás, a kockázat-, megbízhatósági és biztonsági elemzés, a verifikáció és a validáció alapvető módszereit és azok korlátait.
Képességek	egy valós fejlesztési feladatban azonosítod az érdekelt feleket és a problémát, mérhető követelményeket és működési koncepciót írsz, koncepcióváltozatokat dolgozol ki és választasz közülük indokolt módszerrel, elvégzed a legfontosabb kockázati és megbízhatósági elemzéseket, és verifikációs és validációs tervet készítesz. A módszereket táblázatkezelőben és Pythonban is alkalmazni tudod.
Attitűd	a rendszert egészében és a teljes életciklusában szemléled; nyitott vagy más szakterületek és az érdekelt felek nézőpontjára; a döntéseidet adatokkal és forrásokkal támasztod alá; és kritikusan kezeled a modellek, a szoftveres eszközök és a nagy nyelvi modellek eredményeit.
Autonómia és felelősség	egy kisebb fejlesztési projekt rendszermérnöki feladatait önállóan, egy csapat tagjaként végzed; felelősséget vállalsz a munkatermékeid minőségéért és nyomonkövethetőségéért; és felismered, mikor kell egy döntést a hatáskörrel rendelkező érdekelt felekre bízni, különösen a biztonságot és az embert érintő kérdésekben.

Hogyan tanulj a könyvből?

A könyv önálló tanulásra készült. Egy fejezet feldolgozásához a következő sorrendet javasoljuk:

1. Olvasd el a fejezet elején a **tanulási eredményeket** és a V-térképet: ezekből látod, mit kell a fejezet végére tudnod, és hol tartunk a fejlesztésben.
2. Haladj végig a szövegen, és a **futó esetek** dobozainál állj meg: ezek mutatják meg, hogyan működik a módszer egy konkrét rendszeren.
3. Számold végig a **kidolgozott példákat** papíron vagy a Gyakorlati útmutató sablonjaiban; a számok újraszámolása többet tanít, mint az olvasásuk.
4. Válaszolj az **önellenőrző kérdésekre**, mielőtt a megoldást megnéznéd. Ha egy kérdés nem megy, térj vissza a hozzá tartozó szakaszhoz.
5. Végezd el a **projektlépést** a saját féléves projekteden. A projektlépések egymásra épülnek: a félév végére egy kisebb rendszer teljes rendszermérnöki dokumentációja áll össze.

Válogatott útvonalak

A fejezetek önállóan is olvashatók: mindegyik elején ott vannak a tanulási eredmények, és ahol egy korábbi fejezetre épül, ezt hivatkozás jelzi. Néhány javasolt útvonal, ha csak egy-egy témakörben haladnál:

2. táblázat. Javasolt útvonalak a könyvben.

Útvonal	Ajánlott fejezetek
A rendszermérnökség alapjai	1., 2., 4., 5., 6., 23., 24.
Minőségfejlesztés	5., 7., 15., 16., 18., 25.
Döntéstámogatás	5., 9., 10., 14.
Ember a rendszerben	5., 11., 12., 13., 24.
Folyamatok és működés	19., 20., 21., 25.
Informatikai fejlesztés	2., 3., 4., 5., 6., 8., 11., 22., 23., 24.

A futó esettanulmányok

A könyv ugyanazon a három, kitalált, de életszerű rendszeren mutatja be a módszereket:

- **KoboCella:** kollaboratív robotos összeszerelő cella egy beszállító üzemben;
- **VárosiBringa:** közösségi elektromos kerékpárrendszer egy dombos egyetemi városban;
- **ForgácsTech:** egy CNC-alkatrészgyártó üzem digitális átállása.

Az esetek részletes leírása az A. függelékben található.

Kiemelődobozok

A szövegben visszatérő dobozok segítik a tájékozódást:

- **Tanulási eredmények:** mit fogsz tudni a fejezet végére;
- **Fogalom:** pontos meghatározás;
- **Futó eset:** a módszer alkalmazása az esettanulmányokon (kék: KoboCella, zöld: VárosiBringa, lila: ForgácsTech);
- **Kidolgozott példa:** lépésről lépésre végigszámolt feladat;
- **Gyakori buktató:** tipikus hibák;
- **Kitekintés:** a terület kutatási frontja, újabb irányok;
- **Összefoglalás:** a fejezet lényege;
- **Önellenőrzés:** kérdések megoldással;
- **Projektlépés:** a féléves projekt soron következő feladata;
- **Továbbolvasás:** a legfontosabb források.

Források és a Gyakorlati útmutató

A könyv elméleti gerincét a rendszermérnökség nemzetközi szabványai adják: az életciklus-folyamatokat leíró ISO/IEC/IEEE 15288:2023 (ISO/IEC/IEEE 2023), az életciklus-kezelés ISO/IEC/IEEE 24748-1 szabványa (ISO/IEC/IEEE 2024) és a követelménymérnökség ISO/IEC/IEEE 29148 szabványa (ISO/IEC/IEEE 2018). A gyakorlati tudást az INCOSE rendszermérnöki kézikönyve (INCOSE 2023b), a szabadon elérhető rendszermérnöki tudástár (SEBoK) (SEBoK Editorial Board 2025) és a klasszikus tankönyvek (Blanchard és Fabrycky 2011; Kossiakoff és mtsai. 2011) összegzik. Az egyes módszereket mindig az eredeti szakirodalmuk és a rájuk vonatkozó szabványok alapján mutatom be, a minőségügy, a döntéstámogatás, a biztonságtechnika és az emberközpontú tervezés újabb eredményeivel kiegészítve. A példák ipari, szolgáltatási és informatikai rendszerekből származnak. Kiegészítő olvasmányként két szabadon hozzáférhető, sokféle mintát és ellenőrzőlistát tartalmazó gyűjteményt is ajánlunk: a NASA rendszermérnöki kézikönyvét (NASA 2016) és a tervezőmérnököknek szóló NASA-eszköztárat (Goldberg és mtsai. 1994); a fejezetek továbbolvasási listái megadják, melyik részük kapcsolódik az adott témához.

A könyv párja a **Rendszermérnöki eszköztár – Gyakorlati útmutató**. Ez ugyanebben a sorrendben, egységes eszközlapokon mutatja be a módszereket, letölthető sablonokkal, mintapromptokkal és a féléves projekt teljes útmutatójával. A két kötet kölcsönösen hivatkozik egymásra.

A könyv anyaga a Pannon Egyetem Rendszermérnöki ismeretek tantárgyának több éves oktatási tapasztalatára épül.

Szabad felhasználás

A könyv szövege és ábrái a Creative Commons Nevezd meg! – Ne add el! – Így add tovább! 4.0 (CC BY-NC-SA 4.0) licenc alatt használhatók. A forrásállományok és a mellékelt programkódok MIT-licencűek. A könyv minden ábrája vektorgrafika, és a forráskódjuk megtalálható a tárhelyen. A Pannon Egyetem neve, címere és logója az egyetem arculati eleme; a licenc ezekre nem terjed ki.

A könyv kiadásait a Zenodo archiválja. Hivatkozáskor ezt az adatot használd:

> Abonyi János (2026): *Rendszermérnöki ismeretek és módszerek – Ötlettől a működő rendszerig a V-modell mentén* (kézikönyv és gyakorlati útmutató). Pannon Egyetem, Veszprém. DOI: 10.5281/zenodo.24414135

Veszprém, 2026. november

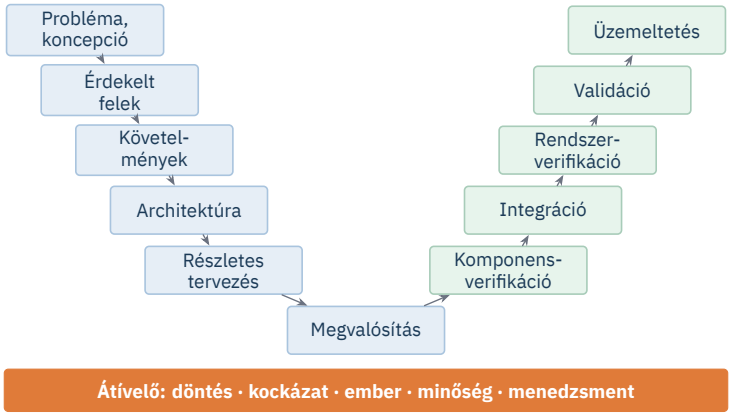
Abonyi János

I. rész – Alapok



Pannon Egyetem
University of Pannonia

1. Rendszerek és rendszerszemlélet



Tanulási eredmények

A fejezet feldolgozása után képes leszel:

- meghatározni, mi a rendszer, és felismerni egy rendszer elemeit, attribútumait, kapcsolatait, határát és környezetét;
- megmagyarázni, mit jelent az emergens tulajdonság, és miért nem vezethető le egy rendszer viselkedése pusztán az elemeiből;
- egy rendszert hierarchikus szintekre bontani, és elhelyezni egy befoglaló rendszerben;
- a műszaki–gazdasági rendszereket alrendszereikkel és irányítási szerkezetükkel leírni;
- megkülönböztetni az egyszerű, a bonyolult és a komplex rendszereket, valamint a rendszerelemzés, a rendszertervezés és a rendszerirányítás feladatát;
- ok-okozati hurokdiagrammal ábrázolni egy rendszer visszacsatolásait;
- elhatárolni egymástól a rendszerszemléletet, a rendszermérnökséget és a mérnöki rendszereket, és leírni a rendszermérnök szerepét és kompetenciáit.

Ez a fejezet a könyv szókincsét alapozza meg. A V-modell minden lépése (2.4. ábra) rendszerekről szól: a bal ágon egy még nem létező rendszert írunk le egyre részletesebben, a csúcson megépítjük az elemeit, a jobb ágon pedig ellenőrizzük, hogy az összerakott egész azt teszi-e, amit vártunk tőle. Ehhez közös nyelv kell. Mit nevezünk rendszernek? Hol van a határa? Miért viselkedik másként az egész, mint a részei? És mit csinál ebben a rendszermérnök?

A fejezet öt kérdésre épül:

- Mi a rendszer, és milyen általános tulajdonságai vannak?
- Hogyan épül fel egy rendszer, és hogyan illeszkedik nagyobb rendszerekbe?
- Miben különbözik egy gyártócella egy városi közlekedési szolgáltatástól, és mit jelent ez a tervezésre nézve?
- Miért kell a visszacsatolásokban gondolkodni?
- Mi a rendszermérnökség, és milyen szakember a jó rendszermérnök?

Bemelegítő: a kilenc pont feladványa

Rajzolj fel kilenc pontot három sorban, három oszlopban. Kösd össze mind a kilencet négy egyenes vonallal úgy, hogy közben nem emeled fel a tollat. A legtöbben először sikertelenül próbálkoznak, mert hallgatólagosan feltételezik, hogy a vonalak nem léphetnek ki a pontok alkotta négyzetből. Ezt a feladat nem mondja. Amint elvetjük a képzelt határt, a megoldás kézenfekvő. Ha már ez megvan, próbáld meg három vonallal is!
A tanulság a rendszermérnökség egyik alapgondolata. A probléma határát mindig tudatosan kell kijelölni. Ha ezt nem tesszük meg, a fejünkben lévő, ki nem mondott határ korlátozza a megoldások körét. A rendszerhatár meghúzásával a 3. fejezet és a 4. fejezet részletesen foglalkozik.

1.1. Mi a rendszer?

A „rendszer” szót mindenki használja, és mindenki mást ért rajta. A mérnöki gyakorlatban ezért pontos meghatározásokra támaszkodunk. A rendszer- és szoftvermérnöki életciklus-folyamatok nemzetközi szabványa szerint a rendszer egymással kölcsönhatásban álló elemek olyan együttese, amely egy vagy több kitűzött cél elérésére szerveződik. A szabvány azt is kiemeli, hogy az elemek sokfélék lehetnek: hardver, szoftver, adat, ember, folyamat, eljárás, létesítmény és anyag (ISO/IEC/IEEE 2023).

A meghatározás fontos részlete, hogy az *emberek* és az *eljárások* is a rendszer elemei. Egy kollaboratív robotcella nem csak a robotból, a csavarozóból és a vezérlésből áll. Az operátor, a munkautasítás és a karbantartási eljárás ugyanúgy része, mint a szervomotor. Ha az embert a rendszer környezetének tekintjük, a tervezés éppen azt a kölcsönhatást hagyja ki, amelyen a teljesítmény sokszor múlik. Ezzel a gondolattal a könyv emberközpontú fejezetei (11–13. fejezet) részletesen foglalkoznak.

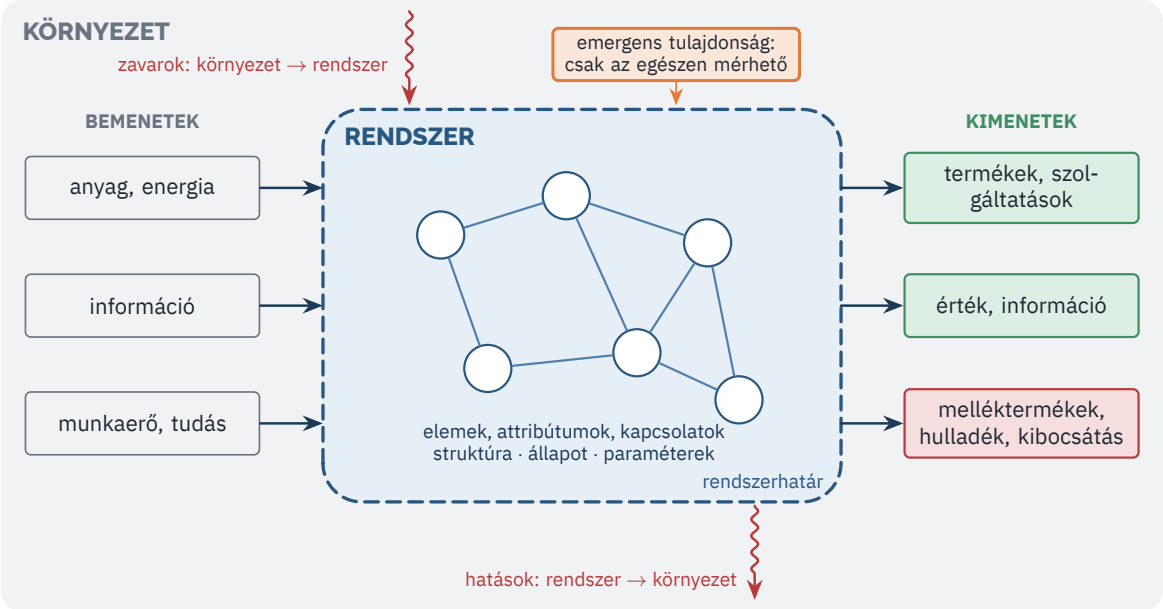
A Nemzetközi Rendszermérnöki Szövetség (INCOSE) 2019-ben két szinten fogalmazta meg a fogalmat (INCOSE 2019). Az *általános* meghatározás szerint a rendszer részek vagy elemek olyan elrendeződése, amely együtt olyan viselkedést vagy jelentést mutat, amilyet az egyes összetevők külön nem. A *mérnöki rendszer* pedig olyan rendszer, amelyet úgy terveztek vagy alakítottak ki, hogy egy várható működési környezettel kölcsönhatásban egy vagy több kitűzött célt érjen el, miközben megfelel a rá vonatkozó korlátoknak. A mérnöki rendszer nem feltétlenül technikai: egy szolgáltatás vagy egy szervezet is lehet az, ha tudatosan, rendszermérnöki módszerekkel alakítják ki.

1.1.1. Elemek, attribútumok, kapcsolatok

Blanchard és Fabrycky klasszikus tankönyve a rendszert három építőelemmel írja le (Blanchard és Fabrycky 2011):

- **Komponensek (elemek):** a rendszer működő részei. Lehetnek fizikai tárgyak (motor, szenzor), szoftvermodulok, emberek vagy eljárások.
- **Attribútumok:** a komponensek és a rendszer jellemzői, amelyek leírják és mérhetővé teszik őket (tömeg, ciklusidő, megbízhatóság, ár, kezelhetőség).
- **Kapcsolatok:** a komponensek közötti kölcsönhatások. Ezeken keresztül áramlik az anyag, az energia és az információ.

A rendszert a **határa** választja el a **környezetétől**. A környezet az, amire a rendszer hat, és ami a rendszerre hat, de amit a fejlesztés során nem tervezünk meg. A határon lépnek be a **bemenetek** (anyag, energia, információ, munkaerő), és lépnek ki a **kimenetek** (termékek, szolgáltatások, érték, de melléktermékek és kibocsátások is). A környezet felől **zavarok** érik a rendszert, a rendszer pedig maga is hat a környezetére (1.1. ábra). A rendszer belső leírásához a Lakatos-jegyzet három fogalmat használ: a **struktúra** az elemek és kapcsolataik elrendeződése, az **állapot** a rendszer pillanatnyi helyzetét leíró változók összessége, a **paraméterek** pedig a működést meghatározó, (közel) állandó jellemzők (Lakatos, é. n.a).



1.1. ábra. A rendszer és környezete. A rendszerhatárt mi jelöljük ki; a határon át lépnek be a bemenetek és lépnek ki a kimenetek, a környezet felől zavarok érik a rendszert. Az emergens tulajdonságok csak az egész rendszeren figyelhetők meg.

ForgácsTech: mi a rendszer és mi a környezet?

A ForgácsTech digitális átállásánál az első vita arról szólt, mi tartozik a fejlesztendő rendszerhez. A tulajdonos szerint „a MES-szoftver”. A termelésvezető szerint „az egész műhely”. A beszélgetés végére a csapat a következő határt húzta meg:

- **A rendszer része:** a rendelés felvételétől a kész munkadarab kiszállításáig tartó információs és anyagáramlás. Ebbe beletartoznak a CNC-gépek adatkapcsolatai, a gyártáselőkészítés, a gépkezelők munkautasításai és a MES-szoftver.
- **A környezet része:** a vevők megrendelési rendszerei, a könyvelőiroda, a nyersanyagbeszállítók és a szerszámgépek gyártói. Ezekkel a rendszernek kapcsolatban kell állnia, de nem tervezzük újra őket.

A határ meghúzása döntés, nem felfedezés. Ugyanazt a műhelyt más fejlesztési célhoz másként határolnánk le. Egy energiahatékonysági projektben például a sűrítettlevegő-hálózat lenne a rendszer, a MES pedig a környezet.

1.1.2. Emergens tulajdonságok

A rendszerszemlélet talán legfontosabb gondolata, hogy az egész több, mint a részek összege. Maier és Rechtin rendszerarchitektúráról szóló könyvének egyik alapgondolata, hogy

a rendszer hozzáadott értéke, amely túlmutat a részek önálló hozzájárulásán, elsősorban a részek közötti kapcsolatokból születik, vagyis abból, ahogyan a részeket összekapcsoljuk (Maier és Rechtin 2009). Az olyan tulajdonságot, amely csak az egész rendszeren jelenik meg, és egyik elemnek sincs meg külön, **emergens tulajdonságnak** nevezzük.

Néhány példa a futó esetekből:

- A KoboCella **ciklusideje** nem a robot sebességéből és nem az operátor ügyességéből adódik, hanem abból, hogyan osztják meg és hangolják össze a munkát. Egy gyorsabb robot semmit sem javít, ha az operátorra kell várnia.
- A VárosiBringa **elérhetősége** (van-e kerékpár a közelben, amikor kell) nem egy kerékpár vagy egy dokkoló tulajdonsága. A flotta, az állomáshálózat, a felhasználói viselkedés és az áthelyező szolgálat együttes eredménye.
- A **biztonság** is emergens tulajdonság. Egy alkatrész lehet megbízható, a rendszer mégis veszélyes lehet, ha az elemek kölcsönhatása váratlan állapotba viszi. Ezzel a gondolattal a 17. fejezet foglalkozik.

Az emergencia két irányban is figyelmeztet. A kívánt tulajdonságokat (teljesítmény, használhatóság, biztonság) nem lehet egy-egy elemre „rátervezni”: az elemek kapcsolatait kell tudatosan alakítani. A nem kívánt tulajdonságok (rezgés, holtpont, túlterhelt operátor) pedig gyakran csak az integráció során bukkannak fel. Ezért kell a V-modell jobb ágán lépésről lépésre integrálni és verifikálni (22. fejezet).

1.1.3. A rendszer általános tulajdonságai

A rendszerelméleti szakirodalom a rendszerek néhány általános tulajdonságát emeli ki (Bertalanffy 1968; Lakatos, é. n.a). Ezek egyben gondolkodási szabályok is a tervező számára.

- **A rendszer célorientált egység.** Van egy rendszeralkotó tényező, amelynek megvalósulása érdekében a rendszer működik. Mérnöki rendszernél ez a kitűzött cél. Ha a cél nem világos, a rendszer sem lesz az.
- **A rendszer egyszerre autonóm és heteronóm.** Vannak saját belső törvényszerűségei (autonóm), de működését a környezet is befolyásolja, és ő maga is hat a környezetére (heteronóm).
- **A rendszer tagolt egész.** Alrendszerekre és elemekre bontható, ezekből összeépíthető, és az elemek külön is vizsgálhatók. Az elemek tulajdonságai megjelennek a rendszerben, de az egésznek vannak olyan tulajdonságai is, amelyek csak rá jellemzők.
- **A rendszer kölcsönhatásban álló elemek összessége.** Nem bontható két részre úgy, hogy a két rész között ne maradjon legalább egy értelmezett kapcsolat. Ha egy „rendszer” két, egymással semmilyen kapcsolatban nem álló részből áll, akkor valójában két rendszerről van szó.
- **A rendszer hierarchikus jellegű.** Maga is alrendszerekből áll, és ő maga is egy nagyobb rendszer része. Ebből adódik a *hierarchiaparadoxon*: egy rendszert akkor tudunk leírni, ha egy magasabb szintű rendszer elemeként írjuk le, ehhez azonban önmagában, alacsonyabb szintű elemek egészeként is le kell tudnunk írni. A gyakorlatban a paradoxont iterációval oldjuk fel: felülről lefelé és alulról felfelé is többször végigmegyünk a hierarchián. A V-modell rekurzív alkalmazása (2.4. szakasz) éppen ezt teszi.

A Lakatos-jegyzet a rendszereket megjelenési formájuk, keletkezésük, a környezethez való viszonyuk és további szempontok szerint is osztályozza (1.1. táblázat). A táblázat jól mutatja, hogy egy rendszer leírásakor sok döntést hozunk. Egy gyártócellát a ciklusidő számításakor determinisztikusnak és statikusnak tekinthetünk, a sorbanállás és a meghibásodások elemzésekor viszont sztochasztikus, dinamikus rendszerként kell kezelnünk.

1.1. táblázat. A rendszerek morfológiai osztályozása a Lakatos-jegyzet nyomán, a futó esetek példáival.

Szempont	Egyik végpont (példa)	Másik végpont (példa)
Megjelenési forma	konkrét (a KoboCella hardvere)	absztrakt (a cella vezérlőprogramja)
Keletkezés	természetes (egy folyó vízgyűjtője)	mesterséges (egy szennyvíztisztító)
Kapcsolat a környezettel	zárt (közelítőleg egy lezárt tartály)	nyitott (minden szolgáltatás)
Időbeli viselkedés	statikus (egy rácsos tartó)	dinamikus (egy szabályozott hajtás)
Meghatározottság	determinisztikus (egy fogaskerékhajtás)	sztochasztikus (a kölcsönzési igények)
Struktúra	merev (egy célgép)	rugalmas (egy átállítható robotcella)
Elemek és kapcsolatok száma	kevés, egyszerű (egy csavarozó egység)	nagyon sok, komplex (egy városi közlekedési hálózat)

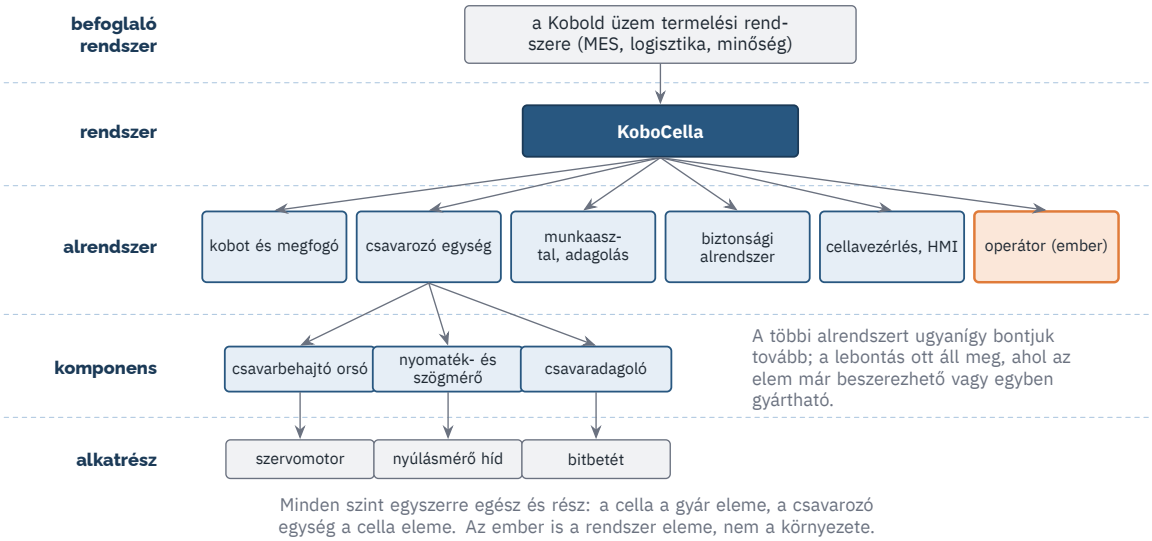
1.2. A rendszer felépítése: hierarchia és befoglaló rendszer

Kossiakoff és szerzőtársai rámutatnak, hogy a „rendszer” fogalma az aggregáció bármely szintjén értelmezhető. Egy telefonközpont a hozzá tartozó vonalakkal rendszer, de egyben a városi telefonhálózat alrendszere is. Minden rendszer egy magasabb szintű rendszer alrendszere, és minden alrendszer maga is tekinthető rendszernek (Kossiakoff és mtsai. 2011, 3.2. szakasz). A komplex rendszerek ezért hierarchikus felépítésűek. Néhány fő, egymással kölcsönhatásban álló elemből, alrendszerből állnak. Ezek egyszerűbb funkcionális egységekből épülnek fel, és így tovább egészen az olyan elemi részekig, mint egy fogaskerék vagy egy izzó.

A hierarchia szintjeinek megnevezése szakterületenként eltér. A könyvben a következő szinteket használjuk (1.2. ábra):

- **befoglaló rendszer:** az a nagyobb rendszer, amelynek a mi rendszerünk egy eleme;
- **rendszer:** a fejlesztés tárgya, amelynek a követelményeit mi határozzuk meg;
- **alrendszer:** a rendszer egy fő funkcionális egysége, gyakran külön csapat vagy beszállító felelőssége;
- **komponens:** önállóan specifikálható, beszerezhető vagy gyártható egység;
- **alkatrész:** a tovább már nem bontott elemi rész.

A gyakorlatban a hierarchiát termékszerkezetként (product breakdown structure, PBS) rögzítjük. A követelményeket a termékszerkezet szintjeihez rendeljük, és a munkalebontási struktúra (WBS) is erre épül, mert termékorientált (Project Management Institute 2019; U.S. Department of Defense 2022). A munkalebontással a 26. fejezet foglalkozik.



1.2. ábra. A KoboCella rendszerhierarchiája. A befoglaló rendszer a gyár termelési rendszere; a csavarozó egység példáján a lebontás egészen az alkatrészekig látható. Az operátor a rendszer eleme.

> **KoboCella: miért kell ismerni a befoglaló rendszert?**
>
> A KoboCella fejlesztőcsapata az első hetekben csak a cellával foglalkozott. A befoglaló rendszer, vagyis a gyár termelési rendszere csak az első tervfelülvizsgálaton került elő, amikor a logisztikai vezető megkérdezte: honnan és milyen gyakran érkeznek a motorfedelek a cellához? Kiderült, hogy a cellát ellátó anyagmozgatás jelenleg óránként egyszer, 120 darabos ládákban szállít. Ez egy 75 másodperces ciklusidejű cellánál mintegy két és fél óra anyagot jelent, amit a cella mellett tárolni kell. A tervezett, kis alapterületű elrendezés ezzel nem fért össze.
> A tanulság: a rendszer bemeneteit és kimeneteit a befoglaló rendszer határozza meg. A befoglaló rendszert nem tervezzük újra, de a kapcsolódási pontjait (3. fejezet) már az első lépésben ismerni kell.

1.2.1. Rendszerek rendszere

Vannak rendszerek, amelyek elemei maguk is önálló, működőképes rendszerek. Ezeket **rendszerek rendszerének** (system of systems, SoS) nevezzük. Maier klasszikus cikke szerint egy együttes akkor tekinthető rendszerek rendszerének, ha elemei működésükben és irányításukban is függetlenek: önállóan is ellátják a feladatukat, és saját tulajdonosuk, saját céljaik szerint működnek, miközben együttesen olyan képességet adnak, amelyet egyikük sem tud egyedül (Maier 1998).

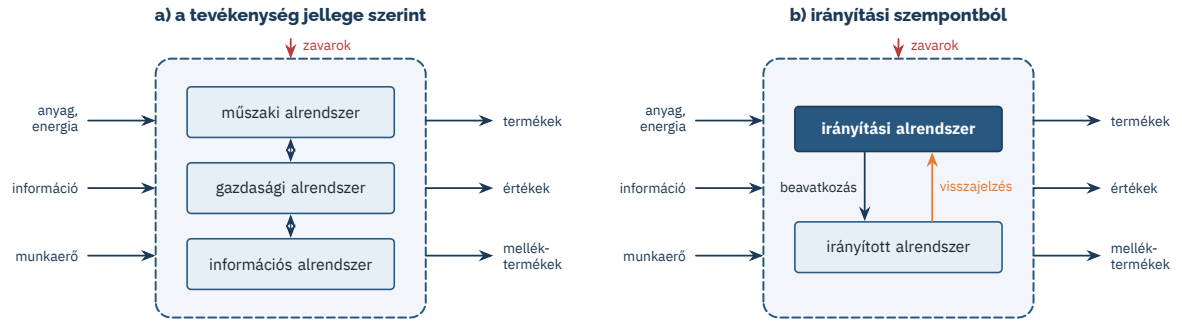
A VárosiBringa jó példa erre. A kerékpárrendszer önállóan is működik, de a valódi értéket akkor teremti, ha a helyi buszhálózattal, a városi kártyarendszerrel, az egyetem beléptető-rendszerével és a navigációs alkalmazásokkal együtt működik. Ezek a rendszerek más tulajdonosok kezében vannak, és nem a VárosiBringa kedvéért változnak. A tervezőnek ezért nem utasítania kell őket, hanem tárgyalnia velük, és szabványos, stabil interfészekre kell építenie. Az érdekeltfél-elemzés (5. fejezet) itt szó szerint rendszerszintű kérdés.

1.3. Műszaki–gazdasági rendszerek

A mérnöki munka tárgya legtöbbször nem pusztán műszaki rendszer. A Lakatos-jegyzet meghatározása szerint a műszaki–gazdasági rendszer bonyolult, nagyon sok elemből és kapcsolatból álló, heterogén összetételű ember–gép szervezet, amelynek célja termékek vagy szolgáltatások előállítása. Különböző természetű alrendszerekből áll, és műszaki alrendszerei dinamikus rendszerek (Lakatos, é. n.a, 1.2. szakasz).

A rendszertechnika az ilyen rendszereket kétféleképpen bontja fel (1.3. ábra):

- **A tevékenység jellege szerint** műszaki, gazdasági és információs alrendszerre. A műszaki alrendszerben anyag és energia előállítása, átvitele, tárolása, átalakítása és felhasználása folyik. A gazdasági alrendszer az értéket, az árut és a szolgáltatást állítja elő. Az információs alrendszer az információt állítja elő és használja fel.
- **Irányítási szempontból** irányítási és irányított alrendszerre. Az irányítási alrendszer információt dolgoz fel, és beavatkozik az irányított alrendszer működésébe. Az irányított alrendszer visszajelzései alapján pedig korrigálja a beavatkozást.



1.3. ábra. A műszaki–gazdasági rendszer kétféle felbontása a Lakatos-jegyzet nyomán: a) műszaki, gazdasági és információs alrendszerek; b) irányítási és irányított alrendszer.

ForgácsTech: hol szakad meg az irányítási kör?

A ForgácsTech műszaki alrendszere (a gépek, a szerszámok, az anyagmozgatás) jól működik, a gépek korszerűek. A gazdasági alrendszer is rendben van: a cég nyereséges, a vevők elégedettek a minőséggel. A gond az információs alrendszerben van. A termelésvezető hétfő reggel papíron osztja ki a heti munkát, de csak péntek délután tudja meg, mi készült el valójában. Az irányítási kör tehát egyhetes késleltetéssel zárul, és a beavatkozások mindig egy hetet késnek.
Ebből a nézőpontból a MES-bevezetés lényege nem egy szoftver telepítése, hanem az irányítási kör lerövidítése: a visszajelzés napi, sőt valós idejű legyen. Ez pontosabb célt ad a fejlesztésnek, mint a kezdeti „kell egy MES” megfogalmazás. A visszacsatolás késleltetésének hatásaival a 1.5. szakasz foglalkozik.

1.4. Egyszerű, bonyolult és komplex rendszerek

A KoboCella és a VárosiBringa egyaránt rendszer, de nagyon különböző tervezési helyzetet jelentenek. A cella határai jól kijelölhetők, a hasonló cellákról van tapasztalat, a lehetséges megoldások száma korlátos, és ha elkészül, kipróbálható: méréssel eldönthető, hogy teljesíti-e a 75 másodperces ciklusidőt. A városi kerékpárrendszer határa elmosódott.

Sok, egymásnak ellentmondó célú érdekelt fél van, a viselkedés a felhasználók döntésein múlik, és nincs egyértelműen „jó” megoldás, csak többé-kevésbé elfogadható.

Snowden és Boone döntéshozatali keretrendszere (a Cynefin-keret) a helyzeteket aszerint sorolja kategóriákba, hogy mennyire ismerhető meg bennük az ok-okozati kapcsolat (Snowden és Boone 2007). A keret fogalmait a rendszerekre alkalmazva hasznos megkülönböztetést kapunk (1.2. táblázat).

- **Egyszerű** helyzetben az ok és az okozat mindenki számára nyilvánvaló, a jó gyakorlat ismert, és alkalmazni kell.
- **Bonyolult** helyzetben az ok-okozati kapcsolat megismerhető, de szakértői elemzés kell hozzá. Több jó megoldás is lehetséges, és ezek elemzéssel összevethetők.
- **Komplex** helyzetben az ok-okozati kapcsolat csak utólag ismerhető fel. A jó megoldást nem lehet előre kielemezni: biztonságosan elbukható kísérletekkel kell kitapogatni, és a felbukkanó mintázatokra kell építeni.
- **Kaotikus** helyzetben nincs felismerhető ok-okozati kapcsolat. Először cselekedni kell, hogy a helyzet stabilizálódjon. Ez a válságkezelés világa.

1.2. táblázat. A bonyolult és a komplex rendszerek tervezési helyzetének összevetése. Saját összeállítás a Cynefin-keret fogalmaira építve (Snowden és Boone 2007).

Jellemző	Bonyolult műszaki rendszer (pl. KoboCella)	Komplex sociotechnikai rendszer (pl. VárosiBringa)
Határ	jól kijelölhető, izolálható	elmosódott, vitatható
Tapasztalat	hasonló rendszerekről van	minden helyzet némileg egyedi
Megoldások köre	korlátos, elemzéssel bejárható	nagyon nagy, előre nem bejárható
Érdekelt felek	kevés, jól azonosítható	sok, egymásnak ellentmondó célokkal
Viselkedés	modellekkel jól előrejelezhető	nehezen előrejelezhető, emergens mintázatok
Ellenőrizhetőség	kipróbálható, mérhető	csak részben tesztelhető, a hatás késve jelentkezik
Jó tervezési stratégia	elemzés, optimalizálás, alapos verifikáció	kis lépések, kísérletek, gyors visszacsatolás, rugalmasság

A különbségtétel a módszerválasztást is meghatározza. Bonyolult rendszereknél a V-modell klasszikus, előre tervező szemlélete jól működik. Komplex rendszereknél is a V-modell mentén haladunk, de rövidebb ciklusokban, inkrementálisan, mérhető kísérletekkel (2.2. szakasz). A VárosiBringa ezért nem 30 állomással indul egyszerre. Először nyolc állomással próbaüzem fut, és a kölcsönzési adatok alapján bővül a hálózat.

> **Komplex probléma bonyolultként kezelve**
>
> Gyakori hiba, hogy egy komplex helyzetet bonyolultnak tekintünk, és azt hisszük, hogy elég alapos elemzéssel előre megtalálható a jó megoldás. Ilyenkor hónapokig készül a „tökéletes” terv, amely az első valós használat után összeomlik, mert a felhasználók másként viselkednek, mint feltételeztük. Az ellenkező hiba is előfordul: egy jól elemezhető, bonyolult műszaki feladatot „próbálgatással” oldanak meg, és drága prototípusok sorát építik meg ott, ahol egy számítás elég lett volna.

1.4.1. Három alapfeladat: elemzés, tervezés, irányítás

A bemenet, a rendszer és a kimenet hármasából három alapfeladat adódik, aszerint, hogy melyik ismeretlen (1.4. ábra) (Lakatos, é. n.a, 3.1. szakasz):

1. **Rendszerelemzés (direkt feladat):** adott a bemenet és a rendszer, a kimenetet keressük. Mennyit termel a meglévő cella, ha ennyi rendelés érkezik?
2. **Rendszertervezés, rendszerszervezés (első inverz feladat):** adott a bemenet és a kívánt kimenet, a rendszert keressük. Milyen cella képes évi 140 ezer motort összeszerelni?
3. **Rendszerirányítás (második inverz feladat):** adott a rendszer és a kívánt kimenet, a bemenetet keressük. Hogyan ütemezzük a rendeléseket, hogy a határidők teljesüljenek?



1.4. ábra. A rendszerrel kapcsolatos három alapfeladat. Narancs színnel a keresett elem látható.

A rendszermérnökség középpontjában az első inverz feladat, a tervezés áll. Az inverz feladatok általában nem egyértelműek: ugyanazt a kimenetet sokféle rendszer előállíthatja. A Lakatos-jegyzet szerint a tervezés a célok és a lehetőségek közötti keresési folyamat, amely funkcionális, teljesítmény- és esztétikai követelmények teljesítésére irányul. A gondos tervező a lehetséges változatokat számba veszi, és közülük választ (Lakatos, é. n.a, 3.1. szakasz). A változatok előállításával a 9. fejezet, a közöttük való választással a 10. fejezet foglalkozik. A tervezés közben azonban folyamatosan direkt feladatokat is megoldunk: minden változatot elemzünk, hogy lássuk, mit tudna.

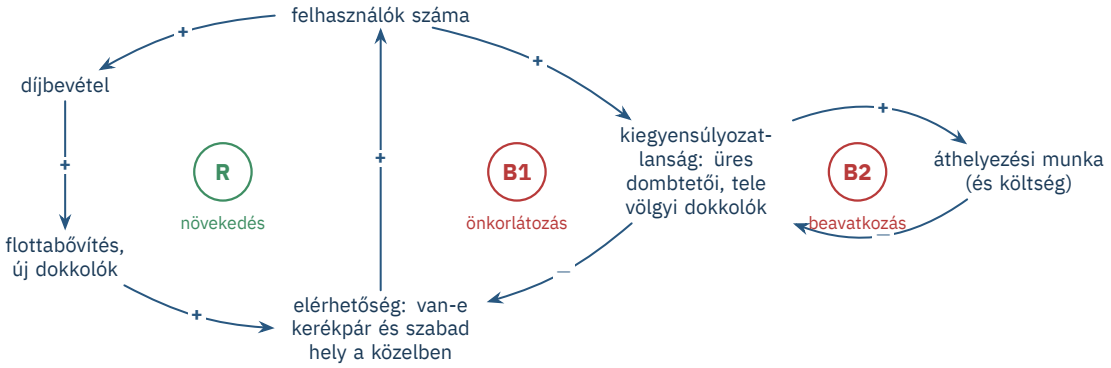
1.5. Gondolkodás visszacsatolásokban

A rendszerszemlélet nem csak a részek és az egész viszonyáról szól, hanem az időbeli viselkedésről is. Bertalanffy általános rendszerelmélete óta tudjuk, hogy a nyitott rendszerek viselkedését a környezettel folytatott csere és a belső visszacsatolások alakítják (Bertalanffy 1968). Donella Meadows rendszergondolkodási bevezetője szerint a rendszer viselkedésének kulcsa a szerkezete: a készletek, az áramlások és a visszacsatolási hurkok. Ugyanaz a szerkezet hasonló viselkedést eredményez, akárkik is a szereplők (Meadows 2008).

Két alapvető visszacsatolási típust különböztetünk meg:

- **Megerősítő (pozitív) visszacsatolás:** a változás önmagát erősíti. Több felhasználó több bevételt hoz, abból több kerékpár lesz, ami még több felhasználót vonz. Az ilyen hurok növekedést vagy összeomlást okoz.
- **Kiegyenlítő (negatív) visszacsatolás:** a változás önmagát korlátozza, és a rendszert egy cél vagy egyensúly felé tereli. Egy termosztát, egy szabályozott hajtás vagy egy árképzési szabály így működik.

A visszacsatolásokat **ok-okozati hurokdiagrammal** ábrázoljuk. A nyilak ok-okozati hatást jelölnek. A „+” jel azt jelenti, hogy a két változó azonos irányba mozdul, a „–” azt, hogy ellentétes irányba. A hurok megerősítő (R), ha páros számú negatív kapcsolat van benne, és kiegyenlítő (B), ha páratlan (Sterman 2000).



1.5. ábra. A VárosiBringa ok-okozati hurokdiagramja. Az R hurok a növekedést hajtja; a B1 hurok azt mutatja, hogy a használat a domborzat miatt maga rontja az elérhetőséget; a B2 hurok az áthelyező szolgálat beavatkozása.

Kidolgozott példa: a VárosiBringa visszacsatolásai

A 1.5. ábra a VárosiBringa három hurkát mutatja.
R – növekedés. Ha nő a felhasználók száma, nő a díjbevétel. Ebből a konzorcium bővítheti a flottát és a dokkolóhálózatot, ami javítja az elérhetőséget, és ez újabb felhasználókat vonz. A hurok minden kapcsolata pozitív, tehát megerősítő.
B1 – önkorlátozás. Bakonyvár dombos város. A felhasználók szívesen gurulnak le a dombról, de kevésbé szívesen tekernek fel, még elektromos rásegítéssel sem. Ha nő a használat, a dombtetői dokkolók kiürülnek, a völgyiek megtelnek. Ez rontja az elérhetőséget: fent nincs kerékpár, lent nincs hova letenni. A rosszabb elérhetőség csökkenti a felhasználók számát. A hurokban egy negatív kapcsolat van, tehát kiegyenlítő: a rendszer maga korlátozza a növekedését.
B2 – beavatkozás. Minél nagyobb a kiegyensúlyozatlanság, annál több áthelyezésre van szükség, és az áthelyezés csökkenti a kiegyensúlyozatlanságot. Ez a hurok is kiegyenlítő, de pénzbe kerül.
Mit tanul ebből a tervező? Ha csak az R hurkot látja, arra jut, hogy minél több kerékpár, annál jobb. A B1 hurok felismerése után más kérdések kerülnek előtérbe. Hogyan lehet a kiegyensúlyozatlanságot olcsóbban kezelni, például díjkedvezménnyel azoknak, akik felfelé visznek kerékpárt? Hol legyenek az állomások, hogy a forgalom kiegyensúlyozottabb legyen? Az áthelyezési költség az üzleti modell egyik fő tétele lesz, amelyet a sorbanállási és szimulációs modellekkel (21. fejezet) becsülünk.

A visszacsatolási hurkok három jellegzetes csapdát rejtenek, amelyekre a rendszermérnöknek figyelnie kell.

- **Késleltetés.** Ha a visszajelzés késve érkezik, a beavatkozás túllő a célon, és lengés alakul ki. A ForgácsTech egyhetes információs késleltetése miatt a termelésvezető mindig a múlt heti helyzetre reagál.
- **Nem szándékolt következmények.** Egy beavatkozás a rendszer egy másik pontján, más hurkon keresztül ellentétes hatást válthat ki. Ha a VárosiBringa díjemeléssel finanszírozná az áthelyezést, az R hurok gyengülne.
- **Tüneti kezelés.** A gyors, tüneti megoldás gyakran elnyomja a valódi okot. Ha a ForgácsTech túlórával pótolja a késéseket, a határidők rövid távon teljesülnek, de az ütemezési probléma rejtve marad. A tüneti kezelésről és a gyökérok-elemzésről a

25. fejezet szól.

1.6. Rendszerszemlélet, rendszermérnökség, mérnöki rendszerek

Három, hasonlóan hangzó fogalmat érdemes pontosan elhatárolni:

- **A rendszerszemlélet** (systems thinking) általános gondolkodásmód. A világot összefüggő rendszerek hálózataként látja, és az elemek helyett a kölcsönhatásokra, a visszacsatolásokra és az egész viselkedésére figyel. Bárki alkalmazhatja: a vezető, az orvos, a közgazdász és a mérnök is.
- **A rendszermérnökség** (systems engineering) a rendszerszemlélet gyakorlati, mérnöki alkalmazása konkrét rendszerek létrehozására. Módszertan és szakma, amely egy rendszert a teljes életciklusán át a megfogalmazott igénytől a kivonásig kísér.
- **A mérnöki rendszerek** (engineering systems) kifejezés azokra a nagy, technológiai, társadalmi és gazdasági szempontból egyaránt összetett rendszerekre utal, amelyek a mérnöki munka tárgyai: energiahálózatok, közlekedési infrastruktúrák, egészségügyi ellátórendszerek. De Weck, Roos és Magee könyve szerint ezek vizsgálata önálló tudományterület, amely a műszaki, a menedzsment- és a társadalomtudományi nézőpontot egyesíti (Weck, Roos, és Magee 2011).

A rendszermérnökségnek több hivatalos meghatározása ismert. Az INCOSE 2019-ben megfogalmazott, az ISO/IEC/IEEE 15288 szabvány 2023-as kiadásában is szereplő meghatározása szerint a rendszermérnökség transzdiszciplináris és integratív megközelítés, amely rendszerelvek és -fogalmak, valamint tudományos, technológiai és menedzsment-módszerek alkalmazásával teszi lehetővé mérnöki rendszerek sikeres megvalósítását, használatát és kivonását (INCOSE 2019; ISO/IEC/IEEE 2023). A *transzdiszciplináris* jelző azt jelenti, hogy a rendszermérnökség nem egyszerűen több szakterület együttműködése. Közös célt, közös fogalmakat és közös gondolkodásmódot teremt, amely túllép az egyes szakterületek határain. Blanchard és Fabrycky klasszikus tankönyve a teljes életciklust helyezi a középpontba: a rendszert a koncepciótól a kivonásig, az életciklus-költségekkel együtt kell tervezni (Blanchard és Fabrycky 2011).

1.6.1. A kiegyensúlyozott nézet

Michael Griffin, a NASA egykori vezetője egy 2007-es előadásában a rendszermérnökséget olyan működőképes rendszer létrehozásának művészeteként és tudományaként írta le, amely gyakran egymásnak ellentmondó korlátok között teljesíti a követelményeket. Szerinte a rendszermérnökség egészben gondolkodó, integráló szakterület: a gépész-, villamos-, szoftver-, ergonómiai és más szakterületek hozzájárulását egymással szemben mérlegeli, hogy olyan összefüggő egész jöjjön létre, amelyet egyetlen szakterület nézőpontja sem ural (Griffin 2007).

A rendszermérnök feladatát Boehm klasszikus kérdéspárja foglalja össze a legtömörebben. Nem elég, ha *jól építjük meg a terméket* (teljesíti a követelményeket); azt is el kell érni, hogy *a jó terméket építsük meg* (szolgálja a működési célokat, és megfeleljen az érdekelt felek elvárásainak) (B. W. Boehm 1984). Az első kérdésre a verifikáció, a másodikra a validáció ad választ a V-modell jobb ágán (23. és 24. fejezet).

A rendszermérnökség szorosan összefonódik a projektmenedzsmenttel. A rendszermérnök a döntések műszaki tartalmáért felel, beleértve azok költség- és ütemezési következményeit is; a projektvezető a költségekért, az ütemtervért, az erőforrásokért és a szerződéses kapcsolatokért. A két szerepkör feladatai sok ponton átfedik egymást: közös terület a tervezés, az érdekelt felek kezelése, a kockázat- és a konfigurációkezelés, valamint a felülvizsgálatok (SEBoK Editorial Board 2025; INCOSE 2023b). Jól működő projektben a kettő kiegészíti egymást, és egyik sem dönt a másik szempontjainak ismerete nélkül.

> **„Elég jó" tervezés**
>
> Gyakran elhangzó mondás, hogy a rendszermérnökség az *éppen elég jó* tervezésének művészete. A gondolat Simon „kielégítő megoldás" (satisficing) fogalmával rokon: összetett helyzetben a döntéshozó nem a matematikai optimumot keresi, hanem olyan megoldást, amely minden lényeges szempontból elég jó (Simon 1956). A mondás nem a hanyagságot dicséri. Azt fejezi ki, hogy egy rendszer nem egyetlen mutatóra optimalizál, hanem a teljesítmény, a költség, a kockázat, az ütemterv, az emberi tényezők és a fenntarthatóság között egyensúlyoz. Egy alrendszer túltervezése, például egy feleslegesen gyors robot, pénzt és időt von el máshonnan, és a rendszer egészét nem javítja. Ezt nevezi a szakirodalom részoptimalizálásnak: az egész rendszert kell optimalizálni, nem egy-egy alrendszert a többi rovására (Blanchard és Fabrycky 2011).

1.7. A rendszermérnök

Ki a rendszermérnök, és mit csinál? Sheard a szakirodalom több mint hatvan szerepleírásából tizenkét jellegzetes rendszermérnöki szerepet gyűjtött össze (Sheard 1996). A legfontosabbak:

- **a követelmények gazdája:** az érdekelt felek igényeiből rendszer- és alrendszerkövetelményeket vezet le, és karbantartja őket;
- **a rendszer tervezője:** felel a rendszer egészének architektúrájáért;
- **elemző:** modellekkel és számításokkal igazolja, hogy a terv teljesítheti a követelményeket;
- **verifikációs és validációs mérnök:** megtervezi és értékeli az ellenőrzéseket;
- **üzemeltetési és logisztikai mérnök:** a használat, a karbantartás, a képzés és a kivonás szempontjait képviseli;
- **az alrendszerek „ragasztója":** az interfészek gazdája, aki a csapatok közötti ellentmondásokat feloldja;
- **kapcsolattartó** a megrendelővel és a felhasználókkal;
- **műszaki vezető:** tervezi és követi a műszaki munkát, kezeli a kockázatokat.

A szerepek nem munkakörök. Egy nagy projektben külön emberek látják el őket, egy kis projektben egyetlen mérnök, vagy éppen a projektvezető. A lényeg az, hogy a rendszermérnöki funkciókat valakinek el kell végeznie.

A rendszermérnök napi munkája ennek megfelelően sokszínű:

- elemzi és pontosítja a követelményeket, és tárgyal róluk az érdekelt felekkel;
- frissíti a rendszertervet, és nyomon követi a változások hatását;
- összehangolja a szakterületi csapatok (gépész, villamos, szoftveres, ergonómus) munkáját;
- interfészeket egyeztet, és feloldja a csapatok közötti ellentmondásokat;
- kockázatokat azonosít, és kompromisszumos döntéseket készít elő;

- teszteket tervez és értékel, hibákat követ nyomon;
- kommunikál a megrendelővel és a felhasználókkal.

1.7.1. Milyen a jó rendszermérnök?

Williams és Derro 2008-ban a NASA-nál 37 olyan rendszermérnököt vizsgált, akiket kollégáik kiemelkedőnek tartottak. Interjúkkal, megfigyeléssel és munkaárnyékolással azt kutatták, milyen viselkedésformák különböztetik meg őket (C. Williams és Derro 2008). A vizsgálat öt nagy témakört azonosított: a vezetést, a hozzáállást és személyes jellemzőket, a kommunikációt, a problémamegoldást és rendszergondolkodást, valamint a műszaki jártasságot. A tanulmány a kiemelkedő rendszermérnökök jellemzői között a következőket sorolja fel (C. Williams és Derro 2008):

- **szellemi kíváncsiság:** az új ismeretek elsajátításának vágya és képessége;
- **az egész látása a részletek elvesztése nélkül:** képes a rendszert egészében látni, mégis elmerülni a részletekben;
- **a változások rugalmas kezelése** és a bizonytalanság elviselése;
- **kivételes kétirányú kommunikáció:** tud figyelni és érthetően magyarázni;
- **sokrétű műszaki jártasság és józan ítélőképesség;**
- **rendszerszintű kapcsolatok kiépítése:** felismeri, hogy egy döntés hol és kit érint;
- **megfelelő mértékű „paranoia”:** a legjobbat várja, de a legrosszabbra is felkészül;
- **jó csapatjátékos és vezető;**
- **magabiztos és döntésképes, de nem arrogáns.**

A cél tehát nem a mindentudó mérnök. A jó rendszermérnök rendszerszinten gondolkodik, kommunikál és dönt, és tudja, mikor kell szakértőt bevonnia.

1.7.2. Kompetenciák és a T alakú szakember

Az INCOSE 2018-ban kiadott kompetenciakerete 36 rendszermérnöki kompetenciát ír le öt csoportban, mindegyiket öt jártassági szinten (INCOSE 2018):

- **alapkompetenciák** (például rendszergondolkodás, életciklusok, képességalapú tervezés, kritikai gondolkodás);
- **szakmai, viselkedési kompetenciák** (például kommunikáció, etika, vezetés, tárgyalás, csapatmunka);
- **műszaki kompetenciák**, amelyek a rendszermérnöki műszaki folyamatokhoz kapcsolódnak (követelmények, architektúra, integráció, verifikáció, validáció és a többi);
- **rendszermérnöki menedzsmentkompetenciák** (tervezés, nyomon követés, döntéskezelés, kockázatkezelés, konfigurációkezelés és a többi);
- **integráló kompetenciák**, amelyekkel mások nézőpontját egységes képbe lehet foglalni (projektmenedzsment, logisztika, minőség, pénzügy).

A rendszermérnöki pályát gyakran a **T alakú szakember** képével írják le. A T függőleges szára egy szakterület mély ismerete: a gépészeté, az irányítástechnikáé, az informatikáé vagy a logisztikáé. A vízszintes szár a széles, rendszerszintű látásmód: a többi szakterület nyelvének megértése, a kommunikáció, a menedzsment és az üzleti szemlélet. A pályakezdő mérnök jellemzően a függőleges szárral kezdi, és a tapasztalattal szélesedik a vízszintes szár. A műszaki menedzser és a mechatronikai mérnök képzés éppen ezt a kettősséget célozza.

KoboCella: ki a rendszermérnök?

A Kobold Hajtástechnika nem alkalmaz „rendszermérnök” nevű munkakört. A KoboCella-projektben a rendszermérnöki feladatokat egy mechatronikai mérnök, a gyártástechnológiai osztály fiatal munkatársa kapta meg, a projektvezető pedig a gyártási igazgató. A mérnök feladata nem az volt, hogy maga tervezze meg a robotpályát vagy a biztonsági vezérlést: ezeket a robotintegrátor és a villamos tervező végezte. Az ő dolga az volt, hogy

- mindenki ugyanazokat a követelményeket ismerje és értse;
- a robotintegrátor, a munkavédelem és az operátorok szempontjai ütközzenek, és az ütközésekből döntések szülessenek;
- minden interfész (mechanikai, villamos, adat- és emberi interfész) gazdája ismert legyen;
- az átadáskor bizonyítani lehessen, hogy a cella teljesíti a vállalt célokat.

Fél év múlva így foglalta össze a tapasztalatait: „Kevesebbet számoltam, mint gondoltam, de sokkal többet beszélgettem. A legfontosabb eszközöm a követelménytáblázat és a felülvizsgálati jegyzőkönyv lett.”

1.8. Az Ipar 4.0-tól az Ipar 5.0-ig

A rendszermérnökség céljai a technológiai és társadalmi környezettel együtt változnak. Az Ipar 4.0 a gyártás digitalizálását, a kiberfizikai rendszereket, az összekapcsolt gépeket és az adatvezérelt döntéseket helyezte előtérbe. Az Európai Bizottság 2021-es jelentése az Ipar 5.0 fogalmával kiegészíti ezt a paradigmát. A jelentés szerint az ipar akkor lehet a jólét ellenálló forrása, ha a technológiai átalakulás három érték köré szerveződik (Breque, De Nul, és Petridis 2021):

- **emberközpontúság:** a technológia az embert szolgálja, a munkavállaló képességeit támogatja, és biztonságosabb, kielégítőbb munkakörnyezetet teremt, nem pedig egyszerűen helyettesíti az embert;
- **fenntarthatóság:** a termelés tiszteletben tartja a bolygó határait, körforgásos és erőforrás-hatékony;
- **reziliencia (ellenállóképesség):** a rendszerek képesek ellenállni a zavaroknak, és helyreállni utánuk, legyen szó ellátási lánc megszakadásáról vagy energiaválságról.

Az Ipar 5.0 tehát nem új technológiák listája, hanem értékválasztás. A technológiák nagyrészt ugyanazok, mint az Ipar 4.0-ban, de a tervezés kiindulópontja a társadalmi és környezeti érték. Ez a rendszermérnök számára konkrét feladatokat jelent. Az érdekelt felek körébe bele kell venni a munkavállalókat és a jövő nemzedékeit (5. fejezet). A funkciómegosztásnál az ember képességeinek támogatására kell törekedni (12. fejezet). A kockázatelemzésnek pedig a rendszer ellenállóképességére is ki kell terjednie (14. fejezet).

Az INCOSE a rendszermérnökség jövőképét felvázoló dokumentumában hasonló irányokat jelöl ki. A jövő rendszereinek a társadalmi igények egyre szélesebb körére kell reagálniuk, okosabbnak, fenntarthatóbbnak, robusztusabbnak és biztonságosabbnak kell lenniük, a fejlesztést pedig egyre inkább modellekre és digitális eszközökre kell építeni (INCOSE 2022). A modellalapú megközelítéssel a következő fejezet foglalkozik.

KoboCella: Ipar 4.0 vagy Ipar 5.0?

A KoboCella első, belső változatú célkitűzése így szólt: „A kézi munkaállomást automatizált cellával váltjuk ki, a létszámot a felére csökkentjük.” Ez tipikus Ipar 4.0-s megfogalmazás. Az üzemi tanács és a munkavédelem bevonása után a célkitűzés megváltozott: „A cella a növekvő igényt az operátorok ergonómiai terhelésének csökkentésével és a minőség javításával szolgálja ki. Az operátor a cella irányítója és minőségellenőre lesz.” A műszaki megoldás, egy kollaboratív robot, mindkét esetben ugyanaz lehet. A követelmények, a funkciómegosztás, a kezelőfelület és a képzési terv azonban egészen másként alakul.

Összefoglalás

- A rendszer egymással kölcsönhatásban álló elemek célra szerveződött együttese. Elemei közé az emberek és az eljárások is beletartoznak. Leírásához elemeket, attribútumokat, kapcsolatokat, határt és környezetet használunk.
- A rendszer hozzáadott értéke főként az elemek kapcsolataiból születik. Az emergens tulajdonságok, mint a ciklusidő, az elérhetőség vagy a biztonság, csak az egészen jelennek meg.
- A rendszerek hierarchikusak: minden rendszer egy befoglaló rendszer része, és alrendszerekre bontható. A befoglaló rendszert ismerni kell, mert az határozza meg a bemeneteket és a kimeneteket. A rendszerek rendszere önálló tulajdonosú, önállóan működő rendszerekből áll.
- A műszaki–gazdasági rendszerek műszaki, gazdasági és információs alrendszerekből, illetve irányítási és irányított alrendszerből állnak.
- A bonyolult rendszerek elemzéssel tervezhetők. A komplex rendszereknél kis lépésekre, kísérletekre és gyors visszacsatolásra van szükség. A tervezés inverz feladat, amelynek általában sok megoldása van.
- A rendszer viselkedését a visszacsatolási hurkok alakítják. A késleltetés, a nem szándékolt következmények és a tüneti kezelés a rendszerszemlélet jellegzetes csapdái.
- A rendszerszemlélet gondolkodásmód, a rendszermérnökség ennek mérnöki alkalmazása, a mérnöki rendszerek pedig a nagy sociotechnikai rendszerek. A rendszermérnök a teljes rendszert látja, kiegyensúlyoz, kommunikál és dönt.
- Az Ipar 5.0 az emberközpontúságot, a fenntarthatóságot és az ellenállóképességet teszi a tervezés kiindulópontjává.

Önellenőrzés

1. Miért fontos, hogy a szabványos rendszerdefiníció az embereket és az eljárásokat is a rendszer elemei közé sorolja? Mondj egy példát a KoboCella esetéből, amikor ennek elhanyagolása hibához vezetne!
2. Mondj egy-egy emergens tulajdonságot a ForgácsTech üzeméből és a saját környezetedből!
3. Egy mobilalkalmazás fejlesztésekor mi lehet a befoglaló rendszer, a rendszer, egy alrendszer és egy komponens?
4. A következő feladatok közül melyik elemzés, melyik tervezés és melyik irányítás? (a) Milyen gyakorisággal kell az áthelyező autónak körbejárnia, hogy a dombtetői állomások ne ürüljenek ki? (b) Hány kerékpár lesz átlagosan egy állomáson reggel 8-kor, ha ismerjük az igényeket és a flottát? (c) Hány állomás és mekkora flotta kell, hogy a kölcsönzések 95%-a sikeres legyen?
5. Rajzold fel a következő hurkot, és döntsd el, megerősítő vagy kiegyenlítő! Ha nő a ForgácsTech határidő-csúszása, nő a túlóra; a túlóra növeli a fáradtságot; a fáradtság növeli a selejtet; a selejt növeli a határidő-csúszást.
6. Miben különbözik a rendszerszemlélet és a rendszermérnökség?
7. Miért nem elegendő, ha egy kollaboratív robotcella célkitűzése a létszám csökkentése? Melyik Ipar 5.0-s értékre hivatkoznál?

Megoldás.

1. Mert a rendszer teljesítménye a technikai elemek és az emberek kölcsönhatásán múlik. Ha az operátort a környezet részének tekintjük, a tervezés kihagyhatja például azt, hogy az alkatrész behelyezése mennyi ideig tart, milyen testhelyzetben végzik, és mit kell az operátornak a kijelzőn látnia. Így a ciklusidő és az ergonómiai célok sem teljesülnek.
2. ForgácsTech: a rendelések átfutási ideje, amely a gépek, az ütemezés, az anyagmozgatás és az információáramlás együtteséből adódik. Saját példa lehet egy csapat hangulata vagy egy városrész forgalmi dugója.
3. Például: befoglaló rendszer a bank teljes ügyfélkiszolgálási rendszere; rendszer a mobilbanki alkalmazás a háttérszolgáltatásaival; alrendszer a bejelentkezési és azonosítási modul; komponens az ujjlenyomat-olvasást kezelő könyvtár.
4. (a) irányítás (adott a rendszer és a kívánt állapot, a beavatkozást keressük); (b) elemzés (adott a rendszer és a bemenet, a kimenetet keressük); (c) tervezés (a rendszert keressük).
5. A hurokban minden kapcsolat pozitív (csúszás → túlóra → fáradtság → selejt → csúszás), tehát megerősítő. A túlóra hosszabb távon rontja a helyzetet; ez a tüneti kezelés csapdájának tipikus esete.
6. A rendszerszemlélet általános gondolkodásmód, amely bármely területen alkalmazható. A rendszermérnökség ennek gyakorlati, mérnöki alkalmazása egy konkrét rendszer életciklusán át, saját folyamatokkal, módszerekkel és felelősségekkel.
7. Mert a cél kizárólag a költségre összpontosít, és figyelmen kívül hagyja a munkavállalókat mint érdekelt feleket. Az emberközpontúság elve szerint a technológiának az ember képességeit kell támogatnia, és biztonságosabb, kielégítőbb munkát kell teremtenie.

Projektlépés: a fejlesztési feladat megfogalmazása

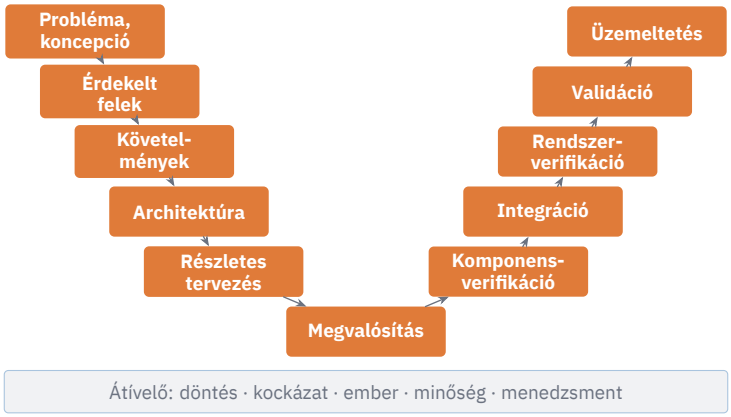
A félév során egy saját fejlesztési feladaton haladsz végig a V-modell lépésein (lásd a Gyakorlati útmutató projektfeladatát). Az első lépésben:

1. válassz egy (elképzelt) szervezetet és egy fejlesztési feladatot a „zöld és digitális átállás” témakörében;
2. írd le néhány mondatban a problémát, a fejlesztés relevanciáját és a várható hozadékot;
3. jelöld ki a fejlesztendő rendszert és a befoglaló rendszert; sorolj fel legalább öt alrendszert;
4. nevezz meg legalább két emergens tulajdonságot, amelyen a fejlesztés sikere múlik;
5. rajzold fel a rendszer legfontosabb visszacsatolási hurkát ok-okozati hurokdiagrammal.

Továbbolvasás

- SEBoK Editorial Board (2025): a rendszermérnökség szabadon elérhető tudásbázisa; bevezetés a rendszerfogalomba és a rendszermérnöki szerepekbe.
- Sheard (1996): a tizenkét rendszermérnöki szerep.
- INCOSE (2019): az INCOSE rendszer- és rendszermérnökség-definíciói magyarázatokkal.
- Kossiakoff és mtsai. (2011), 1–3. fejezet: a komplex rendszerek felépítése és a rendszermérnök feladatai.
- Meadows (2008): olvasmányos bevezetés a rendszergondolkodásba és a visszacsatolási hurkokba.
- Weck, Roos, és Magee (2011): a mérnöki rendszerek mint tudományterület.
- Breque, De Nul, és Petridis (2021): az Európai Bizottság Ipar 5.0-jelentése.
- NASA (2016), 2. fejezet: a rendszer és a rendszermérnökség meghatározása, a rendszermérnök szerepe.

2. A rendszerfejlesztés életciklusa és a V-modell



Tanulási eredmények

A fejezet feldolgozása után képes leszel:

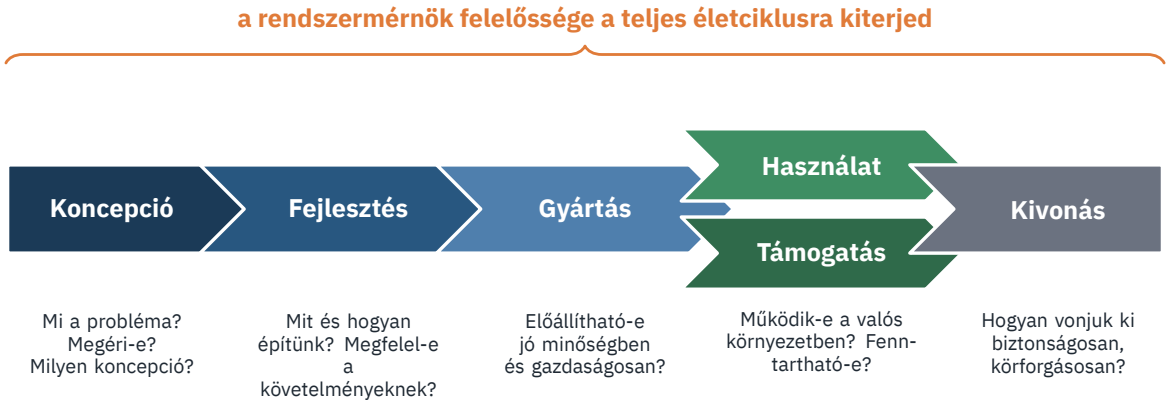
- megnevezni egy műszaki rendszer életciklusának szakaszait, és megindokolni, miért a korai döntések határozzák meg a költségek nagy részét;
- összehasonlítani a vízesés-, a V-, a spirál- és az inkrementális/agilis életciklus-modellt, és egy adott projekthez indokoltan választani közülük;
- elmagyarázni a V-modell bal ágának, csúcsának és jobb ágának feladatait, valamint a két ág közötti vízszintes megfeleltetéseket;
- megkülönböztetni a verifikációt és a validációt, és mindkettőre saját példát adni;
- leírni, hogyan alkalmazzuk a V-modellt rekurzívan a rendszer hierarchiájának szintjein, és mi a szerepe a felülvizsgálatoknak és az alapvonalaknak;
- saját fejlesztési projektedhez elkészíteni a projekt V-tervét.

Ez a fejezet a könyv gerince. Mielőtt bármelyik módszerrel (érdekeltfél-elemzéssel, QFD-vel, FMEA-val vagy folyamatmodellezéssel) foglalkoznánk, látnunk kell azt a pályát, amelyen ezek a módszerek elhelyezkednek. Először a rendszer teljes életét tekintjük át, és megnézzük, miért dől el olyan korán a projekt sorsa. Ezután összevetjük a legfontosabb életciklus-modelleket. A fejezet közepén részletesen bejárjuk a V-modellt, amely a könyv szerkezetét is adja. Végül megmutatjuk, hogyan ismétlődik a V a rendszer különböző szintjein, mit jelentenek a felülvizsgálatok és az alapvonalak, és hogyan néz ki a V a három futó esettanulmányban (A. függelék).

2.1. Egy rendszer élete a koncepciótól a kivonásig

Egy új rendszer nem akkor kezdődik, amikor az első alkatrészt legyártják, és nem is akkor ér véget, amikor a vevő átveszi. A rendszermérnöki szemlélet egyik alapgondolata, hogy a

rendszert a teljes életciklusára tervezzük. Ebbe beletartozik, hogyan gyártják, hogyan használják, hogyan tartják karban, és hogyan vonják ki a forgalomból. Az életciklus-kezelés nemzetközi szabványa és az INCOSE rendszermérnöki kézikönyve hat általános életciklus-szakaszt különböztet meg: koncepció, fejlesztés, gyártás, használat, támogatás és kivonás (2.1. ábra) (ISO/IEC/IEEE 2024; INCOSE 2023b, 2.1.2. szakasz).



2.1. ábra. A műszaki rendszerek általános életciklus-szakaszai és a szakaszok kulcskérdései. A használat és a támogatás (karbantartás, javítás, logisztika) párhuzamosan zajlik. Az INCOSE kézikönyve (INCOSE 2023b) alapján, saját szerkesztés.

A szakaszok tartalma röviden a következő.

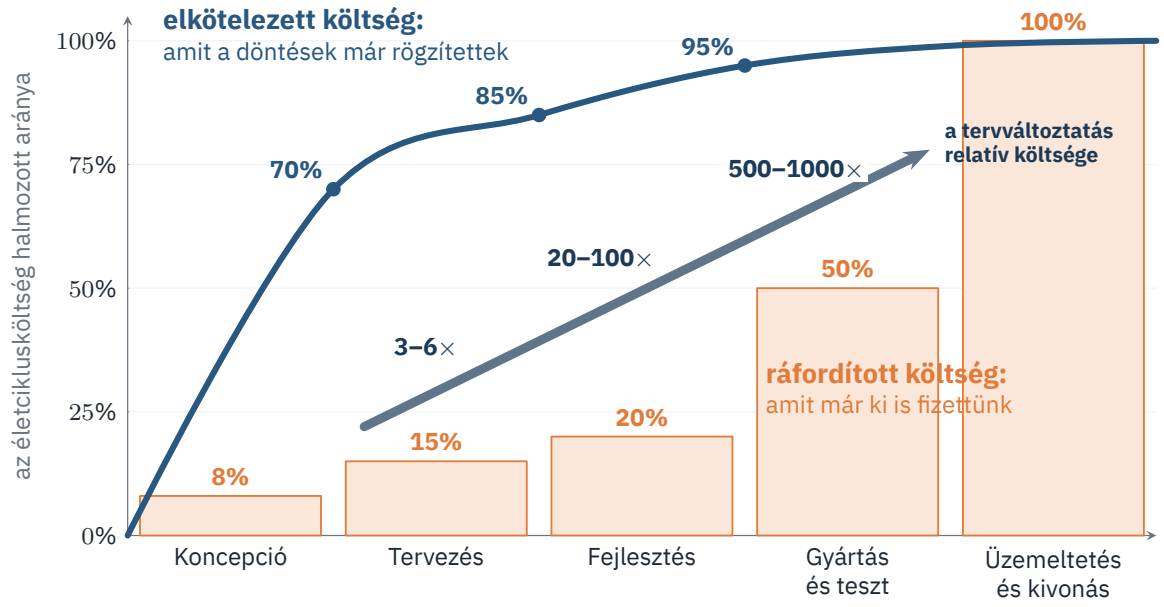
- A **koncepciószakaszban** megértjük a problémát és az érdekelt felek igényeit. Lehetséges megoldási elképzeléseket vázolunk fel, és eldöntjük, érdemes-e egyáltalán belevágni.
- A **fejlesztési szakaszban** a kiválasztott koncepcióból követelmények, architektúra és részletes terv lesz. Ebben a szakaszban készülnek és ellenőrződnek a prototípusok is.
- A **gyártási szakasz** a rendszer előállítását, sorozatgyártásnál a gyártási folyamat felfuttatását jelenti.
- A **használat** során a rendszer azt teszi, amire szántuk. A **támogatás** gondoskodik arról, hogy ezt hosszú távon is meg tudja tenni.
- A **kivonás** a rendszer biztonságos leállítását, szétszerelését, az anyagok visszanyerését vagy ártalmatlanítását foglalja magában.

A szakaszok száma és elnevezése iparáganként és szervezetenként eltér. Az űrkutatásban és a védelmi iparban betűvel vagy számmal jelölt fázisok terjedtek el, egy gyártócég pedig gyakran a saját termékfejlesztési folyamatának lépéseit nevezi meg. A szabvány ezért csak általános keretet ad, amelyet minden szervezet a saját gyakorlatához igazít (ISO/IEC/IEEE 2024). A lényeg mindenhol ugyanaz: a döntéseket a teljes életút következményeinek ismeretében kell meghozni.

2.1.1. Miért dől el korán a projekt sorsa?

A 2.2. ábra a rendszermérnökség talán legtöbbet idézett összefüggését mutatja. Az amerikai Védelmi Beszerzési Egyetem (DAU) vizsgálatai szerint, amelyeket az INCOSE kézikönyve is közöl, a koncepciószakasz végére, a munka mintegy 8%-ának elvégzésekor a meghozott döntések az életciklusköltség körülbelül 70%-át már elkötelezik, a tervezés végére pedig mintegy 85%-át (INCOSE 2015). A ténylegesen elköltött összeg ekkor még

csak töredéke ennek. A tervezés dönti el ugyanis, hogy a rendszert mennyibe kerül majd tesztelni, gyártani, integrálni, üzemeltetni és fenntartani.



2.2. ábra. Az elkötelezett és a ténylegesen ráfordított életciklusköltség, valamint a tervváltoztatás relatív költsége az életciklus előrehaladtával. A számértékek szemléltetők, projektenként eltérnek, a görbék alakja azonban általános. Az INCOSE kézikönyvében közölt, a DAU-tól származó ábra (INCOSE 2015) alapján, saját szerkesztés.

Az ábra másik üzenete a tervváltoztatás költségének meredek növekedése. Egy probléma, amelyet a tervezés során háromszor-hatszor annyiba kerül kijavítani, mint a koncepciónál, a gyártás és a tesztelés idején már nagyságrendekkel drágább. Az üzemeltetés során felfedezett hiba ára akár ezerszeres is lehet. Ennek oka egyszerű: minél később változtatunk, annál több kész munkát kell kidobni és újra ellenőrizni.

Ebből következik a rendszermérnökség egyik legfontosabb gyakorlati tanulsága. **A korai szakaszokba fektetett gondos munka – az igények feltárása, a követelmények tisztázása, a koncepciók összevetése – a teljes projekt legjobb megtérülésű befektetése.**

> **KoboCella: egy korai döntés hosszú árnyéka**
>
> A Kobold Hajtástechnika csapata a koncepciószakaszban dönti el, hogy a kobot és az operátor ugyanazon a munkaterületen, egyidejűleg dolgozik-e, vagy időben elkülönülve, egymás után. Ezen az egyetlen döntésen múlik:
>
> - milyen biztonsági megoldások kellenek (sebesség- és erőkorlátozás vagy elválasztó védőberendezés);
> - mekkora helyet foglal a cella, és milyen lesz az operátor mozgásterülete;
> - mekkora ciklusidő érhető el;
> - milyen kockázatelemzést és milyen tanúsítást kell elvégezni.
>
> Ha a csapat ezt csak az üzembe helyezéskor gondolná újra, a cella elrendezését, a vezérlést, a biztonsági rendszert és a dokumentációt is újra kellene tervezni.

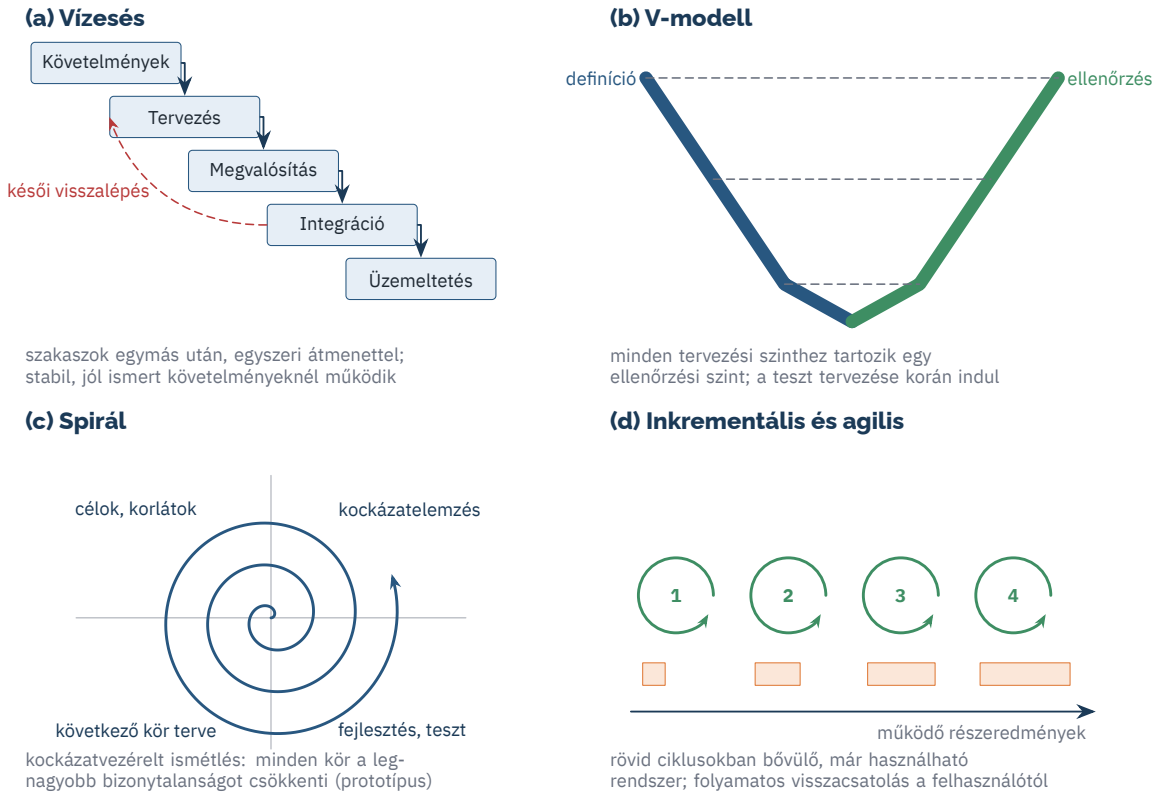
A költség azonban csak az egyik szempont. Ha egy megoldás már jól kiegyensúlyozott, vagyis egyik jellemzője sem javítható a többi rontása nélkül (a 10. fejezet szavával: a Pareto-fronton van), akkor a teljesítmény, a költség és a kockázat nem javítható egyszerre. Ez a rendszermérnök dilemmája:

- adott kockázat mellett a költség csökkentése csak a teljesítmény rovására lehetséges;
- adott költség mellett a kockázat csökkentése szintén teljesítményt kíván;
- adott teljesítmény mellett a költség csökkentése nagyobb kockázatvállalással jár;
- adott teljesítmény mellett a kockázat csökkentése többletköltséggel jár.

Az idő ebben az összefüggésben a költséghez hasonlóan viselkedik. A rendszermérnök feladata, hogy ezeket a kompromisszumokat tudatosan és átláthatóan kezelje. Ehhez nyújtanak segítséget a 10. fejezet döntéstámogató módszerei.

2.2. Életciklus-modellek: hogyan haladunk végig a szakaszokon?

Az életciklus-szakaszok azt írják le, *mi* történik a rendszerrel. Az **életciklus-modell** azt, hogy a fejlesztés során *milyen sorrendben, milyen visszacsatolásokkal és milyen ismétlésekkel* haladunk végig a tevékenységeken. A szakirodalom sokféle modellt ismer. Itt a négy legfontosabbat vetjük össze (2.3. ábra, 2.1. táblázat).



2.3. ábra. Négy alapvető életciklus-modell vázlata: (a) vízesés, (b) V-modell, (c) spirálmodell, (d) inkrementális és agilis fejlesztés. Saját szerkesztés.

A vízesésmodell a tevékenységeket egymást követő szakaszokba rendezi. Minden szakasz egy jól meghatározott dokumentummal zárul, és a következő csak ennek elkészülte után indul. A modell eredetét többnyire Winston Royce 1970-es cikkére vezetik vissza (Royce 1970). Érdekes, hogy Royce éppen ennek a tisztán egyirányú folyamatnak a kockázataira figyelmeztetett. A szakaszok közötti visszacsatolást és egy előzetes, „kétszer építsd meg” jellegű próbaváltozat elkészítését javasolta. A vízesés előnye az átláthatóság és a jó tervezhetőség. Hátránya, hogy a hibák jellemzően csak a végén, az integráció és a tesztelés során derülnek ki, és a menet közben változó követelményeket nehezen kezeli.

A V-modell a vízesés lépéseit V alakba hajlítja. A bal ágon fentről lefelé haladva bontjuk le és határozzuk meg a rendszert, a jobb ágon lentről felfelé haladva építjük össze és ellenőrizzük. A modell ereje a két ág közötti vízszintes kapcsolatokban rejlik: minden meghatározási szinthez tartozik egy ellenőrzési szint, és az ellenőrzés tervét már a meghatározással egy időben elkészítjük. A V alakú ábrázolás szoftverfejlesztési változatát Paul Rook írta le 1986-ban (Rook 1986). A rendszermérnöki „Vee" modellt Kevin Forsberg és Harold Mooz 1991-ben mutatta be (Forsberg és Mooz 1991). Ebben a technikai folyamat a bal felső sarokban a felhasználói igényekkel indul, és a jobb felső sarokban a felhasználó által validált rendszerrel zárul. Ettől függetlenül Németországban a szövetségi informatikai projektek kötelező eljárásrendjévé vált a V-Modell. Mai változata a 2005 óta érvényes V-Modell XT (Friedrich és mtsai. 2009).

A spirálmodellt Barry Boehm írta le (B. W. Boehm 1988). A fejlesztés egymást követő körökben halad. Minden kör a célok és korlátok tisztázásával indul, ezt kockázatelemzés követi, majd a legnagyobb bizonytalanság csökkentésére szolgáló fejlesztési és tesztelési lépés (gyakran prototípus), végül a következő kör megtervezése. A spirálmodell lényege, hogy a kockázatok vezérlik a folyamatot. Ott és annyit kell elemezni és prototípust építeni, ahol a legnagyobb a bizonytalanság. A modell későbbi továbbfejlesztése a fokozatos elköteleződésre épülő spirálmodell (Incremental Commitment Spiral Model) (B. Boehm és mtsai. 2014).

Az inkrementális és az agilis fejlesztés rövid ciklusokban, működő részeredményeket szállítva halad. A 2001-es Agilis kiáltvány (Beck és mtsai. 2001) a folyamatos együttműködést, a működő terméket és a változásokra való gyors reagálást helyezte előtérbe. A szoftverfejlesztésben ez ma elterjedt gyakorlat. A hardvert is tartalmazó, nagyobb rendszerekben viszont jellemzően hibrid megoldás működik. A rendszerszintű követelményeket, az architektúrát és a biztonsági igazolást V-modell szerint kezelik, az egyes alrendszereken, különösen a szoftveren belül pedig agilis ciklusokban dolgoznak. Az INCOSE kézikönyve az agilis rendszermérnökséget már önálló megközelítésként tárgyalja (INCOSE 2023b).

2.1. táblázat. Az életciklus-modellek összehasonlítása. Saját összeállítás Royce (1970), Forsberg és Mooz (1991), B. W. Boehm (1988) és INCOSE (2023b) alapján.

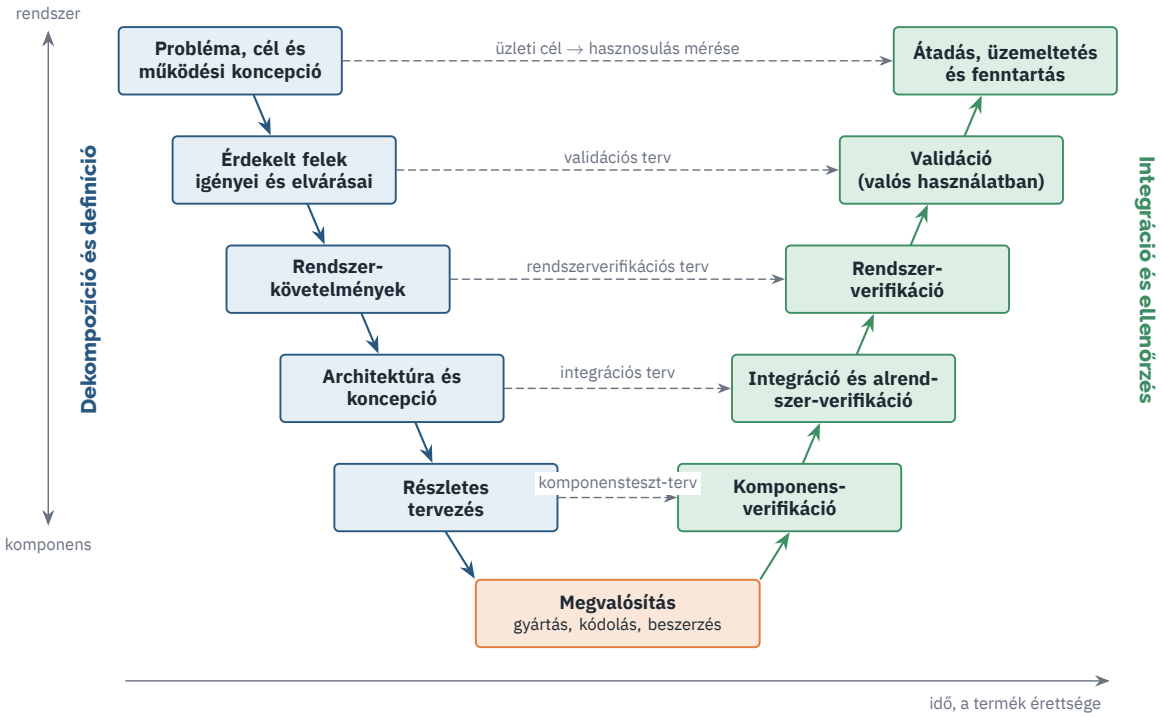
Modell	Mikor jó választás?	Erősségek	Kockázatok
Vízesés	Stabil, jól ismert követelmények; ismert technológia; kis bizonytalanság	Egyszerű, jól tervezhető, jól dokumentált	Késői hibafelismerés; a változás drága
V-modell	Biztonságkritikus vagy szabályozott termékek; sok alrendszer; szigorú igazolási igény	A verifikáció és a validáció korai tervezése; nyomonkövethetőség	Merev, ha szó szerint egyirányú folyamatnak értelmezik
Spirál	Új technológia, nagy műszaki vagy piaci bizonytalanság	Célzott kockázatcsökkentés, prototípusok	Nagyobb irányítási igény; nehezebb ütemezni
Inkrementális/agilis	Gyorsan változó igények; szoftverintenzív rendszerek; korai felhasználói visszajelzés kell	Gyors tanulás, korán használható eredmény	Az architektúra és a rendszerszintű igazolás elsikkadhat

A V-modell nem időrend

Gyakori félreértés, hogy a V-modell egy szigorú időbeli sorrendet ír elő: előbb minden követelményt véglegesíteni kell, aztán minden tervet, és csak a legvégén szabad tesztelni. A V-modell valójában **logikai kapcsolatokat** ábrázol: mi miből következik, és mit mivel ellenőrzünk. A gyakorlatban a V-t sokszor bejárjuk (prototípussal, szimulációval, részleges integrációval), és a jobb ág tevékenységei, például egy kritikus alkatrész korai tesztelése, a bal ági munkával párhuzamosan is indulhatnak. A V és az iteráció tehát nem zárja ki egymást (lásd 2.4. szakasz).

2.3. A V-modell bejárása

A könyv további része a 2.4. ábrán látható V-modellt követi. Az ábra a bal ágon öt meghatározási szintet, a csúcson a megvalósítást, a jobb ágon pedig öt ellenőrzési és üzembe állítási lépést mutat. A szaggatott vízszintes vonalak jelzik, melyik bal ági munkatermék szolgál melyik jobb ági ellenőrzés alapjául.



2.4. ábra. A rendszerfejlesztés V-modellje, ahogyan ez a könyv használja. A bal ág a dekompozíció és a definíció, a jobb ág az integráció és az ellenőrzés. A vízszintes vonalak jelzik, hogy minden meghatározási szint egyben a hozzá tartozó ellenőrzés terve is. Forsberg és Mooz (1991) és INCOSE (2023b) alapján, saját szerkesztés.

2.3.1. A bal ág: a problémától a részletes tervig

A bal ágon minden lépés egy kérdésre felel, és egy munkaterméket (dokumentumot, modellt vagy döntést) állít elő, amely a következő lépés bemenete lesz.

1. **Probléma, cél és működési koncepció.** *Milyen problémát oldunk meg, és miért éri meg?* A munkatermék a szükségletek, célok és célkitűzések rendszere, valamint a

működési koncepció (ConOps). A ConOps a felhasználó szemszögéből írja le, hogyan fogják használni a rendszert, beleértve a rendellenes és a leromlott üzemmódokat is (4. fejezet).
2. **Az érdekelt felek igényei és elvárásai.** *Kiket érint a rendszer, és mit várnak tőle?* Az eredmény az érdekelt felek nyilvántartása és a validált elvárások listája, a hatékonyság mérőszámaival (5. fejezet).
3. **Rendszerkövetelmények.** *Mit kell a rendszernek mérhetően teljesítenie?* Az elvárásokból egyértelmű, ellenőrizhető, a megoldást elő nem író követelmények lesznek, mindegyikhez hozzárendelt verifikációs móddal (6. és 7. fejezet).
4. **Architektúra és koncepció.** *Milyen elemekből, milyen kapcsolatokkal épül fel a rendszer?* A funkciókat formai elemekhez rendeljük, koncepcióváltozatokat generálunk és választunk közülük, a követelményeket pedig lebontjuk az alrendszerekre (8., 9. és 10. fejezet).
5. **Részletes tervezés.** *Pontosan hogyan valósítjuk meg az egyes elemeket?* Ide tartoznak a gyártási rajzok, a kapcsolási rajzok, a szoftvertervek és a tűrések. Ezen a szinten kerülnek előtérbe a megbízhatósági, biztonsági és ember–gép szempontok is (11–18. fejezet).

2.3.2. A csúcs: megvalósítás

A V csúcsán a tervekből valóság lesz. Gyártjuk, beszerezzük vagy kódoljuk az elemeket. Mivel könyvünk műszaki menedzsereknek és mechatronikai mérnököknek is szól, ide soroljuk a gyártó- és szolgáltatófolyamatok tervezését is: a folyamatelemzést, a kapacitástervezést, a folyamatmodellezést és az ütemezést (19–21. fejezet). Ha a fejlesztés tárgya maga egy gyártócella vagy egy szolgáltatás, ezek a módszerek a megvalósítás magját alkotják.

2.3.3. A jobb ág: az integrációtól az üzemeltetésig

A jobb ágon lentről felfelé haladva állítjuk össze és ellenőrizzük a rendszert.

1. **Komponens-verifikáció.** Minden elkészült elemről igazoljuk, hogy megfelel a részletes terv követelményeinek.
2. **Integráció és alrendszer-verifikáció.** Az elemeket lépésenként összeépítjük. Ez az a pont, ahol az interfészhibák jellemzően felszínre kerülnek (22. fejezet).
3. **Rendszerverifikáció.** A teljes rendszerről igazoljuk, hogy minden rendszerkövetelményt teljesít (23. fejezet).
4. **Validáció.** Valós vagy valósághű környezetben, valódi felhasználókkal igazoljuk, hogy a rendszer valóban azt nyújtja, amit az érdekelt felek vártak (24. fejezet).
5. **Átadás, üzemeltetés és fenntartás.** A rendszert átadjuk, beüzemeljük, működtetjük, és mérjük, hogy a várt haszon valóban megvalósul-e. Az üzemeltetési tapasztalatok a folyamatos fejlesztés és a következő fejlesztési ciklus bemenetei lesznek (25. fejezet).

2.3.4. Vízszintes megfeleltetések: korai teszttterv

A V-modell legfontosabb gyakorlati üzenete a vízszintes vonalakban rejlik. **Amikor a bal ágon megfogalmazunk valamit, egyúttal azt is el kell döntenünk, hogyan fogjuk a jobb ágon igazolni.** Ha egy követelményről a megfogalmazás pillanatában nem tudjuk megmondani, milyen vizsgálattal ellenőrizzük majd, akkor az a követelmény valószínűleg

nem elég pontos. Így a korai teszterv nem bürokratikus többletmunka, hanem a bal ági munka minőségének próbája.

Verifikáció és validáció

Az ISO/IEC/IEEE 15288 szabvány a két fogalmat így választja szét (ISO/IEC/IEEE 2023):

- **Verifikáció:** objektív bizonyítékokkal alátámasztott megerősítése annak, hogy a termék teljesíti a rögzített követelményeket. A bizonyítékot vizsgálat, elemzés, bemutatás vagy teszt adja, vagy ezek kombinációja. Boehm rövid kérdésével: *jól építjük-e meg a terméket?* (B. W. Boehm 1984)
- **Validáció:** objektív bizonyítékokkal alátámasztott megerősítése annak, hogy a termék a tervezett használatra, a tervezett környezetben alkalmas, vagyis betölti a rendeltetését, és megfelel a megrendelő és a többi érdekelt fél elvárásainak. Boehm kérdésével: *a jó terméket építjük-e?*

A verifikáció a jóváhagyott követelményekre hivatkozik, a validáció a működési koncepcióra és az érdekelt felek elvárásaira. Validálni nemcsak a végén lehet és kell: a korai szakaszokban modellekkel, makettekkel és szimulációval is feltehető a kérdés, hogy a megfelelő terméket építjük-e a felhasználóinknak (INCOSE 2023b).

VárosiBringa: verifikáció vagy validáció?

Az egyik rendszerkövetelmény így szól: *„A kerékpár kölcsönzésének elindítása az alkalmazásban a QR-kód beolvasásától a zár nyitásáig legfeljebb 5 másodpercig tarthat, a kölcsönzések legalább 95%-ában.”* Ennek ellenőrzése **verifikáció**: tesztkörnyezetben, mért időkkel igazolható. A validáció más kérdést tesz fel: *képesek-e a 65 év feletti lakók esős időben, kesztyűben, segítség nélkül kölcsönözni?* Lehet, hogy az 5 másodperces követelmény teljesül, a felhasználók mégis elakadnak, mert a QR-kód a dokkoló alján, rossz szögben van. A rendszer ilyenkor átmegy a verifikáción, a validáción viszont elbukik. A hiba forrása a bal ág felső szintjein keresendő: valamelyik érdekelt fél elvárását nem tártuk fel, vagy nem fordítottuk le követelménnyé.

2.3.5. A V-modell és a szabványos folyamatok

A V-modell lépései nem egy könyv kitalációi: megfelelnek a nemzetközi szabványok folyamatainak. Az ISO/IEC/IEEE 15288:2023 szabvány harminc folyamatot határoz meg négy csoportban: megállapodási, szervezeti projekttámogató, technikai irányítási és technikai folyamatokat (ISO/IEC/IEEE 2023). A tizennégy technikai folyamat sorra követi a V-t az üzleti vagy küldetéselemzéstől a kivonásig, a nyolc technikai irányítási folyamat pedig a teljes V-n átível. Az INCOSE kézikönyve ugyanezt a folyamatrendszert részletezi gyakorlati tanácsokkal (INCOSE 2023b, 2.3. szakasz). A 2.2. táblázat megmutatja, melyik V-lépéshez melyik szabványos folyamat tartozik, és a könyv melyik fejezete foglalkozik vele.

2.2. táblázat. A könyv V-modelljének megfeleltetése az ISO/IEC/IEEE 15288:2023 folyamatainak (ISO/IEC/IEEE 2023). A folyamatnevek saját fordítások; a megfeleltetés közelítő, mert a szabvány folyamatai egymással párhuzamosan és ismétlődően is futhatnak.

A V-modell lépése	ISO/IEC/IEEE 15288:2023 folyamat	Szakasz	Fejezet
Probléma, cél, működési koncepció	Üzleti vagy küldetéselemzés	6.4.1	4.

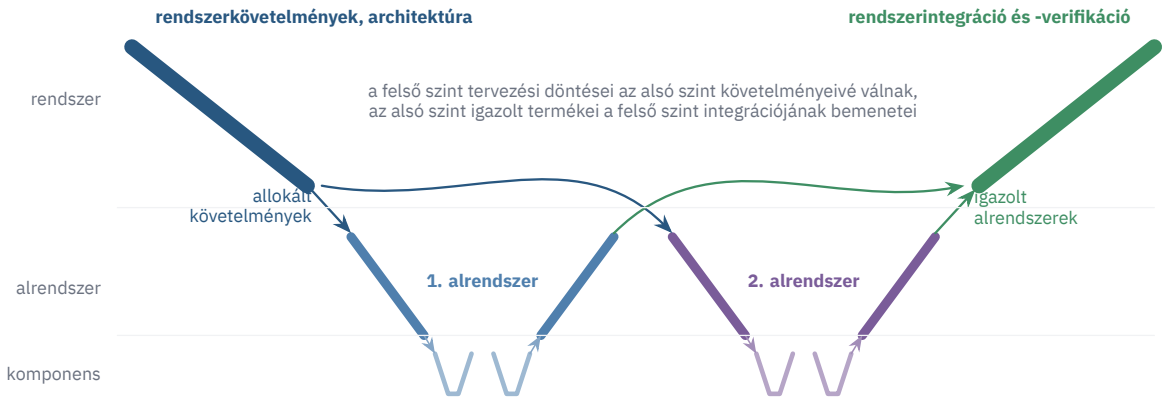
A V-modell lépése	ISO/IEC/IEEE 15288:2023 folyamat	Szakasz	Fejezet
Az érdekelt felek igényei	Az érdekelt felek igényeinek és követelményeinek meghatározása	6.4.2	5.
Rendszerkövetelmények	Rendszerkövetelmények meghatározása	6.4.3	6–7.
Architektúra és koncepció	Rendszerarchitektúra meghatározása; rendszerelemzés	6.4.4; 6.4.6	8–10.
Részletes tervezés	Tervmeghatározás; rendszerelemzés	6.4.5; 6.4.6	11–18.
Megvalósítás	Megvalósítás	6.4.7	19–21.
Integráció	Integráció	6.4.8	22.
Verifikáció (komponens- és rendszerszinten)	Verifikáció	6.4.9	23.
Validáció	Validáció	6.4.11	24.
Átadás, üzemeltetés, fenntartás	Átadás; üzemeltetés; karbantartás; kivonás	6.4.10; 6.4.12–6.4.14	25.
Átívelő tevékenységek	Technikai irányítási folyamatok (projekttervezés, döntéskezelés, kockázatkezelés, konfigurációkezelés stb.)	6.3.1–6.3.8	10., 14., 26.

2.4. A V a V-ben: rekurzió és iteráció

A V-modellt nem egyszer alkalmazzuk a teljes rendszerre. A rendszer hierarchiájának minden szintjén újra lejátszódik (2.5. ábra). A rendszerszintű architektúra eredményeként kialakulnak az alrendszerek, és mindegyik megkapja a rá lebontott, allokált követelményeket. Az alrendszer fejlesztője számára ezek az allokált követelmények jelentik a saját V-jének bal felső sarkát. Az alrendszer saját architektúrát, komponenseket és részletes tervet dolgoz ki, a komponenseket verifikálja, integrálja, végül a kész alrendszert igazoltan visszaadja a rendszerszintű integrációnak.

Az ISO/IEC/IEEE 15288 szabvány a folyamatok alkalmazásának kétféle ismétlődését különbözteti meg (ISO/IEC/IEEE 2023, 5.8.2. szakasz; NASA 2016, 2.3. szakasz). A **rekurzió** azt jelenti, hogy ugyanazokat a folyamatokat a termékszerkezet egymás alatti szintjein újra alkalmazzuk. Az **iteráció** azt jelenti, hogy egy szinten belül többször is végigmegyünk a folyamatokon, amíg a követelmények, a koncepció és a terv összhangba nem kerülnek. A fejlesztés így egymást követő finomításokkal halad: a koncepciószakasz vázlatos körét egyre részletesebb körök követik, és minden kör érettebb, pontosabb eredményt ad.

> **KoboCella: a cella termékszerkezete**
>
> A KoboCella rendszerszintű V-jének architektúralépése öt alrendszert jelöl ki:
>
> 1. a kobot a megfogóval;
> 2. a csavarozóegység a nyomatékszabályozással;



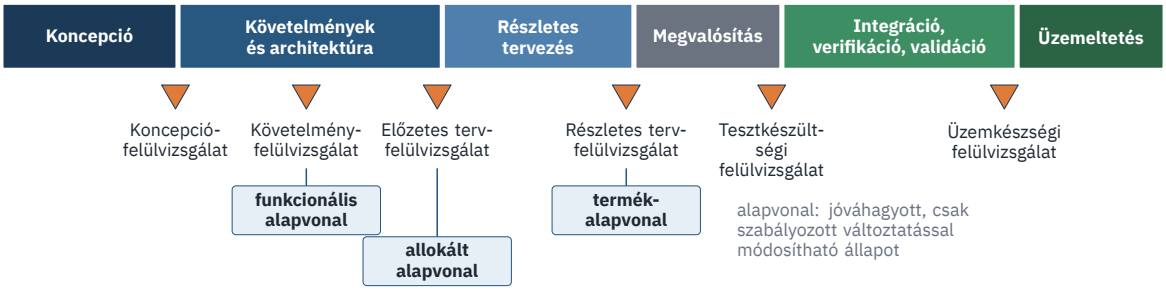
2.5. ábra. A V-modell rekurzív alkalmazása. A rendszerszintű tervezés döntései az alrendszerek követelményeivé válnak, az igazoltan elkészült alrendszerek pedig a rendszerintegráció bemenetei. Saját szerkesztés.

3. a biztonsági rendszer (érzékelők, biztonsági vezérlő);
4. a munkahely és az ergonómiai kialakítás;
5. a cellavezérlés a MES-kapcsolattal.

A csavarozóegységet a beszállító fejleszti. Neki a „8 csavar meghúzása 4,0 ± 0,2 Nm nyomatékkal, 2 másodperces csavaronkénti ciklusidővel, minden meghúzás nyomatékgörbéjének naplózásával” típusú allokált követelmények jelentik a kiindulópontot. A beszállító saját V-t jár be, a cella csapata pedig a kész egység verifikációs jegyzőkönyvét kapja meg, mielőtt integrálja.

2.5. Mérföldkövek, felülvizsgálatok és alapvonalak

Honnan tudjuk, hogy egy szakasz eredménye elég érett ahhoz, hogy a következőre lépjünk? Erre szolgálnak a **technikai felülvizsgálatok**. Ezek előre tervezett, dokumentált bizonyítékokra épülő értékelések, amelyeken az érdekelt felek és a független szakértők közösen döntik el, hogy a munka a következő szakaszba léphet-e (2.6. ábra) (IEEE 2014). A felülvizsgálat nem formalitás: az a pont, ahol a projekt még olcsón korrigálhat.



2.6. ábra. Tipikus technikai felülvizsgálatok és alapvonalak a fejlesztés során. Az IEEE 15288.2 szabvány (IEEE 2014) és a DAU (Defense Acquisition University 2001) gyakorlata alapján, egyszerűsítve.

2.3. táblázat. A technikai felülvizsgálatok áttekintése. Az elnevezések szervezetenként eltérnek; a nagy projektek további felülvizsgálatokat is alkalmaznak, például funkcionális és fizikai konfigurációs auditot (IEEE 2014; Defense Acquisition University 2001; NASA 2016, 3. fejezet).

Felülvizsgálat	Kulcskérdés	Tipikus bizonyítékok	Angol elnevezés
Koncepció-felülvizsgálat	Értjük-e a problémát, és van-e megvalósítható, megérő koncepció?	Szükségletek és célok, működési koncepció, koncepcióváltozatok, előzetes költség- és kockázatbecslés	Alternative Systems Review (ASR)
Követelmény-felülvizsgálat	Teljesek, egyértelműek és ellenőrizhetők-e a rendszerkövetelmények?	Rendszerkövetelmény-specifikáció, nyomonkövethetőség az elvárásokig, verifikációs módok	System Requirements Review (SRR)
Előzetes terv-felülvizsgálat	Teljesítheti-e a választott architektúra az összes követelményt elfogadható kockázattal?	Architektúra, allokált követelmények, interfészek, kompromisszumelemzések, előzetes biztonsági és megbízhatósági elemzések	Preliminary Design Review (PDR)
Részletes terv-felülvizsgálat	Kész-e a terv a gyártásra, kódolásra és beszerzésre?	Részletes tervek, tűrések, FMEA, tesztelési tervek	Critical Design Review (CDR)
Tesztkészültségi felülvizsgálat	Felkészültünk-e a formális tesztelésre?	Tesztelési eljárások, kalibrált eszközök, a tesztobjektum konfigurációja	Test Readiness Review (TRR)
Üzemkészségi felülvizsgálat	Átadható-e a rendszer üzemeltetésre?	Verifikációs és validációs eredmények, kezelési utasítások, képzés, karbantartási terv	System Verification Review (SVR), Operational Readiness Review (ORR)

A felülvizsgálatok egyik legfontosabb kimenete az **alapvonal**. Ez a rendszer leírásának jóváhagyott állapota, amely ettől kezdve csak szabályozott változáskezeléssel módosítható. Az amerikai védelmi gyakorlat a fejlesztési szintekhez igazodva három alapvonalat különböztet meg (Defense Acquisition University 2001):

- a **funkcionális alapvonal** a rendszerszintű, teljesítményben megfogalmazott leírás;
- az **allokált alapvonal** az alrendszerek és komponensek teljesítményleírása;
- a **termékalapvonal** a gyártáshoz szükséges részletes leírás.

Az alapvonal haszna egyszerű: mindenki ugyanarról a változatról beszél. Egy változtatási javaslat hatása (költség, ütemezés, más elemekre gyakorolt következmény) az alapvonalhoz képest értékelhető. A konfigurációkezelést a 26. fejezet tárgyalja részletesen.

Felülvizsgálat mint előadás

Gyakori hiba, hogy a felülvizsgálat a csapat bemutatkozó előadásává válik, ahol mindenki azt hangsúlyozza, ami jól sikerült. A felülvizsgálat akkor éri el a célját, ha **előre rögzített belépési és kilépési feltételei vannak** (milyen dokumentumoknak, eredményeknek kell meglenniük), a bírálók a dokumentumokat előre megkapják, és a nyitott kérdéseket felelőssel és határidővel rögzítik a jegyzőkönyvben.

A felülvizsgálatok száma és formalitása a projekt méretéhez igazítható. Egy féléves hallgatói projektben vagy egy kisvállalati fejlesztésben elegendő lehet három ellenőrzési pont: a koncepció és a követelmények jóváhagyása, a terv jóváhagyása, végül az átadás előtti ellenőrzés. A folyamatok projekthez igazításának (tailoring) a szabványok és a szakirodalom külön figyelmet szentel (ISO/IEC/IEEE 2023; INCOSE 2023b, 4. fejezet; NASA 2016, 3.11. szakasz). Az elv egyszerű: a folyamat szolgálja a projektet, és ne fordítva.

2.6. A három futó eset a V-modellben

Hogyan néz ki mindez a gyakorlatban? A 2.4. táblázat a három futó esettanulmány néhány jellemző munkatermékét rendeli a V-modell lépéseihez. Érdemes megfigyelni, hogy a C eset, a ForgácsTech digitális átállása esetében a „rendszer” elsősorban egy szervezet és annak folyamatai. A V-modell logikája itt is ugyanúgy működik: a problémából követelmények lesznek, azokból terv, a tervből megvalósítás, a megvalósítást pedig ellenőrizni és validálni kell.

2.4. táblázat. Jellemző munkatermékek a V-modell lépéseiben a három futó esettanulmányban.

V-lépés	A: KoboCella	B: VárosiBringa	C: ForgácsTech
Probléma, működési koncepció	Növekvő igény, ergonómiai panaszok, utómunka; a műszak működési forgatókönyvei	Rövid autós utak kiváltása dombos városban; a kölcsönzés és visszaadás forgatókönyvei	Határidő-csúszások, átláthatatlan gépállapot; egy rendelés útja az üzemen át
Érdekelt felek	Operátorok, munkavédelem, vevő, integrátor	Lakosok, önkormányzat, üzemeltető, közlekedési társaság	Tulajdonos, gépkezelők, vevők, MES-szállító
Rendszerkövetelmények	Ciklusidő, első átmenetes kihozatal, ergonómiai terhelés, nyomaték-naplózás	Elérhetőség, kölcsönzési idő, áthelyezési költség, akadálymentesség	Átfutási idő, adatgyűjtési lefedettség, OEE-mérés pontossága
Architektúra	Kobot, csavarozó, biztonsági rendszer, munkahely, vezérlés	Kerékpár, dokkoló, alkalmazás, háttérrendszer, szerviz	Gépcsatlakozások, MES, ütemező modul, terminálok
Részletes terv	Megfogó, biztonsági zónák, kezelőfelület	Dokkolózár, töltés, felhasználói felület	Adatmodell, képernyők, munkautasítások
Megvalósítás	Cella építése, programozás	Gyártás, telepítés, fejlesztés	Bevezetés gépcsoportonként

V-lépés	A: KoboCella	B: VárosiBringa	C: ForgácsTech
Verifikáció	Nyomaték- és ciklusidő-mérés, biztonsági funkciók tesztje	Kölcsönzési idő mérése, terheléses teszt	Adatpontosság ellenőrzése gépnaplóval
Validáció	Próbaüzem valódi műszakban, operátori visszajelzés	Próbaidőszak egy városrészben, felhasználói tesztek	Próbaüzem egy gépcsoporton, gépkezelői tapasztalatok
Üzemeltetés	OEE-követés, karbantartás	Kihasználtság, áthelyezés, javítás	Folyamatos fejlesztés, OEE- és átfutásiidő-trendek

Merre tart a V-modell?

A V-modell logikája (meghatározni, lebontani, megvalósítani, összeépíteni, igazolni) időtálló. Megvalósítása azonban gyorsan változik. Az INCOSE 2022-ben kiadott jövőképe szerint a rendszermérnökség jövője modellalapú (INCOSE 2022). A dokumentumok helyett egy összefüggő rendszermodell hordozza a követelményeket, az architektúrát és az ellenőrzési eredményeket, így a V vízszintes kapcsolatai automatikusan nyomon követhetők. Ehhez kapcsolódik a digitális iker, amely az üzemeltetési adatokat visszavezeti a tervezésbe, és ezzel „bezárja” a V-t. Szintén ide tartozik a folyamatos integráció és telepítés, amely a jobb ág ellenőrzéseinek egy részét automatizálja és naponta futtatja. A nagy nyelvi modellek új lehetőségeket nyitnak a követelmények minőségének ellenőrzésében és a dokumentumok közötti nyomonkövethetőség feltárásában. Ezek eredményei azonban ugyanúgy verifikálandók, mint bármely más mérnöki munkatermék. A modellalapú megközelítést a 3. fejezet tárgyalja.

Összefoglalás

- A rendszert a teljes életciklusára tervezzük a koncepciótól a kivonásig. A korai döntések rögzítik az életciklusköltség nagy részét, és a változtatás költsége a projekt előrehaladtával nagyságrendekkel nő.
- Az életciklus-modell azt írja le, milyen sorrendben és visszacsatolásokkal haladunk. A vízesés egyszerű, de merev. A V-modell az ellenőrzés korai tervezésére épül. A spirálmodell a kockázatokat állítja a középpontba. Az agilis fejlesztés rövid, működő eredményt adó ciklusokkal dolgozik. A gyakorlatban gyakori a hibrid megoldás.
- A V-modell bal ága a problémától a részletes tervig halad, a csúcsa a megvalósítás, a jobb ága a komponens-verifikációtól az üzemeltetésig tart. Minden bal ági szinthez tartozik egy jobb ági ellenőrzés, amelynek tervét már a meghatározáskor elkészítjük.
- A verifikáció azt igazolja, hogy jól építettük meg a terméket, a validáció azt, hogy a jó terméket építettük meg.
- A V rekurzív: a rendszer minden szintjén újra lejátszódik. Iteratív is: egy szinten belül többször bejárjuk. A felülvizsgálatok és az alapvonalak teszik a haladást ellenőrizhetővé és a változtatásokat kezelhetővé.

Önellenőrzés

1. Mit jelent az, hogy a tervezés a költségek nagy részét „elkötelezi”, de csak kis részét költi el? Adj rá saját példát!
2. Milyen projektjellemzők szólnak a spirálmodell mellett a V-modellel szemben?
3. Egy okosotthon-termosztát egyik követelménye: „A beállított hőmérsékletet ±0,5 °C pontossággal kell tartania.” Írj egy verifikációs és egy validációs kérdést ehhez a termékhez!

4. Miért nevezzük a V-modell vízszintes kapcsolatait korai tesztterveknek? Mi a haszna annak, ha a verifikáció módját a követelmény megfogalmazásakor rögzítjük?
5. Mi a különbség a rekurzió és az iteráció között a rendszermérnöki folyamatban?
6. Mi az alapvonal, és miért van rá szükség?

Megoldás.

1. A tervezési döntések (anyagválasztás, architektúra, gyártási technológia) meghatározzák a későbbi gyártási, üzemeltetési és karbantartási költségeket, bár maguk viszonylag olcsók. Példa: ha egy kerékpárdokkoló házát öntött alumíniumból tervezzük, az öntőszerszám, a darabár és a javítás módja is eldől, mielőtt egyetlen darab elkészülne.
2. Új, kipróbálatlan technológia; nagy műszaki vagy piaci bizonytalanság; olyan kockázatok, amelyek csak prototípussal vagy kísérlettel csökkenthetők; változó vagy nehezen megfogható igények.
3. Verifikáció: klímakamrában, adott hőterhelés mellett a termosztát a beállított értéket ±0,5 °C-on belül tartja-e? Validáció: a lakók egy téli hónap alatt kényelmesnek érzik-e a lakást, és képesek-e a programozást segítség nélkül beállítani?
4. Mert a bal ági meghatározással egy időben dől el, hogyan igazoljuk majd a jobb ágon. Ha a verifikáció módja nem rögzíthető, a követelmény jellemzően nem elég pontos vagy nem mérhető. A korai rögzítés ezért a követelmények minőségét is javítja, és időben jelzi a szükséges tesztberendezéseket.
5. Rekurzió: ugyanazokat a folyamatokat a termékszerkezet egymás alatti szintjein (rendszer, alrendszer, komponens) újra alkalmazzuk. Iteráció: egy szinten belül többször végigmegyünk a folyamatokon, amíg a követelmények és a terv összhangba nem kerülnek.
6. A rendszerleírás jóváhagyott állapota, amely csak szabályozott változáskezeléssel módosítható. Ennek köszönhetően mindenki ugyanarról a változatról beszél, és a változtatások hatása az alapvonalhoz mérhető.

Projektlépés: a projekt V-terve

Válassz egy fejlesztési feladatot a Gyakorlati útmutató projektjavaslatai közül, vagy hozd a sajátodat. Készítsd el hozzá a projekt V-tervét:

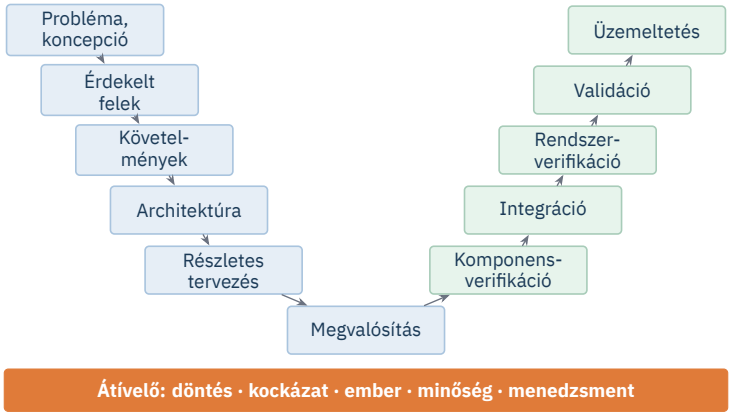
- rajzold fel a V-t a saját rendszeredre;
- minden lépéshez írd fel a várható munkaterméket;
- minden bal ági lépés mellé írd fel, hogyan fogod a jobb ágon ellenőrizni.

Jelölj ki legalább három felülvizsgálati pontot belépési feltételekkel. A részletes útmutatót és a sablont a Gyakorlati útmutató *Projekt V-terve* eszközlapja tartalmazza.

Továbbolvasás

- INCOSE (2023b), 2. fejezet: az életciklus-fogalmak és -modellek részletes, szabványokra épülő tárgyalása.
- ISO/IEC/IEEE (2024): az életciklus-szakaszok és -modellek szabványos kerete.
- SEBoK Editorial Board (2025): az életciklus-modellekről és a felülvizsgálatokról szóló, szabadon elérhető fejezetek.
- Forsberg és Mooz (1991): a rendszermérnöki V-modell eredeti cikke.
- B. W. Boehm (1988): a kockázatvezérelt spirálmodell klasszikus leírása.
- NASA (2016), 2. és 3. fejezet: a rendszermérnöki motor és a projektfázisok gyakorlati leírása.

3. Modellek és modellalapú rendszermérnökség



Tanulási eredmények

A fejezet feldolgozása után képes leszel:

- megfogalmazni, mire valók a modellek a rendszermérnökségben, és miért közelítés minden modell;
- a feladathoz illő modelltípust választani (fizikai, fogalmi, matematikai, szimulációs, adatvezérelt);
- kontextusdiagrammal kijelölni egy rendszer határát, és rendszerezni a külső interfészeit;
- megkülönböztetni a modell verifikációját és validációját;
- összevetni a dokumentumközpontú és a modellalapú rendszermérnökséget, és értelmezni a nézőpont és a nézet fogalmát;
- megnevezni a SysML diagramtípusait és azt, hogy melyik mire való, valamint értelmezni egy egyszerű SysML v2 szöveges modellt;
- megkülönböztetni a digitális modellt, a digitális árnyékot és a digitális ikret.

Az előző fejezet rendszerekről szólt. Ez arról szól, hogyan gondolkodunk róluk, és hogyan beszélünk róluk egymással. A rendszermérnök nem tud egy még nem létező rendszert kipróbálni, és egy létező, nagy rendszeren sem kísérletezhet szabadon. Ezért modelleket használ: egyszerűsített leképezéseket, amelyeken kérdéseket tehet fel, és amelyek segítségével a csapat közös képet alakíthat ki a rendszerről. A V-modell bal ágán a modellek a tervezés eszközei, a jobb ágán az ellenőrzés mércéi: a mérési eredményeket a modell előrejelzéseivel vetjük össze.

A fejezet négy kérdésre keresi a választ:

- Mire jó egy modell, és hogyan válasszuk ki a feladathoz illőt?
- Hogyan jelöljük ki a rendszer határát, és hogyan írjuk le a környezetével fennálló kapcsolatait?
- Honnan tudjuk, hogy egy modellben megbízhatunk?

- Mit jelent a modellalapú rendszermérnökség, a SysML és a digitális iker?

3.1. Miért modellezünk?

A **modell** egy rendszer céltudatosan egyszerűsített leképezése, amely egy meghatározott kérdés megválaszolását szolgálja. A meghatározás két kulcsszava a *céltudatos* és az *egyszerűsített*. Nincs „a” modellje egy rendszernek, csak egy adott kérdéshez illő modellje van. A KoboCella háromdimenziós CAD-modellje alkalmas arra, hogy ellenőrizzük, elér-e a robot minden csavarhoz, de semmit sem mond arról, hány motort szerel össze a cella egy műszakban. Erre egy egyszerű folyamatmodell vagy szimuláció való.

George Box statisztikus sokat idézett megállapítása szerint minden modell hamis, de vannak hasznos modellek (Box 1979). A modell attól hasznos, hogy elhagyja a lényegtelent. Ha egy térkép ugyanolyan részletes lenne, mint a terep, nem lehetne vele tájékozódni. A modellező fő kérdése ezért nem az, hogy a modell „igaz-e”, hanem az, hogy elég jó-e a feltett kérdés megválaszolására.

Epstein sokat idézett írásában az előrejelzés mellett tizenhat további okot sorol fel arra, miért érdemes modellezni (Epstein 2008). A rendszermérnök szempontjából ezek közül négy a legfontosabb:

- **Struktúrában gondolkodunk.** A modellalkotás arra kényszerít, hogy azonosítsuk az elemeket, feltárjuk a kapcsolatokat, és kimondjuk a feltételezéseinket.
- **Megalapozottabban döntünk.** A modellel kiszámíthatjuk, mi történne különböző döntések esetén, és összevethetjük a változatokat, mielőtt bármit megépítenénk.
- **Termékenyebbek leszünk.** A modell olyan kérdéseket is felvet, amelyekre nem gondoltunk, és olyan összefüggéseket is megmutat, amelyek intuitíve nem nyilvánvalók: egyensúlyi, ciklikus, véletlenszerű vagy komplex viselkedést.
- **Óvatosságra intenek.** A modell feltárja a logikai korlátokat, és megmutatja, hol bizonytalan a tudásunk.

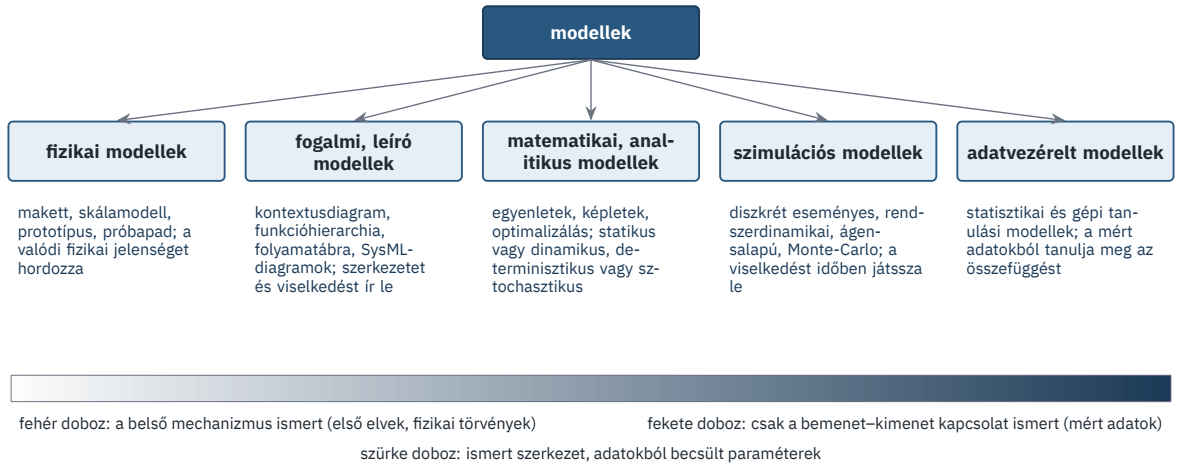
Ehhez jön egy ötödik, a rendszermérnökségben talán a legfontosabb szerep: a **kommunikáció**. Egy jó kontextusdiagram vagy állapotgép-diagram percek alatt közös képet teremt egy gépész, egy programozó, egy ergonómus és egy beszerző között. Szöveggel ehhez oldalakra volna szükség.

> **Modell, szimuláció, digitális iker**
>
> - **Modell:** egy rendszer céltudatosan egyszerűsített leképezése egy kérdés megválaszolására.
> - **Szimuláció:** a modell működtetése, jellemzően időben, hogy megfigyeljük a viselkedését. A szimuláció tehát nem modelltípus, hanem a modell használatának egy módja.
> - **Digitális iker:** a valós rendszerrel automatikus, kétirányú adatkapcsolatban álló digitális modell (3.7. szakasz).

3.2. Modelltípusok

A rendszermérnöki gyakorlatban öt modellcsaládot érdemes megkülönböztetni (3.1. ábra).

- **Fizikai modellek:** makettek, skálamodellek, prototípusok, próbapadok. Előnyük, hogy a valódi fizikai jelenséget hordozzák, így olyan hatásokat is megmutatnak, amelyekre nem gondoltunk. Hátrányuk, hogy drágák, és lassan módosíthatók.
- **Fogalmi, leíró modellek:** kontextusdiagramok, funkcióhierarchiák, folyamatábrák, SysML-diagramok. A rendszer szerkezetét, viselkedését és kapcsolatait írják le, és elsősorban a megértést és a kommunikációt szolgálják.
- **Matematikai, analitikus modellek:** egyenletek és képletek, amelyek a rendszer jellemzői közötti összefüggést írják le. Lehetnek statikusak vagy dinamikusak, determinisztikusak vagy sztochasztikusak, folytonosak vagy diszkrétek. Ide tartoznak a megbízhatósági blokkdiagramok és a sorbanállási képletek is.
- **Szimulációs modellek:** a rendszer viselkedését időben játsszák le. A gyártásban és a logisztikában a diszkrét eseményes szimuláció terjedt el (Law 2015). A visszacsatolásokkal teli, stratégiai szintű kérdésekhez a rendszerdinamikai modellek valók (Sterman 2000). A sok önálló döntéshozó viselkedéséből kibontakozó mintázatokhoz az ágensalapú modellek, a bizonytalanság terjedésének vizsgálatához a Monte-Carlo-szimuláció.
- **Adatvezérelt modellek:** statisztikai és gépi tanulási modellek, amelyek mért adatokból tanulják meg az összefüggéseket. Akkor hasznosak, ha a mechanizmus túl bonyolult az első elvekből történő leíráshoz, de van elég jó minőségű adat.



3.1. ábra. A modellek fő családjai és a fehér doboz–fekete doboz skála. A fehér doboz modell a belső mechanizmusra, a fekete doboz modell a mért bemenet–kimenet kapcsolatra épül.

A modelleket aszerint is osztályozhatjuk, hogy mennyit tudunk a modellezett rendszer belső működéséről. Karplus klasszikus elemzése szerint a modellek egy spektrumon helyezkednek el. A spektrum egyik végén a jól ismert fizikai törvényekre épülő „fehér doboz” modellek állnak (villamos és mechanikai rendszerek), a másik végén a csak megfigyelt adatokra épülő „fekete doboz” modellek (társadalmi és gazdasági rendszerek). A spektrum fekete doboz végén a modellek legfeljebb spekulációra és előrejelzésre alkalmasak, a fehér doboz felé haladva pedig elemzésre, irányításra, végül tervezésre is (Karplus 1977). A Lakatos-jegyzet ugyanezt a spektrumot használja a rendszerek modellezhetőségének jellemzésére (Lakatos, é. n.a). A kettő között vannak a **szürke doboz** modellek: ezek szerkezete első elvekből adódik, paramétereiket viszont mért adatokból becsüljük.

A modellválasztás a kérdésből indul ki, nem az eszközből. A 3.1. táblázat a futó esetek néhány tipikus kérdéséhez rendel modellt.

3.1. táblázat. Kérdések és modelltípusok a futó esetekben.

Kérdés	Modelltípus	Miért ez?
Elér-e a robot minden csavarhoz, és ütközik-e az operátorral? (A)	fizikai és geometriai (CAD, robotszimulátor)	térbeli viszonyokról szól; a kinematika ismert
Mekkora a cella ciklusideje az adott munkamegosztással? (A)	analitikus (idődiagram, kritikus út)	determinisztikus műveleti idők összege és átfedése
Mennyit termel a cella egy műszakban, ha a robot időnként leáll? (A)	diszkrét eseményes szimuláció	véletlen leállások és pufferek kölcsönhatása
Kik és milyen áramlásokkal kapcsolódnak a rendszerhez? (C)	fogalmi (kontextusdiagram)	a cél a határ és az interfészek tisztázása
Hogyan alakul a felhasználószám öt év alatt? (B)	rendszerdinamikai	visszacsatolások és késleltetések uralják
Hány kölcsönzés várható egy új állomáson? (B)	adatvezérelt (regresszió hasonló állomások alapján)	a mechanizmus összetett, de van adat
Milyen hatással jár egy MES-bevezetés a határidő-tartásra? (C)	folyamatmodell és szimuláció	sorbanállás, ütemezés, információs késleltetés

Az eszköz választja a modellt

A kezdő modellező gyakran azzal a modellel dolgozik, amelyik eszközt ismeri. Aki szimulációs programot tanult, mindent szimulálni akar. Aki táblázatkezelőben jártas, mindent táblázatban számol. A jó kérdés mindig az, hogy mi a legegyszerűbb modell, amely a feltett kérdésre elég pontos választ ad. Egy borítékhátoldali becslés sokszor többet ér egy hetekig épített szimulációnál, ha a döntéshez elég a nagyságrend.

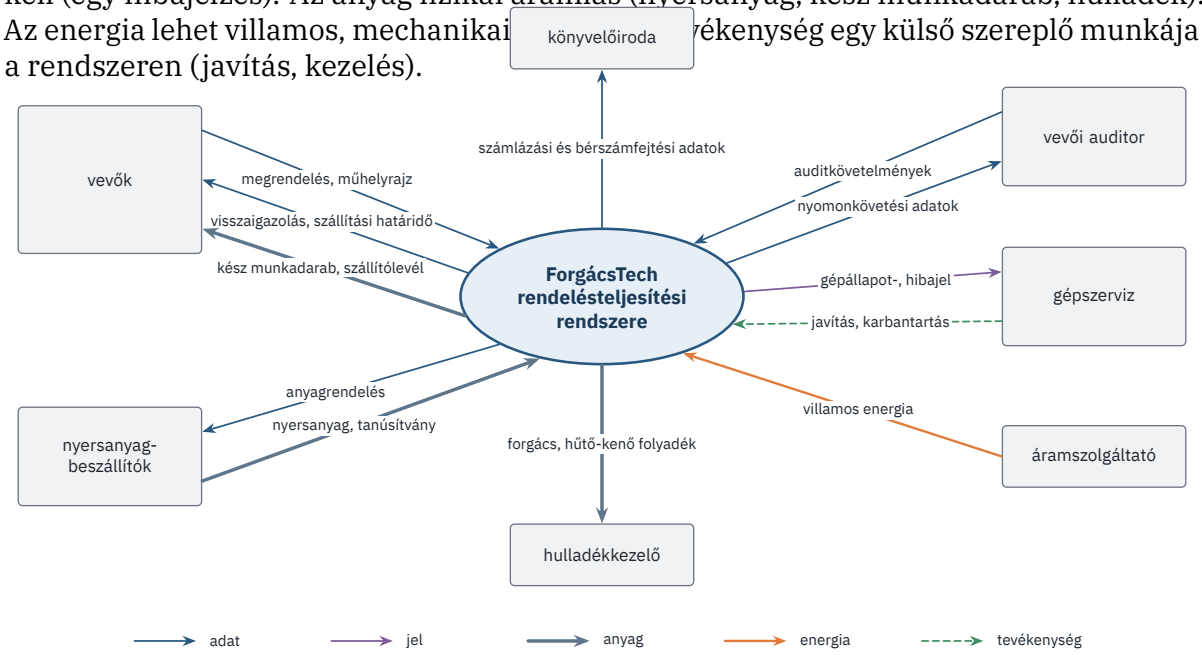
3.3. Rendszerhatár és kontextusdiagram

A legelső és sokszor a legfontosabb modell a **kontextusdiagram**. Kossiakoff és szerzőtársai a rendszermérnök egyik legfontosabb kommunikációs eszközének tartják: megmutatja a rendszer környezetében lévő külső entitásokat és a rendszerrel folytatott kölcsönhatásaikat. A rendszert fekete dobozként ábrázolja, vagyis egyetlen alakzatként középen, belső felépítését elrejtve (Kossiakoff és mtsai. 2011, 3.4. szakasz). A diagram három összetevőből áll:

1. **Külső entitások:** minden szereplő, rendszer vagy objektum, amellyel a rendszer kapcsolatba lép. Ezek a bemenetek forrásai és a kimenetek címzettjei.
2. **Kölcsönhatások:** nyilak, amelyek iránya az áramlás irányát mutatja. Minden nyilat meg kell címkézni azzal, ami a határon átmegy. Kossiakoff szerint az egyirányú nyilak egyértelműbbek, és az olyan semmitmondó címkéket, mint az „adat” vagy a „kommunikáció”, kerülni kell.
3. **A rendszer:** egyetlen alakzat középen, csak a rendszer nevével.

A határon áthaladó kölcsönhatások öt kategóriába sorolhatók: **adat, jel, anyag, energia és tevékenység** (Kossiakoff és mtsai. 2011, 3.4. szakasz). Az adat feldolgozandó információ

(egy megrendelés). A jel valamilyen állapotot vagy eseményt hordoz, amelyre reagálni kell (egy hibajelzés). Az anyag fizikai áramlás (nyersanyag, kész munkadarab, hulladék). Az energia lehet villamos, mechanikai [illegible]vékenység egy külső szereplő munkája a rendszeren (javítás, kezelés).



3.2. ábra. A ForgácsTech rendelésteljesítési rendszerének kontextusdiagramja. A nyilak színe és stílusa a kölcsönhatás típusát jelöli.

Kidolgozott példa: a ForgácsTech kontextusdiagramja

A 3.2. ábra a ForgácsTech rendelésteljesítési rendszerének kontextusdiagramja. Elkészítése négy lépésből állt.

1. A rendszer megnevezése. A csapat szándékosan nem „MES-nek” nevezte a rendszert, hanem rendelésteljesítési rendszernek. A szoftver ugyanis csak egy eleme a rendszernek, amelyhez a gépek, az emberek és az eljárások is hozzátartoznak (1.1. szakasz).

2. A külső entitások összegyűjtése. A csapat az érdekeltfél-listából (5. fejezet) és a meglévő folyamatok bejárásából indult ki. Hét külső entitás maradt: vevők, nyersanyag-beszállítók, könyvelőiroda, vevői auditor, gépszerviz, áramszolgáltató és hulladékkezelő.

3. A kölcsönhatások felrajzolása és típusba sorolása. Minden kapcsolatnál megkérdezték, mi megy át a határon, milyen irányba, és milyen típusú. A „kommunikáció a vevővel” címkéből így három nyíl lett: a megrendelés és a műhelyrajz befelé, a visszaigazolás a szállítási határidővel kifelé, a kész munkadarab a szállítólevéllel szintén kifelé.

4. Ellenőrzés. A csapat végignézte, van-e olyan entitás, amely csak ad, de nem kap semmit, vagy fordítva, és hogy van-e olyan bemenet, amelyből nem lesz kimenet. Kiderült, hogy a hulladékkezelő eredetileg kimaradt. A forgács és a hűtő-kenő folyadék elszállítása pedig jogszabályi nyilvántartási kötelezettséggel jár, így a rendszer információs feladatai közé is bekerült.

A diagram egyetlen oldalon mutatja meg, mivel kell a rendszernek kapcsolatban állnia, és mi nem a fejlesztés tárgya. A MES-szállítóval folytatott első egyeztetésen ez lett a beszélgetés kiindulópontja.

3.3.1. Interfészek és interfészelemek

A kontextusdiagram nyilai a **külső interfészeket** mutatják. A rendszer belsejében, az alrendszerek között ugyanígy **belső interfészek** vannak. Kossiakoff és szerzőtársai az interfészek rendszerezett azonosításához három interfészelem-típust különböztetnek meg (Kossiakoff és mtsai. 2011, 3.5. szakasz):

- **összekötők** (connectors), amelyek lehetővé teszik az áram, a folyadék, az erő és más hatások átvitelét a komponensek között;
- **szigetelők** (isolators), amelyek gátolják az ilyen kölcsönhatásokat;
- **átalakítók** (converters), amelyek megváltoztatják a kölcsönhatást hordozó közeg formáját.

3.2. táblázat. Interfészelemek a KoboCella példáján, Kossiakoff és szerzőtársai rendszerezése nyomán.

Közeg	Összekötő	Szigetelő	Átalakító
villamos	a cellavezérlés kábelei, csatlakozói, kapcsolói	a robot vezérlőszekrényének EMC-árnyékolása	a nyomatékérzékelő analóg–digitális átalakítója
mechanikai	a megfogó gyorscserélő tengelykapcsolója	a munkaasztal rezgéscsillapító lábai	a csavarorsó bolygóműves hajtása
pneumatikus	a megfogó sűrítettlevegő-tömlője és szelepe	a hengerek tömítései	a nyomáscsökkentő szelep
ember–gép	a kezelőpanel és a jelzőlámpa	az átlátszó védőburkolat	az érintőképernyő mint beviteli eszköz

Az interfészelemek egyszerűsége megtévesztő. Kossiakoff és szerzőtársai hangsúlyozzák, hogy a tapasztalatok szerint a rendszerhibák jelentős része az interfészeken keletkezik, és kifejezetten a rendszermérnök felelőssége, hogy az interfészek kompatibilisek és megbízhatók legyenek (Kossiakoff és mtsai. 2011, 3.5. szakasz). A belső interfészek rendszerezett feltárásával, az N²-diagrammal és a tervezési struktúramátrixszal a 8. fejezet, az interfészek kezelésével a 22. fejezet foglalkozik.

> **KoboCella: az elfelejtett ember–gép interfész**
>
> A KoboCella első interfészlistája huszonhárom tételt tartalmazott. Volt benne minden kábel, tömlő és adatkapcsolat a robot, a csavarozó, a biztonsági vezérlés és a MES között. Az operátor egyetlen sorban szerepelt: „HMI – érintőképernyő”. Az ergonómus kérdésére derült ki, hogy az operátor valójában hat különböző módon lép kapcsolatba a cellával. Behelyezi az alkatrészt (anyag), nyugtázza a ciklus indítását (jel), leolvassa a hibakódot (adat), kiveszi és szemrevételezi a kész motort (anyag és tevékenység), feltölti a csavaradagolót (anyag), és vészleállítást kezdeményezhet (jel). Mind a hatnak saját követelményei lettek: elérési távolság, kijelzési idő, kezelőszerv-elhelyezés. Ezekkel az emberközpontú tervezés fejezete (11. fejezet) foglalkozik.

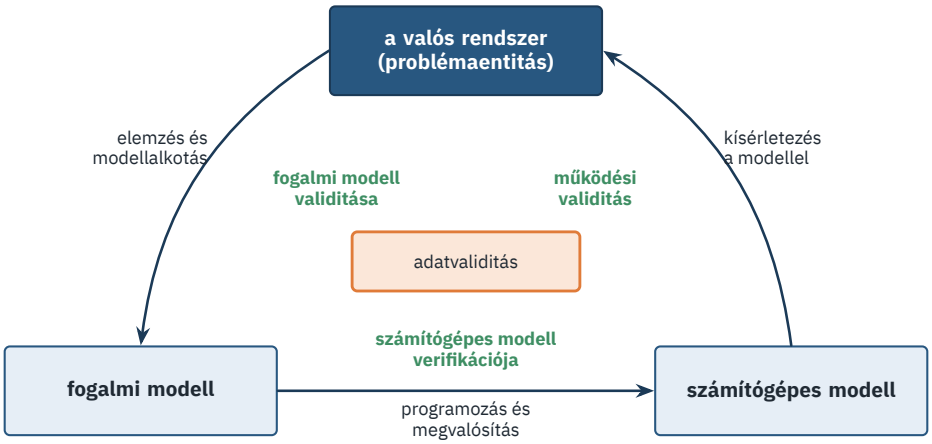
3.4. Megbízhatunk-e a modellben?

Egy modell csak akkor segíti a döntést, ha megbízhatunk benne. A modellek hitelességét ugyanazzal a két kérdéssel vizsgáljuk, mint magát a rendszert (2. fejezet):

- **A modell verifikációja:** helyesen építettük-e meg a modellt? A számítógépes modell azt csinálja-e, amit a fogalmi modell előír, azaz nincs-e programozási, képlet- vagy beállítási hiba?
- **A modell validációja:** a megfelelő modellt építettük-e meg? A modell a felhasználási céljához elegendő pontossággal írja-e le a valós rendszert?

Sargent széles körben használt keretrendszere a modellalkotás folyamatát három elem kapcsolataként írja le: a valós rendszer (a problémaentitás), a fogalmi modell és a számítógépes modell. Ezekhez négy ellenőrzési feladat tartozik (3.3. ábra) (Sargent 2013):

- **a fogalmi modell validitása:** a fogalmi modell feltételezései és elméletei helytállóak, és a modell szerkezete, logikája, okozati viszonyai a felhasználási célhoz megfelelőek;
- **a számítógépes modell verifikációja:** a fogalmi modell programozása és megvalósítása helyes;
- **a működési validitás:** a modell kimenete a felhasználási céljának megfelelő pontosságú a modell alkalmazási tartományában;
- **az adatvaliditás:** a modell építéséhez, értékeléséhez és a kísérletekhez használt adatok megfelelőek és helyesek.



3.3. ábra. A modell verifikációja és validációja Sargent egyszerűsített keretrendszere nyomán.

Sargent egy fontos megszorítást is tesz: egy modell validitását mindig egy adott felhasználási célhoz és egy adott alkalmazási tartományhoz viszonyítva kell megítélni (Sargent 2013). A KoboCella szimulációs modellje lehet érvényes az évi 90–140 ezer darabos tartományban, de semmit sem mond arról, mi történne évi 300 ezer darabnál, amikor egészen más szűk keresztmetszetek lépnének fel.

KoboCella: a szimuláció validálása

A KoboCella diszkrét eseményes szimulációs modelljét a robotintegrátor készítette. A rendszermérnök három lépésben ellenőrizte.

1. **Verifikáció.** Determinisztikus esetben, leállások nélkül, kézzel kiszámolta a várható ciklusidőt, és összevetette a szimuláció eredményével. Egy elírás miatt a csavarozási idő a modellben kétszer szerepelt; ezt kijavították.
2. **Adatvaliditás.** A robot leállási adatait az integrátor egy másik ügyfelénél működő, hasonló cella egyéves naplójából vette. A rendszermérnök ellenőrizte, hogy ott is ugyanaz a csavarozó típus és hasonló darabszám volt-e. A darabszám jóval kisebb volt, ezért a leállási gyakoriságot érzékenységvizsgálattal, a duplájára növelve is kipróbálták.
3. **Működési validitás.** A kézi munkaállomás jelenlegi teljesítményét a modell egyszerűsített változatával is lefuttatták, és összevetették a MES-ből kinyert valós termelési

adatokkal. Az eltérés 3% alatt maradt, ami a beruházási döntéshez elegendő pontosság.

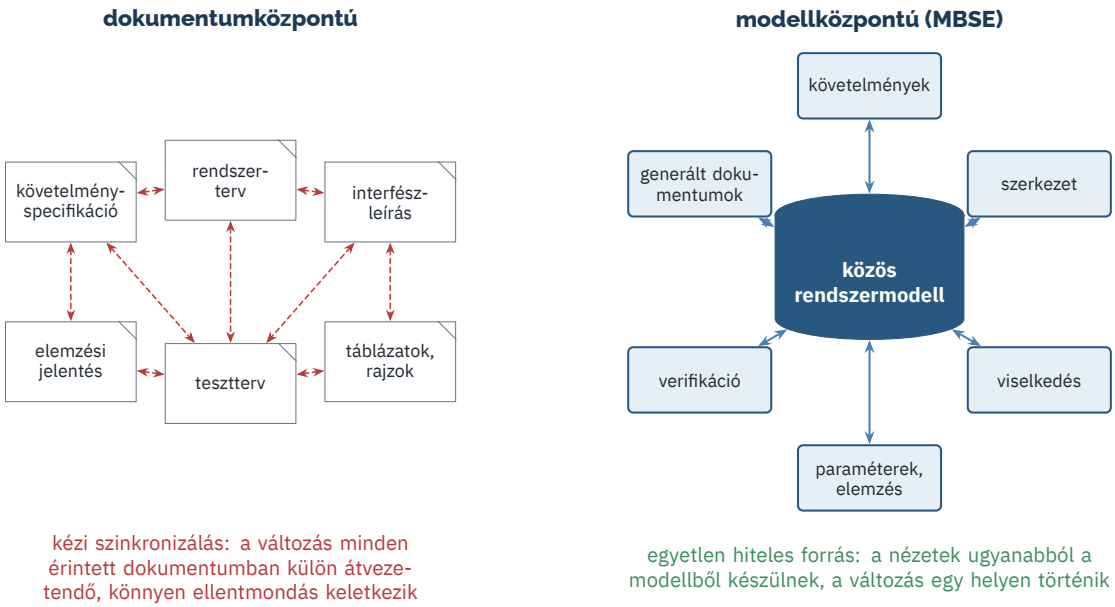
Szép eredmény, ellenőrizetlen modell

A modellek eredményei, különösen a színes szimulációs animációk és a sok tizedesjegyű számok, hamis biztonságérzetet keltenek. A modell nem lesz pontosabb attól, hogy szép. Minden modelleredmény mellé oda kell írni, milyen feltételezésekkel, milyen adatokkal és milyen tartományban érvényes. A modell bemeneti adatainak minőségénél soha nem lehet jobb az eredmény.

3.5. A dokumentumoktól a modellekig: MBSE

A rendszermérnökség hagyományosan dokumentumokra épült. A követelményspecifikáció, a rendszerterv, az interfészleírás, az elemzési jelentés és a teszterv külön-külön dokumentum, amelyeket kézzel kell összhangban tartani. Ha egy követelmény megváltozik, a változást minden érintett dokumentumban át kell vezetni. Nagy rendszereknél ez szinte lehetetlen, és az ellentmondások a későbbi fázisokban drága hibákká válnak (3.4. ábra).

Az INCOSE meghatározása szerint a **modellalapú rendszermérnökség** (model-based systems engineering, MBSE) a modellezés formalizált alkalmazása a rendszerkövetelmények, a tervezés, az elemzés, a verifikáció és a validáció támogatására, a koncepciótervezési fázistól kezdve a fejlesztésen át a későbbi életciklus-szakaszokig (INCOSE 2007). A lényeg a *formalizált* szóban van. Rajzok mindig is voltak. Az MBSE-ben azonban a rendszerről szóló információ egyetlen, egységes szabályok szerint felépített, összefüggő modellben él, és a dokumentumok, diagramok, táblázatok ennek a modellnek a *nézetei*.



3.4. ábra. Dokumentumközpontú és modellközpontú rendszermérnökség. A modellközpontú megközelítésben a különböző nézetek egyetlen közös modellből készülnek.

Az MBSE várható előnyei a következők:

- **Egyetlen hiteles forrás.** A rendszer minden eleme egyszer szerepel a modellben, a nézetek ebből készülnek, így nem kerülhetnek ellentmondásba.
- **Hatáselemzés.** Mivel a követelmények, a funkciók, a szerkezeti elemek és a tesztek össze vannak kapcsolva, egy változás hatása automatikusan nyomon követhető.
- **Korai ellenőrzés.** A modell egy része végrehajtható vagy elemezhető. Az ellentmondások, a hiányzó interfészek és a teljesítményproblémák már a tervezés során kiderülhetnek.
- **Jobb kommunikáció.** Minden érdekelt fél a számára releváns nézetet látja, de mindenki ugyanarról a rendszerről.

Az MBSE nem ingyenes. Modellezési nyelvet és eszközt kell megtanulni, modellezési szabályokat kell kialakítani, és a modellt karban kell tartani. Kis projekteknél a jól szervezett táblázatok és diagramok is betölthetik ugyanezt a szerepet, ha következetesen kezelik őket. Ezt a megközelítést követi a Gyakorlati útmutató is: a sablonok összekapcsolt táblázatai egy egyszerű, de valódi rendszermodellt alkotnak.

3.5.1. Nézőpontok és nézetek

Egy összetett rendszert nem lehet egyetlen ábrán teljes egészében bemutatni. Az architektúraleírás nemzetközi szabványa, az ISO/IEC/IEEE 42010:2022 ezért különbséget tesz a rendszer **architektúrája** (a rendszer alapvető fogalmai és tulajdonságai) és az **architektúraleírás** között, amely az architektúrát kifejező munkatermék (ISO/IEC/IEEE 2022). Az architektúraleírás **nézetekből** áll. Minden nézet egy **nézőpont** szerint készül, amely megadja, mely érdekelt felek mely kérdéseire (aggályaira) ad választ, és milyen modellezési technikákkal.

A KoboCella architektúraleírásában például a következő nézetek szerepelhetnek:

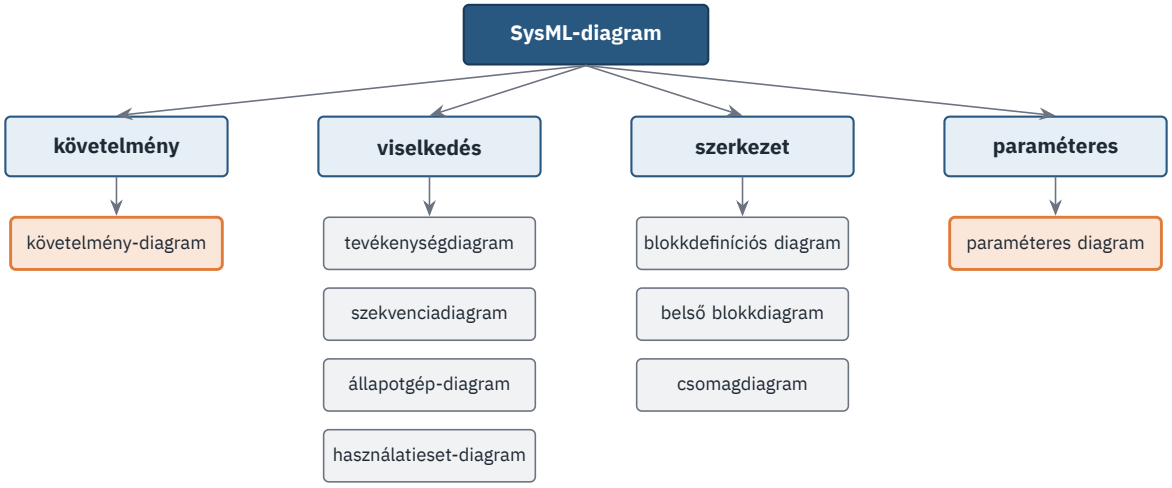
- **funkcionális nézet** a rendszermérnöknek és a robotintegrátornak: milyen funkciókat lát el a cella, és ezeket ki végzi;
- **biztonsági nézet** a munkavédelemnek: hol és hogyan érintkezhet az ember a robottal, milyen védelmi funkciók vannak;
- **kezelői nézet** az operátoroknak: mit lát, mit csinál, mire reagál az operátor egy ciklus során;
- **adatnézet** az informatikának: milyen adatokat küld a cella a MES-nek, milyen gyakran és milyen formátumban.

A nézetek nem független dokumentumok. Ugyanazon modell különböző metszetei, ezért ha a funkciómegosztás megváltozik, a biztonsági, a kezelői és az adatnézet is ennek megfelelően változik.

3.6. A SysML nyelv

A modellalapú rendszermérnökség legelterjedtebb nyelve a **SysML** (Systems Modeling Language). Az első változatot az Object Management Group (OMG) az INCOSE-zal együttműködve, a szoftverfejlesztésben használt UML nyelvre építve dolgozta ki. A SysML 1.x kilenc diagramtípust határoz meg négy csoportban (3.5. ábra) (Friedenthal, Moore, és Steiner 2014):

- **követelménydiagram:** a követelmények és kapcsolataik (levezetés, teljesítés, verifikáció);
- **viselkedési diagramok:** tevékenységdiagram (folyamatok, funkciók áramlása), szekvenciadiagram (üzenetváltások időrendje), állapotgép-diagram (állapotok és átmenetek), használatieset-diagram (a felhasználók céljai);
- **szerkezeti diagramok:** blokkdefiníciós diagram (a rendszer elemei és hierarchiája), belső blokkdiagram (az elemek közötti kapcsolatok és interfészek), csomagdiagram (a modell szervezése);
- **paraméteres diagram:** a rendszer jellemzői közötti matematikai összefüggések, amelyek mérnöki elemzésekhez kapcsolhatók.



A SysML 1.x kilenc diagramtípusa. Narancs: a SysML-nek az UML-hez képest új diagramtípusai. A SysML v2 ugyanezeket a nézeteket egységes szemantikájú modellből, grafikus és szöveges jelöléssel, szabványos programozói felülettel (API) állítja elő.

3.5. ábra. A SysML 1.x diagramtípusai Friedenthal és szerzőtársai rendszerezése nyomán.

3.6.1. SysML v2

A SysML 1.x egyik gyengesége, hogy szemantikája nem teljesen formális, ezért a különböző eszközökben készült modellek nehezen cserélhetők és elemezhetők. Az OMG 2025 júliusában véglegesen elfogadta a nyelv új, második főváltozatát. A SysML v2 egy új, alkalmazásfüggetlen alapnyelvre, a KerML-re (Kernel Modeling Language) épül, és egy szabványos programozói felületet (Systems Modeling API and Services) is meghatároz. A formális specifikációkat 2025 szeptemberében tették közzé (OMG 2025). A SysML v2 legfontosabb újdonságai:

- **formális szemantika:** a modell jelentése egyértelmű, ezért a modellek gépileg ellenőrizhetők és elemezhetők;
- **szöveges és grafikus jelölés:** ugyanaz a modell szövegként is írható és olvasható, így verziókezelő rendszerben ugyanúgy kezelhető, mint a programkód;
- **szabványos API:** a modell programokból (például Pythonból) is lekérdezhető és módosítható, így szimulációs és elemző eszközökhöz kapcsolható.

A következő szöveges SysML v2 modell a KoboCella néhány elemét és egy követelményt írja le:

```
package KoboCella {
    private import ScalarValues::Real;

    part def Cella {
        part kobot : Kobot;
        part csavarozo : CsavarozoEgyseg;
        part operator : Operator;
        attribute ciklusido : Real;      // másodperc
    }
    part def Kobot;
    part def Operator;
    part def CsavarozoEgyseg {
        attribute nyomatek : Real;       // newtonméter
    }

    requirement def CiklusidoKovetelmeny {
        doc /* A cella ciklusideje legfeljebb 75 s. */
        subject c : Cella;
        require constraint { c.ciklusido <= 75.0 }
    }
}
```

A modellben a `part def` egy rendszerelem típusát (definícióját) adja meg, a `part` egy ilyen típusú részt, az `attribute` egy jellemzőt, a `requirement def` pedig egy követelményt, amelynek alanya a cella, és amely a ciklusidőre korlátot ír elő. Ebből a szöveges leírásból a modellező eszköz blokkdefiníciós diagramot és követelménydiagramot is elő tud állítani.

A SysML mellett más megközelítések is elterjedtek. Az Arcadia-módszer és a hozzá tartozó nyílt forráskódú Capella eszköz például saját, a műveleti elemzéstől a fizikai architektúráig vezető nézetrendszert kínál (Voirin 2017). A módszer és a nyelv megválasztásánál a legfontosabb szempont az, hogy a csapat és a partnerek ugyanazt használják, és a modellezésnek legyen egyértelmű célja.

> **Nagy nyelvi modellek a modellezésben**
>
> A nagy nyelvi modellek (LLM-ek) szöveges leírásokból kontextusdiagram-vázlatot, interfészlistát, sőt SysML v2 vagy BPMN szöveges modellt is tudnak javasolni. Ez gyorsítja az első vázlat elkészítését, különösen akkor, ha sok leírás, jegyzőkönyv vagy szabályzat áll rendelkezésre. A generált modell azonban csak *javaslat*. Ugyanúgy ellenőrizni kell, mint bármely más modellt. A külső entitásoknak a valóságban is létezniük kell, a nyilak irányának és tartalmának helytállónak kell lennie, és a modellnek a felhasználási céljához kell illeszkednie. A Gyakorlati útmutató mintapromptjai ezért mindig tartalmaznak ellenőrző lépést is. A modell gazdája az ember marad, a hiteles forrás pedig a csapat által jóváhagyott modell.

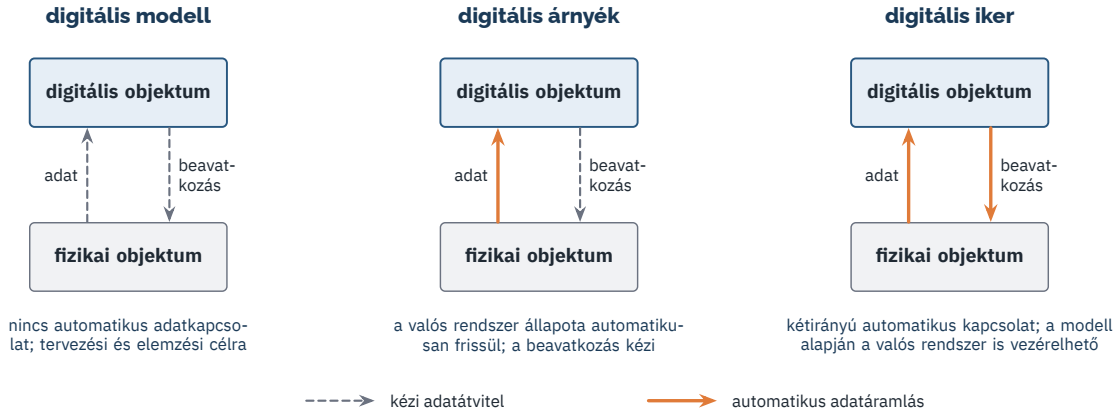
3.7. Digitális fonal és digitális iker

A modellalapú megközelítés logikus folytatása, hogy a modellek ne csak a tervezés idején éljenek, hanem az egész életcikluson át kapcsolatban maradjanak a valós rendszerrel. A **digitális fonal** (digital thread) az életciklus különböző szakaszaiban keletkező adatok és modellek összekapcsolt láncolatát jelenti. A követelményektől a terveken és a gyártási adatokon át az üzemeltetési mérésekig minden visszakövethető. Az INCOSE jövőképe a

modellalapú, digitális mérnöki munkát a szakma egyik fő fejlődési irányaként jelöli meg (INCOSE 2022).

A **digitális iker** fogalmát Grieves és Vickers írta le részletesen: virtuális információs konstrukciók együttese, amely egy tervezett vagy már létező fizikai terméket teljes körűen leír, és az életciklus során kapcsolatban áll vele. A szerzők szerint a digitális iker abban segít, hogy a nem várt és nem kívánt emergens viselkedést még a valós rendszerben való megjelenése előtt felismerjük és megelőzzük (Grieves és Vickers 2017). A kifejezést a gyakorlatban nagyon sokféle értelemben használják. Kritzinger és szerzőtársai ezért az adatkapcsolat automatizáltsága szerint három szintet különböztetnek meg (3.6. ábra) (Kritzinger és mtsai. 2018):

- **Digitális modell:** a fizikai és a digitális objektum között nincs automatikus adatkapcsolat. Az adatcsere kézi, a modell tervezési és elemzési célra szolgál.
- **Digitális árnyék:** a fizikai objektum állapota automatikusan frissül a digitális objektumban, a visszairányú beavatkozás azonban kézi.
- **Digitális iker:** az adatáramlás mindkét irányban automatikus. A digitális objektumban bekövetkező változás a fizikai objektum állapotát is módosíthatja.



3.6. ábra. A digitális modell, a digitális árnyék és a digitális iker megkülönböztetése az adatkapcsolat automatizáltsága szerint, Kritzinger és szerzőtársai nyomán.

A gyártás területén az ISO 23247 szabványsorozat ad keretrendszert a digitális ikrekhez. Első része 2021-ben jelent meg, és a gyártási digitális ikrek áttekintését, fogalmait és általános elveit tartalmazza (ISO 2021).

ForgácsTech: digitális iker vagy digitális árnyék?

A MES-szállító ajánlatában „a teljes üzem digitális ikre” szerepelt. A rendszermérnöki átvilágítás során kiderült, hogy a rendszer valójában a következőket tudja:

- a gépek állapotát (fut, áll, hibás) és a darabszámokat percenként automatikusan gyűjti, és ebből valós idejű OEE-kijelzőt készít;
- a gépek programjait és beállításait nem módosítja, a beavatkozás továbbra is a gépkezelő és a termelésvezető dolga.

Kritzinger és szerzőtársai osztályozása szerint ez **digitális árnyék**, és a ForgácsTech céljaihoz pontosan ez kell. A rendszer lerövidíti az irányítási kört (1.3. szakasz), de az ember marad a döntéshozó. Valódi digitális iker, amely például automatikusan átütemezi a gépeket, csak akkor volna indokolt, ha a gyártás jól modellezhető, nagy sorozatú és stabil lenne. A ForgácsTech egyedi és kis sorozatú gyártására ez nem igaz. A pontos szóhasználat itt nem szőrszálhasogatás: a „digitális iker” elvárás alapján a cég jóval drágább és bonyolultabb

rendszert rendelt volna meg, mint amire szüksége van.

Összefoglalás

- A modell egy rendszer céltudatosan egyszerűsített leképezése egy kérdés megválaszolására. Minden modell közelítés. A jó modell a feltett kérdéshez elég pontos és a lehető legegyszerűbb.
- Öt modellcsaládot különböztetünk meg: fizikai, fogalmi, matematikai, szimulációs és adatvezérelt modelleket. A fehér doboz modellek a mechanizmusra, a fekete doboz modellek a mért adatokra épülnek.
- A kontextusdiagram fekete dobozként ábrázolja a rendszert, és megmutatja a külső entitásokat és a velük folytatott, címkézett kölcsönhatásokat (adat, jel, anyag, energia, tevékenység). Az interfészelemek összekötők, szigetelők vagy átalakítók. A hibák jelentős része az interfészeken keletkezik.
- A modell verifikációja azt kérdezi, jól építettük-e meg a modellt, a validációja azt, hogy a megfelelő modellt építettük-e meg. A validitás mindig egy felhasználási célhoz és tartományhoz kötött.
- Az MBSE-ben a rendszerinformáció egyetlen, formalizált modellben él. A dokumentumok és diagramok ennek nézetei, amelyek az érdekelt felek kérdéseihez igazodó nézőpontok szerint készülnek.
- A SysML a legelterjedtebb rendszermodellezési nyelv. A 2025-ben elfogadott SysML v2 formális szemantikájú, szövegesen és grafikusan is írható, és szabványos API-ja van.
- A digitális modell, a digitális árnyék és a digitális iker az adatkapcsolat automatizáltságában különbözik. A pontos szóhasználat a követelmények és a költségek szempontjából is fontos.

Önellenőrzés

1. Miért nincs „a” modellje egy rendszernek? Mondj két különböző kérdést a VárosiBringa esetéből, amelyekhez különböző modell kell!
2. Melyik modellcsaládot választanád, és miért? (a) Hány dokkolóhely kell egy állomáson, hogy reggel 95% eséllyel le lehessen tenni a kerékpárt? (b) Milyen alakú legyen a kerékpár kormánya, hogy kényelmes legyen? (c) Hogyan hat a díjszabás változása a használatra három év alatt?
3. Mi a hiba a következő kontextusdiagram-címkében: „adatcsere a bankkal”? Javíts rajta!
4. Sorold be a következő interfészelemeket (összekötő, szigetelő, átalakító): hőszigetelő burkolat, USB-kábel, mikrofon, tengelykapcsoló, tűzgátló fal, frekvenciaváltó.
5. Egy szimulációs modell kiválóan visszaadja a ForgácsTech tavalyi átfutási idejét. Mondhatjuk-e ebből, hogy a modell érvényes a MES bevezetése utáni helyzet előrejelzésére? Miért?
6. Mi a különbség az architektúra és az architektúraleírás, illetve a nézőpont és a nézet között?
7. Egy hőerőmű folyamatirányító rendszere a mért adatokból percenként frissíti a kazán szimulációs modelljét, és a modell alapján automatikusan módosítja a tüzelőanyag-adagolás alapjelét. Digitális modell, árnyék vagy iker?

Megoldás.

1. Mert a modell mindig egy kérdéshez készül, és elhagyja a kérdés szempontjából lényegtelent. Például: „hol legyenek az állomások?” (térinformatikai és adatvezérelt keresletmodell), illetve „mennyi áthelyezés kell naponta?” (szimuláció vagy sorbanállási modell).
2. (a) Sztochasztikus analitikus modell (sorbanállási vagy valószínűségi modell) vagy diszkrét eseményes szimuláció, mert véletlen érkezésekről és távozásokról van szó. (b) Fizikai modell (prototípus) felhasználói próbákkal, mert az érzékelt kényelem nehezen számítható. (c) Rendszerdinamikai modell, mert visszacsatolások és késleltetések határozzák meg a hosszú távú viselkedést.
3. Kétirányú és semmitmondó. Helyette például két nyíl: „fizetési kérelem” a rendszertől a bank

felé, és „fizetési visszaigazolás” a banktól a rendszer felé.
4. Szigetelő: hőszigetelő burkolat, tűzgátló fal. Összekötő: USB-kábel, tengelykapcsoló. Átalakító: mikrofon (akusztikus jelet villamossá alakít), frekvenciaváltó (a villamos energia frekvenciáját alakítja át).
5. Nem feltétlenül. A validitás egy alkalmazási tartományra vonatkozik. A MES bevezetése megváltoztatja az információáramlást és az ütemezést, ezért a modell szerkezetét is módosítani kell, és a fogalmi modell validitását az új helyzetre újra meg kell vizsgálni.
6. Az architektúra a rendszer alapvető fogalmai és tulajdonságai, az architektúraleírás pedig az ezt kifejező munkatermék. A nézőpont egy nézet elkészítésének előírása (kinek, milyen kérdésére, milyen technikával), a nézet pedig egy adott nézőpont szerint elkészült leírás.
7. Digitális iker, mert az adatáramlás mindkét irányban automatikus.

Projektlépés: kontextusdiagram

Saját fejlesztési feladatodhoz:

1. nevezd meg a rendszert úgy, hogy a neve a feladatát fejezze ki, ne egy megoldást;
2. gyűjtsd össze a külső entitásokat (legalább hatot), és rajzold fel a kontextusdiagramot;
3. minden kölcsönhatást címkézz meg konkrétan, és sorold típusba (adat, jel, anyag, energia, tevékenység);
4. ellenőrizd, hogy minden entitásnak van-e értelmes bemenete vagy kimenete, és hogy kimaradt-e valamilyen környezeti hatás (hulladék, kibocsátás, zaj);
5. írd le, milyen modellekkel válaszolnád meg a projekted két legfontosabb kérdését.

Az eszközt és a sablont a Gyakorlati útmutató kontextusdiagram-eszközlapja tartalmazza.

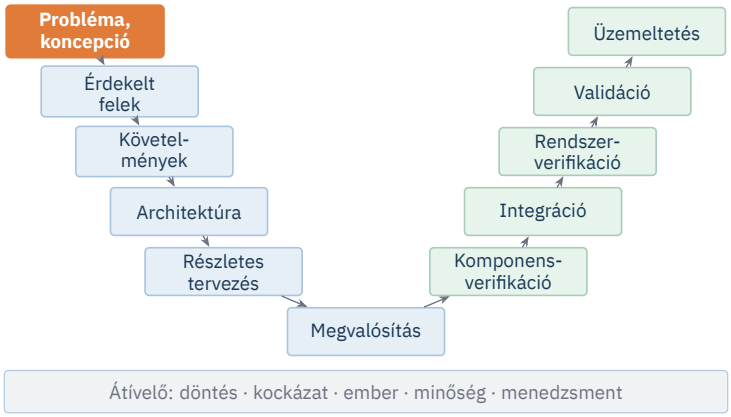
Továbbolvasás

- Kossiakoff és mtsai. (2011), 3. fejezet: a rendszer környezete, a kontextusdiagram és az interfészelemek.
- Sargent (2013): a szimulációs modellek verifikációjának és validációjának áttekintése.
- Friedenthal, Moore, és Steiner (2014): a SysML 1.x nyelv részletes, gyakorlati bevezetője.
- OMG (2025): a SysML v2 specifikációja és nyílt referencia-implementációja.
- Kritzinger és mtsai. (2018): a digitális iker fogalmának rendszerezése a gyártásban.
- ISO/IEC/IEEE (2022): az architektúraleírás fogalmai.

II. rész – A V bal ága: a problémától a tervig

Pannon Egyetem
University of Pannonia

4. A probléma meghatározása és a működési koncepció



Tanulási eredmények

A fejezet feldolgozása után képes leszel:

- szétválasztani a problémateret és a megoldásteret, és felismerni, ha egy probléma-megfogalmazás valójában megoldást tartalmaz;
- problémafával feltárni egy probléma okait és hatásait, és célfává alakítani;
- a „van–nincs” elemzéssel pontosan körülhatárolni egy problémát;
- tömör, mérhető problémamegfogalmazást írni, rögzítve a korlátokat, a feltételezéseket és a siker kritériumait;
- felismerni a „gonosz” (nehezen meghatározható) problémákat, és a tervezési stratégiát ehhez igazítani;
- működési koncepciót (ConOps) készíteni üzemmódokkal, normál és rendellenes forgatókönyvekkel, valamint használati esetekkel.

A V-modell bal ága nem a követelményekkel kezdődik, hanem egy kérdéssel: *milyen problémát oldunk meg?* Ez a lépés gyakran rövidebb a kelleténél. A megrendelő hozza a kész megoldást („kell egy MES”, „automatizáljuk az állomást”, „csináljunk egy alkalmazást”), és a csapat azonnal nekilát. A 2. fejezetben láttuk, hogy az életciklus-költségek nagy része már a korai döntésekkel elkötelezetté válik. A rosszul megfogalmazott probléma ezért a legdrágább hiba: a legjobb megvalósítás sem segít, ha rossz kérdésre ad választ.

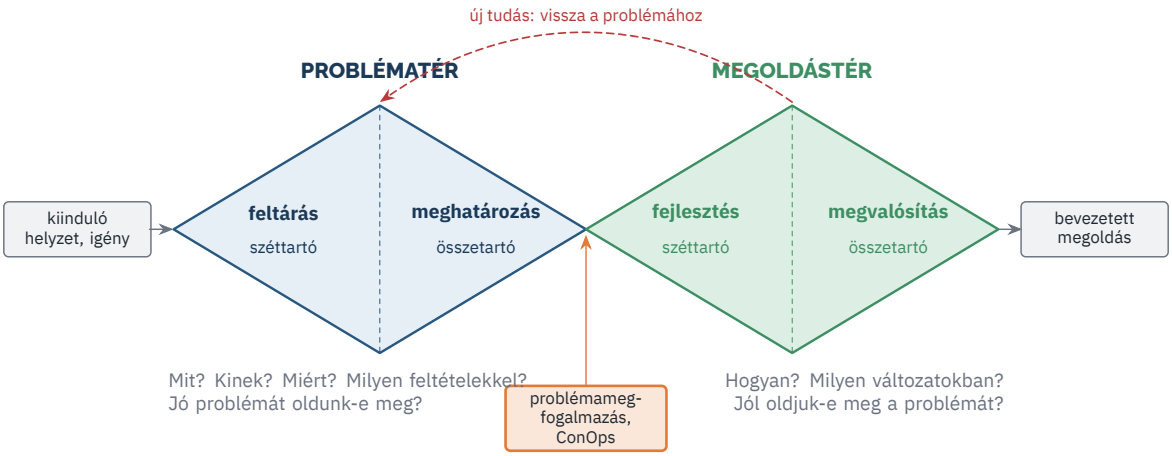
A fejezet négy kérdésre épül:

- Hogyan válasszuk szét, *mit* akarunk elérni, és *hogyan* akarjuk elérni?
- Hogyan tárjuk fel egy probléma okait és hatásait, és hogyan fogalmazzuk meg pontosan?
- Mit kezdjünk azokkal a problémákkal, amelyeket nem lehet egyszer s mindenkorra meghatározni?
- Hogyan írjuk le a jövőbeli rendszer használatát úgy, hogy abból követelmények születhessenek?

4.1. Problématér és megoldástér

A tervezési munka két, egymástól elválasztandó térben zajlik. A **problématér** arról szól, *mit* kell elérni, *kinek*, *miért* és *milyen feltételekkel*. A **megoldástér** arról, *hogyan* érjük el. A két tér összemosása a rendszermérnökség egyik leggyakoribb hibája. Az INCOSE követelményírási útmutatója is erre figyelmeztet: a követelmény azt fogalmazza meg, *mire* van szükség, és ne azt, *hogyan* kell megvalósítani (INCOSE 2023a). Ha egy követelmény mégis megoldást ír elő, érdemes megkérdezni, *miért* van rá szükség: a válasz gyakran a valódi követelményhez vezet.

A brit Design Council 2005-ben közzétett, 2019-ben továbbfejlesztett „dupla gyémánt” modellje ezt a szétválasztást teszi szemléletessé (4.1. ábra) (Design Council 2019). Az első gyémánt a problématér: előbb széttartó gondolkodással feltárjuk a helyzetet, aztán összetartó gondolkodással pontosan meghatározzuk a problémát. A második gyémánt a megoldástér: előbb sok lehetséges megoldást dolgozunk ki, aztán kiválasztjuk és megvalósítjuk a legjobbat. A két gyémánt határán áll a problémamegfogalmazás és a működési koncepció. A modell nem lineáris: a megoldástérben szerzett tudás gyakran visszavezet a problémához.



4.1. ábra. A problématér és a megoldástér szétválasztása a Design Council „dupla gyémánt” modellje nyomán. Mindkét térben előbb széttartó, aztán összetartó gondolkodás következik.

A V-modell nyelvén: a bal ág első két lépése (a probléma és az érdekelt felek igényei) az első gyémánt, a koncepcióváltozatok kidolgozása és kiválasztása (9. és 10. fejezet) a második. A követelmények (6. fejezet) a kettő között közvetítenek. A problématérből származnak, de már a megoldás tulajdonságait írják elő.

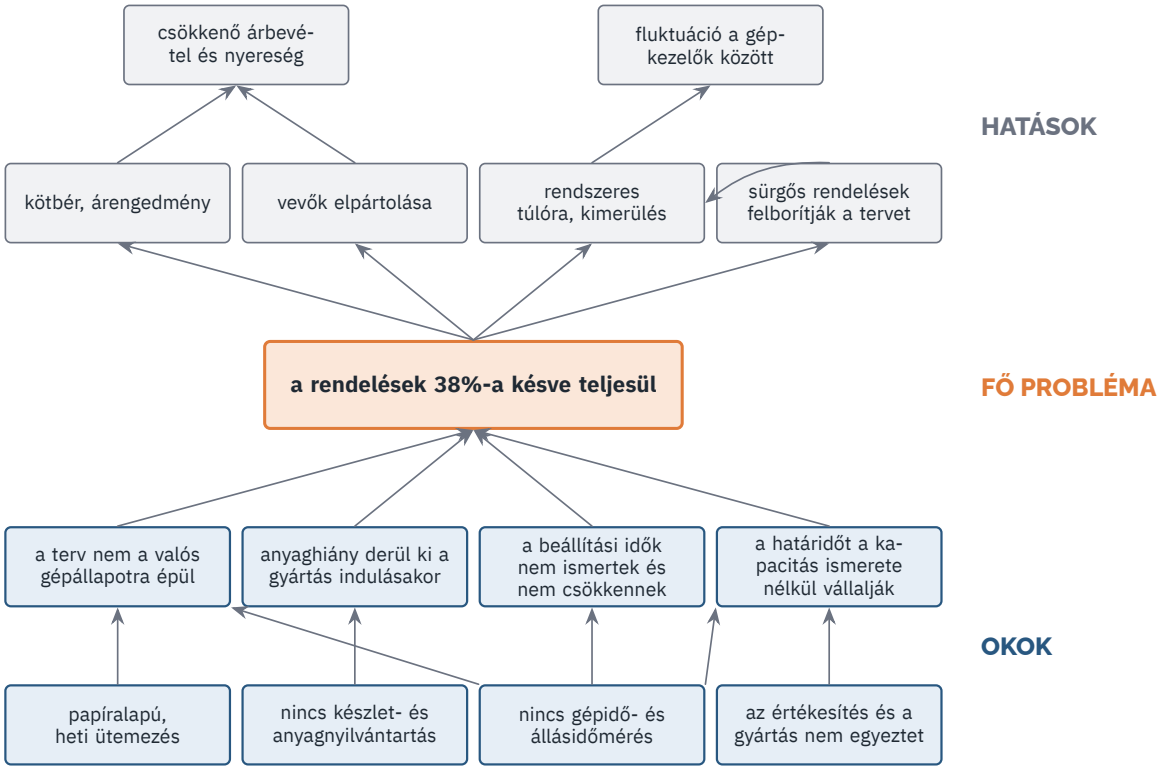
> **ForgácsTech: „kell egy MES”**
>
> A ForgácsTech tulajdonos-ügyvezetője egy szakmai kiállításon látott MES-rendszerrel érkezett a projektindító megbeszélésre. A rendszermérnök felírta a táblára: „Probléma: nincs MES.” Aztán megkérdezte: „Ha holnap reggelre lenne egy MES, miben lenne más a cég helyzete egy év múlva?”
> A válaszok sorra a valódi problémát rajzolták ki: „Nem csúsznának a határidők.” „Tudnánk, mennyit állnak a gépek.” „Nem kellene minden pénteken túlórázni.” „Nem vesztenénk el a legnagyobb vevőnket.” A „nincs MES” egy *hiányzó megoldás* volt. A probléma a megbízhatatlan határidő-tartás, amelynek a MES csak az egyik lehetséges orvossága. A továbbiakban kiderült, hogy legalább ugyanannyit számít az értékesítés és a gyártás egyeztetése és a beállítási idők csökkentése. Ezekhez nem kell szoftver.

4.2. A probléma feltárása

4.2.1. Problémafa és célfa

A **problémafa** a probléma okait és hatásait rendezi ok-okozati hierarchiába. Az Európai Bizottság projektciklus-menedzsment útmutatója a logikai keretmátrixos tervezés első lépéseként írja le. A középpontba a fő problémát (fókuszproblémát) helyezzük, alá az okait, fölé a hatásait, a nyilak pedig az ok-okozati kapcsolatokat mutatják (European Commission 2004). Az elkészítés lépései:

1. **Gyűjtsük össze a problémákat** az érdekelt felekkel közösen, kártyákon. Egy kártya egy problémát tartalmazzon, negatív állapotként megfogalmazva („a rendelések 38%-a késve teljesül"). Kerüljük a hiányzó megoldás formát („nincs MES").
2. **Válasszuk ki a fő problémát**, amelyhez a többiek okként vagy hatásként kapcsolódnak.
3. **Rendezzük az okokat** a fő probléma alá: mi okozza ezt? És mi okozza az okot? Addig haladjunk lefelé, amíg befolyásolható okokhoz nem érünk.
4. **Rendezzük a hatásokat** a fő probléma fölé: mi következik ebből?
5. **Ellenőrizzük a logikát:** minden nyíl mentén igaz-e, hogy az alsó elem *okozza* a felsőt?



4.2. ábra. A ForgácsTech problémafája. A fő probléma alatt az okok, fölötte a hatások láthatók. A gépidő- és állásidőmérés hiánya több okhoz is hozzájárul.

A problémafából **célfa** készül: minden negatív állapotot a kívánt pozitív állapottá fogalmazunk át. A fő probléma így fő céllá válik („a rendelések legalább 95%-a határidőre teljesül"), az okokból eszközcélok lesznek („a terv a valós gépállapotra épül"), a hatásokból pedig a magasabb szintű, átfogó célok („nő az árbevétel"). Az útmutató arra is figyelmeztet, hogy nem minden eszközcél vállalható egyetlen projektben. A célfából ki kell választani azt az ágat vagy ágakat, amelyekre a projekt összpontosít (European Commission 2004).

Kidolgozott példa: a ForgácsTech problémafája

A 4.2. ábra a ForgácsTech kétórás problémaelemző workshopjának eredménye. Ezen a termelésvezető, két gépkezelő, a gyártáselőkészítő, az értékesítő és a tulajdonos vett részt.
A fő probléma. A csapat a „határidő-csúszás” mellett döntött, és az elmúlt fél év szállítási adataiból számszerűsítette: a rendelések 38%-a teljesült késve.
Az okok. Négy közvetlen okot azonosítottak. A legmélyebb szinten egy ok több ágon is megjelent: a gépidő és az állásidő mérésének hiánya. Emiatt nem lehet a valós gépállapotra tervezni, a beállítási időket nem lehet csökkenteni, mert senki sem tudja, mennyik, és a határidőt is a kapacitás ismerete nélkül vállalják. Az ilyen, több ágon visszatérő ok különösen jó beavatkozási pont.
A célfa és a projekt hatóköre. Az átfogalmazás után a csapat két ágat választott, ezekre összpontosít a projekt: a gépállapot és az állásidő mérését (ehhez kell a MES adatgyűjtő része), valamint a kapacitásalapú határidő-vállalást (ehhez az értékesítés és a gyártás közös heti egyeztetése és egy egyszerű kapacitástábla kell). Az anyagnyilvántartás bevezetését egy későbbi szakaszra halasztották.

4.2.2. A probléma körülhatárolása: a „van–nincs” elemzés

Kepner és Tregoe problémaelemzési módszerének központi eszköze a probléma pontos specifikációja. Azt vizsgálja, *mi* a probléma, *hol*, *mikor* és *milyen mértékben* jelentkezik, és ugyanezekre a kérdésekre azt is, hol, mikor és mennyiben *nem* jelentkezik, holott jelentkezhetne (Kepner és Tregoe 1965). A „van” és a „nincs” közötti különbség gyakran közvetlenül az ok felé mutat.

4.1. táblázat. A ForgácsTech határidő-csúszásának „van–nincs” elemzése.

Kérdés	VAN (hol jelentkezik?)	NINCS (hol nem, bár jelentkezhetne?)	Különbség, nyom
Mi?	késés a kis sorozatú, egyedi rendeléseknél	a három legnagyobb vevő ismétlődő rendeléseinél ritka	az ismétlődő rendeléseknek kész programja és beállítási lapja van
Hol?	a többtengelyes megmunkáló központokon	a hagyományos esztergákon alig	a megmunkáló központok beállítása hosszabb és változékonyabb
Mikor?	főleg a hónap második felében	a hónap elején ritka	a hónap második felében gyűlnek a sürgős rendelések
Mennyi?	átlagosan 4 munkanap, a rendelések 38%-a	–	a késések fele a beállításnál és az anyagbeszerzésnél keletkezik

A 4.1. táblázat jól mutatja, mit ad az elemzés. A probléma nem általános „szervezetlenség”. Kimutathatóan a változatos, egyedi rendeléseknél és a hosszú beállítású gépeken jelentkezik. Ez a beállítási idők mérését és csökkentését (a 19. fejezetben tárgyalt SMED-szemléletet) is a középpontba állítja.

4.3. A probléma megfogalmazása

A feltárás eredményét tömör **problémamegfogalmazásba** foglaljuk. Ez a projekt egyik legfontosabb dokumentuma. Minden érdekelt félnek ugyanazt kell értenie belőle, és minden későbbi döntést hozzá lehet mérni. A jó problémamegfogalmazás a következő elemeket tartalmazza:

- **a jelenlegi helyzet:** mi a probléma, kit érint, hol és mikor jelentkezik, mekkora (számszerűen, ha lehet);
- **a kívánt helyzet:** mit akarunk elérni, mérhető módon, határidővel;
- **a jelentőség:** miért fontos, mi történik, ha nem oldjuk meg;
- **a hatókör:** mi tartozik a feladathoz, és mi nem;
- **a korlátok:** a megoldásnak kötelezően betartandó feltételek;
- **a feltételezések:** amit igaznak fogadunk el, de nem tudunk biztosan;
- **a siker kritériumai:** miből látjuk majd, hogy a probléma megoldódott.

A siker kritériumai azt határozzák meg, mit kell a rendszernek elérnie ahhoz, hogy sikeresnek tekintsük, és milyen jól kell ezt elérnie. Az érdekelt felek elvárásait rögzítik, ezért a korlátokkal együtt a legfelső szintű követelmények forrásai. Az érdekelt felek nézőpontjából megfogalmazott, mérhető sikerkritériumokat a szakirodalom a hatékonyság mérőszámainak (measures of effectiveness, MOE) nevezi: ezek a tervezett használati környezetben, adott feltételek mellett mutatják meg, mennyire teljesíti a rendszer a küldetését (Roedler és Jones 2005; NASA 2016, 4.1. szakasz). Ezekkel a 5.6. szakasz részletesen foglalkozik.

ForgácsTech: a problémamegfogalmazás

Jelenlegi helyzet. Az elmúlt hat hónapban a ForgácsTech rendeléseinek 38%-a teljesült késve, átlagosan négy munkanappal. A késés főként a kis sorozatú, egyedi rendeléseket és a többtengelyes megmunkáló központokat érinti. A heti ütemterv papíralapú, a gépállapotról és az állásidőkről nincs adat, a határidőket a kapacitás ismerete nélkül vállalják.
Kívánt helyzet. Tizennyolc hónapon belül a rendelések legalább 95%-a teljesüljön határidőre, a késve teljesülő rendelések átlagos késése legfeljebb egy munkanap legyen, és a cég minden gépre ismerje a tényleges kihasználtságot és az állásidő okait.
Jelentőség. A két legnagyobb vevő a következő keretszerződés feltételeül szabta a 95%-os határidő-tartást és a rendelések nyomonkövethetőségét. A két vevő az árbevétel mintegy 40%-át adja.
Hatókör. A rendelés felvételétől a kiszállításig tartó információs és anyagáramlás. Nem része a feladatnak a könyvelési rendszer cseréje és a gépek felújítása.
Korlátok. A beruházás felső határa 45 millió forint. A bevezetés alatt a termelés nem állhat le egy hétnél tovább. A gépek vezérlőinek gyártói garanciáját nem szabad veszélyeztetni.
Feltételezések. A gépek többsége rendelkezik adatkapcsolati lehetőséggel (ellenőrizendő a régebbi esztergáknál). A gépkezelők elfogadják a digitális munkautasításokat (az üzemi megbeszélésen visszajelzés kérendő).
A siker kritériumai. Határidőre teljesülő rendelések aránya ≥ 95% három egymást követő hónapban; átlagos késés ≤ 1 munkanap; a gépek állásidejének legalább 90%-ához ok rendelve.

4.3.1. Korlátok és feltételezések

A **korlát** olyan feltétel, amelyet a megoldásnak mindenképpen teljesítenie kell, és amelyről a tervezők nem dönthetnek. Ilyen a költségkeret, a határidő, egy jogszabály, egy szabvány vagy egy meglévő rendszerhez való illeszkedés. Az ISO/IEC/IEEE 29148 szabvány szerint

a korlát kívülről előírt megkötés, amely a követelményeket, a tervet, a megvalósítást vagy magát a fejlesztési folyamatot korlátozza (ISO/IEC/IEEE 2018). A korlát teljesüléséről költség- vagy teljesítményszempontok miatt sem lehet lemondani. A korlátok szűkítik a megoldásteret. Ezért fontos, hogy csak a valódi korlátokat rögzítsük, és ne emeljünk korláttá egy megszokott vagy kényelmes megoldást.

A **feltételezés** olyan állítás, amelyet a tervezés során igaznak fogadunk el, de nem tudjuk biztosan. Ha egy feltételezés hamisnak bizonyul, a rá épülő követelmények és tervezési döntések is megdőlhetnek. A feltételezéseket ezért érdemes nyilvántartani, és a követelmények jóváhagyása előtt megerősíteni: ki ellenőrzi őket, és mikorra. A meg nem erősített feltételezés kockázat, és a kockázatlistára kerül (14. fejezet).

4.3.2. Gonosz problémák

Rittel és Webber 1973-as tanulmánya óta **gonosz problémáknak** (wicked problems) nevezzük azokat a társadalmi és tervezési problémákat, amelyeket nem lehet egyértelműen, végérvényesen megfogalmazni (Rittel és Webber 1973). A szerzők tíz jellemzőjük közül néhány különösen fontos a rendszermérnök számára:

- **Nincs végleges megfogalmazásuk.** A probléma megértése és a megoldás keresése ugyanaz a folyamat: minden megoldási kísérlet újabb oldalát mutatja meg a problémának.
- **Nincs leállási szabályuk.** Nem lehet megmondani, mikor oldottuk meg őket. Csak azt, mikor elég jó a helyzet.
- **A megoldásaik nem igazak vagy hamisak, hanem jobbak vagy rosszabbak**, és ezt különböző érdekelt felek eltérően ítélik meg.
- **Minden megoldás egyszeri, következményekkel járó beavatkozás.** Nincs lehetőség próbálkozásra és tévedésre következmények nélkül.
- **Minden gonosz probléma lényegében egyedi**, ezért a máshol bevált megoldás nem vihető át gondolkodás nélkül.

A VárosiBringa ilyen probléma. A „belvárosi forgalom csökkentése” mögött ellentmondó érdekek és viselkedési mintázatok állnak. A kerékpárrendszer bevezetése maga is megváltoztatja a problémát: más lesz a parkolási igény, más a buszok kihasználtsága, mások lesznek a panaszok. A gonosz problémákra nem az a válasz, hogy lemondunk a pontos megfogalmazásról. Kisebb, visszafordítható lépésekben kell haladni (próbaüzem, bővíthető hálózat), a problémamegfogalmazást rendszeresen újra kell vizsgálni, és az érdekelt feleket végig be kell vonni. Ez a 1.4. szakaszban tárgyalt komplex rendszerek tervezési stratégiája.

4.4. A működési koncepció

A problémamegfogalmazás azt mondja meg, mit kell elérni. A **működési koncepció** (concept of operations, ConOps) azt írja le, hogyan fogják a jövőbeli rendszert használni ahhoz, hogy ezt elérjék. A ConOps a koncepciószakasz elején készül. Felhasználóközpontú dokumentum: a rendszer mennyiségi és minőségi jellemzőit a felhasználók, a megrendelő, a fejlesztők és a támogató szervezetek számára egyaránt érthetően írja le, és a felhasználó szervezetét, feladatait és céljait egységes rendszerszemléletben mutatja be (IEEE 1998; NASA 2016, 4.1. szakasz). Működési nézőpontból, jellemzően időbeli sorrendben írja

le, hogyan fogják a rendszert használni, ezért a felhasználókhoz kapcsolódó követelmények és az architektúra kiindulópontja. A ConOps-dokumentum tartalmát eredetileg az IEEE 1362 útmutató rögzítette; ezt 2011-ben a követelménymérnökség szabványa, az ISO/IEC/IEEE 29148 váltotta fel. A szabvány jelenlegi változata a szervezeti szintű működési koncepciót (ConOps) és a rendszer szintű működési koncepciót (OpsCon) külön információs elemként kezeli (ISO/IEC/IEEE 2018).

A ConOps legfontosabb tulajdonsága, hogy **a felhasználó nyelvén szól**. Nem azt írja le, milyen alkatrészekből áll a rendszer, hanem azt, hogy ki, mikor, milyen helyzetben, mit csinál vele, és mit tapasztal közben. Ezért a nem mérnök érdekelt felek is el tudják olvasni, ki tudják egészíteni, és jóvá tudják hagyni. A követelményekhez hasonlóan a ConOps is megvalósításfüggetlen: azt mondja meg, mit kell a rendszernek és elemeinek tennie, és nem azt, hogyan.

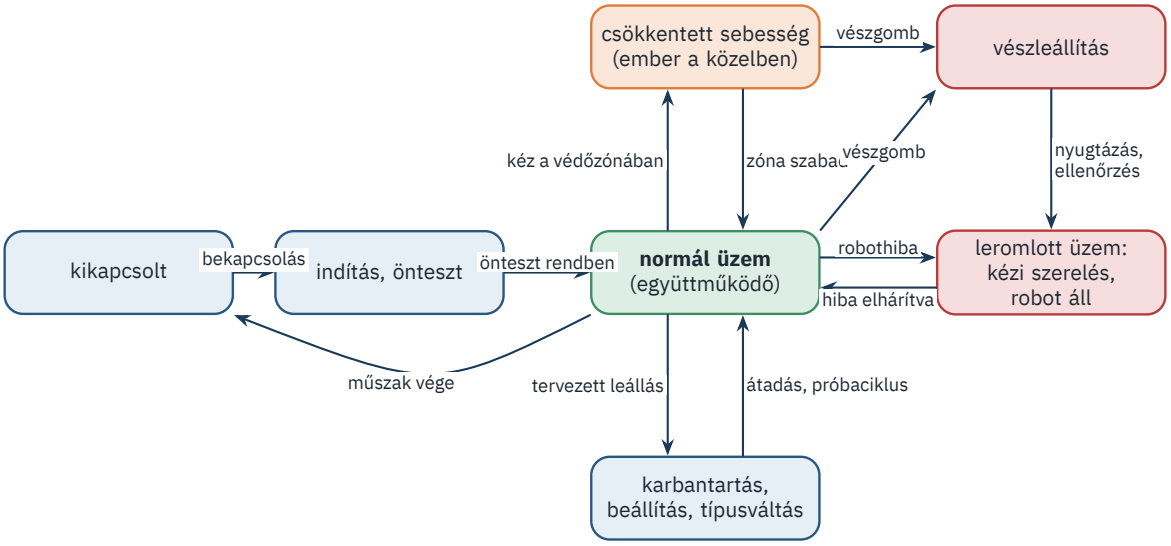
Az IEEE 1362 útmutató vázlata a mai napig a ConOps-dokumentumok legelterjedtebb szerkezete (IEEE 1998). Kissé egyszerűsítve a következő fejezetekből áll:

1. **Hatókör:** a rendszer azonosítása, a dokumentum célja, a rendszer rövid áttekintése.
2. **Hivatkozott dokumentumok.**
3. **A jelenlegi rendszer vagy helyzet:** háttér és célok; működési szabályok és korlátok; a jelenlegi működés leírása; üzemmódok; felhasználói csoportok; támogató környezet.
4. **A változtatás indoka és jellege:** milyen hiányosságok miatt kell az új rendszer, és mi változik.
5. **A tervezett rendszer koncepciója:** célok; működési szabályok és korlátok; a rendszer leírása; üzemmódok; felhasználói csoportok; támogató környezet (üzemeltetés, karbantartás, javítás).
6. **Működési forgatókönyvek:** normál és rendellenes körülmények között.
7. **A hatások összefoglalása:** a várható hatás a működésre, a szervezetre és a fejlesztés ideje alatti munkára.
8. **A tervezett rendszer elemzése:** az előnyök, a hátrányok és korlátok, valamint a mérlegelt változatok.

4.4.1. Üzemmódok

A ConOps-vázlat a jelenlegi és a tervezett rendszer leírásában is külön pontot szán az **üzemmódoknak** (IEEE 1998), vagyis azoknak a működési módoknak vagy konfigurációknak, amelyekre a rendszernek életciklusa során szüksége lehet. Ide tartoznak a fejlesztéshez és a betanításhoz szükséges teszt- és oktatási módok, a karbantartási mód, valamint a rendellenes helyzetek kezelésére szolgáló módok is. Az üzemmódokat és a köztük lévő átmeneteket állapotgép-diagrammal ábrázoljuk (3.6. szakasz).

> **KoboCella: miért kell leromlott üzemmód?**
>
> A KoboCella első koncepciója két állapotot ismert: „működik” és „áll”. A ConOps-workshopon a műszakvezető megkérdezte: „Mi történik, ha a robot hétfő reggel elromlik, és a szerviz csak szerdán jön?” A gyártási terv szerint a cella napi mintegy 610 motort szerel. Két nap kiesés több mint 1200 motor, ami a vevő felé szerződésszegés.
> Így került a koncepcióba a **leromlott üzemmód** (4.3. ábra): a robot áll, de a cella kézi munkaállomásként tovább működik, csökkentett ütemben. Ez a döntés több követelményt is szült. A munkaasztal magasságának kézi szereléshez is megfelelőnek kell lennie. A csavarozó egységnek kézi üzemben is használhatónak kell lennie, nyomatékellenőrzéssel. Az operátoroknak évente legalább egyszer gyakorolniuk kell a kézi szerelést. Ha a leromlott üzemmód nem a ConOps-ban, hanem csak az üzembe helyezéskor merül fel, ezek a követelmények



zöld: névleges üzem · narancs: korlátozott üzem · piros: rendellenes vagy leromlott üzem

4.3. ábra. A KoboCella üzemmódjai és a köztük lévő átmenetek. A diagram a normál, a korlátozott és a rendellenes üzemet egyaránt mutatja.

drága átalakítást jelentettek volna.

4.4.2. Működési forgatókönyvek

A **működési forgatókönyv** egy konkrét helyzet időbeli lefolyásának leírása: ki, mit, milyen sorrendben csinál, és hogyan reagál a rendszer. A forgatókönyveknek a normál és a rendellenes körülményeket is le kell fedniük, és együtt minden várt funkcióra és képességre ki kell terjedniük: a hiányzó forgatókönyv hiányzó követelményt jelent. Jó gyakorlat, ha minden forgatókönyv saját azonosítót kap, mert így a belőle levezetett követelmények visszakövethetők.

Három forgatókönyvtípust érdemes megkülönböztetni:

- **normál (névleges) forgatókönyv:** minden a terv szerint történik;
- **rendellenes forgatókönyv:** valami eltér a tervtől (hibás bemenet, váratlan esemény, kezelési hiba), és a rendszernek ezt kezelnie kell;
- **leromlott üzemű forgatókönyv:** a rendszer egy része nem működik, de a rendszer csökkentett képességgel tovább üzemel.

Kidolgozott példa: a VárosiBringa egy normál és egy rendellenes forgatókönyve

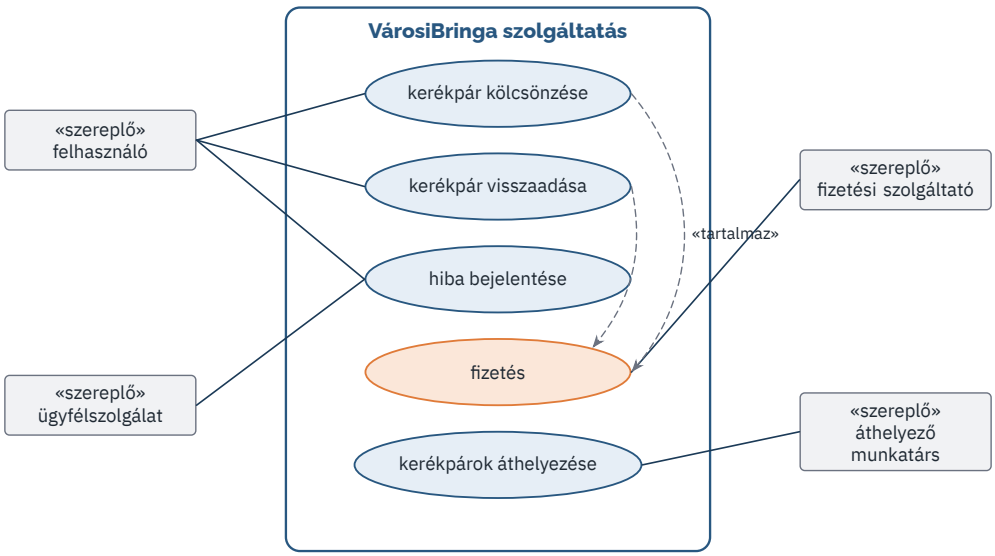
[FK-01] Reggeli kölcsönzés (normál). Hétköznap reggel 7:40-kor Anna, másodéves hallgató, a lakótelepi állomáson az alkalmazással kiválaszt egy kerékpárt. A rendszer megjeleníti a kerékpár töltöttségét (82%) és a kampusz állomásán várhatóan szabad helyek számát (6). Anna jóváhagyja a kölcsönzést. A dokkoló két másodpercen belül kioldja a zárat. Anna 12 perc alatt felér a kampuszra, a kerékpárt az ottani dokkolóba tolja. A zár rögzíti a kerékpárt, és a rendszer egy percen belül lezárja a kölcsönzést. Anna értesítést kap a díjról, amelyet a közös bérlete miatt nem kell megfizetnie.
[FK-07] Tele van a céldokkoló (rendellenes). Egy vizsgaidőszaki reggelen a kampusz dok-

kolója 7:55-re megtelik. Bence a helyszínen nem tudja letenni a kerékpárt. A forgatókönyv szerint a rendszer a dokkolótól 30 méteren belül felismeri a helyzetet, felajánlja a legközelebbi szabad helyet (a könyvtárnál, 250 méterre), és tizenöt perc ingyenes többletidőt ad. Ha Bence ezt nem fogadja el, a kampuszon kijelölt túlcsorduló területen is leteheti a kerékpárt egy virtuális zárolással. Ekkor az áthelyező szolgálat riasztást kap.
A második forgatókönyv három követelményt is felszínre hozott, amelyekre a csapat korábban nem gondolt: a helymeghatározás pontossága, a túlcsorduló terület kezelése és az áthelyező szolgálat valós idejű riasztása. A rendellenes forgatókönyvek általában több új követelményt hoznak, mint a normálok.

4.4.3. Használati esetek

A **használati eset** (use case) egy szereplő céljának elérését írja le a rendszerrel folytatott kölcsönhatás révén. Cockburn klasszikus könyve szerint a használati eset a rendszer viselkedésére vonatkozó szerződés, amely leírja, hogyan reagál a rendszer a különböző feltételek mellett egy elsődleges szereplő kérésére. Központi eleme a **fő sikerforgatókönyv**, amely a cél elérésének leggyakoribb útja, és a **kiterjesztések**, amelyek az eltéréseket és a hibakezelést írják le (Cockburn 2001).

A használati esetek áttekintésére a használatieset-diagram szolgál (4.4. ábra). A diagram megmutatja a szereplőket, a használati eseteket és a köztük lévő kapcsolatokat. A szereplő lehet ember, szervezet vagy másik rendszer. A diagram önmagában kevés információt hordoz: az értéke a hozzá tartozó szöveges leírásokban van.



4.4. ábra. A VárosiBringa néhány használati esete. A „fizetés” eset más esetek része («tartalmaz» kapcsolat).

4.2. táblázat. A HE-01 használati eset leírása Cockburn szerkezete szerint.

Elem	Tartalom
Azonosító, név	HE-01 Kerékpár kölcsönzése
Elsődleges szereplő	felhasználó
Cél	egy feltöltött kerékpárral elindulni egy állomásról
Előfeltétel	a felhasználó regisztrált, érvényes fizetési móddal vagy bérlettel

Elem	Tartalom
Fő sikerforgatókönyv	1. A felhasználó kiválaszt egy kerékpárt az alkalmazásban vagy a dokkolón. 2. A rendszer megjeleníti a töltöttséget és a díjat. 3. A felhasználó jóváhagyja a kölcsönzést. 4. A rendszer ellenőrzi a fizetési jogosultságot. 5. A dokkoló kioldja a zárat. 6. A felhasználó elviszi a kerékpárt; a rendszer elindítja a kölcsönzést.
Kiterjesztések	4a. A fizetési jogosultság nem érvényes: a rendszer felajánlja a fizetési mód frissítését. 5a. A zár nem old ki: a rendszer másik kerékpárt ajánl, és hibajegyet nyit. 6a. A kerékpárt egy percen belül nem viszik el: a rendszer visszazárja és törli a kölcsönzést.
Utófeltétel	a kölcsönzés aktív, a kerékpár a felhasználóhoz van rendelve

A ConOps forgatókönyvei és a használati esetek egymást erősítik. A forgatókönyv egy konkrét történet konkrét szereplőkkel. A használati eset általánosítja ezt, és rendszerezetten végiggondolja az eltéréseket. Mindkettő a követelmények egyik fő forrása. Minden lépésből és minden kiterjesztésből egy vagy több követelmény születhet, és a validáció során ugyanezek a forgatókönyvek lesznek a tesztek alapjai (24. fejezet).

Megoldás a forgatókönyvben

A forgatókönyv akkor jó, ha a felhasználó céljáról és tapasztalatáról szól. Ha már gombokat, menüpontokat és adatbázismezőket ír le, akkor a megoldásteret írja le, és szűkíti a lehetőségeket. A „Anna megnyomja a zöld »Kölcsönzés« gombot a kezdőképernyő jobb alsó sarkában” mondat felhasználói felület terv, nem működési koncepció. A ConOps-ba ennyi kell: „Anna jóváhagyja a kölcsönzést.”

Nagy nyelvi modellek a forgatókönyvek bővítésében

A rendellenes forgatókönyvek felderítése fárasztó és könnyen hiányos munka. A nagy nyelvi modellek ebben hasznos segítők lehetnek: egy normál forgatókönyvből rövid idő alatt sok lehetséges eltérést javasolnak („mi van, ha…”). A javaslatok minősége azonban változó, és a modell nem ismeri a valós üzemeltetési körülményeket. A Gyakorlati útmutató ConOps-eszközlapjának mintapromptja ezért arra kéri a modellt, hogy a javaslatait valószínűség és következmény szerint is jelölje, a csapat pedig az érdekelt felekkel együtt dönti el, melyik forgatókönyv kerül a ConOps-ba.

Összefoglalás

- A problématér azt írja le, mit, kinek, miért és milyen feltételekkel kell elérni, a megoldástér azt, hogyan. A kettő összemosása a rendszermérnökség egyik leggyakoribb hibája. A „kell egy X” típusú kérés hiányzó megoldás, nem probléma.
- A problémafa a fő probléma okait és hatásait rendezi ok-okozati hierarchiába, a célfa ennek pozitív átfogalmazása. A több ágon visszatérő okok jó beavatkozási pontok.
- A „van–nincs” elemzés a probléma pontos körülhatárolásával az okok felé terel.
- A problémamegfogalmazás tartalmazza a jelenlegi és a kívánt helyzetet, a jelentőséget, a hatókört, a korlátokat, a feltételezéseket és a siker kritériumait. A meg nem erősített feltételezések kockázatok.
- A gonosz problémák nem fogalmazhatók meg végérvényesen. Kis, visszafordítható lépésekkel, folyamatos újrafogalmazással és az érdekelt felek bevonásával kezelhetők.
- A működési koncepció a felhasználó nyelvén írja le, hogyan fogják a rendszert használni. Része az üzemmódok leírása, valamint a normál, a rendellenes és a leromlott üzemű forgatókönyvek. A használati esetek ezeket rendszerezik. Mindkettő a követelmények és a validációs tesztek forrása.

Önellenőrzés

1. Mi a probléma a következő megfogalmazással: „A problémánk, hogy nincs automata raktárunk.” Fogalmazd át!
2. Készíts kis problémafát a következő helyzetről: egy egyetemi menza előtt ebédidőben 25 perces sorok állnak. Legalább három okot és két hatást adj meg!
3. Mi a különbség a korlát és a feltételezés között? Mondj egy-egy példát a KoboCella esetéből!
4. Miért tekinthető a VárosiBringa gonosz problémának, a KoboCella pedig nem? Mi következik ebből a projekt ütemezésére?
5. Írj egy rendellenes forgatókönyvet a KoboCella számára arra az esetre, amikor a csavaradagoló kifogy!
6. Melyik mondat tartozik a ConOps-ba, és melyik nem? (a) „A karbantartó a heti ellenőrzést 20 perc alatt elvégzi.” (b) „A robot vezérlőszekrénye IP54 védettségű.” (c) „Ha a robot leáll, az operátor kézi üzemben folytatja a szerelést.”

Megoldás.

1. A megfogalmazás egy hiányzó megoldást nevez meg. A valódi probléma például: „A komissiózás átlagosan 4 óráig tart, és a tételek 2%-a hibás; emiatt a délután beérkező rendeléseket csak másnap tudjuk kiszállítani.” Ezt automata raktár, jobb raktári elrendezés vagy vonalkódos ellenőrzés is megoldhatja.
2. Például. Okok: egyetlen pénztár; a diákok a pult előtt választanak a menüből; minden diák 12:00 és 12:30 között érkezik (az órarend miatt). Hatások: a diákok elkésnek a délutáni óráról; sokan inkább nem esznek. A fő probléma: ebédidőben 25 perces várakozás.
3. A korlát kötelező feltétel, amelyről a tervezők nem dönthetnek (például a cellának el kell férnie a jelenlegi 6 × 4 méteres helyen). A feltételezés olyan állítás, amelyet igaznak fogadunk el, de ellenőrizni kell (például hogy a gyár MES-rendszere fogadja a cella adatait a meglévő interfészen).
4. A VárosiBringa problémáját különböző érdekelt felek eltérően értik, a beavatkozás maga is megváltoztatja a problémát, és nincs egyértelmű megoldása. A KoboCella célja jól megfogalmazható és mérhető, a megoldás kipróbálható. A VárosiBringa ezért próbaüzemmel és lépcsőzetes bővítéssel, rendszeres felülvizsgálattal ütemezendő.
5. Például: [FK-12] A csavaradagoló kiürül egy ciklus közben. A cella a ciklust befejezi a meglévő csavarokkal, ha lehet; ha nem, a darabot félkész állapotban, jelölve a kiadóba teszi. A kezelőpanel és a jelzőlámpa jelzi a hiányt. Az operátor feltölti az adagolót, nyugtázza a jelzést, és a félkész darabot a következő ciklusban befejezteti. A rendszer naplózza az eseményt a MES felé.
6. (a) és (c) a ConOps-ba tartozik, mert a használatot írja le. (b) a megoldás tulajdonsága, legfeljebb egy környezeti követelményből levezetett tervezési döntés.

Projektlépés: problémamegfogalmazás és működési koncepció

Saját fejlesztési feladatodhoz:

1. készíts problémafát (legalább egy fő probléma, négy ok, három hatás), és alakítsd át célfává; jelöld ki, mire összpontosít a projekt;
2. írd meg a problémamegfogalmazást a fejezetben bemutatott hét elemmel; a kívánt helyzet és a siker kritériumai legyenek mérhetők;
3. sorold fel a rendszer üzemmódjait, és rajzolj róluk állapotdiagramot;
4. írj legalább két normál és két rendellenes forgatókönyvet azonosítóval;
5. készíts egy használati eset leírást fő sikerforgatókönyvvel és legalább három kiterjesztéssel.

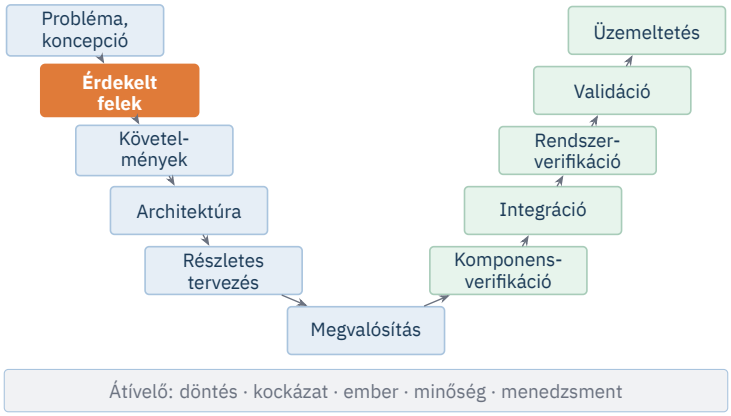
Az eszközöket és a sablonokat a Gyakorlati útmutató interjú- és ConOps-eszközlapjai tartalmazzák.

Továbbolvasás

- IEEE (1998): a ConOps-dokumentum klasszikus vázlata, részletes magyarázattal.
- ISO/IEC/IEEE (2018): a ConOps és az OpsCon helye a követelménymérnökség folyamatában.

- European Commission (2004): a problémafa és a célfa módszere a projektciklus-menedzsmentben.
- Kepner és Tregoe (1965): a problémaelemzés klasszikus módszere, a „van–nincs” specifikáció.
- Rittel és Webber (1973): a gonosz problémák fogalmának eredeti forrása.
- Cockburn (2001): a használati esetek írásának gyakorlati kézikönyve.
- NASA (2016), 4.1. szakasz és S függelék: a működési koncepció szerepe és egy lehetséges vázlata.

5. Érdekelt felek elemzése



Tanulási eredmények

A fejezet feldolgozása után képes leszel:

- meghatározni, kik egy rendszer érdekelt felei, és megkülönböztetni a megrendelőt, a felhasználót és a további érdekelteket;
- rendszerezetten azonosítani egy fejlesztés érdekelt feleit az életciklus-szakaszok és a hagymamodell segítségével, a „csendes” és a negatív érdekelteket is beleértve;
- priorizálni az érdekelt feleket a hatalom–érdekeltség rács és a fontossági (szaliencia-) modell alapján, és mindegyikhez bevonási stratégiát rendelni;
- kiválasztani és alkalmazni az elvárások feltárásának megfelelő módszereit (interjú, megfigyelés, workshop, kérdőív, adatelemzés, prototípus);
- Kano-kérdőívvel osztályozni az elvárásokat, és kiszámítani az elégedettségi együtthatókat;
- az elvárásokból szükséglet–cél–célkitűzés rendszert és a hatékonyság mérőszámait (MOE) levezetni.

A V-modell bal ága az érdekelt felekkel kezdődik (2.4. ábra). Ennek egyszerű oka van: egy rendszer akkor sikeres, ha valakinek értéket teremt, és nem árt indokolatlanul másoknak. Ha nem tudjuk, kik ezek a „valakik” és „mások”, akkor a legjobb műszaki megoldás is a semmibe vezethet.

A fejezet négy kérdésre épül:

- Kik az érdekelt felek, és miért nem elég a megrendelőre figyelni?
- Hogyan azonosíthatjuk őket rendszerezetten, hogy senki fontos ne maradjon ki?
- Kire mennyi figyelmet fordítsunk?
- Hogyan tárjuk fel, osztályozzuk és fogalmazzuk meg az elvárásaikat úgy, hogy a következő lépésben (6. fejezet) mérhető követelmények születhessenek belőlük?

A fejezet végén az érdekelt felek elvárásait a hatékonyság mérőszámaivá alakítjuk. Ezek lesznek a validáció mércéi a V jobb ágán.

5.1. Kik az érdekelt felek?

Az érdekelt fél (stakeholder) fogalma a stratégiai menedzsmentből érkezett a mérnöki gyakorlatba. R. Edward Freeman klasszikus meghatározása szerint érdekelt fél minden olyan csoport vagy egyén, amely befolyásolni tudja a szervezet céljainak elérését, vagy amelyet ez befolyásol (Freeman 1984). Az ISO/IEC/IEEE 15288 szabvány a rendszerfejlesztésre szabva fogalmaz: érdekelt fél az az egyén vagy szervezet, akinek joga, részesedése, igénye vagy érdeke fűződik a rendszerhez, illetve ahhoz, hogy a rendszer az igényeinek és elvárásainak megfelelő tulajdonságokkal rendelkezzen (ISO/IEC/IEEE 2023).

Mindkét definíció kétirányú. Érdekelt az is, aki *hat* a rendszerre (finanszírozza, engedélyezi, megépíti), és az is, *akire* a rendszer hat (használja, elszenvedi a zaját, elveszíti miatta a munkáját). A rendszermérnöki gyakorlatban három csoportot érdemes megkülönböztetni.

- **A megrendelő** az, aki a terméket kéri és jellemzően fizet érte. Az, hogy ki a megrendelő, a termékszerkezet szintjétől függ. Az ISO/IEC/IEEE 15288 megrendelő és szállító szerepekben gondolkodik, és ezek a szerepek a termékszerkezet minden szintjén, akár egy szervezeten belül is megjelennek (ISO/IEC/IEEE 2023, 6.1. szakasz). A rendszer legfelső szintjén a vásárló szervezet a megrendelő, egy alrendszer fejlesztője számára viszont az a csapat, amely az alrendszert a nagyobb egységbe integrálja.
- **A felhasználók és üzemeltetők** azok, akik a rendszert közvetlenül működtetik vagy használják. Ők érzékelik leginkább a rendszer erősségeit és hibáit. Gyakran mégsem ők döntenek a beszerzésről.
- **A további érdekelt felek** tágabb kereteket szabnak (hatóságok, szabványalkotók), életciklus-szolgáltatásokat nyújtanak (karbantartók, beszállítók, hulladékkezelők), vagy a rendszer hatásait viselik (szomszédok, a környezet, a jövő nemzedékei).

> **Igény, elvárás, követelmény**
>
> A három fogalmat a mindennapi nyelv gyakran összemossa. A rendszermérnökségben azonban mást jelentenek, és a különbség a V-modell lépéseit is tükrözi.
>
> - **Igény (szükséglet):** az a probléma vagy hiány, amelyet az érdekelt fél megoldani szeretne. Megoldásfüggetlen: *„biztonságosan és fáradság nélkül akarok feljutni a dombon lévő egyetemre”*.
> - **Elvárás:** az érdekelt fél elképzelése arról, mit és milyen feltételekkel nyújtson a rendszer. Lehet minőségi és pontatlan: *„legyen mindig kerékpár a közelben”*.
> - **Követelmény:** a rendszerre vonatkozó egyértelmű, mérhető, ellenőrizhető állítás, amelyet az elvárásokból vezetünk le: *„csúcsidőben az állomások legalább 90%-án legyen legalább két kölcsönözhető kerékpár”*. A követelményekkel a 6. fejezet foglalkozik.

Miért nem elég a megrendelőre figyelni? Mert a megrendelő csak egy hang a sok közül, és sokszor nem is ismeri a többiek szempontjait. Az érdekelt felek igényeinek feltárása a V-modell első technikai lépése, és minden további munka erre épül (ISO/IEC/IEEE 2023, 6.4.2. szakasz). Ha a megrendelő, a többi érdekelt fél és a rendszermérnök a projekt elején közösen rögzíti, mit fog tudni a termék, azzal megelőzhető, hogy a követelmények a fejlesztés során ellenőrizetlenül bővüljenek (az angol szakirodalom ezt *requirements creep* néven ismeri). A kimaradt érdekelt fél nem tűnik el. Később jelentkezik, jellemzően a legrosszabb pillanatban: a hatósági engedélyezésnél, az üzembe helyezéskor vagy egy sajtóhírben.

VárosiBringa: a kezdeti lista

Az önkormányzat projektcsapata az első megbeszélésen négy érdekelt felet sorolt fel: a város vezetését, az üzemeltetőt, a „felhasználókat” és a közlekedési társaságot. Egy órás, rendezett gondolkodás után a lista harmincnál több szereplőre bővült. Ide kerültek többek között:

- az idősebb és a mozgáskorlátozott lakók;
- a belvárosi boltosok, akik a dokkolók helyét féltik;
- a gyalogosok, akiket a járdára hagyott kerékpárok zavarnak;
- a rendőrség;
- az energiaszolgáltató;
- a kerékpárokat rongáló csoportok.

A felhasználók csoportja is szétvált diákokra, ingázókra és turistákra, mert eltérő a napszakos használatuk, a fizetési szokásuk és a technikai jártasságuk. A következő szakasz azt mutatja meg, hogyan lehet ezt a bővítést szisztematikusan, nem pedig szerencsén múlóan elvégezni.

5.2. Az érdekelt felek rendszerezett azonosítása

Az érdekelt felek azonosítása kreatív és fegyelmezett munka egyszerre. A kreativitás ahhoz kell, hogy a nem nyilvánvaló szereplők is eszünkbe jussanak, a fegyelem ahhoz, hogy a keresés teljes legyen. Három, egymást kiegészítő nézőpont segít ebben: az életciklus, a hagymamodell és néhány célzott ellenőrző kérdés.

5.2.1. Az életciklus nézőpontja

Az érdekelt felek köre a rendszer életciklusa során változik. Az ISO/IEC/IEEE 15288 szabvány ezért azt írja elő, hogy azokat az érdekelt feleket kell azonosítani, akiknek a rendszer *teljes életciklusa* során érdekük fűződik hozzá (ISO/IEC/IEEE 2023, 6.4.2. szakasz). A 5.1. táblázat a 2.1. szakasz életciklus-szakaszai szerint sorolja fel a jellemző érdekelteket. Ha a listánkon nincs olyan szereplő, aki a gyártással, a karbantartással vagy a kivonással foglalkozik, akkor valószínűleg kihagytunk valakit.

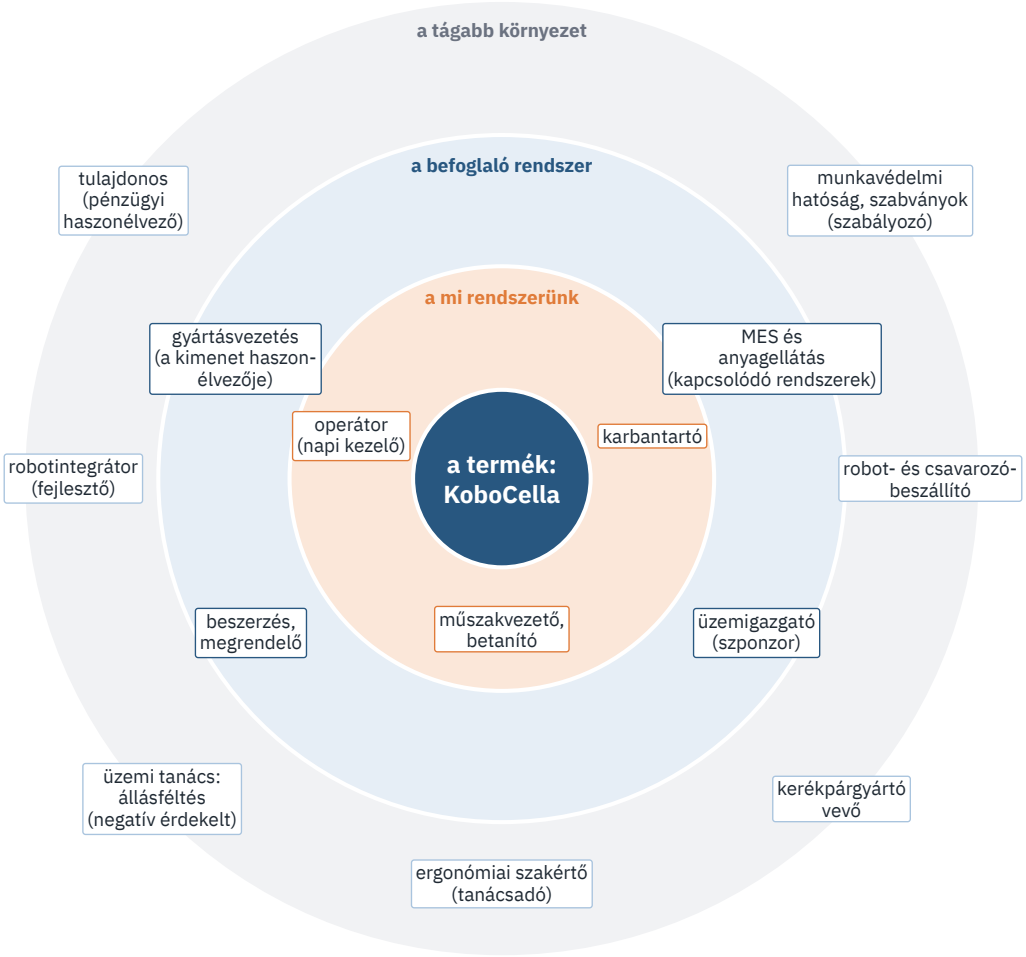
5.1. táblázat. Jellemző érdekelt felek az életciklus-szakaszokban. Saját összeállítás.

Életciklus-szakasz	Jellemző érdekelt felek
Koncepció	Szponzor és stratégiai döntéshozók, megrendelő, potenciális felhasználók, finanszírozók, szabályozó hatóságok
Fejlesztés	Megrendelő, mérnöki szakterületek, biztonsági és munkavédelmi szakemberek, beszállítók, gyártás-előkészítés, tesztelők
Gyártás	Gyártás, minőségügy, logisztika, beszállítók, munkavállalók és érdekképviseletük
Használat	Felhasználók és üzemeltetők, ügyfélszolgálat, a kimenet haszonélvezői, a rendszer hatásait viselő szomszédok
Támogatás	Karbantartók, alkatrészellátás, betanítók, szoftverfrissítést végzők
Kivonás	Hulladékkezelők és újrahasznosítók, környezetvédelmi hatóságok, a közvélemény

5.2.2. A hagymamodell

Ian Alexander a rendszer köré rajzolt koncentrikus körökkel, „hagymával” rendszerezi az érdekelt feleket (Alexander 2005). A középpontban maga a termék áll, amely önmagában nem érdekelt fél. Körülötte három gyűrű helyezkedik el (5.1. ábra).

- **A mi rendszerünk** gyűrűjében azok vannak, akik a terméket közvetlenül működtetik: a napi kezelők, a karbantartók és az üzemeltetést segítők (betanítók, ügyfélszolgálat). A termék és ezek a szerepek együtt alkotják azt a működő rendszert, amely a szolgáltatást ténylegesen nyújtja.
- **A befoglaló rendszer** gyűrűjébe tartoznak a termék kimenetének haszonélvezői, a kapcsolódó rendszerekért felelősök, a megrendelő és a projekt szponzora, aki a fejlesztést elindítja, finanszírozza és védi.
- **A tágabb környezet** gyűrűjében találjuk a pénzügyi és a politikai haszonélvezőket, a szabályozókat, a fejlesztőket, a tanácsadókat, a beszállítókat, valamint a **negatív érdekelteket**. Ők azok, akiknek a rendszer árthat, vagy akik ártani akarnak neki.



5.1. ábra. Az érdekelt felek hagymamodellje a KoboCella példáján. Alexander (2005) szerepkategóriái alapján, saját szerkesztés.

A hagymamodell legfőbb haszna, hogy **szerepekben** gondolkodtat, nem személyekben. Ugyanaz a személy több szerepben is megjelenhet: egy műszakvezető egyszerre lehet kezelő, betanító és a változtatással szembeni ellenállás hangja. Egy szerepet pedig több csoport is betölthet. Alexander megfigyelése szerint a projektek kudarca gyakran nem

rossz követelményekből fakad, hanem abból, hogy egy szerep nézőpontja teljesen kimaradt. Ilyenkor a hozzá tartozó forgatókönyvek és követelmények egész csoportja hiányzik (Alexander 2005).

5.2.3. Ellenőrző kérdések

Az alábbi kérdések a „ki maradt ki?” vizsgálatot segítik. Érdemes mindegyiket feltenni, és minden válaszhoz konkrét csoportot vagy szervezetet rendelni.

1. Ki **fizet** a rendszerért, és ki **fizeti az üzemeltetését**?
2. Ki **használja** naponta, alkalmanként, kivételes helyzetben? Ki használja **rendellenes körülmények között** (hiba, vészhelyzet, rossz idő)?
3. Ki **építi, telepíti, karbantartja, frissíti és vonja ki**?
4. Ki **szabályozza, engedélyezi, ellenőrzi**?
5. Kinek az **érdekeit sérti**, kit **zavar**, kinek a **munkáját változtatja meg**?
6. Kinek a **rendszerével kapcsolódik**? Melyik meglévő folyamatot, adatot, infrastruktúrát érinti?
7. Kinek **nincs hangja**, mégis érinti? Ilyen a természeti környezet, a jövő nemzedékei, a fogyatékossággal élők, a digitálisan kevésbé jártasak.
8. Ki **akadályozhatja** a sikert, ha kimarad (hatóság, munkavállalói érdekképviselet, sajtó, közösségi média)?

> **A „felhasználó” mint egyetlen csoport**
>
> Az egyik leggyakoribb hiba, hogy a felhasználókat egyetlen homogén csoportként kezeljük. A VárosiBringa diák felhasználói este és hétvégén, mobilról, bankkártyával fizetnek. Az ingázók reggel és délután, bérlettel. A turisták alkalmanként, idegen nyelvű felületet és egyszerű regisztrációt várnak. Az idősebb lakók számára a nagy betűméret, a stabil kerékpár és a személyes segítség lehetősége a döntő. Ezek a csoportok eltérő elvárásokat, eltérő forgatókönyveket és eltérő követelményeket jelentenek. A felhasználói csoportok jellemzésére a 11. fejezet perszónái adnak eszközt.

5.3. Kire mennyi figyelmet? Az érdekelt felek priorizálása

Harminc érdekelt fél mindegyikével nem lehet ugyanolyan intenzitással foglalkozni. A priorizálás nem azt jelenti, hogy egyes érdekelteket figyelmen kívül hagyunk. Azt jelenti, hogy tudatosan választjuk meg, kit vonunk be a döntésekbe, kit tájékoztatunk, és kinek a helyzetét figyeljük. Két egymást kiegészítő eszközt mutatunk be.

5.3.1. A hatalom–érdekeltség rács

A rács az érdekelt feleket két szempont szerint helyezi el: mekkora a **hatalmuk** (befolyásuk) a rendszer sorsára, és mekkora az **érdekeltségük**, vagyis mennyire foglalkoztatja őket a rendszer. Az ötletet Mendelow vetette fel a környezetfigyelés kapcsán (Mendelow 1981). A négy negyedet és a hozzájuk tartozó stratégiákat Eden és Ackermann dolgozta ki (Eden és Ackermann 1998; Ackermann és Eden 2011).

- **Kulcsszereplők** (players, nagy hatalom és nagy érdekeltség): velük szorosan együtt kell dolgozni, és be kell vonni őket a döntésekbe.

- **Érintettek** (subjects, kis hatalom, nagy érdekeltség): tájékoztatni és bevonni kell őket. Az érdekeiket gyakran csak összefogva tudják érvényesíteni, ezért érdemes számolni a szövetségeikkel.
- **Keretadók** (context setters, nagy hatalom, kis érdekeltség): elégedettségüket fenn kell tartani, és érdemes felkelteni az érdeklődésüket. Gyakran éppen ők alakítják a projekt környezetét (szabályozás, finanszírozás).
- **Tömeg** (crowd, kis hatalom és kis érdekeltség): elég figyelemmel kísérni őket, de számolni kell azzal, hogy helyzetük megváltozhat.

Ackermann és Eden arra is felhívja a figyelmet, hogy a rács nem statikus térkép. Egy érdekelt fél helyzete változik, sőt tudatosan változtatható (Ackermann és Eden 2011). Egy támogató, de gyenge érintett csoport megerősíthető, és így kulcsszereplővé válhat. Egy közömbös keretadó érdeklődése felkelthető. Egy esetleges ellenfél pedig korai bevonással semlegesíthető.

Kidolgozott példa: a VárosiBringa hatalom–érdekeltség rácsa

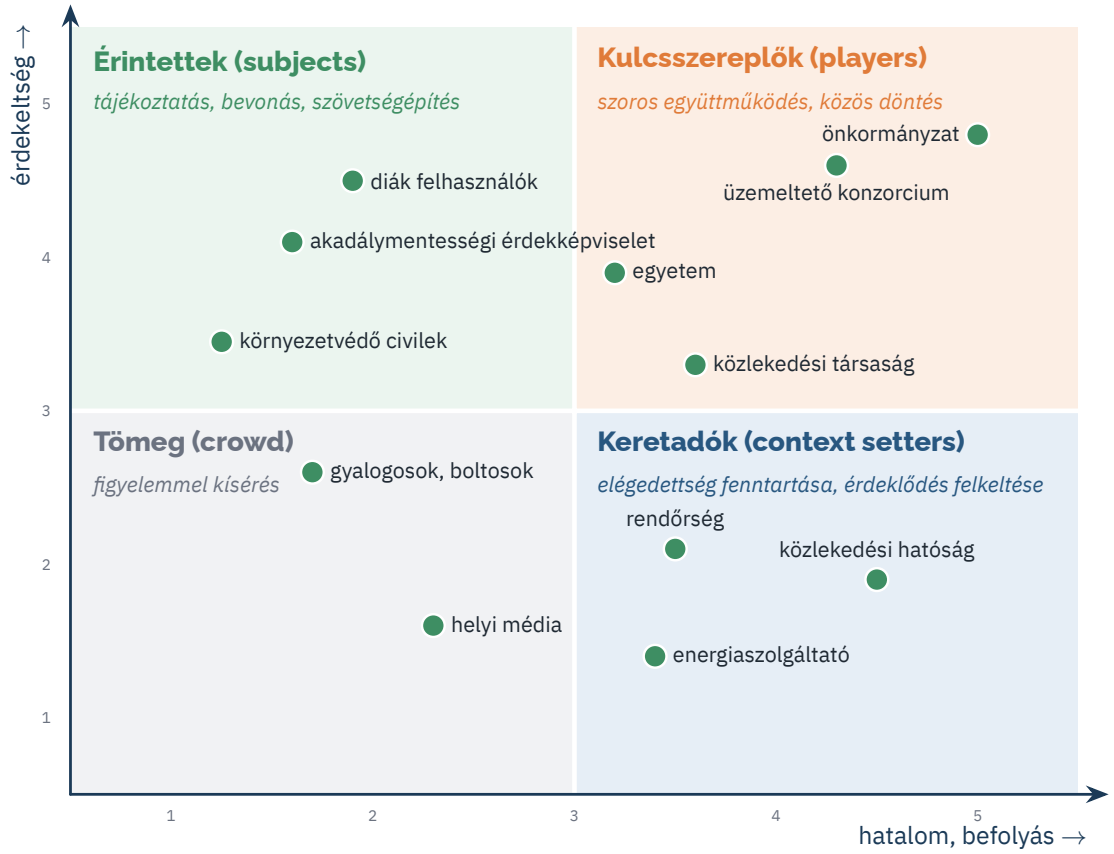
A projektcsapat négy tagja egymástól függetlenül pontozta a tizenkét legfontosabbnak ítélt érdekelt felet 1-től 5-ig mindkét szempont szerint. Ezután megbeszélték az eltéréseket, és az átlagot rögzítették (5.2. táblázat). A pontozás szempontjai a következők voltak.

- **Hatalom:** van-e döntési jogköre (engedély, finanszírozás, szerződés), rendelkezik-e nélkülözhetetlen erőforrással (terület, hálózat, ügyfélkör), és képes-e mozgósítani a közvéleményt.
- **Érdekeltség:** mennyire változtatja meg a rendszer a mindennapjait vagy az üzleti eredményét, és mennyire foglalkozik vele aktívan.

5.2. táblázat. A VárosiBringa érdekelt feleinek pontozása (a csapat átlaga, szemléltető adatok).

Érdekelt fél	Hatalom	Érdekeltség	Negyed	Indoklás (röviden)
Önkormányzat	5,0	4,8	kulcsszereplő	finanszíroz, közterületet biztosít, politikai felelősséget visel
Üzemeltető konzorcium	4,3	4,6	kulcsszereplő	szerződéses partner, az üzleti eredmény az övé
Közlekedési társaság	3,6	3,3	kulcsszereplő	bérletrendszer, átszállópontok; partner és versenytárs is
Egyetem	3,2	3,9	kulcsszereplő	sok potenciális felhasználó, telephelyet adhat
Diák felhasználók	1,9	4,5	érintett	a fő célcsoport, de szervezetlen
Akadálymentességi érdekképviselet	1,6	4,1	érintett	jogi és erkölcsi súly, kis közvetlen hatalom
Környezetvédő civilek	1,25	3,45	érintett	támogatók, a közvéleményt mozgósíthatják
Gyalogosok, boltosok	1,7	2,6	tömeg	a járdahasználat miatt később felerősödhet
Helyi média	2,3	1,6	tömeg	egy-egy rossz hír felnagyíthatja a problémákat
Energiaszolgáltató	3,4	1,4	keretadó	hálózati csatlakozás, töltési díjak
Közlekedési hatóság	4,5	1,9	keretadó	jóváhagyások, közúti szabályok
Rendőrség	3,5	2,1	keretadó	közterületi rend, lopások, balesetek

Az eredményt a 5.2. ábra mutatja. Két tanulság rögtön adódik. Az akadálymentességi érdekképviselet és a környezetvédők érintettek: kis hatalmuk miatt könnyű lenne őket csak tájékoztatni. Korai bevonásukkal azonban a projekt hiteles szövetségeseket nyerhet, és elkerülheti a későbbi, költséges áttervezést, például a dokkolók akadálymentes kialakításánál. A gyalogosok és a boltosok ma a tömegbe tartoznak, de ha a kerékpárok elárasztják a járdákat, gyorsan érintettekké, a helyi média segítségével akár keretadókká válhatnak.



5.2. ábra. A VárosiBringa érdekelt feleinek hatalom–érdekeltség rácsa. A negyedek elnevezése Eden és Ackermann (1998) és Ackermann és Eden (2011) alapján; a bevonási stratégiák a szakirodalom általános ajánlásait követik. Szemléltető adatok, saját szerkesztés.

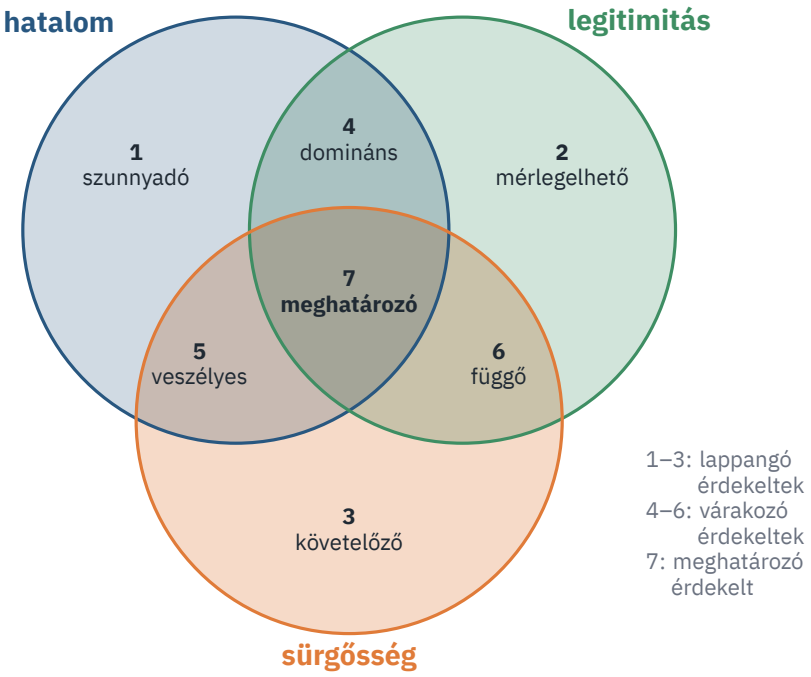
5.3.2. A fontossági (szaliencia-) modell

A hatalom–érdekeltség rács egyszerű és szemléletes, de két fontos szempontot nem lát: azt, hogy jogos-e az érdekelt fél követelése, és azt, hogy mennyire sürgető. Mitchell, Agle és Wood ezért három tulajdonság alapján határozza meg, mennyire „fontos” (szalienciás) egy érdekelt fél a döntéshozók számára (Mitchell, Agle, és Wood 1997):

- **hatalom:** képes-e elérni, hogy akaratát érvényesítse (kényszerrel, erőforrásokkal vagy szimbolikus eszközökkel);
- **legitimitás:** társadalmilag elfogadott, jogos-e a követelése (törvény, szerződés, erkölcsi jog);
- **sürgősség:** mennyire időkritikus a követelése, és mennyire fontos neki.

A három tulajdonság kombinációi hét érdekeltfél-osztályt adnak (5.3. ábra, 5.3. táblázat). Minél több tulajdonság jellemző valakire, annál nagyobb figyelmet kap a döntéshozóktól.

A modell legfontosabb tanulsága azonban a *dinamika*. Az érdekelt fél egyik osztályból a másikba léphet, ha megszerez egy tulajdonságot, és ez a projekt szempontjából előre jelezhető kockázat.



5.3. ábra. Az érdekelt felek fontossági modellje: a hatalom, a legitimitás és a sürgősség kombinációiból adódó hét osztály. Mitchell, Agle, és Wood (1997) alapján, saját szerkesztés.

5.3. táblázat. A fontossági modell hét osztálya példákkal. Az osztályok Mitchell, Agle, és Wood (1997) szerint; a magyar elnevezések saját fordítások.

Osztály (angol név)	Tulajdonságok	Példa a futó esetekből
Szunnyadó (dormant)	hatalom	Az anyavállalat központi beszerzése, amely jelenleg nem foglalkozik a KoboCellával, de leállíthatja a beruházást
Mérlegelhető (discretionary)	legitimitás	A ForgácsTech egyik kisebb vevője, amely szívesen látna online rendelési nyomkövetést
Követelőző (demanding)	sürgősség	Egy VárosiBringa-dokkoló melletti lakó, aki minden nap panaszt tesz a zaj miatt
Domináns (dominant)	hatalom + legitimitás	A munkavédelmi hatóság a KoboCella esetében
Veszélyes (dangerous)	hatalom + sürgősség	A VárosiBringa kerékpárjait szervezetten rongáló csoport
Függő (dependent)	legitimitás + sürgősség	A KoboCella operátorai, akiknek egészségi panaszai jogosak és sürgetőek, de döntési hatalmuk nincs
Meghatározó (definitive)	mindhárom	A KoboCella üzemigazgatója (szponzor, határidővel), vagy az operátorok, ha az üzemi tanács felkarolja ügyüket

KoboCella: a függő érdekeltből meghatározó lesz

Az operátorok kezdetben függő érdekeltek: követelésük, a fizikai terhelés csökkentése, legitim és sürgető, de hatalmuk nincs. Ha az üzemi tanács felkarolja az ügyet, és munkavédelmi vizsgálatot kezdeményez, az operátorok hatalmat is szereznek, és meghatározó érdekelté válnak. A projektcsapat ezt a fordulatot nem kivárja, hanem megelőzi. Két operátort a kezdetektől bevon a cella kialakításába. Ez nemcsak a konfliktust csökkenti: a tapasztalt operátorok olyan rejtett tudást hoznak (például a fedél illesztésének apró trükkjeit), amely nélkül a funkciómegosztás rossz lenne (12. fejezet).

A két modell eredménye végül egy **bevonási tervben** összegződik. A terv minden érdekelt félhez megadja:

- a bevonás módját (közös döntés, konzultáció, tájékoztatás, figyelés);
- a csatornát;
- a gyakoriságot;
- a felelős kapcsolattartót.

A Gyakorlati útmutató *Hatalom–érdekeltség és fontossági elemzés* eszközlapja ehhez sablont is ad.

5.4. Az elvárások feltárása

Ha tudjuk, kik az érdekelt felek, meg kell tudnunk, mit várnak a rendszertől. Ez nehezebb, mint amilyennek látszik. Az emberek nem mindig tudják megfogalmazni, mire van szükségük. Gyakran egy megoldást mondanak a probléma helyett. A legfontosabb elvárásaikat pedig sokszor meg sem említik, mert annyira magától értetődőnek tartják őket. A feltárásnak ezért több módszere van, és a jó gyakorlat ezeket kombinálja (5.4. táblázat).

5.4. táblázat. Az elvárások feltárásának módszerei. Az első öt sor Dumas és mtsai. (2018) folyamatfeltárási módszereit követi, saját bővítéssel.

Módszer	Erősségek	Korlátok	Mikor különösen hasznos?
Dokumentum-elemzés	Strukturált információ; nem igényli az érdekeltek idejét	Elavult lehet; nem a megfelelő részletességű	Korai tájékozódás, szabályozási háttér
Helyszíni megfigyelés	Gazdag, a környezetbe ágyazott kép; rejtett tudás feltárása	Időigényes; a megfigyeltek másképp viselkedhetnek; kevés eset	Munkafolyamatok, kezelési helyzetek
Interjú	Részletes, mély megértés	Az érdekeltek idejét igényli; több kör kell	Kulcsszereplők, szakértők, konfliktusok feltárása
Workshop	Az eltérő nézetek közvetlen ütköztetése és feloldása	Sok érdekelt egyidejű részvételét igényli	Ellentmondó elvárások, közös célkép

Módszer	Erősségek	Korlátok	Mikor különösen hasznos?
Kérdőív	Sok válaszadó, számszerűsíthető eredmény	Új szempontot ritkán tár fel; alacsony válaszarány	Elvárások súlyozása, Kano-elemzés
Adatelemzés, automatikus feltárás	Nagy esetszám, objektív adatok	Adatminőség; csak a meglévő működést mutatja	Működő rendszer vagy folyamat fejlesztése
Prototípus, makett	Konkrét visszajelzés; a ki nem mondott elvárások felszínre kerülnek	Korán egy megoldás felé terelhet	Kezelőfelületek, ergonómia, szolgáltatás-élmény

5.4.1. Igény a megoldás mögött

A feltárás egyik legfontosabb készsége, hogy a kimondott megoldás mögött meghalljuk a valódi igényt. Ha egy érdekelt fél megoldást mond, mindig érdemes megkérdezni: *mit tenne lehetővé ez a megoldás, és miért fontos ez?* Ez nem jelenti a javaslat elutasítását. Azt jelenti, hogy a megoldást az igényhez mérjük, és nyitva hagyjuk a lehetőséget más, esetleg jobb megoldások előtt.

ForgácsTech: „kell egy tablet minden gép mellé”

Az egyik interjúban egy tapasztalt CNC-gépkezelő határozottan kijelentette, hogy minden gép mellé tablet kell. A beszélgetés során kiderült, mi áll e mögött:

- naponta többször is az irodába kell mennie a következő rendelés műhelyrajzáért;
- a papírrajzok között előfordul, hogy nem a legfrissebb változat van nála;
- a szerszámcseréhez szükséges adatokat egy füzetben vezeti.

A valódi igény tehát így szól: *„a következő feladat aktuális rajzához és technológiai adataihoz a gép mellől, keresgélés nélkül akarok hozzáférni”*. Erre a tablet csak az egyik lehetséges válasz. Ugyanígy szóba jöhet a gépvezérlőn megjelenő munkautasítás vagy egy közös terminál a gépcsoportnál. Az igény megfogalmazása a változatok összevetésének lehetőségét is megnyitja (10. fejezet).

5.4.2. Az interjú gyakorlata

Az interjú a leggyakrabban használt feltárási módszer. Néhány bevált szabály jelentősen javítja az eredményét.

- **Készülj fel**, de ne kérdőívet olvass fel. Egy rövid témavázlat jobb, mint egy merev kérdéssor.
- **Konkrét esetekről kérdezz.** A „Hogyan szokta…?” kérdés általánosságokat hoz. A „Mesélje el, legutóbb mi történt, amikor…” kérdés valós helyzetet, valós problémákat.
- **Kerüld a sugalmazó kérdéseket.** A „Ugye fontos lenne, ha…?” kérdésre szinte mindenki igennel felel.
- **Kérdezd meg, miért** – türelmesen, akár többször is –, amíg a megoldástól el nem jutsz az igényig.
- **Kérdezz a kivételekről.** Mi történik hiba, rossz idő, csúcsidő, betegség esetén? A rendellenes helyzetek gyakran a legfontosabb követelményeket hozzák.

- **Rögzíts szó szerint**, és az interjú végén foglald össze, amit megértettél. Az összefoglalót küldd vissza jóváhagyásra. Ez az elvárás első validálása.

> **Nagy nyelvi modellek az érdekeltfél-elemzésben**
>
> A nagy nyelvi modellek gyorsan készítenek hosszú listákat az érdekelt felekről és tipikus elvárásaikról, és interjúvázlatok, kérdőívek megfogalmazásában is segíthetnek. Ez hasznos ellenőrző lista lehet, különösen a „ki maradt ki?" kérdéshez. Két korlátjukat azonban látni kell. Listáik a tipikusat adják vissza, nem a konkrét helyzetet: a helyi szereplők, a nem nyilvános konfliktusok és a kimondatlan érdekek nem szerepelnek bennük. Másodszor a modell válasza nem forrás. Egy érdekelt fél elvárását csak maga az érdekelt fél erősítheti meg. A Gyakorlati útmutató mintapromptjai ezért a modell kimenetét mindig emberi ellenőrzéssel és az érdekeltekkel való egyeztetéssel kapcsolják össze.

5.5. Az elvárások osztályozása: a Kano-modell

Nem minden elvárás egyforma. Vannak, amelyeknek a hiánya felháborodást kelt, a megléte viszont senkit sem lelkesít. Vannak, amelyek minél jobban teljesülnek, annál elégedettebb a felhasználó. És vannak, amelyekre senki sem számít, de ha megjelennek, meglepetésszerű örömet okoznak. Noriaki Kano és munkatársai 1984-ben megjelent modellje ezt a nem lineáris összefüggést teszi kezelhetővé (Kano és mtsai. 1984). A modell a termékjellemzőket aszerint osztályozza, hogyan függ az elégedettség a jellemző teljesülésétől (5.4. ábra).

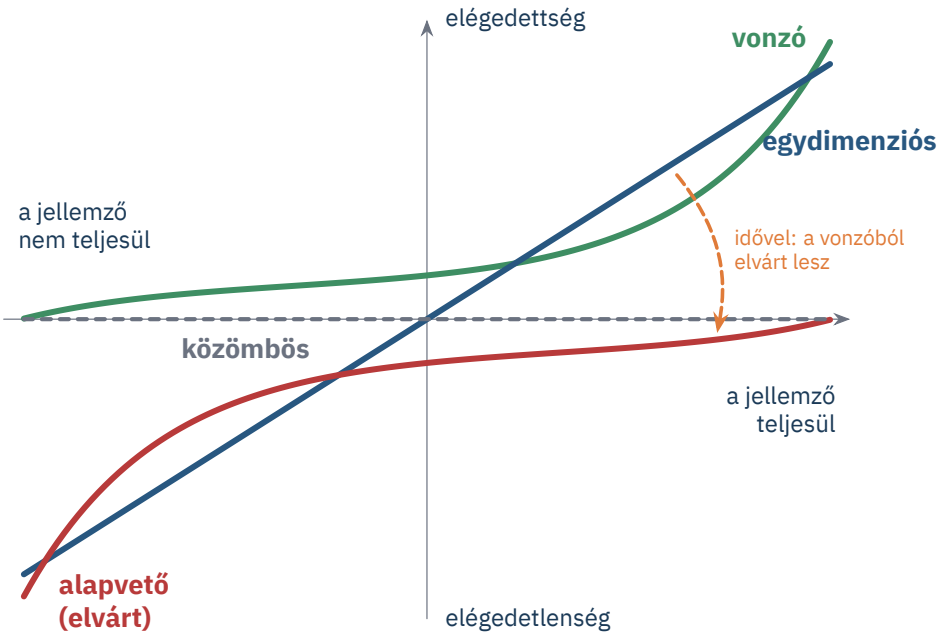
- **Alapvető (elvárt) jellemzők** (must-be): hiányuk erős elégedetlenséget okoz, meglétük viszont csak a semleges állapotot biztosítja. Ezek a piacra lépés feltételei.
- **Egydimenziós jellemzők** (one-dimensional): az elégedettség nagyjából arányosan nő a teljesülés mértékével. Erről szólnak a versenytársakkal való összehasonlítások.
- **Vonzó jellemzők** (attractive): hiányukat senki sem kéri számon, meglétük viszont kiemelkedő elégedettséget okoz.
- **Közömbös jellemzők** (indifferent): sem meglétük, sem hiányuk nem befolyásolja érdemben az elégedettséget.
- **Fordított jellemzők** (reverse): egyesek számára éppen a hiányuk kívánatos.

Az osztályok nem állandók. A ma vonzó jellemzőből idővel egydimenziós, majd alapvető elvárás lesz, ahogy a felhasználók hozzászoknak. A valós idejű elérhetőség jelzése a közösségi kerékpárrendszerekben néhány éve még újdonság volt, ma már inkább elvárásnak számít. Ezt az életciklus-jelenséget a szakirodalom külön is vizsgálta (Löfgren, Witell, és Gustafsson 2011). A gyakorlati következmény az, hogy a Kano-elemzést időről időre meg kell ismételni.

5.5.1. A Kano-kérdőív és kiértékelése

Az osztályozáshoz jellegzetes kérdőívet használnak (Berger és mtsai. 1993). Minden jellemzőről két kérdést teszünk fel:

- **funkcionális kérdés:** *„Mit érezne, ha a kerékpár valós idejű elérhetősége látszana az alkalmazásban?"*
- **diszfunkcionális kérdés:** *„Mit érezne, ha a kerékpár valós idejű elérhetősége nem látszana az alkalmazásban?"*



5.4. ábra. A Kano-modell: az elégedettség és a jellemző teljesülésének kapcsolata a különböző jellemzőtípusoknál. Kano és mtsai. (1984) és Berger és mtsai. (1993) alapján, saját szerkesztés.

Mindkét kérdésre öt válasz közül lehet választani: *tetszene*; *elvárom, magától értetődő*; *közömbös*; *elviselném*; *nem tetszene*. A két válasz kombinációja a 5.5. táblázat szerint adja a jellemző osztályát az adott válaszadónál.

5.5. táblázat. A Kano-kiértékelő táblázat. Jelölések: A = alapvető, E = egydimenziós, V = vonzó, Kö = közömbös, F = fordított, K = kétséges (ellentmondó válasz). Berger és mtsai. (1993) alapján.

Funkcionális \ diszfunkcionális	Tetszene	Elvárom	Közömbös	Elviselném	Nem tetszene
Tetszene	K	V	V	V	E
Elvárom	F	Kö	Kö	Kö	A
Közömbös	F	Kö	Kö	Kö	A
Elviselném	F	Kö	Kö	Kö	A
Nem tetszene	F	F	F	F	K

A válaszadók osztálybesorolásait jellemzőnként összesítjük. A jellemző osztálya jellemzően a leggyakoribb kategória. Ha két kategória közel azonos gyakoriságú, az a felhasználói csoportok eltérő ízlésére utal, és érdemes a csoportokat külön elemezni. Az eredmény kifejezőbb, ha kiszámítjuk az elégedettségi együtthatókat is (Berger és mtsai. 1993). A képletekben V, E, A és $Kö$ a vonzó, az egydimenziós, az alapvető és a közömbös besorolások száma (az angol szakirodalom ugyanezt A, O, M és I betűkkel jelöli):

$$CS^{+} = \frac{V + E}{V + E + A + Kö}, \qquad CS^{-} = -\frac{E + A}{V + E + A + Kö}. \tag{5.1}$$

A CS^{+} (0 és 1 között) azt mutatja, mennyire növeli az elégedettséget a jellemző megléte. A CS^{-} (–1 és 0 között) azt, mennyire csökkenti a hiánya. A fordított és a kétséges válaszokat

a képlet nem veszi figyelembe.

Kidolgozott példa: a VárosiBringa Kano-elemzése

A projektcsapat 120 lakóval töltetett ki Kano-kérdőívet öt lehetséges szolgáltatásjellemzőről. A válaszok összesítését és az együtthatókat a 5.6. táblázat mutatja (szemléltető adatok).

5.6. táblázat. Kano-elemzés a VárosiBringa öt szolgáltatásjellemzőjére (120 válaszadó, szemléltető adatok). V = vonzó, E = egydimenziós, A = alapvető, Kö = közömbös, F = fordított, K = kétséges.

Jellemző	V	E	A	Kö	F	K	Osztály	CS^+	CS^-
Megbízható zár és kölcsönzés	6	22	80	10	0	2	alapvető	0,24	−0,86
Valós idejű elérhetőség az alkalmazásban	18	62	28	10	0	2	egydimenziós	0,68	−0,76
Közös bérlet a helyi buszokkal	58	24	8	26	2	2	vonzó	0,71	−0,28
Telefontöltő a kormányon	30	10	4	70	4	2	közömbös	0,35	−0,12
Hangjelzés a kerékpár indulásakor	6	6	4	54	46	4	közömbös (sok fordított)	0,17	−0,14

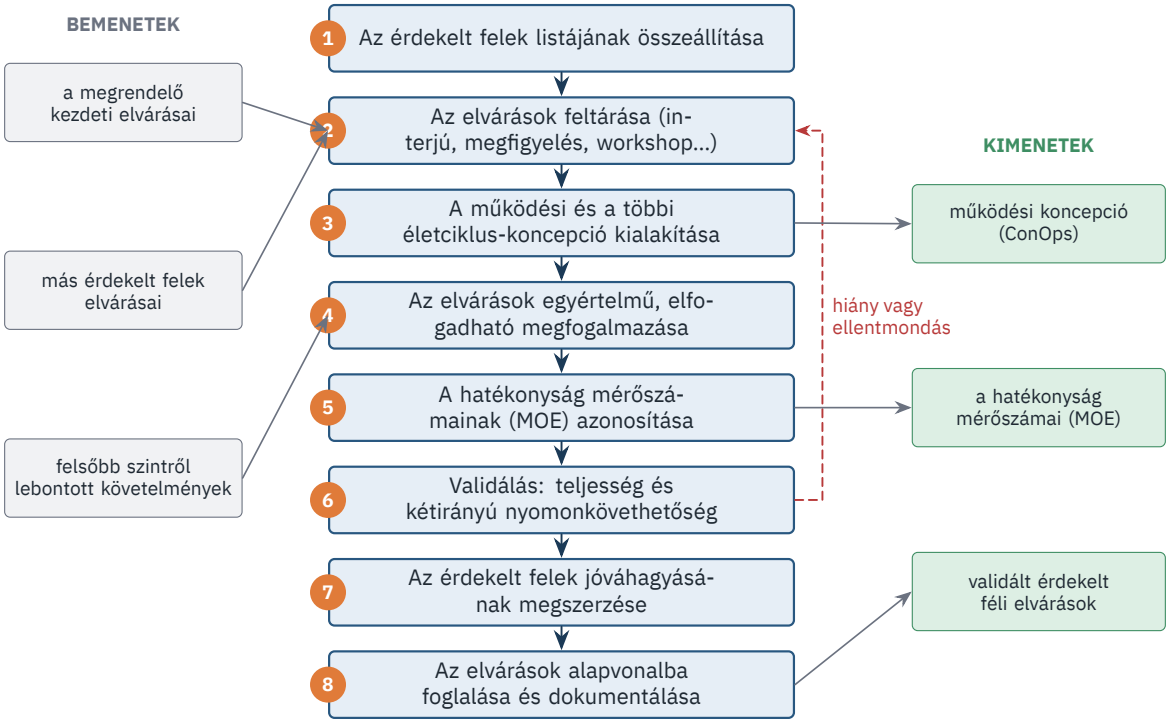
Nézzük például a közös bérletet. Itt $V = 58$, $E = 24$, $A = 8$, $Kö = 26$, így a nevező $58 + 24 + 8 + 26 = 116$. Ebből $CS^+ = (58 + 24)/116 \approx 0{,}71$ és $CS^- = -(24 + 8)/116 \approx -0{,}28$. A jellemző tehát nagy elégedettséget okozhat, hiánya viszont alig okoz elégedetlenséget: tipikus vonzó jellemző.
A tanulságok a tervezés számára:

- A **megbízható zár és kölcsönzés** alapvető. Hiánya súlyos elégedetlenséget okoz ($CS^- = -0{,}86$), túlteljesítése viszont nem hoz többletelégedettséget. A cél a biztos teljesítés, nem a kiemelkedés.
- A **valós idejű elérhetőség** egydimenziós. Itt érdemes a versenyképes szintet keresni, mert minden javulás megtérül az elégedettségben.
- A **közös bérlet** vonzó. Nem kötelező, de a projekt egyik leghatásosabb megkülönböztető eleme lehet, és ráadásul a közlekedési társaság, egy kulcsszereplő együttműködését is igényli.
- A **telefontöltő** a válaszadók többségének közömbös. Erőforrást csak akkor érdemes rá fordítani, ha olcsó.
- A **hangjelzésnél** a közömbös és a fordított válaszok aránya közel azonos. Ez megosztottságra utal. Érdemes megvizsgálni, mely csoportok utasítják el (például a csendes lakóövezetek lakói), és lehet-e a jelzést helyzettől függővé tenni.

A Kano-modell jól kapcsolódik a következő lépésekhez. Az osztályok segítenek a vevői elvárások súlyozásában a minőség házában (7. fejezet). Matzler és Hinterhuber ezt az integrációt kifejezetten javasolja (Matzler és Hinterhuber 1998). A modell a koncepciók értékelésében is segít (10. fejezet). A Gyakorlati útmutató *Kano-elemzés* eszközlapja kérdőívsablont, kiértékelő táblázatot és egy interaktív kalkulátort is tartalmaz.

5.6. Az elvárásoktól a mérhető célokig

Az érdekelt felek elvárásai még nem követelmények. Pontatlanok, gyakran ellentmondanak egymásnak, és nem mindig mérhetők. A V-modell második lépésének feladata, hogy ezekből egy egyeztetett, nyomon követhető és jóváhagyott elvárásrendszer legyen, amely a követelmények meghatározásának bemenete. Az ISO/IEC/IEEE 15288 szabvány erre önálló folyamatot határoz meg (ISO/IEC/IEEE 2023, 6.4.2. szakasz; NASA 2016, 4.1. szakasz). Tevékenységeit és feladatait nyolc lépésbe rendezve a 5.5. ábra mutatja.



5.5. ábra. Az érdekelt felek elvárásainak meghatározási folyamata bemenetekkel és kimenetekkel. Az ISO/IEC/IEEE 15288 szabvány 6.4.2. pontja alapján, saját szerkesztés (ISO/IEC/IEEE 2023; NASA 2016, 4.1–1. ábra).

5.6.1. Szükséglet, cél, célkitűzés

Az elvárások rendezésének bevált kerete a szükséglet–cél–célkitűzés hierarchia. Kidolgozásához az INCOSE igény- és követelménykezelési útmutatója részletes eljárást ad: a probléma megfogalmazásából indul, és a küldetésen, a célokon és a célkitűzéseken át jut el a mérőszámokig (Wheatcraft, Ryan, és Edner Katz 2025).

- A **szükséglet** egyetlen állítás arról, milyen problémát akarunk megoldani. Nem tartalmazza a megoldást.
- A **célok** kifejtik, mit kell elérni a szükséglet kielégítéséhez. Nem kell számszerűeknek lenniük, de lehetővé kell tenniük annak megítélését, hogy elértük-e őket.
- A **célkitűzések** konkrét, mérhető célszintek, amelyek egy-egy célhoz kapcsolódnak.

A jó célkitűzés Doran sokat idézett szempontjai szerint SMART (Doran 1981):

1. **konkrét** (specific): egy meghatározott eredményre vonatkozik, és azt mondja meg, *mit* kell elérni, nem azt, *hogyan*;

2. **mérhető** (measurable): van mérőszáma és célértéke, így a teljesülése ellenőrizhető;
3. **hozzárendelhető** (assignable): megmondható, ki felel az eléréséért;
4. **reális** (realistic): a rendelkezésre álló erőforrásokkal elérhető, de nem triviális;
5. **időhöz kötött** (time-related): kijelöli, mikorra kell teljesülnie.

A hierarchia ereje a nyomonkövethetőségben rejlik. Ha egy cél nem szolgál egyetlen szükségletet sem, vagy egy célkitűzés egyetlen célt sem, akkor nem tartozik a rendszerbe.

VárosiBringa: szükséglet, cél, célkitűzés

Szükséglet: Bakonyvár lakói és látogatói a belvárosi, legfeljebb néhány kilométeres utakat autó nélkül, a domborzattól függetlenül, kényelmesen tehessék meg.
Célok és célkitűzések (részlet):

- *C1 – Elérhetőség:* a kerékpár ott legyen, ahol szükség van rá.
  - Csúcsidőben az állomások legalább 90%-án legyen legalább két kölcsönözhető kerékpár.
- *C2 – Hatás:* a rendszer valóban kiváltson autós utakat.
  - A bevezetést követő második évben a belvárosi, 3 km-nél rövidebb autós utak aránya legalább 10%-kal csökkenjen a kiinduló felméréshez képest.
- *C3 – Befogadás:* a rendszer mindenki számára használható legyen.
  - A 65 év felettiek körében a sikeres első kölcsönzések aránya segítség nélkül is érje el a 85%-ot.

5.6.2. A hatékonyság mérőszámai

A célkitűzések közül azokat, amelyek a rendszer egészének sikerét az érdekelt felek szemszögéből mérik, a **hatékonyság mérőszámainak** (measures of effectiveness, MOE) nevezzük. A fenti példában a C2 célkitűzés tipikus MOE. Nem azt mondja meg, milyen gyors a kerékpár vagy hány állomás van, hanem azt, hogy a rendszer eléri-e a célját. A MOE-k a validáció mércéi lesznek a V jobb ágán. A teljesítmény mérőszámaival (MOP) és a műszaki teljesítménymutatókkal (TPM), amelyek a MOE-kból vezethetők le, a 6. fejezet foglalkozik.

5.6.3. Ellentmondó elvárások

A VárosiBringa esetében az önkormányzat sűrű állomáshálózatot szeretne, az üzemeltető alacsony beruházási és üzemeltetési költséget, a gyalogosok szabad járdát, az akadálymentességi érdekképviselet pedig azt, hogy egyetlen dokkoló se szűkítse a járdát egy kerekesszék számára szükséges szélesség alá. Az ellentmondások a természetes velejárói minden többszereplős rendszernek. A rendszermérnök feladata nem az, hogy eltüntesse őket, hanem hogy **láthatóvá és kezelhetővé tegye** őket.

1. **Tedd explicitté** az ellentmondást: melyik elvárás melyikkel ütközik, és miért.
2. **Keresd a mögöttes érdeket.** Az állomássűrűség mögött az elérhetőség igénye áll, amit esetleg kevesebb, de jobb helyen lévő állomás is kielégít.
3. **Fogalmazd meg mérhetően** mindkét oldalt, hogy a változatok összevethetők legyenek.

4. **Rögzítsd, ki dönt.** A döntési jog tisztázása az érdekeltfél-elemzés része, a döntés módszertana a 10. fejezet témája.

A folyamat végén az érdekelt felek jóváhagyják a validált elvárásokat. Ezek alapvonalba kerülnek, és ettől kezdve szabályozott változáskezeléssel módosíthatók (2.5. szakasz). Az érdekelt feleket nemcsak a kezdeti szakaszban, hanem a projekt minden szakaszában érdemes bevonni, mert rendszeres visszajelzésük önkorrekciós visszacsatolásként működik. Az ISO/IEC/IEEE 15288 is előírja, hogy az elemzett követelményeket vissza kell juttatni az érdekelt felekhez, és meg kell erősíttetni velük, hogy igényeiket helyesen rögzítettük (ISO/IEC/IEEE 2023, 6.4.2. szakasz).

Összefoglalás

- Érdekelt fél mindenki, aki hat a rendszerre, vagy akire a rendszer hat. A megrendelő csak egy közülük. A kimaradt érdekelt fél később, rendszerint drágán jelentkezik.
- Az azonosítást három nézőpont segíti: az életciklus-szakaszok, a hagymamodell szerepei és néhány célzott ellenőrző kérdés. A felhasználókat ne kezeljük egyetlen homogén csoportként.
- A hatalom–érdekeltség rács négy negyedet ad (kulcsszereplők, érintettek, keretadók, tömeg). A fontossági modell a hatalom, a legitimitás és a sürgősség alapján hét osztályt különböztet meg. Mindkettő dinamikus, és bevonási tervben ölt testet.
- Az elvárások feltárására többféle módszer szolgál. A kimondott megoldás mögött mindig a valódi igényt keressük.
- A Kano-modell az elvárásokat alapvető, egydimenziós, vonzó, közömbös és fordított kategóriába sorolja. Az elégedettségi együtthatók a tervezési prioritásokat segítik.
- Az elvárásokból szükséglet–cél–célkitűzés hierarchia és a hatékonyság mérőszámai (MOE) lesznek. Ezek a követelmények forrásai és a validáció mércéi.

Önellenőrzés

1. Miért kétirányú az érdekelt fél fogalma? Mondj egy-egy példát a KoboCella esetéből mindkét irányra!
2. A hagymamodell melyik gyűrűjébe sorolnád a ForgácsTech gépkezelőit, a MES-szállítót és a vevő minőségügyi auditorát?
3. Mi a különbség a hatalom–érdekeltség rács és a fontossági modell szemlélete között? Mit lát az egyik, amit a másik nem?
4. Fogalmazd át igénnyé a következő megoldásjavaslatot: „A dokkolókra kell egy érintőképernyő.”
5. Egy jellemzőre 100 válasz érkezett: $V = 20$, $E = 15$, $A = 45$, $Kö = 18$, $F = 2$, $K = 0$. Melyik osztályba tartozik, és mennyi a CS^{+} és a CS^{-}?
6. Miért nem követelmény még egy jól megfogalmazott célkitűzés sem?

Megoldás.

1. A rendszerre hat például az üzemigazgató (finanszíroz, határidőt szab) és a munkavédelmi hatóság (engedélyez). A rendszer hat az operátorokra (megváltozik a munkájuk és a terhelésük) és az üzemi tanácsra (a munkahelyek jövője miatt).
2. A gépkezelők a „mi rendszerünk” gyűrűjébe tartoznak (napi kezelők), a MES-szállító a tágabb környezetbe (fejlesztő vagy beszállító), a vevői auditor a befoglaló rendszerbe vagy a tágabb környezetbe, attól függően, hogy a vevő a kimenet haszonélvezőjeként vagy szabályozóként lép fel.
3. A rács két mennyiségi szempont szerint rendez, és a bevonás intenzitását adja meg. A fontossági modell a követelés jogosságát és sürgősségét is vizsgálja, és megmutatja, hogyan léphet egy érdekelt fél egyik osztályból a másikba, például hogyan válhat egy függő érdekelt meghatározóvá.
4. Például: „A dokkolónál kölcsönző felhasználó, okostelefon nélkül is, a helyszínen tudjon kerékpárt kölcsönözni és segítséget kérni.” Ezt képernyő, kártyaolvasó vagy telefonos segélyvonal is kielégítheti.
5. A leggyakoribb kategória az alapvető ($A = 45$). A nevező $20 + 15 + 45 + 18 = 98$, így

$CS^{+} = 35/98 \approx 0{,}36$ és $CS^{-} = -60/98 \approx -0{,}61$.

6. Mert az érdekelt fél nézőpontjából fogalmazza meg a kívánt eredményt, és még nem a rendszerre vonatkozó, a megoldás szintjén ellenőrizhető állítás. A célkitűzésekből a követelmények meghatározása során vezetjük le a rendszer által teljesítendő, egyértelmű követelményeket.

Projektlépés: érdekelt felek és elvárások

Saját fejlesztési feladatodhoz:

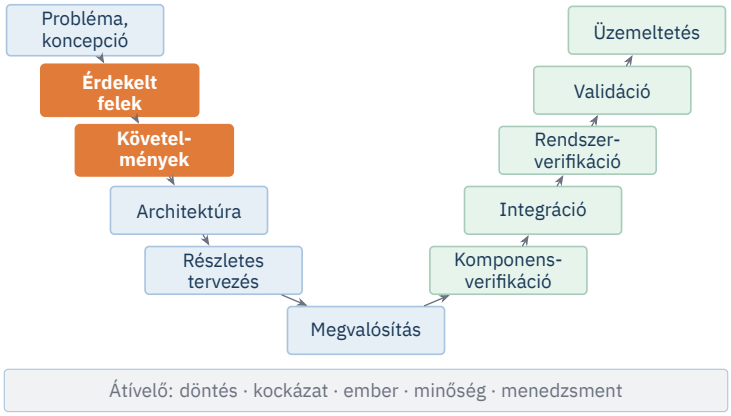
1. azonosíts legalább tíz érdekelt felet az életciklus-táblázat, a hagymamodell és az ellenőrző kérdések segítségével (legalább hármat a rendszeren kívülről);
2. készítsd el a hatalom–érdekeltség rácsot, és jelöld ki a kulcsszereplőket;
3. a két legfontosabb érdekelt féllel készíts rövid interjút, és fogalmazz meg legalább öt elvárást igényként;
4. fogalmazd meg a szükségletet, legalább három célt és célonként legalább egy mérhető célkitűzést; jelöld meg, melyik MOE.

A sablonokat a Gyakorlati útmutató érdekeltfél-eszközlapjai tartalmazzák.

Továbbolvasás

- Wheatcraft, Ryan, és Edner Katz (2025): az igények feltárásának és a szükséglet–cél–célkitűzés rendszer kidolgozásának részletes útmutatója.
- Alexander (2005): a hagymamodell és az érdekeltfél-szerepek részletes leírása.
- Ackermann és Eden (2011): az érdekelt felek stratégiai kezelése elmélet és gyakorlat alapján.
- Mitchell, Agle, és Wood (1997): a fontossági modell eredeti cikke.
- Berger és mtsai. (1993): a Kano-módszer gyakorlati útmutatója.
- NASA (2016), 4.1. szakasz: az érdekelt felek elvárásainak meghatározása a rendszermérnöki folyamatban.

6. Követelmények meghatározása



Tanulási eredmények

A fejezet feldolgozása után képes leszel:

- megkülönböztetni az elvárást, a követelményt, a célt és a korlátot, és a követelményt mértékként és értékként megfogalmazni;
- osztályozni a követelményeket típusuk szerint (funkcionális, teljesítmény-, interfész-, környezeti, emberi tényezőkre és minőségi jellemzőkre vonatkozó);
- egyedi követelményeket és követelményhalmazokat a jó követelmény ismérvei szerint megfogalmazni és ellenőrizni;
- strukturált mondatmintákkal (EARS) egyértelmű követelményeket írni;
- a hatékonyság, a teljesítmény és a műszaki teljesítmény mérőszámait (MOE, MOP, TPM) levezetni és nyomon követni;
- a követelményeket lebontani, allokálni, kétirányúan visszakövetni, és verifikációs módot rendelni hozzájuk;
- a követelményváltozásokat szabályozottan kezelni.

A probléma meghatározása (4. fejezet) és az érdekelt felek elvárásainak feltárása (5. fejezet) után elérkeztünk a V-modell bal ágának talán legtöbbet emlegetett lépéséhez: a követelményekhez. A követelmények közvetítenek a problématér és a megoldástér között. Az érdekelt felek nyelvéből a mérnökök nyelvére fordítják le az elvárásokat, és egyben a V jobb ágának mércéi is: a verifikáció azt vizsgálja, hogy a kész rendszer teljesíti-e őket.

A 2. fejezet költséggörbéi szerint a követelményekben elkövetett hiba a legdrágább, mert a fejlesztés minden későbbi lépése rá épül. Ezért kell az érdekelt feleket rávenni arra, hogy elvárásaikat a tervezés megkezdése előtt alaposan végiggondolják, és ezért kell a követelményeket a fejlesztés korai szakaszában rendszeresen felülvizsgálni. Csak így szűrhetők ki időben a félreértések és az ellentmondások, és csak így becsülhető megbízhatóan a költség és az ütemterv. A fejezet ehhez ad eszközöket.

A fejezet négy kérdésre keres választ:

- Mi a követelmény, és milyen fajtái vannak?

- Mitől jó egy követelmény, és hogyan fogalmazzuk meg?
- Hogyan mérjük a rendszer sikerét és a fejlesztés előrehaladását?
- Hogyan bontjuk le, követjük vissza és kezeljük a követelményeket a fejlesztés során?

6.1. Mi a követelmény?

A követelménymérnökség nemzetközi szabványa szerint a követelmény olyan állítás, amely egy igényt és a hozzá tartozó korlátokat és feltételeket fogalmazza meg vagy fordítja le (ISO/IEC/IEEE 2018). A mennyiségi követelményeket Gilb nyomán érdemes két részre bontani: arra, *amit* mérünk, és arra a szintre, amelyet el kell érni (Gilb 2005). Ebből egyszerű, a gyakorlatban jól használható képlet adódik:

$$\text{követelmény} = \text{mérték} + \text{érték}. \tag{6.1}$$

A **mérték** az, amit mérünk (ciklusidő, rendelkezésre állás, tömeg, válaszidő). Az **érték** az, amit a mértéknek teljesítenie kell (legfeljebb 75 s, legalább 95%). Ehhez gyakran egy **célérték** is társul, amely a kívánatos, de nem kötelező szintet jelöli. A mérték lehet folytonos (idő, tömeg), diszkrét vagy logikai (van-e vészleállító), sőt minőségi is (kezelhetőség egy adott skálán), de mindig mérhetőnek, ellenőrizhetőnek kell lennie. A mérték formája lehet arányos („a selejtarány legalább 50%-kal csökkenjen”), abszolút („legfeljebb 75 s”) vagy valószínűségi („90%-os valószínűséggel legfeljebb 2 perc”).

A követelményeket a megfogalmazásuk is megkülönbözteti más állításoktól. Az ISO/IEC/IEEE 29148 szabvány az angol nyelvű dokumentumokra négy segédigét különít el (ISO/IEC/IEEE 2018):

- a **„shall”** (magyarul: *kell*) kötelező, verifikálandó követelményt jelöl;
- a **„should”** (magyarul: *célszerű*, *kellene*) kívánatos, de nem kötelező célt vagy preferenciát jelöl;
- a **„may”** (magyarul: *lehet*, *megengedett*) javaslatot vagy lehetőséget jelöl;
- a **„will”** (magyarul: jelen vagy jövő idejű kijelentés) tényt, jövőbeli állapotot vagy szándékot jelöl, nem követelményt.

A könyvben ennek megfelelően a követelményeket „…-nak/-nek … kell …” szerkezettel írjuk, a célokat „célérték” vagy „célszerű” kifejezéssel jelöljük, a ténymegállapításokat pedig kijelentő módban.

Elvárás, követelmény, cél, korlát

- **Elvárás:** az érdekelt fél elképzelése arról, mit nyújtson a rendszer; lehet pontatlan és minőségi („ne kelljen hajolgatni”).
- **Követelmény:** a rendszerre vonatkozó egyértelmű, ellenőrizhető állítás („a munkaasztalon végzett minden műveletnek legfeljebb 20°-os törzshajlítással elvégezhetőnek kell lennie”).
- **Cél:** kívánatos, de nem kötelező szint („célérték: 15°”).
- **Korlát:** a megoldást szűkítő, a tervezők által nem módosítható feltétel („a cellának el kell férnie a jelenlegi 6 × 4 méteres területen”).

6.2. A követelmények típusai

A műszaki követelmények sokféle forrásból származnak. Az ISO/IEC/IEEE 29148 szabvány rendszerkövetelmény-specifikációjának vázlata külön pontot szán többek között a funkcionális, a használhatósági, a teljesítmény-, az interfész-, az üzemmódokra és állapotokra vonatkozó, a fizikai, a környezeti, a védelmi, a jogszabályi és az életciklus-támogatási követelményeknek (ISO/IEC/IEEE 2018). Ha valamelyik típus kimarad, a követelményhalmaz nem lesz teljes. A leggyakoribb típusok a következők:

- **Funkcionális követelmények:** mit kell a rendszernek tennie („a cellának minden csavar meghúzási nyomatékát rögzítenie kell”).
- **Teljesítménykövetelmények:** milyen jól kell a funkciót ellátnia („…±2% pontossággal”). A funkcionális követelmény önmagában ritkán elegendő, ezért a funkcionális és a teljesítménykövetelmény általában párban jár.
- **Interfészkövetelmények:** hogyan kell a rendszernek más rendszerekkel kapcsolódnia (a kontextusdiagram interfészlistájából származnak, 3.3. szakasz).
- **Környezeti követelmények:** milyen környezeti feltételek között kell működnie, illetve mit kell elviselnie (hőmérséklet, por, rezgés, elektromágneses zavarok), beleértve a szállítást, a tárolást és a telepítést is.
- **Emberi tényezőkre vonatkozó követelmények:** a kezelhetőség, az ergonómia, a terhelés és a hibatűrés követelményei (11. fejezet).
- **Megbízhatósági, karbantarthatósági, biztonsági követelmények:** a rendszer működésének folytonosságára és veszélytelenségére vonatkoznak (16. és 17. fejezet).
- **Korlátok:** költség, határidő, jogszabályok, szabványok, meglévő rendszerekhez való illeszkedés.

A szoftverintenzív rendszereknél jól használható keret az ISO/IEC 25010:2023 termékminőségi modellje. Kilenc minőségi jellemzőt határoz meg, ezek mindegyike egy-egy követelménycsoport forrása lehet (6.1. táblázat) (ISO/IEC 2023). A 2023-as kiadás a korábbiakhoz képest a biztonságot (safety) önálló jellemzőként vette fel, a használhatóságot interakciós képességgé, a hordozhatóságot rugalmassággá bővítette.

6.1. táblázat. Az ISO/IEC 25010:2023 kilenc termékminőségi jellemzője a VárosiBringa példáin.

Minőségi jellemző	Kérdés	Példa a VárosiBringa alkalmazására
funkcionális megfelelőség	Megvannak-e a szükséges funkciók, és helyesen működnek-e?	a díj számítása minden díjtételre helyes
teljesítményhatékonyság	Elég gyors-e, elég-e a kapacitása?	a kölcsönzés jóváhagyása 2 s-on belül
kompatibilitás	Együttműködik-e más rendszerekkel?	a buszbérlet-adatbázissal való adatcsere
interakciós képesség	Képesek-e a felhasználók hatékonyan és elégedetten használni?	idősebb felhasználók is segítség nélkül kölcsönöznek
megbízhatóság	Működik-e, amikor kell?	havi rendelkezésre állás legalább 99,5%
védelem (security)	Védi-e az adatokat és a hozzáférést?	a fizetési adatokat csak a fizetési szolgáltató tárolja

Minőségi jellemző	Kérdés	Példa a VárosiBringa alkalmazására
karbantarthatóság	Könnyen módosítható, javítható-e?	új díjszabás bevezetése kódmódosítás nélkül
rugalmasság	Alkalmazkodik-e új környezethez, igényekhez?	újabb városba telepíthető
biztonság (safety)	Elkerüli-e a veszélyes helyzeteket?	menet közben nem jelenít meg figyelmet elvonó értesítést

6.3. A jó követelmény

A követelmény minőségéről két szinten beszélünk: az egyes követelmények és a követelményhalmaz szintjén. Az ISO/IEC/IEEE 29148:2018 és az INCOSE követelményírási útmutatója szerint az **egyedi követelmény** akkor jól formált, ha (ISO/IEC/IEEE 2018; INCOSE 2023a):

- **szükséges:** kihagyása hiányt okozna, amelyet más követelmény nem pótol;
- **megfelelő szintű:** részletessége illeszkedik ahhoz a szinthez (rendszer, alrendszer, komponens), amelyre vonatkozik;
- **egyértelmű:** csak egyféleképpen értelmezhető;
- **teljes:** nem szorul kiegészítésre ahhoz, hogy érthető legyen;
- **egyedi (szinguláris):** egyetlen képességet, jellemzőt, korlátot vagy minőségi tényezőt fogalmaz meg;
- **megvalósítható:** a korlátok (költség, idő, technológia, fizika) között elérhető;
- **verifikálható:** objektív módon igazolható, hogy a rendszer teljesíti;
- **helyes:** pontosan tükrözi a forrásául szolgáló igényt vagy felsőbb szintű követelményt;
- **szabályos:** megfelel a projektben elfogadott mintának és stílusnak.

A **követelményhalmaz** egészének teljesnek, következetesnek (ellentmondásmentesnek), megvalósíthatónak, érthetőnek és validálhatónak kell lennie (ISO/IEC/IEEE 2018). Az INCOSE útmutatója ehhez hozzáteszi, hogy a halmaz ne legyen redundáns: minden követelmény egyszer és csak egyszer szerepeljen, és a fogalmakat, mértékegységeket egységesen használjuk (INCOSE 2023a).

6.3.1. Írási szabályok a gyakorlatban

Az INCOSE követelményírási útmutatója a jó tulajdonságokat konkrét, ellenőrizhető írási szabályokra bontja (INCOSE 2023a; NASA 2016, C függelék). A legfontosabbak, a magyar nyelvhez igazítva:

- **Cselekvő szerkezet:** „A [termék]nak … kell …”, és nem szenvedő szerkezet.
- **Egy követelmény, egy állítás:** egy alany és egy állítmány; egy mondatban egyetlen „kell”.
- **Tűrésekkel:** a mennyiségi értékekhez tartozzon tűrés vagy korlát (legalább, legfeljebb, ± érték).
- **Megvalósításfüggetlen:** mit kell elérni, és nem hogyan. Ha a követelmény megoldást ír elő, kérdezzük meg, miért kell.

- **Megfelelő alany:** a követelmény alanya az a rendszer vagy elem legyen, amelyre vonatkozik. Az „Az operátornak … kell” típusú mondatok szinte mindig működési állítások, amelyek a ConOps-ba tartoznak.
- **Pozitív megfogalmazás:** a „nem szabad” helyett lehetőleg azt írjuk le, aminek teljesülnie kell.
- **Indoklással:** minden követelményhez tartozzon érthető indoklás a feltételezésekkel együtt.
- **TBD helyett TBR:** a „később meghatározandó” (TBD) értékek helyett inkább egy legjobb becslést adjunk meg „pontosítandó” (TBR) jelöléssel. Rögzítsük, ki és mikorra pontosítja.
- **Kétértelmű kifejezések nélkül:** kerülendő a „szükség szerint”, a „stb.”, az „és/vagy” és a „többek között”.

A magyar nyelvű követelményekben jellemzően a következő kifejezések okoznak kétértelműséget: *megfelelő, gyors, rugalmas, felhasználóbarát, könnyen, lehetőleg, optimális, maximalizál, minimalizál, kellően, általában, szükség szerint, stb., és/vagy*. A Gyakorlati útmutató követelménysablonja ezeket automatikusan jelzi.

6.2. táblázat. Rossz és jobb követelmény-megfogalmazások.

Rossz megfogalmazás	Mi a gond?	Jobb megfogalmazás
A cellának gyorsnak kell lennie.	nincs mérték és érték; nem verifikálható	A cellának egy motort legfeljebb 75 s alatt kell összeszerelnie.
A rendszer legyen felhasználóbarát.	kétértelmű; nem verifikálható	Az első alkalommal kölcsönzők legalább 90%-ának segítség nélkül, 3 percen belül kell tudnia kölcsönözni.
A cellának robottal kell felhelyeznie a fedelet és meghúznia a csavarokat.	megoldást ír elő; két követelmény egyben	A cellának a nyolc csavart 4,0 ± 0,2 Nm nyomatékkal kell meghúznia. (A fedél felhelyezése külön követelmény, a megoldás nyitva marad.)
Az operátornak ellenőriznie kell a kész motort.	működési leírás, nem a rendszerre vonatkozik	A cellának minden kész motor nyomatékadatait a motor azonosítójához rendelve kell rögzítenie.
Az alkalmazásnak stb. adatokat kell megjelenítenie.	hiányos, „stb.”	Az alkalmazásnak a kiválasztott kerékpár töltöttségét és a becsült díjat a kölcsönzés jóváhagyása előtt meg kell jelenítenie.
A dokkoló ne legyen hangos.	negatív, nincs mérték	A dokkoló működés közbeni hangnyomásszintje 1 m-en nem haladhatja meg a 60 dB(A)-t.

6.3.2. Strukturált mondatminták: EARS

A természetes nyelvű követelmények egyik legnagyobb gyengesége a kétértelműség. Mavin és szerzőtársai a Rolls-Royce-nál, repülőgép-hajtómű irányítórendszerének légialkalmassági előírásait elemezve figyelték meg, hogy a jól megírt követelmények néhány

visszatérő mintát követnek. Ezekből alakították ki az EARS (Easy Approach to Requirements Syntax) mondatmintákat (Mavin és mtsai. 2009). A minták néhány kulcsszóval kötik meg a mondat szerkezetét, de természetes nyelvűek maradnak. A 6.3. táblázat a minták magyar változatát mutatja.

6.3. táblázat. Az EARS-mondatminták magyar változata. A minták kombinálhatók is (például állapot és esemény együtt).

Minta	Szerkezet	Példa
állandó (mindenkor érvényes)	A ‹rendszer›nek ‹válasz› kell.	A cellának minden csavar meghúzási nyomatékát rögzítenie kell.
eseményvezérelt	Amikor ‹kiváltó esemény›, a ‹rendszer›nek ‹válasz› kell.	Amikor a felhasználó jóváhagyja a kölcsönzést, a dokkolónak 2 s-on belül ki kell oldania a zárat.
állapotvezérelt	Amíg ‹állapot›, a ‹rendszer›nek ‹válasz› kell.	Amíg személy tartózkodik a védőzónában, a kobot sebességének legfeljebb 250 mm/s-nak kell lennie.
nem kívánt viselkedés	Ha ‹nem kívánt feltétel›, akkor a ‹rendszer›nek ‹válasz› kell.	Ha egy csavar meghúzási nyomatéka a tűrésen kívül esik, akkor a cellának a motort hibásként kell megjelölnie.
opcionális jellemző	Ahol ‹jellemző megléte›, a ‹rendszer›nek ‹válasz› kell.	Ahol az állomás napelemes töltéssel működik, a dokkolónak a töltöttséget a háttérrendszernek óránként jelentenie kell.

6.3.3. Követelményattribútumok

A követelmény nem csak egy mondat. A nyilvántartásban minden követelményhez attribútumok tartoznak, amelyek kezelhetővé és visszakövethetővé teszik. Az INCOSE útmutatója az indoklást a legfontosabb attribútumnak tartja: aki később módosítani akarja a követelményt, ebből tudja meg, miért született (INCOSE 2023a). Érdemes külön megjelölni a **meghatározó követelményeket** (key driving requirements, KDR) is. Ezek teljesítése a prioritásuktól függetlenül erősen befolyásolja a tervet, a költséget vagy az ütemtervet. Rokon fogalom a **kulcsfontosságú teljesítményparaméter** (key performance parameter, KPP): olyan jellemző, amelynek küszöbértékét ha a rendszer nem éri el, a koncepciót vagy akár a projektet is újra kell értékelni (Roedler és Jones 2005). A leggyakoribb attribútumok:

- **azonosító** (egyedi és változatlan, például R-05);
- **szöveg** és **indoklás** (miért kell, milyen feltételezésekre épül);
- **forrás** (melyik elvárásból, szabványból vagy felsőbb szintű követelményből származik);
- **típus** és **prioritás** (például kötelező, fontos, kívánatos);
- **verifikációs mód** (vizsgálat, elemzés, bemutatás, teszt) és **verifikációs szint**;
- **felelős** (aki a követelményért felel);
- **állapot** (javasolt, jóváhagyott, alapvonalban, verifikált, törölt) és **KDR-jelölés**.

KoboCella: egy követelmény teljes leírása

Attribútum	Érték
Azonosító	R-01
Szöveg	A cellának egy motort legfeljebb 75 s alatt kell összeszerelnie.
Indoklás	Két műszakos munkarendben (műszakonként 7,5 óra nettó munkaidő, évi 230 munkanap) évi 3450 óra áll rendelkezésre, így az évi 140 ezer motorhoz a vevői ütemidő 3450 × 3600 / 140 000 ≈ 88,7 s. A tervezett 85%-os berendezéshatékonyság mellett a ciklusidő legfeljebb 88,7 × 0,85 ≈ 75 s lehet.
Forrás	EF-01 (üzemigazgató): évi 140 ezer motor; MOE-1
Típus, prioritás	teljesítmény; kötelező; KDR
Verifikációs mód és szint	teszt (próbaüzem 3 × 8 órán át), rendszerszint; előtte elemzés (idődiagram, szimuláció)
Felelős	rendszermérnök; a robotintegrátor a robotalrendszerért
Állapot	alapvonalban (SRR után)

6.4. A siker mérése: MOE, MOP, TPM

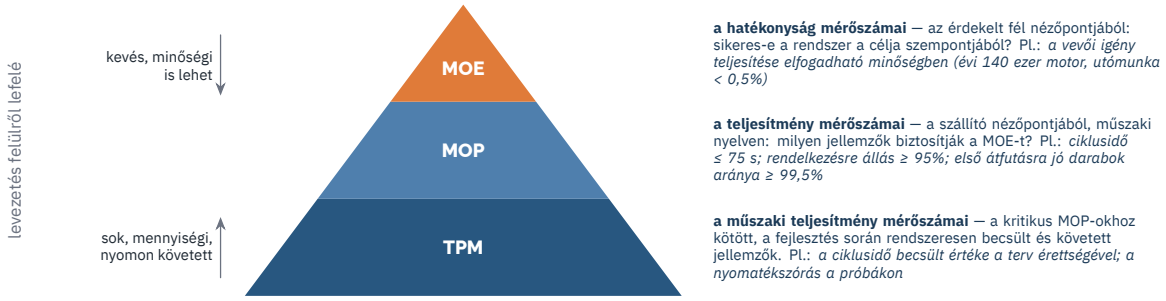
A követelmények egy része a rendszer sikerét méri. Ezeket három szinten szokás megkülönböztetni (6.1. ábra). Kossiakoff és szerzőtársai meghatározásai szerint (Kossiakoff és mtsai. 2011, 6. fejezet):

- **A hatékonyság mérőszáma** (measure of effectiveness, MOE) a rendszer egészének teljesítményét jellemző minőségi vagy mennyiségi mérőszáma, amely azt mutatja, milyen mértékben éri el a rendszer a céljait meghatározott feltételek mellett. A MOE mindig a rendszer egészére vonatkozik.
- **A teljesítmény mérőszáma** (measure of performance, MOP) a rendszer egy jellemzőjének vagy egy alrendszer teljesítményének mennyiségi mérőszáma. Jellemzően a rendszer egésze alatti szinten méri a fizikai teljesítményt.

Az INCOSE műszaki mérési útmutatója a két fogalmat a nézőponttal is megkülönbözteti (Roedler és Jones 2005). A MOE-k a megrendelő és a felhasználó nézőpontjából fogalmazódnak meg: azt mérik, sikeres-e a rendszer a tervezett működési környezetben. A MOP-ok a MOE-kből származnak, de műszakibb nyelven, a szállító nézőpontjából írják le a rendszer fizikai vagy funkcionális jellemzőit. Egy MOE teljesítéséhez jellemzően több, mennyiségi és mérhető MOP kell, még akkor is, ha maga a MOE minőségi.

- **A műszaki teljesítmény mérőszámai** (technical performance measures, TPM) a MOP-okhoz kapcsolódó vagy azokból levezetett fizikai vagy funkcionális jellemzők, amelyeket a siker szempontjából kritikusnak tekintünk. A megvalósítás során rendszeresen összevetjük az aktuálisan elért értéküket vagy legjobb becslésüket azzal, amit az adott időpontra terveztünk. Így igazolják az előrehaladást, és korán jelzik

azokat a hiányosságokat, amelyek egy kritikus követelmény teljesülését veszélyeztetik, vagy költség- és ütemezési kockázatot jelentenek (Roedler és Jones 2005; NASA 2016, 4.2. szakasz).



6.1. ábra. A MOE, a MOP és a TPM viszonya a KoboCella példáján. Felülről lefelé a mérőszámok száma nő, és egyre műszakibbá válnak.

Kossiakoff és szerzőtársai szerint egy MOE vagy MOP három részből áll: a mérőszámból, a mértékegységből és a feltételekből vagy környezetből, amelyben a mérőszám érvényes (Kossiakoff és mtsai. 2011, 6. fejezet). A „ciklusidő” önmagában nem teljes mérőszám. Teljes alakja: „ciklusidő (másodperc), a referencia-motortípusra, betanult operátorral, folyamatos üzemben”.

6.4.1. A TPM-ek nyomon követése

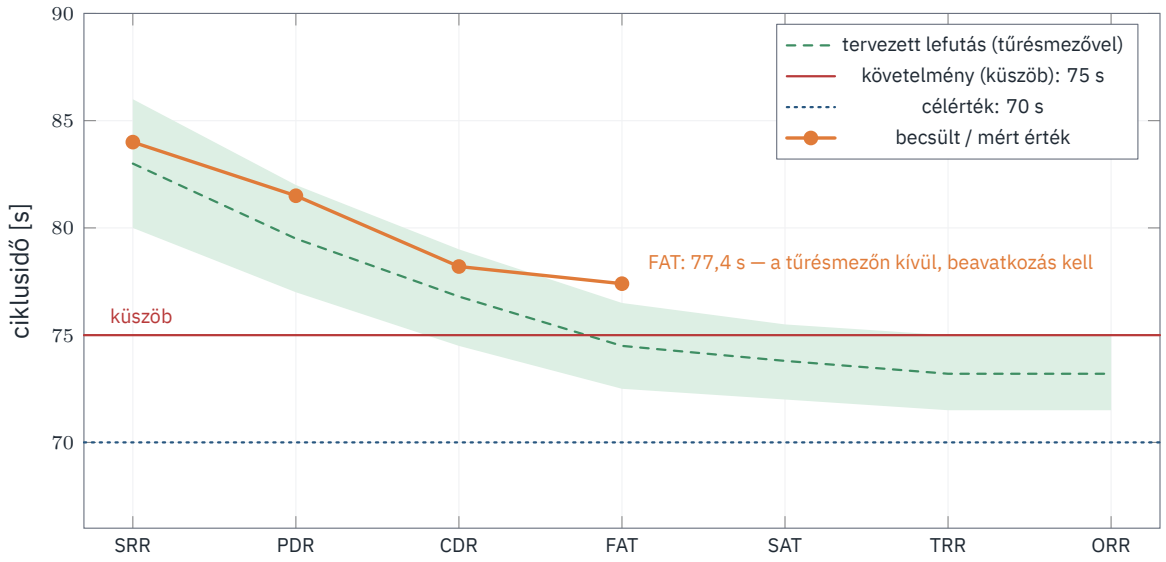
A TPM-eket a fejlesztés során rendszeresen, például minden felülvizsgálaton becsüljük vagy mérjük, és egy tervezett lefutással és tűrésmezővel vetjük össze (6.2. ábra). A tervezett lefutás azt mutatja, hogyan várjuk a jellemző alakulását a terv érlelődésével. A korai becslések általában kedvezőtlenebbek, és az optimalizálással javulnak. Ha a becsült érték kilép a tűrésmezőből, beavatkozásra van szükség, még akkor is, ha a végső követelmény teljesítése elvileg még lehetséges.

> **KoboCella: mit jelzett a TPM?**
>
> A gyári átvételi teszten a cella ciklusideje 77,4 s lett. Ez még csak 2,4 s-mal haladta meg a követelményt, a tervezett tűrésmezőt (legfeljebb 76,5 s) azonban már átlépte. A csapat nem várt a helyszíni telepítésig. Az idődiagram elemzése megmutatta, hogy a kobot a csavarozás után feleslegesen visszatér a nyugalmi helyzetébe, mielőtt a kész motort kivenné. A pálya átírása 2,8 s-ot hozott, a helyszíni átvételen a ciklusidő 74,1 s lett. Ha a csapat csak a végső követelményt figyeli („a 75 s majd a helyszínen is teljesül”), a hiba az üzembe helyezéskor derült volna ki, drágábban és határidő-csúszással.

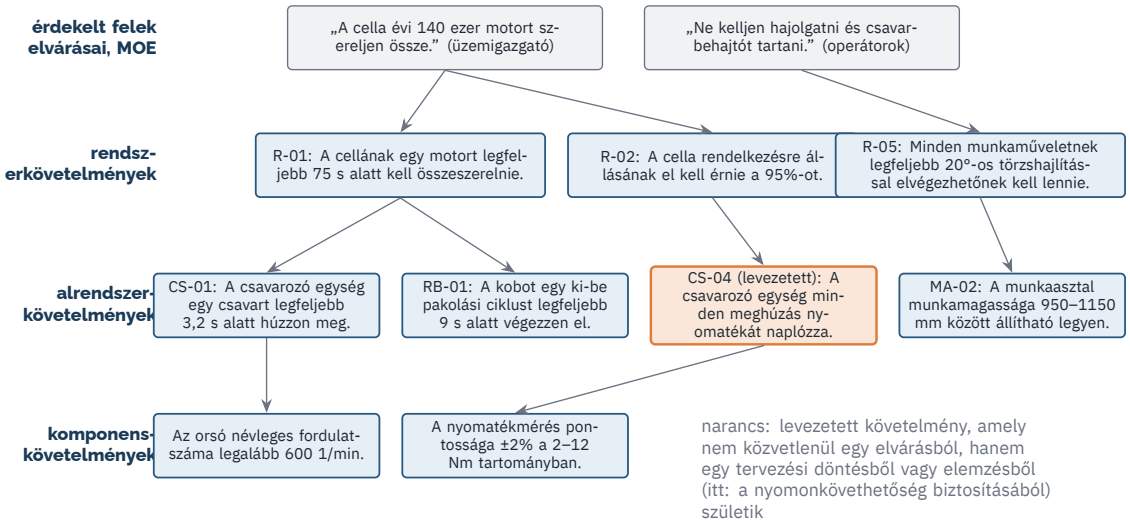
6.5. Lebontás, allokáció, nyomonkövethetőség

A rendszerszintű követelményeket le kell bontani az alrendszerek és a komponensek szintjére (6.3. ábra). Az eredmény egy követelményhierarchia: a felső szinten azt írjuk le, *mit* akarunk elérni (a hogyant nem specifikáljuk), az alsóbb szinteken pedig a választott architektúra minden elemére, hardverre és szoftverre is lebontjuk a követelményeket.

A lebontás két fontos fogalma:



6.2. ábra. A KoboCella ciklusidejének TPM-követése. A gyári átvételi teszten (FAT) mért érték kilépett a tervezett tűrésmezőből; a csapat beavatkozott, mielőtt a helyszíni átvételre (SAT) került volna sor.



6.3. ábra. A KoboCella követelményhierarchiájának részlete. A követelmények az érdekelt felek elvárásaiból a rendszer-, alrendszer- és komponensszintre bomlanak le.

- **Allokáció:** egy rendszerszintű követelmény teljesítésének szétosztása az alrendszerek között. A teljesítményjellemzőknél ez gyakran **költségvetés** formáját ölti: a teljes ciklusidő, tömeg, energiafogyasztás vagy hibavalószínűség az alrendszerek részesedéseinek összege. A költségvetésben mindig **tartalékot** (margint) kell hagyni a bizonytalanságokra.
- **Levezetett követelmény:** olyan követelmény, amely nem közvetlenül egy érdekelt fél elvárásából, hanem egy tervezési döntésből vagy elemzésből születik. Ide tartoznak a választott architektúra és a tervezési döntések következményei, valamint a szervezetben jó gyakorlatként elfogadott belső előírások. A levezetett követelménynek is van forrása: az a döntés vagy elemzés, amelyből született, és ezt a nyilvántartásban rögzíteni kell.

6.5. táblázat. A KoboCella ciklusidő-költségvetése a koncepció szakaszában. A szemrevételezés és a következő darab előkészítése a csavarozással párhuzamosan zajlik, ezért nem szerepel a soros összegben.

Ciklusidő-összetevő	Végrehajtó	Allokált idő [s]
motorház és fedél behelyezése a befogóba	operátor	18,0
a fedél pozicionálása és rögzítése	kobot	9,0
8 csavar meghúzása (8 × 3,2 s)	csavarozó egység	25,6
a kész motor kivétele a kiadóba	kobot	6,0
soros összeg		**58,6**
követelmény (R-01)		**75,0**
tartalék		**16,4 (≈ 22%)**

A 6.5. táblázat koncepciószintű költségvetésében 22% a tartalék. Ez a koncepció szakaszában indokolt, mert az idők még becslések. A FAT-on mért 77,4 s azt mutatta, hogy a tartalékot a valóságban a nem tervezett mozgások és várakozások felélték. Ezért kell a költségvetést a TPM-ekkel együtt követni.

6.5.1. Kétirányú nyomonkövethetőség

A követelménykezelés lényege a változások kezelése és a **kétirányú nyomonkövethetőség** fenntartása az érdekelt felek igényei, a rendszer-, alrendszer- és komponenskövetelmények, a tervdokumentumok és a verifikáció között (ISO/IEC/IEEE 2018; ISO/IEC/IEEE 2023, 6.4.3. szakasz; NASA 2016, 6.2. szakasz). A kétirányúság azt jelenti, hogy mindkét irányban követhető a lánc (6.4. ábra):

- **előre (lefelé):** minden elvárásnak és felsőbb szintű követelménynek van-e alacsonyabb szintű követelménye, tervezési eleme és verifikációja? Ez a **lefedettség** kérdése.
- **visszafelé (felfelé):** minden követelménynek és tervelemnek van-e forrása? Ha egy követelménynek nincs szülője, és nem is dokumentált levezetett követelmény, akkor vagy a visszakövetés hibás, vagy a követelmény felesleges „aranyozás” (gold plating), amelyet törölni kell.

A nyomonkövethetőséget a gyakorlatban **követelmény-nyomonkövetési mátrixban** (requirements traceability matrix, RTM) vagy követelménykezelő eszközben rögzítjük. A mátrixból automatikusan kiszűrhetők az árva (forrás nélküli) követelmények, a lebontatlan felsőbb szintű követelmények és a verifikáció nélküli követelmények. A Gyakorlati útmutató nyomonkövetési sablonja és notebookja ezeket az ellenőrzéseket végzi el.



6.4. ábra. A kétirányú nyomonkövethetőség lánca a KoboCella egy ergonómiai elvárásán. Előre a lefedettséget, visszafelé a forrást ellenőrizzük.

6.5.2. A verifikáció módjának hozzárendelése

Már a követelmény megfogalmazásakor el kell döntenünk, hogyan fogjuk igazolni a teljesülését. Ha erre nem tudunk válaszolni, a követelmény nem verifikálható, és újra kell fogalmazni. A szabványok és a szakirodalom négy alapvető verifikációs módot különböztetnek meg (ISO/IEC/IEEE 2018; INCOSE 2023b; NASA 2016, 5.3. szakasz):

- **vizsgálat (inspection):** szemrevételezés vagy dokumentumellenőrzés (van-e vészleállító, megvannak-e a feliratok);
- **elemzés (analysis):** számítás, modell vagy szimuláció (megbízhatósági elemzés, ciklusidő-szimuláció);
- **bemutatás (demonstration):** a funkció működésének megfigyelése mérés nélkül vagy kevés méréssel (a kezelő végre tudja-e hajtani a típusváltást);
- **teszt (test):** ellenőrzött körülmények közötti mérés (ciklusidő-mérés próbaüzemben, nyomatékmérés).

A verifikációs módokkal és a követelmény-verifikációs mátrixszal a 23. fejezet foglalkozik részletesen.

6.6. A követelmények változásának kezelése

A követelmények változnak. Új érdekelt felek jelentkeznek, a piac megváltozik, a tervezés során kiderül, hogy egy követelmény nem teljesíthető, vagy egy másik fontosabb. A cél nem a változás megakadályozása, hanem a **szabályozott** változáskezelés. Minden változtatási kérelmet a jóváhagyás előtt értékelni kell: hogyan hat a műszaki megoldásra, a költségekre és az ütemtervre (ISO 2017). A követelményeknél ez konkrétan azt jelenti, hogy meg kell vizsgálni a hatást az architektúrára, a tervre, az interfészekre, a ConOps-ra, valamint a felsőbb és alsóbb szintű követelményekre.

A rendszerkövetelmény-felülvizsgálat (SRR) után a követelmények formális konfigurációkezelés alá kerülnek. Ettől kezdve minden változtatást a konfigurációkezelési bizottság (CCB) vagy egy hasonló testület hagy jóvá. A 2.1. szakasz költséggörbéje szerint minél később változik egy követelmény, annál nagyobb költség- és ütemezési többletet okoz, ezért a késői változtatásokat különösen gondosan kell értékelni. A hatáselemzést megkönnyíti, ha naprakész a teljesítménytartalékok listája, tudjuk, mely szakértőket kell bevonni, és vezetjük a kockázatnyilvántartást. A konfigurációkezeléssel a 26. fejezet foglalkozik.

A követelmények burjánzása

A követelmények burjánzása (requirements creep) a követelmények fokozatos, ellenőrizetlen bővülése: „ha már úgyis…, akkor legyen benne az is”. Egyenként minden bővítés kicsinek tűnik, együtt azonban felborítják a költségvetést és az ütemtervet. Ellenszere a jóváhagyott alapvonal, a változtatási kérelmek hatáselemzése és az, hogy minden új követelményhez forrást és indoklást kérünk. Ami nem vezethető vissza egy érdekelt fél elvárására, az aranyozás.

Automatizált követelményellenőrzés

A követelményírási szabályok jelentős része formalizálható: a kétértelmű kifejezések, a több „kell” egy mondatban, a szenvedő szerkezet, a hiányzó mértékegység vagy tűrés, a TBD jelölés szabályokkal felismerhető. Az INCOSE útmutatójának szabályait több kutatás és eszköz is automatizált ellenőrzéssé alakította, és a nagy nyelvi modellek újabb lehetőséget adnak a megfogalmazási hibák felismerésére és javítási javaslatok megfogalmazására. Az automatikus ellenőrzés azonban csak a *megfogalmazás* minőségét vizsgálja. Azt, hogy egy követelmény *helyes*-e, vagyis a valódi igényt fejezi-e ki, csak az érdekelt felekkel közösen lehet eldönteni. A Gyakorlati útmutató követelményellenőrző notebookja szabályalapú ellenőrzést mutat be.

Összefoglalás

- A követelmény egy igényt és a hozzá tartozó feltételeket megfogalmazó, ellenőrizhető állítás: mérték és érték, gyakran célértékkel. A követelményt („kell”) meg kell különböztetni a céltól („célszerű”), a ténymegállapítástól és a korláttól.
- A követelmények típusai: funkcionális, teljesítmény-, interfész-, környezeti, emberi tényezőkre vonatkozó, megbízhatósági, biztonsági követelmények és korlátok. A szoftverintenzív rendszereknél az ISO/IEC 25010 kilenc minőségi jellemzője jó ellenőrzőlista.
- Az egyedi követelmény legyen szükséges, megfelelő szintű, egyértelmű, teljes, egyedi, megvalósítható, verifikálható, helyes és szabályos. A halmaz legyen teljes, következetes, megvalósítható, érthető és validálható.
- Az EARS-minták (állandó, eseményvezérelt, állapotvezérelt, nem kívánt viselkedés, opcionális jellemző) egyértelműbbé teszik a természetes nyelvű követelményeket. Minden követelményhez attribútumok tartoznak: azonosító, indoklás, forrás, prioritás, verifikációs mód, felelős, állapot.
- A MOE az érdekelt fél nézőpontjából méri a sikert, a MOP a szállító nézőpontjából a teljesítményt, a TPM a kritikus jellemzőket követi a fejlesztés során. A TPM tűrésmezőből való kilépése korai figyelmeztetés.
- A követelményeket lebontjuk, allokáljuk (tartalékkal), kétirányúan visszakövetjük, és verifikációs módot rendelünk hozzájuk. A forrás nélküli követelmény aranyozás.
- Az SRR után a követelmények változását konfigurációkezelés szabályozza. A késői változások különösen drágák.

Önellenőrzés

1. Bontsd mértékre és értékre: „A VárosiBringa alkalmazásának a kölcsönzést 2 másodpercen belül jóvá kell hagynia az esetek 99%-ában.” Milyen formájú a mérték?
2. Javítsd ki a következő követelményeket: (a) „A dokkolónak megfelelő méretűnek kell lennie.” (b) „Az alkalmazás jelenítse meg a térképet és a díjakat, és tegye lehetővé a fizetést.” (c) „A karbantartónak hetente ellenőriznie kell a kerékpárok fékjeit.”
3. Fogalmazd meg EARS-mintával: a kobotnak meg kell állnia, ha az operátor kezét érzékeli a csavarozási zónában.
4. Mi a különbség a MOE és a MOP között? Adj egy-egy példát a ForgácsTech esetéből!
5. Egy cella tervezett ciklusidő-költségvetése 61 s, a követelmény 70 s. Mekkora a tartalék másodpercben és százalékban? Mit tennél, ha a CDR-en a becslés 68 s lenne?

6. Mit tesz a rendszermérnök egy olyan követelménnyel, amelynek nincs szülője a nyomonkövetési mátrixban?
7. Miért drágább egy követelményváltozás a részletes terv után, mint a koncepció szakaszában?

Megoldás.

1. Mérték: a kölcsönzés jóváhagyásának ideje; érték: legfeljebb 2 s az esetek legalább 99%-ában. A mérték valószínűségi formájú.
2. (a) Például: „A dokkoló járdára merőleges kiterjedése legfeljebb 0,8 m lehet.” (b) Három külön követelmény: a térkép megjelenítése, a díjak megjelenítése, a fizetés lehetősége, mindegyik saját teljesítményjellemzővel. (c) Ez működési leírás. A rendszerre vonatkozó változat például: „A kerékpár fékrendszerének heti ellenőrzése legfeljebb 5 percet igényelhet egy kerékpárnál, szerszám nélkül.” A karbantartási rend a ConOps vagy a karbantartási terv része.
3. „Ha a csavarozási zónában emberi kéz jelenlétét érzékeli, akkor a kobotnak legfeljebb 100 ms-on belül meg kell állnia.” (nem kívánt viselkedés minta; a reakcióidő értékét a kockázatértékelés határozza meg)
4. A MOE az érdekelt fél szempontjából méri a sikert, a rendszer egészére vonatkozik (például: „a rendelések legalább 95%-a határidőre teljesül”). A MOP műszaki jellemző, amely a MOE eléréséhez járul hozzá (például: „a gépállapot-adatok legfeljebb 1 perc késéssel jelennek meg”, „a beállítási idő átlaga legfeljebb 35 perc”).
5. 9 s, ami a követelmény ≈ 12,9%-a. Ha a CDR-en a becslés 68 s, a tartalék 2 s-ra (≈ 3%) csökkent. Ez korai figyelmeztetés: meg kell vizsgálni, mely összetevők nőttek, és optimalizálni kell, illetve a TPM-et sűrűbben kell követni.
6. Megvizsgálja, hogy elfogadható saját levezetésű követelmény-e (egy tervezési döntés vagy jó gyakorlat következménye). Ha igen, ezt dokumentálja és jóváhagyatja. Ha nem, vagy a visszakövetés hibás és pótolni kell, vagy a követelmény aranyozás, és törölni kell.
7. Mert a részletes terv után egy követelményváltozás már elkészült terveket, beszerzett alkatrészeket, megírt szoftvert és interfészeket érint, és ezeket újra kell tervezni, verifikálni és dokumentálni. Az életciklus-költségek elkötelezettsége (2. fejezet) ekkor már magas.

Projektlépés: követelményspecifikáció

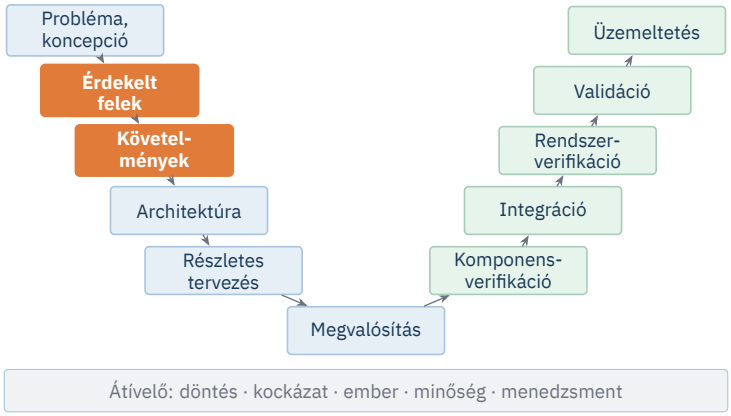
Saját fejlesztési feladatodhoz:

1. vezess le az érdekelt felek elvárásaiból és a ConOps forgatókönyveiből legalább 15 rendszerkövetelményt, köztük legalább hármat a funkcionális és a teljesítménykövetelményeken kívüli típusokból;
2. minden követelményhez add meg az attribútumokat (azonosító, indoklás, forrás, prioritás, verifikációs mód);
3. legalább öt követelményt fogalmazz meg EARS-mintával;
4. határozz meg két MOE-t, mindegyikhez legalább két MOP-ot, és válassz ki egy TPM-et, amelyet a projekt során követnél;
5. bonts le egy rendszerkövetelményt alrendszer-szintre, és készíts hozzá költségvetést tartalékkal;
6. futtasd le a követelményellenőrzést a sablonnal vagy a notebookkal, és javítsd a jelzett hibákat.

Továbbolvasás

- Roedler és Jones (2005): a MOE, a MOP, a TPM és a KPP fogalma és használata.
- ISO/IEC/IEEE (2018): a követelménymérnökség szabványa, a jó követelmény ismérvei.
- INCOSE (2023a): az INCOSE követelményírási útmutatója szabályokkal és példákkal.
- Mavin és mtsai. (2009): az EARS-mondatminták eredeti cikke.
- Kossiakoff és mtsai. (2011), 6. fejezet: a MOE és a MOP fogalma és szerkezete.
- NASA (2016), 4.2. és 6.2. szakasz, C függelék: a műszaki követelmények meghatározása, a követelménykezelés és a követelményírás ellenőrzőlistája.

7. A vevő hangja a tervezésben: QFD és a minőségház



Tanulási eredmények

A fejezet feldolgozása után képes leszel:

- elmagyarázni, mire való a minőségfunkció-telepítés (QFD), és hol a helye a V-modellben;
- felépíteni a minőségházat: vevői elvárások és fontosságuk, műszaki jellemzők, kapcsolati mátrix, tető, versenytárs-értékelés, műszaki fontosság és célértékek;
- kiszámítani a műszaki jellemzők abszolút és relatív fontosságát, és értelmezni az eredményt;
- a tetőmátrix alapján felismerni a műszaki jellemzők közötti ellentmondásokat, és kompromisszumot vagy innovatív megoldást keresni;
- a Kano-modell eredményeit beépíteni a minőségházba;
- leírni a négyfázisú QFD-t, és megnevezni a módszer korlátait és újabb, hálózatalapú kiterjesztéseit.

Az érdekelt felek elvárásai (5. fejezet) a vevő nyelvén szólnak: „mindig legyen kerékpár a közelben”, „ne kelljen hajolgatni”. A követelmények (6. fejezet) a mérnök nyelvén: állomássűrűség, ciklusidő, munkamagasság. A kettő közötti fordítás nem magától értetődő. Melyik műszaki jellemző hat leginkább a vevői elégedettségre? Melyikre érdemes a legtöbb fejlesztési erőforrást fordítani? Mekkora legyen a célérték, hogy versenyképesek legyünk? Hol ütköznek egymással a műszaki jellemzők?

A minőségfunkció-telepítés (quality function deployment, QFD) erre a fordításra ad rendszerezett eljárást. Egyetlen táblázatban kapcsolja össze a piac elvárásait, a kielégítésüket befolyásoló műszaki jellemzőket és a versenytársak termékeinek színvonalát. Segít az új termék műszaki paramétereinek meghatározásában, és dokumentációs eszköz is: utólag is áttekinthető marad belőle, hogyan és miért születtek a tervezési döntések.

A fejezet három kérdésre épül:

- Hogyan épül fel és hogyan készül a minőségház?
- Hogyan értelmezzük az eredményét, és hogyan jutunk el belőle a célértékekig?
- Mik a módszer korlátai, és hogyan bővíthető?

7.1. A QFD eredete és célja

Akao Jodzsi, a módszer egyik megalkotója szerint a QFD olyan módszer, amely a vevő elégedettségét célzó tervezési minőséget fejleszti ki, és a vevő igényeit tervezési célokká és a gyártás során alkalmazandó fő minőségbiztosítási pontokká fordítja le (Akao 1990). A módszer az 1970-es évek elején Japánban született. Hauser és Clausing nagy hatású cikke szerint a minőségház első alkalmazása 1972-re, a Mitsubishi kobei hajógyárára tehető, a Toyota és beszállítói pedig sokféle irányban továbbfejlesztették (Hauser és Clausing 1988). A cikk egy szemléletes adatot is közöl: a Toyota karosszériagyártó vállalatánál az új modellek gyártásindítási és előgyártási költségei 1977 januárja és 1984 áprilisa között, a QFD bevezetésével párhuzamosan, a korábbi szint 39%-ára csökkentek (Hauser és Clausing 1988).

A QFD középpontjában a **minőségház** (house of quality) áll. Hauser és Clausing szerint a minőségház a marketing és a mérnöki tervezés közötti kommunikáció eszköze: olyan közös térkép, amelyen a különböző felelősségű szakemberek a bizonyítékok mintázatára hivatkozva vitathatják meg a tervezési prioritásokat (Hauser és Clausing 1988).

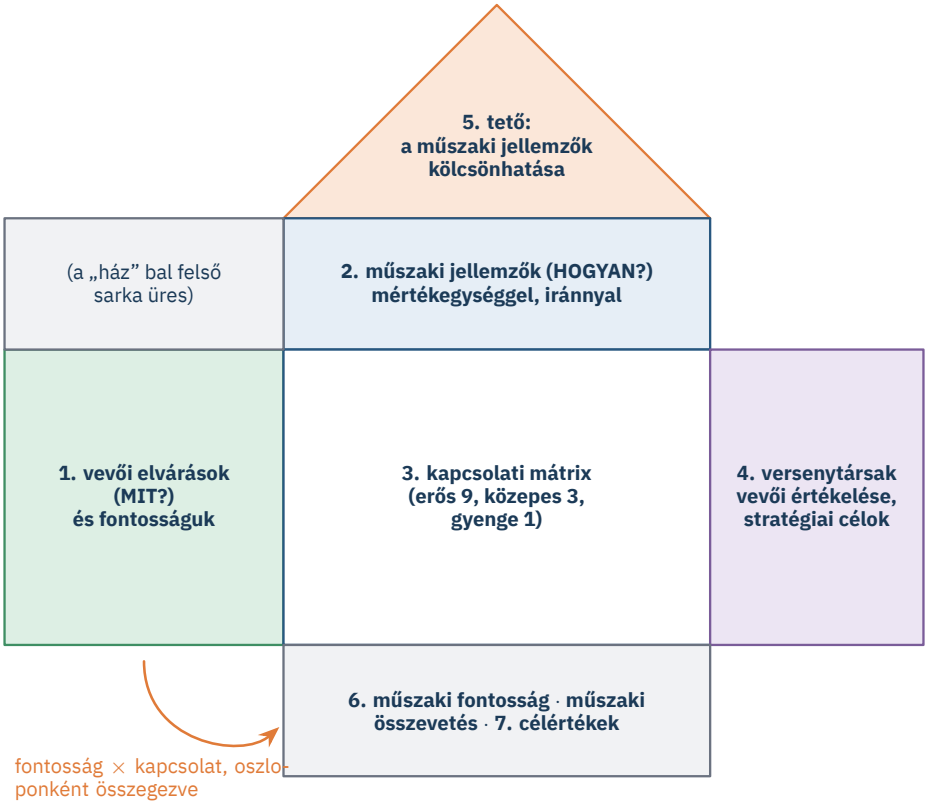
Chan és Wu irodalmi áttekintése szerint a módszer előnyei és hátrányai a következőkben foglalhatók össze (Chan és Wu 2002):

- **Előnyök:** rendszerezett felépítés, amelyet a minőségtervezésben érdekelt minden vállalati terület használhat; a termék tulajdonságai jól lefedik a vevő követelményeit; a fejlesztési erőforrások a vevő számára fontos jellemzőkre összpontosulnak; a területek közötti kommunikáció javul.
- **Hátrányok:** a szakértői csapat idő- és költségigényes egyeztetést igényel; a minőségház már viszonylag egyszerű terméknél is bonyolulttá válhat; a súlyok és a kapcsolati erősségek szubjektív becslések.

7.2. A minőségház felépítése

A minőségház hét „helyiségből” áll (7.1. ábra). A számozás egyben a kitöltés sorrendje is.

1. **Vevői elvárások (MIT?) és fontosságuk.** A ház bal oldala. Az elvárásokat a vevő nyelvén, de már rendezetten soroljuk fel, jellemzően 5–15 tételt. Mindegyikhez fontossági súly tartozik, amelyet felmérésből, interjúkból vagy a Kano-elemzésből (5.5. szakasz) határozunk meg.
2. **Műszaki jellemzők (HOGYAN?).** A ház „mennyezete”. Mérhető, a megoldástól lehetőleg független jellemzők mértékegységgel és a javulás irányával (nagyobb vagy kisebb a jobb). Ezek a leendő MOP-ok (6.4. szakasz).
3. **Kapcsolati mátrix.** A ház közepe. Minden elvárás–jellemző párra megállapítjuk, mennyire hat a műszaki jellemző az elvárás teljesülésére. A szokásos skála: erős (9), közepes (3), gyenge (1) vagy nincs kapcsolat (üres).



7.1. ábra. A minőségház felépítése. A számok a kitöltés javasolt sorrendjét jelölik.

4. **Versenytársak vevői értékelése.** A ház jobb oldala. A vevők hogyan értékelik a saját (vagy tervezett) termékünket és a versenytársakét az egyes elvárások szerint, 1–5 skálán. Ebből látszik, hol vagyunk lemaradásban, és hol tudunk kitűnni.
5. **Tető: a műszaki jellemzők kölcsönhatása.** A tetőmátrix azt mutatja, hogy egy műszaki jellemző javítása segíti (+) vagy rontja (–) egy másik jellemző teljesítését.
6. **Műszaki fontosság és műszaki összevetés.** A ház alja. A műszaki jellemzők fontossága a kapcsolati mátrixból számítható. Ide kerülhet a saját és a versenytárs termékek mért műszaki értékeinek összevetése is.
7. **Célértékek.** Az elemzés eredménye: minden műszaki jellemzőhöz egy célérték, amely a fejlesztés követelménye lesz.

7.2.1. A műszaki fontosság kiszámítása

A j-edik műszaki jellemző **abszolút fontossága** az elvárások fontosságával súlyozott kapcsolati erősségek összege:

$$F_j = \sum_{i=1}^{n} w_i \, r_{ij}, \tag{7.1}$$

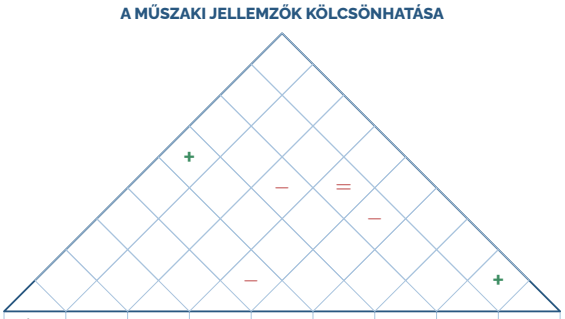
ahol w_i az i-edik vevői elvárás fontossága, $r_{ij} \in \{0, 1, 3, 9\}$ pedig a kapcsolat erőssége. A **relatív fontosság** az abszolút fontosság aránya az összeghez:

$$F_j^{\text{rel}} = \frac{F_j}{\sum_{k=1}^{m} F_k} \cdot 100\%. \tag{7.2}$$

A 9–3–1 skála szándékosan aszimmetrikus: az erős kapcsolatokat jobban kiemeli, mint egy 3–2–1 skála, és így élesebben elkülönülnek a valóban fontos műszaki jellemzők.

7.3. Kidolgozott példa: a VárosiBringa minőségháza

A VárosiBringa csapata a Kano-felmérés és a fókuszcsoportok alapján hat vevői elvárást választott ki, és kilenc műszaki jellemzőt rendelt hozzájuk. A kész minőségházat a 7.2. ábra mutatja.



VEVŐI ELVÁRÁSOK	fontosság	M1 Állomássűrűség a belvárosban ↑	M2 Kerékpár állomásonként ↑	M3 A kölcsönzés átfutási ideje ↓	M4 Akkumulátor-hatótáv ↑	M5 A rásegítés nyomatéka ↑	M6 Áthelyezési körök naponta ↑	M7 A zár feltörési ellenállása ↑	M8 A kerékpár tömege ↓	M9 Fékút 25 km/h-ról ↓	VEVŐI ÉRTÉKELÉS (1–5) saját	A	B
E1 Mindig legyen kerékpár a közelben	5	●	●		○		●				2	3	4
E2 Gyors, egyszerű kölcsönzés	4	○		●				△			3	2	5
E3 Megbízható zár, ne lopják el	4			△				●			4	3	2
E4 Fáradság nélkül fel lehessen jutni a dombra	5				●	●			○		3	1	5
E5 Megfizethető díj, közös bérlet	3	△	○				○				4	5	2
E6 Biztonságos kerékpár (fékek, világítás)	4								○	●	3	4	2
abszolút fontosság		60	54	40	60	45	54	40	27	36			
relatív fontosság [%]		**14,4**	13,0	9,6	**14,4**	10,8	13,0	9,6	6,5	8,7			
célérték		≥ 3	≥ 10	≤ 20	≥ 50	≥ 50	≥ 4	≥ 5	≤ 25	≤ 5			
mértékegység		ál-lomás/km²	db	s	km	Nm	kör/nap	perc	kg	m			

● erős (9) ○ közepes (3) △ gyenge (1) tető: + támogatja, – gyengíti, = erősen gyengíti egymást; ↑ nagyobb, ↓ kisebb a jobb

7.2. ábra. A VárosiBringa minőségháza. A versenytársak: A – a szomszédos város hagyományos (nem elektromos) közösségi kerékpárrendszere; B – szabadon parkoló elektromos roller szolgáltatás.

A számítás lépésről lépésre

Az elvárások fontossága. A csapat a 120 fős felmérés 1–5 skálájú fontossági értékeléseit kerekítette. A legfontosabb az elérhetőség (E1) és a dombos terep leküzdése (E4), mindkettő 5. A Kano-elemzés megerősítette ezt: az elérhetőség egydimenziós, a zár alapvető jellemző.
A kapcsolati mátrix. A csapat soronként haladt, és minden cellánál megkérdezte: ha ezt a műszaki jellemzőt jelentősen javítjuk, mennyivel javul az elvárás teljesülése? Az elérhetőségre (E1) az állomássűrűség (M1), az állomásonkénti kerékpárszám (M2) és az áthelyezési körök száma (M6) hat erősen (9). Az akkumulátor-hatótáv (M4) közepesen (3) hat, mert a lemerült kerékpár nem kölcsönözhető.
A műszaki fontosság. Az állomássűrűség (M1) abszolút fontossága az 7.1. egyenlet szerint:

$$F_{M1} = 5 \cdot 9 + 4 \cdot 3 + 3 \cdot 1 = 45 + 12 + 3 = 60.$$

Ugyanígy $F_{M4} = 5 \cdot 3 + 5 \cdot 9 = 60$, $F_{M2} = F_{M6} = 54$, $F_{M5} = 45$, $F_{M3} = F_{M7} = 40$, $F_{M9} = 36$ és $F_{M8} = 27$. Az összeg 416, így a relatív fontosság például $F_{M1}^{\text{rel}} = 60/416 \approx 14{,}4\%$.
Értelmezés. A két legfontosabb műszaki jellemző az állomássűrűség és az akkumulátor-hatótáv (14,4–14,4%), őket az állomásonkénti kerékpárszám és az áthelyezés követi (13,0–13,0%). A rendszer sikerét tehát elsősorban a hálózat és a kerékpárok energiaellátása határozza meg, nem a kerékpár tömege vagy a zár feltörési ellenállása. Ez nem jelenti, hogy a zár nem fontos. Alapvető jellemzőként teljesítenie kell egy elfogadható szintet, de azon túl a fejlesztése nem növeli számottevően az elégedettséget.
A versenytársak értékelése. A vevők a tervezett rendszert az elérhetőségben (2) mindkét versenytársnál gyengébbnek látják, a díj és a zár szempontjából viszont jobbnak. A domb leküzdésében az elektromos roller (B) a mérce (5). A stratégiai következtetés: a nyitó hálózatnak elég sűrűnek kell lennie, a rásegítés teljesítményének pedig el kell érnie a roller szintjét.
A tető. A tetőmátrix két jelentős ellentmondást mutat. A hatótáv növelése nagyobb akkumulátort, így nagyobb tömeget jelent (M4–M8, erősen gyengíti). Az erősebb rásegítés növeli a fogyasztást, így csökkenti a hatótávot (M5–M4), és a motor tömege is nő (M5–M8). Ezek a 7.4. szakaszban tárgyalt kompromisszumok.
A célértékek. Az elemzés alapján a csapat kilenc célértéket rögzített (a ház alsó sora), amelyek a rendszerkövetelmények közé kerültek. Például: belvárosi állomássűrűség legalább 3 állomás/km², akkumulátor-hatótáv legalább 50 km, rásegítési nyomaték legalább 50 Nm, tömeg legfeljebb 25 kg.

7.4. A tető: ellentmondások és kompromisszumok

A tetőmátrix a minőségház talán legértékesebb, mégis gyakran elhanyagolt része. Megmutatja, hogy a műszaki jellemzők nem függetlenek egymástól: mely paraméterek gyengítik, és melyek erősítik egymást.

A negatív kapcsolatokkal háromféleképpen bánhatunk:

1. **Kompromisszum.** A két jellemző közül a fontosabbnak adunk elsőbbséget, és a másikra elfogadható célértéket szabunk. A VárosiBringánál a hatótáv relatív fontossága (14,4%) jóval nagyobb a tömegénél (6,5%), ezért a csapat inkább a nagyobb akkumulátort választotta, és a tömegre 25 kg-os felső határt szabott.
2. **Rendszerszintű megoldás.** A konfliktust a rendszer egy másik szintjén oldjuk fel. A hatótáv–tömeg ellentmondás enyhíthető úgy is, hogy az akkumulátort nem kell nagyra méretezni, mert a dokkolók töltenek. A követelmény így „napi 50 km” helyett „két dokkolás között 50 km” lehet.

3. **Innovatív megoldás.** Az ellentmondás egy új műszaki elvvel megszüntethető. A TRIZ-módszertan (9. fejezet) éppen az ilyen műszaki ellentmondások rendszerezett feloldására való. A tetőmátrix negatív kapcsolatai ezért közvetlen bemenetei a koncepcióalkotásnak.

A pozitív kapcsolatok is fontosak, mert egymást erősítő hatást jeleznek. A VárosiBringánál a sűrűbb állomáshálózat (M1) rövidebb áthelyezési utakat jelent, így ugyanakkora áthelyezési kapacitással több kör tehető meg (M6).

7.5. A Kano-modell és a QFD

A minőségház bal oldalán a vevői elvárások fontossága áll. Egyszerűbb esetben ezt közvetlenül egy felmérés fontossági értékeléseiből vesszük. A Kano-modell (5.5. szakasz) azonban rámutat, hogy a fontosság és az elégedettségre gyakorolt hatás nem ugyanaz. Az alapvető jellemzők hiánya erős elégedetlenséget okoz, de a teljesítésükön túli javítás alig növeli az elégedettséget. A vonzó jellemzők viszont a vevő által kimondottan nem kért, mégis nagy elégedettséget okozó lehetőségek.

Matzler és Hinterhuber javasolta, hogy a Kano-modell eredményeit építsük be a QFD-be (Matzler és Hinterhuber 1998). A gyakorlatban ez három dolgot jelent:

- **Az alapvető jellemzőkre** elég a versenyképes, elfogadható szintet célozni. Ezeket nem érdemes a fontossági súlyok alapján túlfejleszteni.
- **Az egydimenziós jellemzőkben** érdemes a versenytársak fölé célozni, mert itt a teljesítmény arányosan növeli az elégedettséget.
- **A vonzó jellemzők** a megkülönböztetés lehetőségei. A minőségházban akkor is érdemes szerepeltetni őket, ha a felmérésben kimondott fontosságuk alacsony.

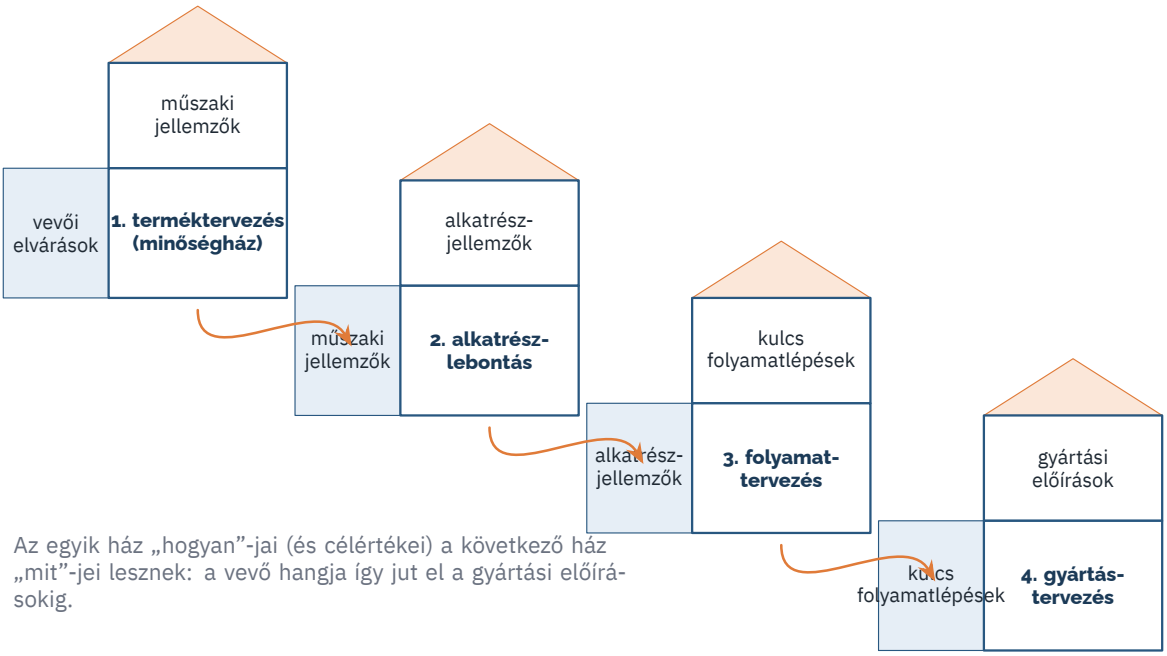
A VárosiBringánál ezért került be a vevői elvárások közé a közös bérlet (E5). A felmérésben mért fontossága közepes (3) volt, de a Kano-elemzés vonzó jellemzőként azonosította, és a helyi közlekedési társasággal kötött megállapodás olcsón megvalósíthatóvá tette.

7.6. A négyfázisú QFD

A minőségház csak az első lépés. Hauser és Clausing leírása szerint a házak egymáshoz kapcsolhatók: az egyik ház „hogyan”-jai a következő ház „mit”-jeivé válnak. Így a vevő hangja egészen a gyártásig eljut (7.3. ábra) (Hauser és Clausing 1988):

1. **terméktervezés:** vevői elvárások → műszaki jellemzők (a minőségház);
2. **alkatrész-lebontás:** műszaki jellemzők → az alkatrészek jellemzői;
3. **folyamattervezés:** alkatrész-jellemzők → kritikus folyamatlépések és paramétereik;
4. **gyártástervezés:** folyamatlépések → gyártási előírások, ellenőrzési tervek.

A négy fázis jól illeszkedik a V-modell bal ágához. Az első ház a rendszerkövetelmények szintje, a második az alrendszerek és komponensek allokációja (6.5. szakasz), a harmadik és a negyedik pedig a V csúcsa, a megvalósítás és a gyártás. A gyakorlatban ritkán készül mind a négy ház teljes részletességgel. Gyakoribb, hogy a minőségház mellett csak a kritikus műszaki jellemzőkre készül alkatrész- vagy folyamatszintű lebontás.



Az egyik ház „hogyan”-jai (és célértékei) a következő ház „mit”-jei lesznek: a vevő hangja így jut el a gyártási előírásokig.

7.3. ábra. A négyfázisú QFD: az egymáshoz kapcsolt házak a vevői elvárásokat lépésről lépésre a gyártási előírásokig vezetik.

KoboCella: a második ház

A KoboCella minőségházában a csavarkötés minősége (a nyomaték pontossága) lett a legfontosabb műszaki jellemző, mert erősen hat a vevő minőségi elvárására és a selejtcsökkentésre is. A csapat erre az egy jellemzőre elkészítette a második házat. Ebben a nyomatékpontosság a „mit”, a csavarozó egység alkatrészjellemzői (az orsó fordulatszám-szabályozása, a nyomatékérzékelő pontossága, a bitbetét kopásállósága, a csavaradagolás megbízhatósága) pedig a „hogyan”. Kiderült, hogy a nyomatékérzékelő pontossága és a bitbetét kopása együtt határozza meg a nyomatékszórást. Ezért a bitbetét cseréjének gyakorisága a karbantartási terv kritikus eleme lett. Ilyen összefüggés a rendszerszintű minőségházból nem látszott volna.

7.7. Korlátok és újabb kiterjesztések

A QFD széles körben elterjedt, de a gyakorlatban több korlátja is ismert:

- **Szubjektivitás.** A kapcsolati erősségek és a fontossági súlyok csapatbecslések. Kis eltérések megváltoztathatják a rangsort, ezért érdemes érzékenységvizsgálatot végezni. Az egyenlő fontosságú jellemzők között a rangsort csak óvatosan szabad értelmezni.
- **Méret.** Sok elvárásnál és jellemzőnél a mátrix gyorsan átláthatatlanná válik, akár egy viszonylag egyszerű terméknél is. Ilyenkor az elvárásokat hierarchiába kell rendezni, és csak a legfontosabbakkal kell dolgozni.
- **A tető kihasználatlansága.** Az egyszerű összegzés (7.1. egyenlet) nem veszi figyelembe a műszaki jellemzők közötti kölcsönhatásokat, így a közvetett hatások rejtve maradnak.
- **Költségvakság.** A műszaki fontosság nem mond semmit arról, mennyibe kerül egy jellemző javítása. Egy kevésbé fontos jellemző olcsó javítása több elégedettséget hozhat egységnyi ráfordításra, mint egy fontos jellemző drága javítása.

Hálózatalapú QFD

Az utóbbi két korlátra a hálózattudomány és az optimalizálás eszközei adnak választ. Abonyi és Czvetkó tanulmánya a QFD két új megfogalmazását javasolta (Abonyi és Czvetkó 2022). Az első a QFD-t hálózati folyamproblémaként írja fel, két céllal: lineáris optimalizálással a vevői igények kielégítéséből származó hasznot maximalizálja, illetve minimális költségű folyamként a vevői követelmények teljesítéséhez szükséges intézkedések összköltségét minimalizálja. A második a QFD-t hipergráfként ábrázolja, amely egyszerre képes leírni a kapcsolati és a korrelációs (tető-) mátrix kölcsönhatásait, és a legnagyobb hatású tényezőket centralitási mérőszámokkal azonosítja. A szerzők eredményei szerint a hipergráf-alapú ábrázolás feltárta a fejlesztési paraméterek és a vevői követelmények közötti közvetett kölcsönhatásokat is.
A szervezeti fenntarthatóság és az Ipar 5.0-megoldások összekapcsolására ugyanez a kutatócsoport hálózat- és QFD-alapú keretrendszert dolgozott ki. Ez azt segíti eldönteni, mely Ipar 5.0-megoldások járulnak hozzá leginkább a fenntarthatósági teljesítmény javításához (Czvetkó, Sebestyén, és Abonyi 2025). A Gyakorlati útmutató QFD-notebookja a tetőmátrixot is figyelembe vevő, közvetett hatásokat számító egyszerű kiterjesztést mutat be.

A minőségház mint űrlap

A minőségház értéke nem a kitöltött táblázat, hanem az a vita, amely a kitöltése közben zajlik. Ha egy ember egyedül, egy délután alatt kitölti, a ház a saját előfeltevéseit rögzíti, és hamis tudományosságot kölcsönöz nekik. A kapcsolati mátrixot és a tetőt mindig vegyes csapat (marketing, tervezés, gyártás, szerviz) töltse ki, és minden vitás cellánál rögzítsék az indoklást.

Összefoglalás

- A QFD a vevő hangját fordítja le műszaki jellemzőkre és célértékekre. Központi eszköze a minőségház, amely a marketing és a mérnöki tervezés közös térképe.
- A minőségház hét része: vevői elvárások és fontosságuk, műszaki jellemzők, kapcsolati mátrix (9–3–1), versenytárs-értékelés, tető, műszaki fontosság és célértékek.
- A műszaki jellemzők abszolút fontossága a fontossággal súlyozott kapcsolati erősségek összege, a relatív fontosság ennek százalékos aránya. Az eredmény megmutatja, hova érdemes a fejlesztési erőforrásokat összpontosítani.
- A tető negatív kapcsolatai ellentmondásokat jeleznek. Ezek kompromisszummal, rendszerszintű megoldással vagy innovációval (TRIZ) oldhatók fel.
- A Kano-modell segít a célértékek megválasztásában: az alapvető jellemzőknél elég az elfogadható szint, az egydimenziósaknál a versenytársak fölé kell célozni, a vonzók a megkülönböztetés lehetőségei.
- A négyfázisú QFD a vevő hangját a gyártási előírásokig vezeti. A módszer korlátai a szubjektivitás, a méret, a közvetett hatások figyelmen kívül hagyása és a költségvakság. A hálózatalapú kiterjesztések ezek egy részére adnak választ.

Önellenőrzés

1. Mi a különbség a vevői elvárás és a műszaki jellemző között? Melyik kerül a minőségház bal oldalára, és melyik a mennyezetére?
2. Számítsd ki az M5 (rásegítés nyomatéka) és az M8 (tömeg) abszolút fontosságát a VárosiBringa minőségházából!
3. Ha az E3 (megbízható zár) fontosságát 4-ről 5-re emelnénk, hogyan változna az M7 abszolút és relatív fontossága?
4. A tetőmátrixban az M3 (kölcsönzési idő) és az M7 (zár feltörési ellenállása) között negatív kapcsolat van. Mi lehet ennek a műszaki oka, és hogyan oldanád fel?
5. Miért nem érdemes egy alapvető jellemzőt a minőségház fontossági súlya alapján túlfejleszteni?
6. Mi a szerepe a második, alkatrész-szintű háznak? Adj példát a ForgácsTech esetéből!

Megoldás.

1. A vevői elvárás a vevő nyelvén fogalmazott kívánság (például „gyors kölcsönzés”), a bal oldalra kerül. A műszaki jellemző mérhető, mértékegységgel és iránnyal rendelkező mennyiség (például „a kölcsönzés átfutási ideje, s, kisebb a jobb”), a mennyezetre kerül.
2. $F_{M5} = 5 \cdot 9 = 45$ (csak az E4-gyel van erős kapcsolata). $F_{M8} = 5 \cdot 3 + 4 \cdot 3 = 27$ (E4 és E6 közepes kapcsolattal).
3. $F_{M7} = 4 \cdot 1 + 5 \cdot 9 = 49$ lenne (a korábbi 40 helyett). Az összeg 416-ról 426-ra nő (az M7-é 9-cel, az M3-é 1-gyel, 41-re nő), így az M7 relatív fontossága $49/426 \approx 11{,}5\%$ lenne a korábbi 9,6% helyett.
4. Egy erősebb, összetettebb zár (például több reteszelési pont, mechanikus kulcs mellett elektronikus azonosítás) lassabb nyitást és több hibalehetőséget jelenthet. Feloldás lehet a gyorsan nyitó elektromechanikus zár és a fizikai feltöréssel szembeni ellenállás szétválasztása: a zárszerkezet edzett, a kioldás elektronikus és gyors. A rendszerszintű megoldás a lopás utólagos felderítése (helymeghatározás), így a zárra kisebb terhelés jut.
5. Mert az alapvető jellemzők elégedettségre gyakorolt hatása telítődik: a hiányuk nagy elégedetlenséget okoz, de az elfogadható szinten túli javításuk alig növeli az elégedettséget. A ráfordítás jobb helyen van az egydimenziós vagy a vonzó jellemzőknél.
6. A második ház a rendszerszintű műszaki jellemzőket az alkatrészek vagy alrendszerek jellemzőire bontja le, és megmutatja, melyik alkatrészjellemző a kritikus. A ForgácsTech-nél például a „gépállapot-adatok késleltetése” műszaki jellemzőt a gépvezérlő adatkapcsolata, a hálózati infrastruktúra és a MES adatgyűjtő moduljának jellemzőire lehetne lebontani.

Projektlépés: a minőségház

Saját fejlesztési feladatodhoz készíts minőségházat:

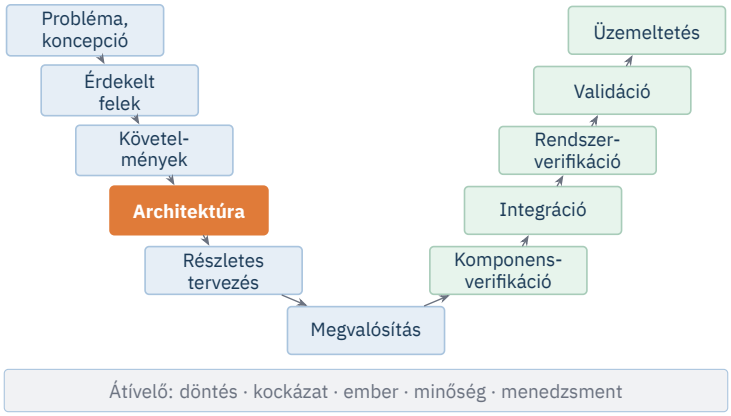
1. válassz ki 5–8 vevői elvárást, és rendelj hozzájuk fontossági súlyt (lehetőleg felmérés vagy Kano-elemzés alapján);
2. határozz meg 5–8 mérhető műszaki jellemzőt mértékegységgel és iránnyal;
3. töltsd ki a kapcsolati mátrixot és a tetőt, és rögzítsd a vitás cellák indoklását;
4. értékeld a saját koncepciódat és legalább egy versenytárs megoldását a vevő szemével;
5. számítsd ki a műszaki fontosságot, és határozz meg célértékeket;
6. írd le a két legfontosabb tanulságot és a tető egy ellentmondásának feloldási javaslatát.

A sablon és a notebook a Gyakorlati útmutató QFD-eszközlapján található.

Továbbolvasás

- Hauser és Clausing (1988): a minőségház klasszikus, ma is olvasmányos bemutatása.
- Akao (1990): a QFD módszertanának alapműve a módszer egyik megalkotójától.
- Matzler és Hinterhuber (1998): a Kano-modell beépítése a QFD-be.
- Abonyi és Czvetkó (2022): a QFD hálózati folyam- és hipergráf-alapú megfogalmazása.
- Czvetkó, Sebestyén, és Abonyi (2025): hálózat- és QFD-alapú keretrendszer az Ipar 5.0 és a szervezeti fenntarthatóság kapcsolatára.

8. Funkció, forma, koncepció: a rendszer architektúrája



Tanulási eredmények

A fejezet feldolgozása után képes leszel:

- megkülönböztetni a rendszer funkcióját, formáját és koncepcióját, és megoldássemlegesen megfogalmazni egy funkciót;
- elhelyezni az architektúra kialakítását (a logikai dekompozíciót) a V-modellben, és felsorolni a lépéseit;
- funkcionális elemzést végezni: funkciófát, funkcionális folyamatábrát (FFBD) és IDEF0-ábrát készíteni;
- a funkciókat a forma elemeihez allokálni, és felismerni az allokációból születő levezetett követelményeket;
- N²-diagrammal azonosítani az alrendszerek közötti interfészeket, és ismertetni az interfészkezelés feladatait;
- tervezési struktúramátrixot (DSM) készíteni, klaszterezéssel modulokat kialakítani, és értékelni egy architektúra modularitását.

Az előző fejezetek a probléma térben mozogtak: megértettük, mit várnak az érdekelt felek (5. fejezet), és ezt mérhető követelményekké fordítottuk (6. és 7. fejezet). A követelmények azt mondják meg, *mit* kell tudnia a rendszernek. Arról, hogy *milyen részekből áll*, és ezek *hogyan dolgoznak együtt*, még semmit nem mondanak. Ezt a lépést nevezzük architektúra-tervezésnek; az ISO/IEC/IEEE 15288 szabvány az architektúra-meghatározás folyamatának hívja (ISO/IEC/IEEE 2023, 6.4.4. szakasz).

Az architektúra a rendszer legnehezebben megváltoztatható döntéseinek összessége. Ha egy alkatrészt rosszul méreteztünk, kicseréljük. Ha viszont a funkciókat rosszul osztottuk szét az alrendszerek között, vagy túl sok és túl bonyolult interfészt hoztunk létre, a hiba a rendszer egész életén végigkíséri: nehéz lesz integrálni, tesztelni, karbantartani és továbbfejleszteni. Blanchard és Fabrycky ezt a fejlesztés alapvető ellentmondásaként írja le: a fejlesztés elején a legkönnyebb változtatni, de ekkor a legkevesebb a rendszerre

vonatkozó tudásunk (Blanchard és Fabrycky 2011). Az architektúrát tehát akkor kell kialakítanunk, amikor a legkevesebbet tudunk, ezért rendszerezett módszerekre van szükség.

A fejezet négy kérdésre épül:

- Mit jelent a funkció, a forma és a koncepció, és hogyan kapcsolódnak egymáshoz?
- Hogyan bontjuk le a rendszer funkcióit, és hogyan ábrázoljuk a lefutásukat?
- Hogyan rendeljük a funkciókat a forma elemeihez, és hogyan azonosítjuk az interfészeket?
- Mitől jó egy architektúra, és hogyan alakíthatunk ki modulokat?

8.1. Funkció, forma és koncepció

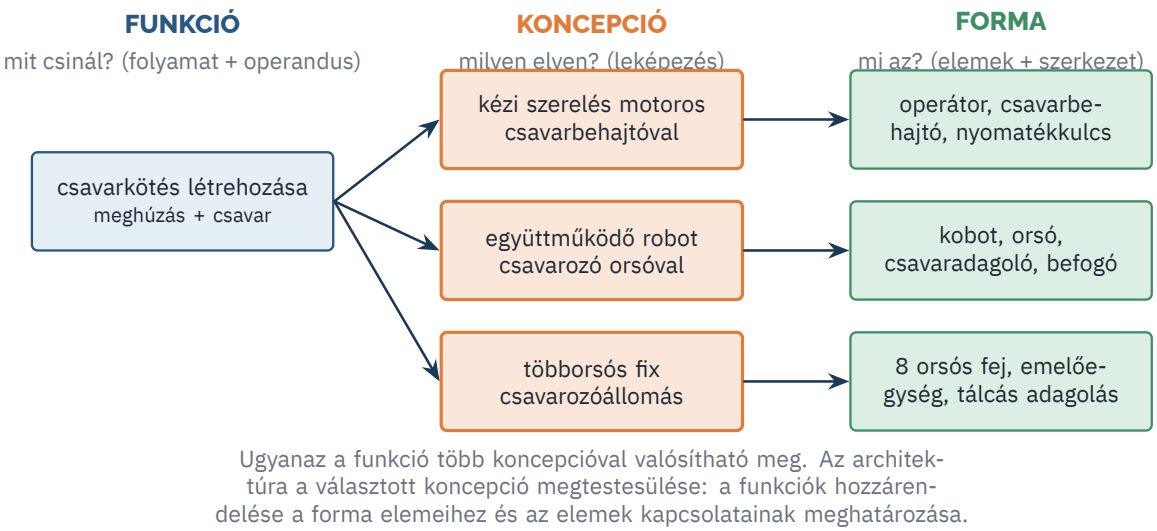
A fejezet Crawley és szerzőtársai rendszerarchitektúra-felfogására épül (Crawley, Cameron, és Selva 2016). Ebben a felfogásban minden rendszer három nézőpontból írható le.

Funkció, forma, koncepció

- **Funkció:** az, amit a rendszer *csinál*. Tevékenység, folyamat, amely létrehoz vagy átalakít valamit. Crawley és szerzőtársai szerint a funkció két részből áll: a **folyamatból** (process, ige) és az **operandusból** (operand, főnév), amelyen a folyamat végbemegy (Crawley, Cameron, és Selva 2016). A „csavarkötés létrehozása” funkcióban a létrehozás a folyamat, a csavarkötés az operandus.
- **Forma:** az, ami a rendszer *van*. Az elemek összessége és a fizikai vagy logikai valóság tagoltsága: alak, konfiguráció, struktúra. A forma az, ami megvalósul (gyártják, megírják, összeszerelik), és gyakran, de nem mindig látható.
- **Koncepció:** a rendszer elképzelése, amely a funkciót a formára képezi le. Crawley és szerzőtársai szerint a koncepció a működési elveket hordozza, és egyben a forma absztrakciója (Crawley, Cameron, és Selva 2016). A koncepció nem maga a termék, hanem az a gondolati leképezés, amely megmutatja, miként valósul meg a funkció egy adott formában.

Crawley és szerzőtársai erre a három fogalomra építik az architektúra definícióját is: az architektúra a koncepció megtestesülése, vagyis a fizikai és információs funkciók hozzárendelése a forma elemeihez, valamint az elemek egymás közötti és a környezettel való kapcsolatainak meghatározása (Crawley, Cameron, és Selva 2016). A koncepcionális tervezés feladata tehát hármas: megalkotni a koncepciót, hozzárendelni a funkciókat a megfelelő formai elemekhez, és meghatározni az elemek közötti kapcsolatokat.

8.1. ábra a KoboCella egyik funkcióján mutatja be a három fogalom kapcsolatát. A „csavarkötés létrehozása” funkció legalább három koncepcióval valósítható meg: kézi szereléssel, együttműködő robottal vagy többorsós fix csavarozóállomással. Mindegyik koncepció más formát eredményez, más elemekkel és más kapcsolatokkal. A koncepció megválasztása határozza meg a rendszer teljesítményének, költségének és kockázatainak nagyobbik részét. Ezért kell a koncepciók között tudatosan választani (10. fejezet), és nem az első ötletnél megállni (9. fejezet).



8.1. ábra. Funkció, koncepció és forma a KoboCella csavarozási funkcióján. Ugyanaz a funkció több koncepcióval valósítható meg, és mindegyik más formához vezet.

8.1.1. A megoldássemleges funkciómegfogalmazás

A funkció megfogalmazásában rejlik az architektúra-tervezés egyik leggyakoribb csapdája. Ha a funkciót a megoldás nevével írjuk le („robotos csavarozás”), a koncepciót már előre eldöntöttük, és a többi lehetőség fel sem merül. Crawley és szerzőtársai ezért **megoldássemleges** (solution-neutral) funkciómegfogalmazást javasolnak: a funkciót úgy kell leírni, hogy az operandus és a kívánt állapotváltozás szerepeljen benne, a megvalósítás módja viszont ne (Crawley, Cameron, és Selva 2016). A „robotos csavarozás” helyett tehát „csavarkötés létrehozása előírt nyomatékkal”. Ebből a megfogalmazásból kiindulva a kézi szerelés, a kobot, a fix állomás, de akár a csavarkötés elhagyása (például pattintott vagy ragasztott kötés) is versenybe szállhat.

Az amerikai védelmi beszerzési egyetem rendszermérnöki alapjegyzete egyszerű ellenőrzési szabályt ad: a funkciók legyenek igék. Ha egy funkció főnévként jelenik meg, akkor valami baj van a funkciók megértésével (Defense Acquisition University 2001, 5. fejezet). Magyarul ez a gyakorlatban igei szerkezetet jelent („csavar adagolása”, „védőzóna figyelése”), amelyből egyértelműen kiolvasható a folyamat és az operandus.

> **Tépőzár és szögesdrót: a természet mint koncepcióforrás**
>
> Két klasszikus példa mutatja, hogyan születik új koncepció egy funkció megoldássemleges megfogalmazásából (Vogel 1998). A tépőzárat George de Mestral svájci mérnök találta fel, miután egy vadászat után mikroszkóp alatt megvizsgálta a kutyája bundájába akadt bogáncsok horgas végződéseit. A szögesdrót a 19. századi Amerikában a legelők elkerítésére született. A telepesek addig tüskés sövényekkel próbálkoztak, de ezek lassan nőttek; az első szögesdrót-szabadalom 1868-ban kifejezetten azt a célt fogalmazta meg, hogy a drótkerítés a tüskés sövényhez hasonló legyen. Mindkét esetben a funkció („két felület oldható összekapcsolása”, „állatok távol tartása”) vezetett el egy addig szokatlan formához. A természet ihlette tervezés módszereivel a 9. fejezet foglalkozik.

8.2. Az architektúra helye a V-modellben

Az ISO/IEC/IEEE 15288 szabvány két folyamatot szentel ennek a lépésnek. Az **architektúra-meghatározás** (architecture definition) célja, hogy architektúraváltozatokat hozzon létre, kiválassza közülük azt, amely a legjobban kezeli az érdekelt felek szempontjait és teljesíti a rendszerkövetelményeket, és ezt egymással összhangban lévő nézetekben és modellekben rögzítse. A **tervmeghatározás** (design definition) pedig annyi részletet ad meg a rendszerről és elemeiről, amennyi a megvalósításukhoz szükséges (ISO/IEC/IEEE 2023, 6.4.4. és 6.4.5. szakasz). A V-modellben ez a bal ág második harmada: a rendszerkövetelmények és az alrendszerek, komponensek tervezése között helyezkedik el (2. fejezet).

Az architektúra-meghatározás a szabvány szerint hat tevékenységből áll (ISO/IEC/IEEE 2023, 6.4.4. szakasz):

1. az architektúra-meghatározás előkészítése (az érdekelt felek szempontjainak és a legfontosabb követelményeknek az összegyűjtése);
2. az architektúra nézőpontjainak kidolgozása;
3. a jelölt architektúrák modelljeinek és nézeteinek elkészítése, a funkciók és jellemzők hozzárendelése az architektúra elemeihez;
4. az architektúra és a részletes tervezés összekapcsolása;
5. a jelölt architektúrák értékelése és a választás;
6. a kiválasztott architektúra kezelése: jóváhagyás, alapvonal, nyomonkövethetőség.

Három gondolat érdemel külön figyelmet. Az első, hogy az architektúra arról szól, hogyan illeszkednek az elemek egymáshoz és az egészhez, és nem az egyes elemek belső működéséről. Ha az elemek közötti kapcsolatok jól meghatározottak, az elemek egymástól függetlenül fejleszthetők, mégis együtt fognak működni (Baldwin és Clark 2000). A második, hogy nem egy, hanem *több* architektúraváltozatot kell kidolgozni, mert csak így van miből választani; a szabvány is változatokban gondolkodik (ISO/IEC/IEEE 2023, 6.4.4. szakasz). A harmadik, hogy az architektúra kialakítása kreatív, rekurzív és iteratív munka. A funkciók szinte mindig többféleképpen bonthatók le, ezért az eredmény nagymértékben függ a csapat kreativitásától, tudásától és tapasztalatától. Maier és Rechtin ezért nevezi az architektúra-tervezést egyszerre tudománynak és művészetnek (Maier és Rechtin 2009).

A rekurzió azt jelenti, hogy ugyanazt az eljárást minden szinten megismételjük: a rendszer architektúrájából alrendszer-követelmények születnek, az alrendszerek architektúrájából komponenskövetelmények, és így tovább (2.4. szakasz). Az iteráció azt jelenti, hogy egy alsóbb szinten felismert nehézség visszahathat a felsőbb szintű döntésekre. A rendszermérnöknek ezért nyitottnak kell maradnia a korábban rögzített architektúra és követelmények megváltoztatására. Véglegesíteni akkor érdemes őket, amikor a rendszer minden szinten definiált, és minden követelményről kiderült, hogy megvalósítható, verifikálható és ellentmondásmentes.

> **Architektúra és architektúraleírás**
>
> Az ISO/IEC/IEEE 42010:2022 szabvány megkülönbözteti egy entitás *architektúráját* és az azt kifejező *architektúraleírást* (architecture description) (ISO/IEC/IEEE 2022). Az architektúra a rendszer alapvető tulajdonsága, akkor is létezik, ha senki nem írta le. Az architektúraleírás az a munkatermék, amely nézőpontok és nézetek segítségével rögzíti. A fejezet eszközei (funkciófa, FFBD, IDEF0, N²-diagram, DSM) egy-egy nézetet adnak; egyik sem írja le egyedül a teljes architektúrát. A nézőpontok és a modellalapú leírás kapcsolatát a 3.5. szakasz tárgyalja.

8.3. Funkcionális elemzés és funkciófa

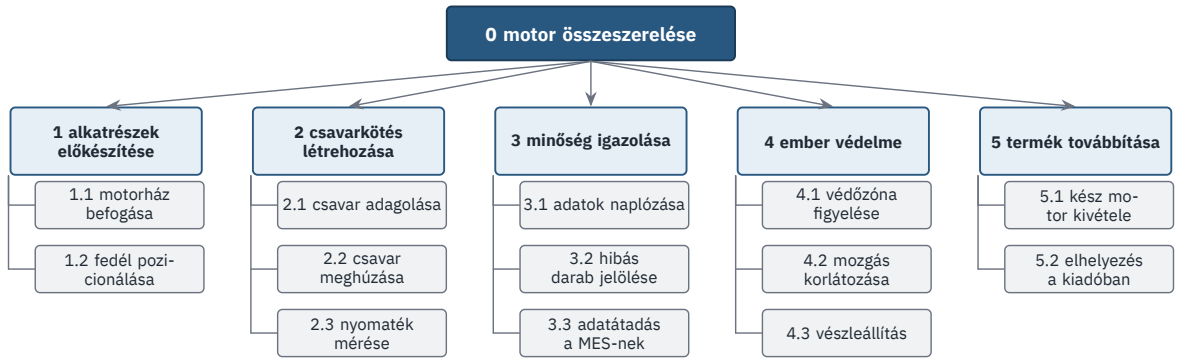
A funkcionális elemzés az architektúra kialakításának és a funkcionális követelmények lebontásának elsődleges módszere. Felülről lefelé haladó, rendszerezett eljárás, amely azonosítja, leírja és egymáshoz kapcsolja azokat a funkciókat, amelyeket a rendszernek a céljai eléréséhez el kell végeznie (Defense Acquisition University 2001, 5. fejezet; NASA 2016, 4.3. szakasz). Kiindulópontja a működési koncepció (ConOps, 4.4. szakasz). Három fő lépésből áll:

1. a legfelső szintű követelmények funkciókká fordítása, vagyis annak meghatározása, milyen funkciókat kell elvégezni a követelmények teljesítéséhez;
2. a funkciók lebontása és allokálása a termékszerkezet alacsonyabb szintjeire;
3. a funkcionális és az alrendszerek közötti interfészek azonosítása és leírása.

Érdemes minden azonosított funkciót a bemeneteivel, kimeneteivel, lehetséges hibamódjaival, a hiba következményeivel és az interfészkövetelményekkel együtt leírni, a funkciókat pedig logikai sorrendbe rendezni, hogy a rendszer bármely előírt használata végigkövethető legyen a kezdettől a végéig. Ez a leírás egyúttal a későbbi hibamód-elemzés (15. fejezet) kiindulópontja is.

8.3.1. A funkciófa

A funkcionális lebontás legegyszerűbb ábrázolása a **funkciófa** (functional hierarchy). A fa gyökere a rendszer fő funkciója, ennek gyermekei a részfunkciók, és így tovább, amíg el nem érjük azt a szintet, ahol egy funkció már egyértelműen egy formai elemhez rendelhető. A funkciókat számozással azonosítjuk (1, 1.1, 1.1.1), mert a számozás a teljes elemzésen végigvihető, és lehetővé teszi a lefelé és felfelé irányuló nyomonkövetést (Defense Acquisition University 2001, 5. fejezet).



Igei szerkezetű funkciók, megoldásmentesen: a fa azt mutatja, mit kell a cellának tennie, nem azt, hogy melyik elem teszi.

8.2. ábra. A KoboCella funkciófája két szinten. A funkciók igei szerkezetűek és megoldássemlegesek.

> **KoboCella: a funkciófa**
>
> A KoboCella csapata a működési koncepció forgatókönyveiből indult ki (4.4. szakasz), és öt főfunkciót azonosított (8.2. ábra): az alkatrészek előkészítését, a csavarkötés létrehozását, a minőség igazolását, az ember védelmét és a termék továbbítását. Az első változatban a második főfunkció „kobotos csavarozás” volt. A felülvizsgálaton a minőségügyi vezető megkérdezte, miért nem szerepel a fában a „kézi csavarozás” mint lehetőség. Ebből derült ki,

hogy a csapat a fa felírásakor már eldöntötte a koncepciót. A megoldássemleges átfogalmazás után a csavarkötés létrehozása három részfunkcióra bomlott: csavar adagolása, csavar meghúzása, nyomaték mérése. Ezek közül csak a második kötődik közvetlenül a kobothoz. A negyedik főfunkció („ember védelme") külön ágat kapott, mert a biztonsági funkciók allokációja és verifikációja szigorúbb szabályokat követ (17. fejezet). Az első két szint 5 főfunkciót és 13 részfunkciót tartalmaz. Ez kezelhető méret: a csapat egyetlen falitáblán át tudta tekinteni.

8.3.2. Szintenként hány elem?

Gyakran idézett hüvelykujjszabály, hogy egy jól áttekinthető rendszer egy szintjén 5–9 rész szerepeljen. A szabály Miller klasszikus tanulmányára vezethető vissza, amely szerint az emberi abszolút ítélőképesség és az azonnali emlékezet terjedelme hét plusz-mínusz kettő elem körül van (Miller 1956). A későbbi kutatások ezt finomították. Cowan áttekintése szerint a rövid távú emlékezet valódi kapacitása, ha a csoportosítás és az ismétlés lehetőségét kizárjuk, inkább három-öt egység (Cowan 2001). A rendszermérnöki gyakorlat számára a tanulság nem egy pontos szám, hanem az, hogy ha egy szinten tizenöt-húsz elem szerepel, akkor a lebontás valószínűleg kihagyott egy közbülső szintet, és a csapat nem fogja tudni átlátni a kapcsolatokat.

8.3.3. Teljesítményallokáció a funkciókra

A funkcionális architektúra nemcsak azt mutatja meg, milyen funkciókat kell elvégezni, hanem azt is, milyen teljesítménykövetelmények tartoznak hozzájuk. A DAU-jegyzet szerint a funkcionális architektúra a rendszer funkcionális és teljesítménykövetelményeinek felülről lefelé haladó lebontása, amely a funkciók logikai sorrendjét és a hozzájuk rendelt teljesítménykövetelményeket is tartalmazza (Defense Acquisition University 2001, 5. fejezet). A KoboCella esetében ez a ciklusidő-költségvetés (6.5. táblázat): a 75 másodperces rendszerszintű követelményt a funkciókhoz és végrehajtóikhoz rendelt időkre bontottuk. A teljesítményallokáció tehát a funkciófa és a követelménylebontás (6.5. szakasz) közös terméke.

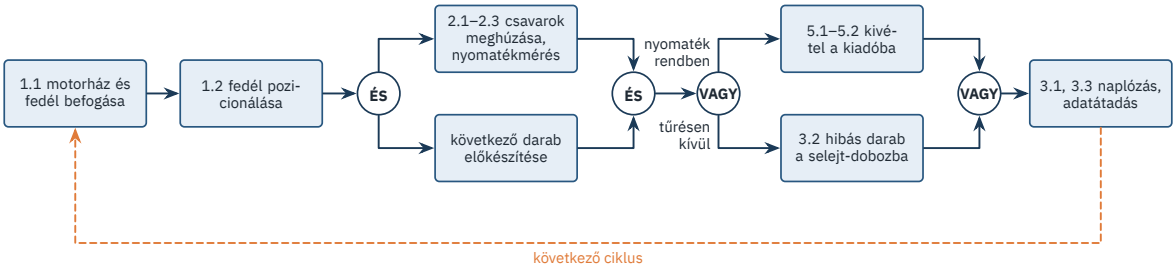
8.4. A funkciók lefutása: FFBD és IDEF0

A funkciófa a funkciók hierarchiáját mutatja meg, a sorrendjüket viszont nem. A rendszer viselkedésének megértéséhez azt is tudnunk kell, melyik funkció melyik után következik, mi fut párhuzamosan, és hol ágazik el a folyamat. Erre két hagyományos ábrázolás terjedt el.

8.4.1. Funkcionális folyamatábra (FFBD)

A **funkcionális folyamatábra** (functional flow block diagram, FFBD) célja a DAU-jegyzet szerint az, hogy a rendszerkövetelményeket funkcionális fogalmakkal írja le. Az ábra funkcióorientált, nem megoldásorientált, és segítségével ellenőrizhető, hogy az életciklus valamennyi funkciója le van-e fedve, a tevékenységek sorrendje és a kritikus tervezési interfészek pedig meg legyenek határozva (Defense Acquisition University 2001, 5–A. kiegészítés). Az FFBD fő szabályai:

- minden funkciót egyetlen, folytonos vonalú doboz jelöl, amely meghatározott, véges, elkülönülő tevékenységet ír le;
- a funkciók számozása szintenként következetes, és a funkciófa számozását követi;
- a folyamat általában balról jobbra halad, a nyilak a funkcionális sorrendet jelölik (nem időbeli szünetet vagy közbülső tevékenységet);
- a kör alakú összegzőpontok ÉS, illetve VAGY kapcsolatot jelölnek. Az ÉS párhuzamos funkciókat jelent, amelyeknek mind teljesülniük kell a továbblépéshez; a VAGY alternatív utakat, amelyek közül egy is elegendő;
- az elágazásoknál a továbbhaladási feltételt (például „megfelelt” vagy „nem felelt meg”) a kimenő vonal mellé írjuk.



8.3. ábra. A KoboCella egy ciklusának FFBD-je. Az ÉS-pont a csavarozás és a következő darab előkészítésének párhuzamosságát, a VAGY-pont a nyomatékellenőrzés két kimenetét jelöli.

A 8.3. ábra jól mutatja, mire jó az FFBD a ciklusidő-költségvetés mellett. A ciklusidőt a *soros* útvonal határozza meg. A párhuzamos ágon futó funkciók (itt a következő darab előkészítése) nem növelik, amíg rövidebbek a velük párhuzamos ágnál. Ugyanez az ábra mutatja meg azt is, hogy a hibás darab kiválogatása külön útvonal. Ha ez gyakran előfordul, a ciklusidő eloszlása kétpúpú lesz, amit a TPM-követésnél (6.4. szakasz) figyelembe kell venni.

8.4.2. IDEF0

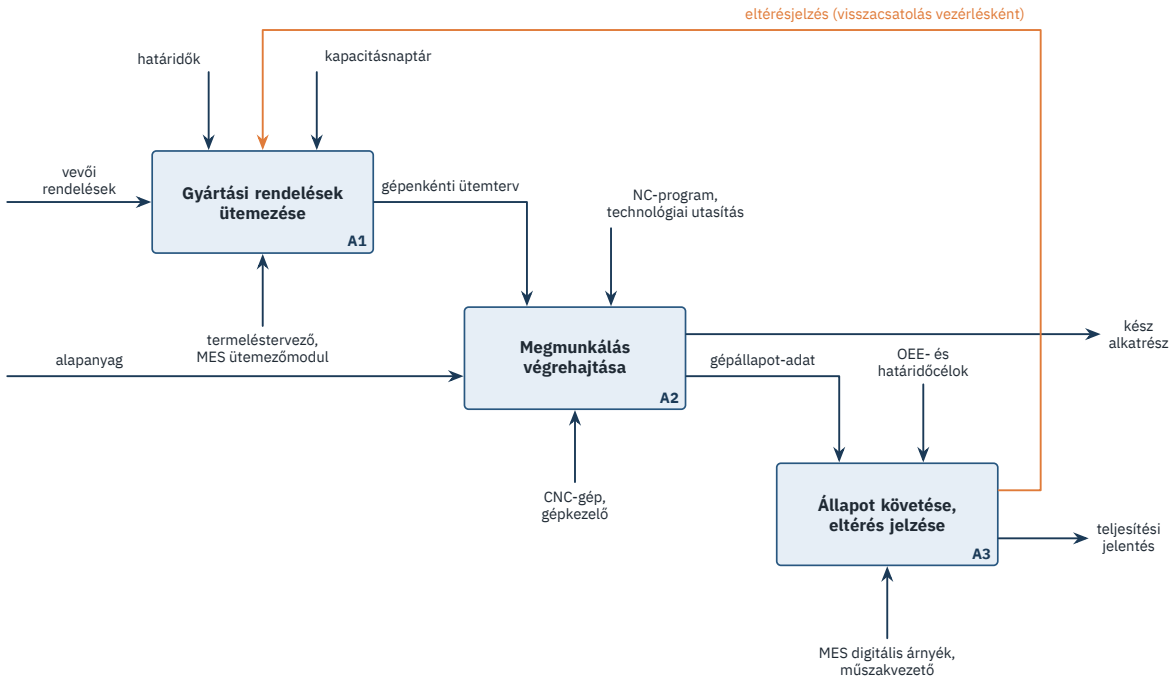
Az FFBD a funkciók sorrendjét mutatja, de nem mutatja meg, milyen információ vagy anyag áramlik közöttük, mi szabályozza őket, és mi hajtja végre őket. Erre szolgál az **IDEF0** (integration definition for function modeling) módszer, amelyet az amerikai szabványügyi intézet FIPS 183 szabványa rögzít (NIST 1993). Az IDEF0-ban minden funkció egy doboz, és a dobozhoz csatlakozó nyíl helye jelöli a nyíl szerepét (Defense Acquisition University 2001, 5–B. kiegészítés):

- **bemenet** (input, balról): az az anyag vagy adat, amelyet a funkció átalakít;
- **vezérlés** (control, felülről): az a szabály, előírás vagy információ, amely a funkció végrehajtását szabályozza;
- **kimenet** (output, jobbra): a funkció eredménye;
- **mechanizmus** (mechanism, alulról): az az erőforrás (ember, gép, szoftver), amely a funkciót végrehajtja.

A négy szerep angol kezdőbetűiből született az ICOM rövidítés. Az IDEF0-modell hierarchikus: a legfelső szintű, egyetlen dobozból álló kontextusábrát (A-0) gyermekábrák bontják ki, amelyeken a dobozok jellemzően lépcsőzetesen, balról fentről jobbra lefelé helyezkednek el. Az IDEF0 különösen alkalmas információs és üzleti folyamatok, valamint gyártási folyamatok elemzésére, ahol a vezérlés és a mechanizmus megkülönböztetése lényeges.

ForgácsTech: az ütemezés IDEF0-ábrája

A ForgácsTech csapata a késések okainak elemzéséhez IDEF0-ábrát készített a rendelésteljesítés magjáról (8.4. ábra). Az ábra három funkciót tartalmaz: a gyártási rendelések ütemezését, a megmunkálás végrehajtását és az állapot követését. A legfontosabb felismerés a visszacsatoló nyílhoz kötődött. A jelenlegi működésben az eltérésjelzés (például egy gép váratlan leállása) a műszakvezető fejében és szóbeli közlésekben létezett, és csak a következő heti ütemezésnél jutott vissza az A1 funkcióhoz. Az ábrán ez a vezérlőnyíl hiányzott. A MES digitális árnyéka (mechanizmus az A3 funkcióban) azt teszi lehetővé, hogy az eltérésjelzés vezérlésként, órákon belül visszajusson az ütemezéshez. Ez a felismerés lett a ForgácsTech-koncepció egyik központi eleme.



8.4. ábra. A ForgácsTech rendelésteljesítésének IDEF0-ábrája. A narancs visszacsatolás az eltérésjelzés, amely vezérlésként tér vissza az ütemezéshez.

8.5. Allokáció: a funkciótól a formáig

A funkcionális elemzés eredménye a funkcionális architektúra. A fizikai architektúrához az **allokáció** vezet el: annak eldöntése, hogy melyik funkciót melyik formai elem (alrendszer, komponens, ember vagy szoftver) végzi el. Kossiakoff és szerzőtársai szerint a rendszer építőelemekre bontását funkcionális allokációnak nevezik, és ez a rendszermérnökség egyik alapvető eszköze (Kossiakoff és mtsai. 2011, 1.2. szakasz).

Az allokáció eredménye legegyszerűbben egy **allokációs táblázatban** rögzíthető, amelynek soraiban a funkciók, oszlopaiban a formai elemek állnak (8.1. táblázat). A táblázat két dolgot azonnal megmutat. Ha egy funkciónak nincs gazdája, a funkció elveszett. Ha egy funkciónak több gazdája van, azok között interfész keletkezik, amelyet meg kell tervezni.

8.1. táblázat. A KoboCella funkcióinak allokációja a forma elemeihez. A több elemhez rendelt funkciók interfészeket hoznak létre.

Funkció	Kobot, megfogó	Csavarozó egység	Asztal, adagolás	Biztonsági alrendszer	Cellavezérlés, HMI	Operátor
1.1 motorház befogása			×			×
1.2 fedél pozicionálása	×					
2.1 csavar adagolása			×			
2.2 csavar meghúzása	×	×				
2.3 nyomaték mérése		×				
3.1 adatok naplózása					×	
3.2 hibás darab jelölése					×	×
3.3 adatátadás a MES-nek					×	
4.1 védőzóna figyelése				×		
4.2 mozgás korlátozása	×			×		
4.3 vészleállítás				×		×
5.1–5.2 kivétel a kiadóba	×					

8.5.1. Moduláris és integrált architektúra

Ulrich klasszikus tanulmánya szerint a termékarchitektúra az a séma, amely szerint a termék funkciója a fizikai komponensekhez rendelődik. Pontosabban három dolgot foglal magában: a funkcionális elemek elrendezését, a funkcionális elemek fizikai komponensekre való leképezését és a kölcsönhatásban álló komponensek közötti interfészek specifikációját (Ulrich 1995). Ulrich két alaptípust különböztet meg:

- **Moduláris architektúra:** a funkcionális elemek és a fizikai komponensek között egy-egy megfeleltetés van, és a komponensek közötti interfészek lazán csatoltak. Egy komponens megváltoztatása nem igényli a többi megváltoztatását.
- **Integrált architektúra:** a leképezés összetett (egy komponens több funkciót lát el, vagy egy funkció több komponensen oszlik meg), és/vagy az interfészek szorosan csatoltak (Ulrich 1995).

Egyik típus sem eleve jobb a másiknál. A moduláris architektúra a változatosságot, a cserélhetőséget, a párhuzamos fejlesztést és a karbantarthatóságot támogatja. Az integrált architektúra egy-egy teljesítményjellemzőt (tömeg, méret, költség nagy darabszámnál) jobban optimalizálhat, mert nem kell a szabványos interfészek „árát” megfizetni. A tervezőnek azt kell eldöntenie, hogy a rendszer életciklusában melyik tulajdonság a fontosabb.

A 8.1. táblázatban a KoboCella vegyes képet mutat. A csavarozó egység és a biztonsági alrendszer jó modulok: néhány, jól körülhatárolt funkciót látnak el. A kobot viszont négy főfunkcióban is részt vesz (pozicionálás, meghúzás, mozgáskorlátozás, kivétel). Ez a kobotos koncepció lényegéből fakad, de azt is jelenti, hogy a kobot programjának bármely változtatása több funkciót érinthet, és ezért a regressziós tesztelés (23. fejezet) itt különösen fontos.

8.5.2. Levezetett követelmények

Az allokáció új követelményeket szül. Ezek a levezetett követelmények a kiválasztott architektúrából fakadnak, ezért nem szerepelhettek a folyamat bemenetét adó követelmények között (6.5. szakasz). Az eredeti és a levezetett követelményeket egyaránt allokálni kell az architektúra elemeihez és a funkciókhoz.

> **KoboCella: követelmények az allokációból**
>
> A „mozgás korlátozása” funkció a kobothoz és a biztonsági alrendszerhez is hozzá van rendelve. A rendszerszintű R-11-es követelmény szerint a kobot sebessége az együttműködési zónában legfeljebb 250 mm/s lehet (6. fejezet). Az allokációból két levezetett követelmény született. A biztonsági alrendszernek a zónába lépést meghatározott reakcióidőn belül jeleznie kell a kobot vezérlőjének, a kobot vezérlőjének pedig a jelzés vételét követően meghatározott időn belül a csökkentett sebességre kell váltania. A két idő összegének a kockázatértékelésben meghatározott határ alatt kell maradnia. Egyik követelmény sem szerepelt az érdekelt felek elvárásai között: mindkettő az architektúra következménye. Ugyanez a felosztás egy interfészt is létrehozott a biztonsági alrendszer és a kobot vezérlője között, amelyet a következő szakasz N²-diagramja is mutat.

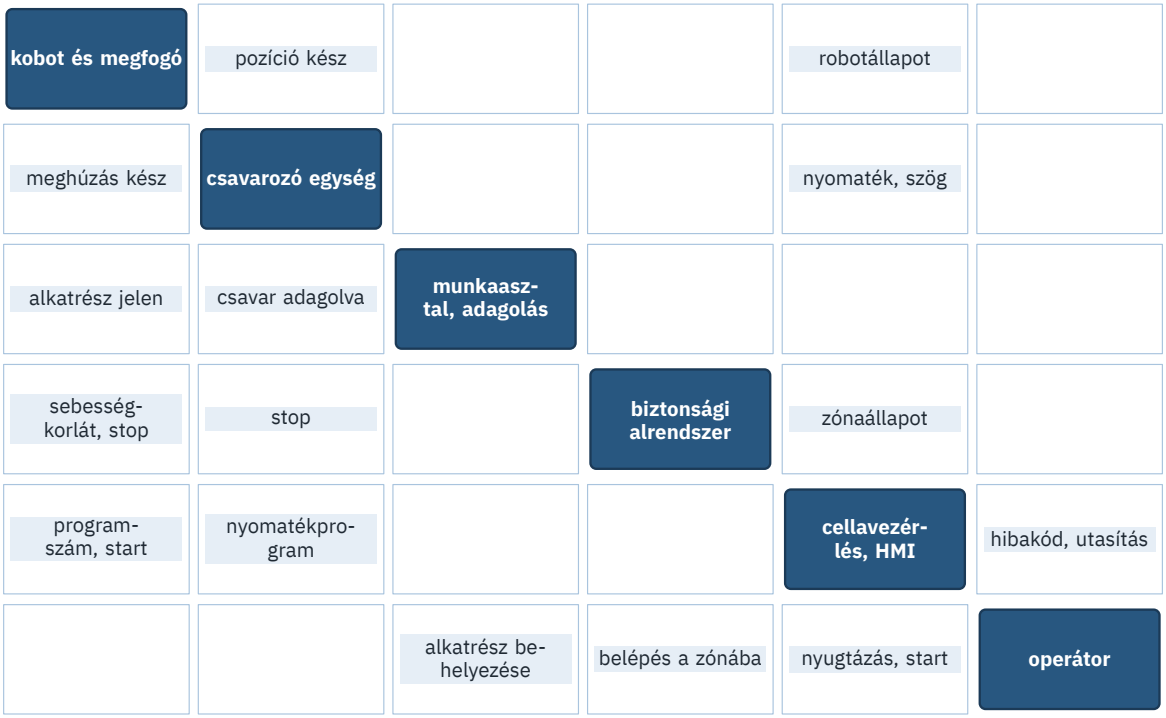
8.6. Interfészek: N²-diagram és interfészkezelés

Kossiakoff és szerzőtársai szerint a rendszertervezés egyik legfontosabb lépése az építőelemek funkcionális és fizikai összekapcsolásának meghatározása. Ehhez korán azonosítani kell az összes jelentős funkcionális kölcsönhatást, és azt, hogyan lehet a funkcionális elemeket úgy csoportosítani, hogy az erősen kölcsönható elemek egy csoportba kerüljenek, a csoportok közötti kölcsönhatások pedig a lehető legegyszerűbbek legyenek (Kossiakoff és mtsai. 2011, 4.4. szakasz). A külső interfészek azonosításának eszköze a kontextusdiagram (3.3. szakasz). A belső interfészeké az N²-diagram.

8.6.1. Az N²-diagram

Az **N²-diagramot** (N-négyzet diagram) Lano dolgozta ki az 1970-es években a TRW vállalatnál, eredetileg szoftverrendszerek interfészeinek elemzésére (Lano 1977). Nevét onnan kapta, hogy N elem esetén N × N cellából áll. Felépítése:

- az **átlóban** az alrendszerek (vagy funkciók) állnak;
- az átlón kívüli cellák az interfészeket jelölik. A szokásos olvasat szerint az *i*-edik sorban álló cella az *i*-edik elem *kimenetét* tartalmazza, amely a cella oszlopában álló elem *bemenete* lesz. A kimenetek tehát vízszintesen, a bemenetek függőlegesen haladnak, az óramutató járásával egyező irányban;
- az üres cella azt jelenti, hogy a két elem között nincs közvetlen interfész.



Az átlón az alrendszerek; a sorokban az alrendszer kimenetei, az oszlopokban a bemenetei (az óramutató járásával egyező olvasat). Az üres cella: nincs közvetlen interfész. Az operátor sora és oszlopa az ember–gép interfészeket mutatja.

8.5. ábra. A KoboCella N²-diagramja hat alrendszerrel. A sorokban az alrendszerek kimenetei, az oszlopokban a bemenetei szerepelnek.

Az N²-diagram (8.5. ábra) gyorsan megmutatja az interfészek számát és eloszlását. A sűrű sorok és oszlopok azokat az elemeket jelölik, amelyek sok más elemmel kommunikálnak: ezek az integráció kritikus pontjai. A KoboCella diagramján a cellavezérlés sora és oszlopa a legsűrűbb, ami egy központi vezérlésű architektúránál várható. Az operátor sora és oszlopa az ember–gép interfészeket mutatja. Ezeket ugyanolyan gondossággal kell megtervezni, mint a műszaki interfészeket (11. fejezet).

Az N²-diagram a funkciók szintjén is alkalmazható. Ilyenkor az átlóban a funkciók állnak, a cellákban az egymásnak átadott anyag, energia vagy információ. Pahl és szerzőtársai funkcióstruktúrája ugyanezt a gondolatot fejezi ki: a részfunkciókat az anyag-, energia- és jeláramok kapcsolják össze (Pahl és mtsai. 2007). A cellák tartalmának típusba sorolásához a kontextusdiagramnál megismert öt kategória (adat, jel, anyag, energia, tevékenység) itt is használható (3.3. szakasz).

8.6.2. Interfészkezelés

Az interfészek azonosítása csak az első lépés. Az interfészkezelés különösen akkor fontos, ha a munka több csapat vagy beszállító között oszlik meg: ilyenkor az interfész az a pont, ahol a felelősségek találkoznak, és ahol a legtöbb félreértés születik (Kossiakoff és mtsai. 2011, 3.5. szakasz). A külső és a belső interfészeket már a fejlesztés korai szakaszában, a működési koncepció alapján érdemes azonosítani, és meghatározni a forrásukat, a céljukat és az átadott anyag, energia vagy információ jellemzőit. Ahogy az architektúra kirajzolódik, új interfészek jelennek meg, a meglévők pedig változnak.

Az interfészeket dokumentálni kell. A nagy projektekben ennek bevett formája az interfészkövetelmény-dokumentum (IRD), amely azt rögzíti, mit kell az interfésznek tudnia, és az interfészszabályozó dokumentum vagy rajz (ICD), amely a megállapodott műszaki megoldást írja le. Ezek a konfigurációkezelés hatálya alá tartoznak, vagyis csak szabályozott változáskezeléssel módosíthatók. Az interfészek összehangolásáért jellemzően egy **interfész-munkacsoport** felel, amelyben az érintett felek műszaki képviselői ülnek (Defense Acquisition University 2001; NASA 2016, 6.3. szakasz). Az interfészek integráció közbeni verifikációjával a 22. fejezet foglalkozik.

> **A „majd megbeszélik" interfész**
>
> Gyakori hiba, hogy az interfész két oldalát két különböző csapat vagy beszállító fejleszti, és mindkettő azt feltételezi, hogy a részleteket „majd megbeszélik". Az interfész így senkié, és az eltérések csak az integrációnál derülnek ki, amikor a javításuk már drága. Minden interfésznek legyen gazdája, legyen dokumentálva legalább az átadott jel vagy anyag, annak formátuma, tartománya és időzítése, és minden változtatását mindkét fél hagyja jóvá.

8.7. Tervezési struktúramátrix és modularitás

Az N²-diagram rokona a **tervezési struktúramátrix** (design structure matrix, DSM). A DSM-et Steward vezette be 1981-ben, eredetileg a tervezési folyamat lépéseinek sorba rendezésére (Steward 1981). A DSM négyzetes mátrix, amelynek sorai és oszlopai ugyanazokat az elemeket tartalmazzák, egy cella pedig azt jelzi, hogy a sor eleme függ-e az oszlop elemétől (vagy fordítva, a használt konvenciótól függően). Az N²-diagrammal szemben a DSM cellái általában nem szöveges leírást, hanem jelölést vagy számot (a kapcsolat erősségét) tartalmaznak, ezért a DSM algoritmikusan elemezhető.

Eppinger és Browning összefoglaló könyve szerint a DSM négy fő alkalmazási területe a termékarchitektúra (komponensek és interfészeik), a szervezeti architektúra (csapatok és kommunikációjuk), a folyamatarchitektúra (tevékenységek és információáramlásuk), valamint ezek összekapcsolása többtartományos mátrixokban (Eppinger és Browning 2012). Browning korábbi áttekintése a statikus (komponens- és csapatalapú) és az időalapú (tevékenység- és paraméteralapú) DSM-eket különböztette meg: az előbbieket jellemzően **klaszterezéssel**, az utóbbiakat **sorba rendezéssel** elemzik (Browning 2001). 8.2. táblázat összefoglalja a típusokat.

8.2. táblázat. A DSM típusai Browning és Eppinger–Browning alapján.

DSM-típus	Elemek	Kapcsolat	Fő elemzési művelet	Felhasználás
termékarchitektúra (komponensalapú)	alrendszerek, komponensek	fizikai, energia-, anyag- vagy információs interfész	klaszterezés	modulok kialakítása, interfészkezelés
szervezeti (csapatalapú)	csapatok, szakterületek	kommunikációs igény	klaszterezés	csapatszerkezet, koordináció
folyamat (tevékenységalapú)	tervezési tevékenységek	információátadás	sorba rendezés, a visszacsatolások csökkentése	tervezési folyamat ütemezése
paraméteralapú	tervezési paraméterek	paraméterek közötti függés	sorba rendezés	részletes tervezés sorrendje

8.7.1. Modularitás és klaszterezés

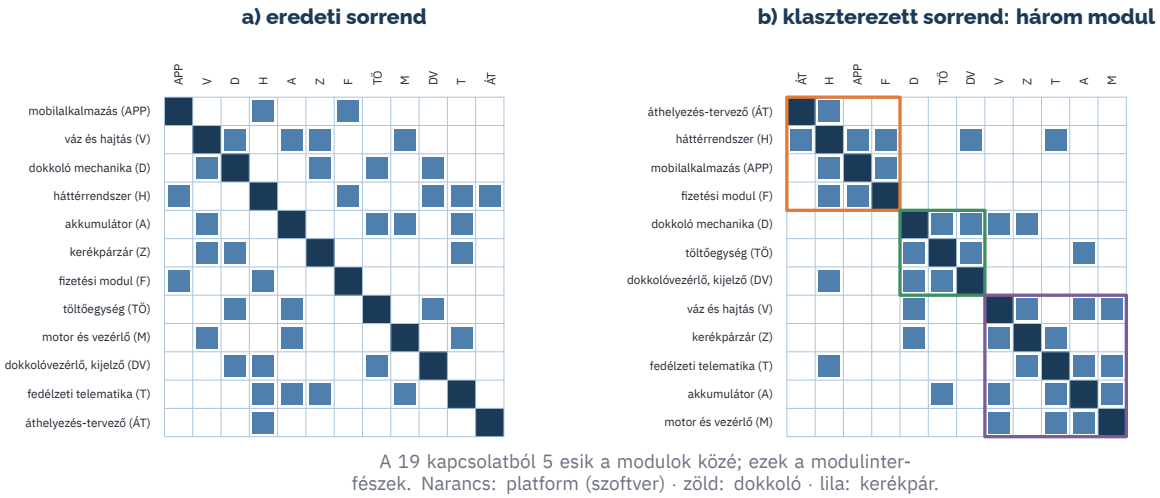
A modularitás gondolata a rendszerelmélet egyik legrégebbi eredményére vezethető vissza. Simon nagy hatású tanulmánya szerint a komplex rendszerek jellemzően hierarchikusak és **majdnem felbonthatók** (nearly decomposable): az alrendszereken belüli kölcsönhatások általában erősebbek, mint az alrendszerek közöttiek (Simon 1962). Kossiakoff és szerzőtársai a modularitást a rendszerkomponensek kölcsönös függetlenségének mértékeként határozzák meg. A rendszermérnökség egyik alapvető célja a nagyfokú modularitás elérése, hogy az interfészek és a kölcsönhatások a lehető legegyszerűbbek legyenek, ami a gyártást, az integrációt, a tesztelést, a karbantartást, a megbízhatóságot és a későbbi továbbfejlesztést is segíti (Kossiakoff és mtsai. 2011, 1.2. szakasz). A modulok tervezésének két mércéje az **összetartás** (cohesion), amely a modulon belüli elemek összetartozását méri, és a **csatolás** (coupling), amely a modulok közötti kölcsönhatásokat méri. A kívánatos a szoros összetartás és a laza csatolás (Kossiakoff és mtsai. 2011, 11. fejezet).

A komponensalapú DSM-ben a modulok kialakítása **klaszterezési** feladat: a sorokat és oszlopokat úgy kell átrendezni, hogy a kapcsolatok az átló körüli blokkokba tömörüljenek, a blokkokon kívül pedig minél kevesebb kapcsolat maradjon. A blokkok a modulok, a blokkokon kívüli kapcsolatok a modulinterfészek. Erre számos algoritmus létezik, a kézi átrendezéstől a hierarchikus klaszterezésen át a hálózattudományi közösségkeresésig. Ez utóbbiak közül Newman modularitási mértéke a legismertebb, amely azt méri, mennyivel

több kapcsolat esik a csoportokon belülre, mint amennyi véletlenszerű kapcsolódás esetén várható lenne (Newman 2006).

VárosiBringa: modulok a DSM-ből

A VárosiBringa csapata 12 rendszerelemet és közöttük 19 kölcsönös kapcsolatot azonosított: a kerékpár elemeit (váz és hajtás, akkumulátor, motor és vezérlő, kerékpárzár, fedélzeti telematika), a dokkolóállomás elemeit (dokkoló mechanika, töltőegység, dokkolóvezérlő és kijelző) és a szoftverplatform elemeit (mobilalkalmazás, háttérrendszer, fizetési modul, áthelyezés-tervező). Az eredeti, felsorolási sorrendben (8.6a. ábra) a kapcsolatok szétszórtan helyezkednek el, és nehéz megmondani, mely elemek tartoznak össze.
A csapat a szomszédsági mintázatok hasonlósága alapján hierarchikus klaszterezést végzett (a módszert a Gyakorlati útmutató DSM-eszközlapja és notebookja részletezi). Az átrendezett mátrixban (8.6b. ábra) három modul rajzolódott ki: a platform (áthelyezés-tervező, háttérrendszer, mobilalkalmazás, fizetési modul), a dokkoló (mechanika, töltőegység, vezérlő) és a kerékpár (váz és hajtás, kerékpárzár, telematika, akkumulátor, motor). A 19 kapcsolatból 14 a modulokon belülre esik, és csak 5 a modulok közé: a zár és a dokkoló mechanika, a váz és a dokkoló mechanika, az akkumulátor és a töltőegység, a telematika és a háttérrendszer, valamint a dokkolóvezérlő és a háttérrendszer között. Ez az öt kapcsolat lett a három beszállítói csomag közötti interfészszerződések tárgya.



8.6. ábra. A VárosiBringa DSM-je eredeti (a) és klaszterezett (b) sorrendben. A keretezett blokkok a modulok, a blokkokon kívüli cellák a modulinterfészek.

8.7.2. Tervezési szabályok és a szervezet tükre

Baldwin és Clark a modularitás gazdasági és szervezeti következményeit vizsgálták. Szerintük a moduláris tervezés kulcsa, hogy a tervezési paramétereket két csoportra osztjuk. A **látható tervezési szabályok** (az architektúra, az interfészek és a megfelelőségi tesztek) mindenkire kötelezőek, és korán rögzíteni kell őket. A **rejtett paramétereket** a modulok fejlesztői a szabályok keretein belül szabadon alakíthatják. Ez teszi lehetővé a modulok párhuzamos és egymástól független fejlesztését, és azt, hogy egy modul jobb változata a rendszer többi részének módosítása nélkül beépíthető legyen (Baldwin és Clark 2000).

Az architektúra és a szervezet összefügg. Conway már 1968-ban megfogalmazta azt a megfigyelést, amelyet ma Conway-törvényként ismerünk: a rendszereket tervező szervezetek olyan terveket hoznak létre, amelyek a saját kommunikációs szerkezetüket másolják

(Conway 1968). Sosa, Eppinger és Rowles egy nagy repülőgép-hajtómű fejlesztésében vetették össze a tervezési interfészeket a csapatok közötti kommunikációval. Két esetet vizsgáltak: amikor egy ismert tervezési interfészhez nem tartozik csapatkommunikáció, és amikor egy megfigyelt kommunikáció nem magyarázható tervezési interfésszel. Eredményeik szerint a kettő illeszkedését a szervezeti és rendszerhatárok, az interfészek erőssége, a közvetett kapcsolatok és a rendszer modularitása befolyásolja (Sosa, Eppinger, és Rowles 2004). A gyakorlati tanulság az, hogy a termék-DSM és a szervezeti DSM összevetése megmutatja azokat az interfészeket, amelyekért senki nem felel. Ezekhez kell interfészgazdát rendelni.

> **ForgácsTech: architektúra és szervezet**
>
> A ForgácsTech MES-bevezetésénél a csapat a koncepció szakaszában két architektúrát vetett össze. Az elsőben egyetlen beszállító szállítja a MES-t a gépi adatgyűjtéssel és az ütemező-modullal együtt. A másodikban a gépi adatgyűjtést a gépgyártók saját megoldásai végzik, a MES pedig csak fogadja az adatokat. A második változat olcsóbbnak tűnt, de az N²-diagram megmutatta, hogy a gépi adatgyűjtés és a MES között hat különböző interfész jönne létre (gépenként eltérő protokollal), és ezek gazdája egyik szerződésben sem lett volna egyértelmű. A csapat ezért az első változat felé hajlott, a végső döntést azonban az 10. fejezetben bemutatott értékeléssel hozta meg.

8.8. Mitől jó egy architektúra?

Az architektúra minőségét végső soron az dönti el, mennyire teljesíti a követelményeket és az érdekelt felek elvárásait a teljes életciklus során. A koncepció szakaszában azonban, amikor még nincs mit mérni, a fejezetben tárgyalt eszközökből több jellemző is kiolvasható:

- **Teljesség:** minden funkciónak van gazdája az allokációs táblázatban, és minden követelmény legalább egy funkcióhoz kötődik.
- **Áttekinthetőség:** szintenként kezelhető számú elem, egyértelmű funkciónevek, következetes számozás.
- **Modularitás:** szoros összetartás a modulokon belül és laza csatolás a modulok között. A DSM-ben a kapcsolatok nagy része a blokkokon belülre esik.
- **Egyszerű interfészek:** kevés, jól definiált, lehetőleg egyirányú interfész. Kossiakoff és szerzőtársai szerint a moduláris architektúrák a karbantartható és továbbfejleszthető rendszerek kulcsai (Kossiakoff és mtsai. 2011, 4.4. szakasz).
- **Tesztelhetőség:** a modulok külön is verifikálhatók, és az integráció lépésenként, modulonként végezhető.
- **Illeszkedés a szervezethez:** minden modulinterfésznek van felelőse, és a szervezeti felosztás nem vág ketté erősen csatolt elemeket.
- **Tartalék és bővíthetőség:** a kritikus erőforrásokon (ciklusidő, számítási kapacitás, energia, hely) van tartalék, és a várható bővítések nem igényelik az architektúra átalakítását.

Ezek a jellemzők gyakran ellentmondanak egymásnak és a teljesítménykövetelményeknek. A modulárisabb megoldás általában drágább vagy nehezebb, a legkisebb költségű megoldás pedig gyakran integrált. Ezért az architektúra-tervezés nem egyetlen helyes válasz megkeresése, hanem alternatívák létrehozása (9. fejezet) és tudatos kiválasztása (10. fejezet).

Modellalapú és adatvezérelt architektúra-tervezés

A modellalapú rendszermérnökségben (3.5. szakasz) az allokáció nem külön táblázatban, hanem a rendszermodellben él: a SysML nyelvben a funkciókat leíró viselkedési elemek és a formát leíró szerkezeti elemek közötti allokációs kapcsolatok formájában. Ennek előnye, hogy a funkciófa, az allokációs táblázat, az N²-diagram és a DSM ugyanabból a modellből generálható, így nem kerülhetnek ellentmondásba egymással. A DSM-ek elemzésében a hálózattudomány eszközei (közösségkeresés, centralitás, a változások terjedésének modellezése) egyre nagyobb szerepet kapnak. Segítségükkel nagy, több száz elemű rendszereknél is azonosíthatók a modulok és azok az elemek, amelyeken keresztül egy változás a legtöbb más elemre átterjedhet.

Összefoglalás

- A rendszer három nézőpontból írható le: a funkció az, amit csinál (folyamat és operandus), a forma az, ami van (elemek és szerkezet), a koncepció pedig a kettő közötti leképezés. Az architektúra a koncepció megtestesülése: a funkciók hozzárendelése a forma elemeihez és az elemek kapcsolatainak meghatározása.
- A funkciókat megoldássemlegesen, igei szerkezettel kell megfogalmazni, különben a koncepciót a funkció felírásakor már eldöntöttük.
- Az architektúra-meghatározás a követelményekből funkcionális elemzéssel hozza létre az architektúrát és a levezetett követelményeket. A munka kreatív, rekurzív és iteratív, és mindig több architektúraváltozatot kell kidolgozni.
- A funkciófa a funkciók hierarchiáját, az FFBD a sorrendjüket és párhuzamosságukat, az IDEF0 a bemeneteiket, vezérléseiket, kimeneteiket és mechanizmusaikat mutatja.
- Az allokáció a funkciókat a forma elemeihez rendeli. Ulrich szerint moduláris az architektúra, ha a leképezés egy-egy megfeleltetés és az interfészek lazán csatoltak; integrált, ha a leképezés összetett vagy az interfészek szorosan csatoltak. Az allokációból levezetett követelmények születnek.
- Az N²-diagram a belső interfészeket azonosítja (kimenetek a sorokban, bemenetek az oszlopokban). Az interfészeket dokumentálni és konfigurációkezelés alatt tartani kell, és mindegyiknek gazdája kell legyen.
- A DSM algoritmikusan elemezhető. A komponensalapú DSM klaszterezése modulokat ad. A jó architektúra szoros összetartású és lazán csatolt, és a szerkezete illeszkedik a fejlesztő szervezethez.

Önellenőrzés

1. Fogalmazd meg megoldássemlegesen a következő funkciókat: (a) „kerékpárzár bezárása mobilalkalmazással”, (b) „OEE-kijelzés a műhelyben”, (c) „robotos alkatrész-kivétel”!
2. Mi a különbség a funkciófa és az FFBD között? Melyik kérdésre válaszol az egyik, és melyikre a másik?
3. Az FFBD-n egy ÉS-pont után két párhuzamos ág fut: az egyik 25 s, a másik 12 s. Mennyivel járul hozzá a két ág a ciklusidőhöz? Mi változik, ha a második ág 30 s-ra nő?
4. Egy IDEF0-dobozba a „karbantartási utasítás” nyíl felülről, a „karbantartó” nyíl alulról érkezik. Mi a szerepük? Mi lenne a hiba, ha a karbantartási utasítást bemenetként rajzolnánk?
5. A 8.1. táblázat alapján melyik alrendszer a legkevésbé moduláris, és miért? Milyen következménye van ennek a tesztelésre?
6. Egy N²-diagramon a 3. sor 5. oszlopában az „üzemállapot” felirat áll. Mit jelent ez?
7. Egy 10 elemű termék-DSM-ben 24 kapcsolat van. A klaszterezés után 18 esik a modulokon belülre. Hány modulinterfész maradt? Mit tennél, ha két modul között 4 kapcsolat is van?

Megoldás.

1. (a) „Kerékpár rögzítése jogosulatlan elvitel ellen” (az azonosítás módja nyitott marad). (b) „Gépek teljesítményének láthatóvá tétele a döntéshozók számára”. (c) „Kész alkatrész

továbbítása a kiadóba”.

2. A funkciófa azt mutatja meg, milyen részfunkciókból áll egy funkció (hierarchia, „miből áll?”). Az FFBD azt, milyen sorrendben, milyen párhuzamosságokkal és elágazásokkal hajtódnak végre a funkciók („mi után mi következik?”).
3. A párhuzamos ágak közül a hosszabb határozza meg a ciklusidőt, ezért a két ág együtt 25 s-mal járul hozzá. Ha a második ág 30 s-ra nő, ez lesz a kritikus ág, és a hozzájárulás 30 s-ra nő, vagyis a ciklusidő 5 s-mal hosszabb lesz.
4. A karbantartási utasítás vezérlés: szabályozza, hogyan kell a funkciót végrehajtani. A karbantartó mechanizmus: ő hajtja végre. Bemenet az, amit a funkció átalakít (például a karbantartandó gép). Ha az utasítást bemenetként rajzolnánk, az ábra azt állítaná, hogy a funkció az utasítást alakítja át, ami félrevezető.
5. A kobot: négy főfunkcióban is szerepel (pozicionálás, meghúzás, mozgáskorlátozás, kivétel), így a leképezés nem egy-egy megfeleltetés. A kobot programjának bármely módosítása után több funkció regressziós tesztelésére van szükség.
6. A 3. sorban álló elem kimenete az „üzemállapot”, amely az 5. oszlopban álló elem bemenete. Vagyis a 3. elem üzemállapot-információt küld az 5. elemnek.
7. 24 − 18 = 6 modulinterfész maradt. Ha két modul között 4 kapcsolat van, érdemes megvizsgálni, hogy nem egyetlen modulról van-e szó, vagy hogy egy elem nem rossz modulba került-e. Ha az összevonás nem lehetséges, ezt a modulpárt közös interfészgazdához kell rendelni, és az interfészeket kiemelten kell kezelni.

Projektlépés: az architektúra első változata

Saját fejlesztési feladatodhoz:

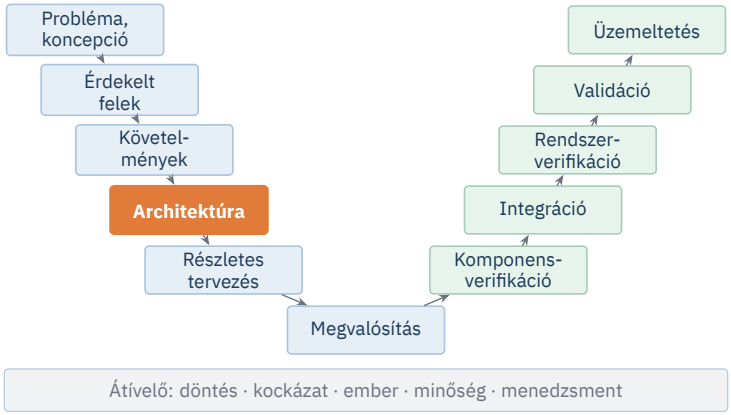
1. fogalmazd meg a rendszer fő funkcióját megoldássemlegesen, és készíts legalább kétszintű funkciófát, szintenként legfeljebb 7–9 elemmel;
2. rajzold fel a fő működési forgatókönyv FFBD-jét ÉS- és VAGY-pontokkal;
3. írj le legalább két koncepciót, és az egyikhez készíts allokációs táblázatot;
4. készíts N²-diagramot az alrendszerekről, és jelöld meg a három legkritikusabb interfészt;
5. ha legalább 8 elemed van, készíts DSM-et, és klaszterezéssel javasolj modulokat;
6. írd le, milyen levezetett követelmények születtek az allokációból.

A funkciófa és a DSM sablonja, valamint a notebookok a Gyakorlati útmutató funkciófa- és N²/DSM-eszközlapján találhatók.

Továbbolvasás

- Crawley, Cameron, és Selva (2016): a funkció–forma–koncepció felfogás és a rendszerarchitektúra alapvető tankönyve.
- Maier és Rechtin (2009): a rendszerarchitektúra-tervezés heurisztikái és gyakorlata.
- Ulrich (1995): a termékarchitektúra és a moduláris–integrált megkülönböztetés klasszikus tanulmánya.
- Eppinger és Browning (2012): a DSM-módszerek kézikönyve sok ipari példával.
- Simon (1962): a hierarchikus, majdnem felbontható rendszerek elmélete.
- Baldwin és Clark (2000): a modularitás gazdasági és szervezeti következményei.
- NASA (2016), 4.3–4.4. és 6.3. szakasz: a logikai dekompozíció, a tervezési megoldás meghatározása és az interfészkezelés folyamata.

9. Koncepciógenerálás és kreatív problémamegoldás



Tanulási eredmények

A fejezet feldolgozása után képes leszel:

- megkülönböztetni a divergens és a konvergens gondolkodást, és megmagyarázni, miért kell őket időben szétválasztani;
- felismerni a kreativitás tipikus gátjait, köztük a tervezési fixációt, és tudatosan ellenük dolgozni;
- megtervezni és levezetni egy ötletrohamot, egy 6-3-5 ötletírást, egy névleges csoportos ülést és egy Delphi-felmérést, és indokolni, mikor melyiket érdemes választani;
- morfológiai dobozzal rendszerezetten koncepciókat előállítani, és kezelni a kombinatorikus robbanást;
- a TRIZ segítségével megfogalmazni egy műszaki vagy fizikai ellentmondást, és feltalálói elvekkel megoldási irányokat keresni;
- alkalmazni a hat gondolkodó kalap módszerét, a természet ihlette tervezést és a számítógépes koncepciókeresést, és értékelni egy ötletgeneráló folyamat eredményességét.

Az előző fejezet azzal zárult, hogy az architektúra-tervezés nem egyetlen helyes válasz megkeresése, hanem alternatívák létrehozása és tudatos kiválasztása. Mindig több architektúraváltozatot kell kidolgozni, hogy legyen miből választani (8.2. szakasz). Honnan jönnek ezek a koncepciók?

Erre senki nem tud pontos választ adni, és a kreativitás nem is gépiesíthető teljesen. Pasteur sokat idézett mondása szerint azonban a megfigyelésben a véletlen csak a felkészült elmének kedvez, vagyis a kreativitás feltételei megteremthetők: vannak módszerek, amelyek segítenek több, változatosabb és jobb ötletet előállítani, és vannak jól ismert csapdák, amelyek elkerülhetők. Ez a fejezet ezekkel a módszerekkel és csapdákkal foglalkozik.

A koncepcióalkotás két lépésből áll:

- **koncepciógenerálás:** olyan megoldásváltozatok előállítása, amelyek képesek teljesíteni a funkciót és a követelményeket;
- **koncepció-kiválasztás:** a változatok közül annak kiválasztása, amely a követelmények teljesítése mellett a legnagyobb értéket adja, és összhangban van a jelenlegi és a várható szabályozással és szabványokkal.

Ez a fejezet az elsővel, a következő (10. fejezet) a másodikkal foglalkozik. A fejezet négy kérdésre épül:

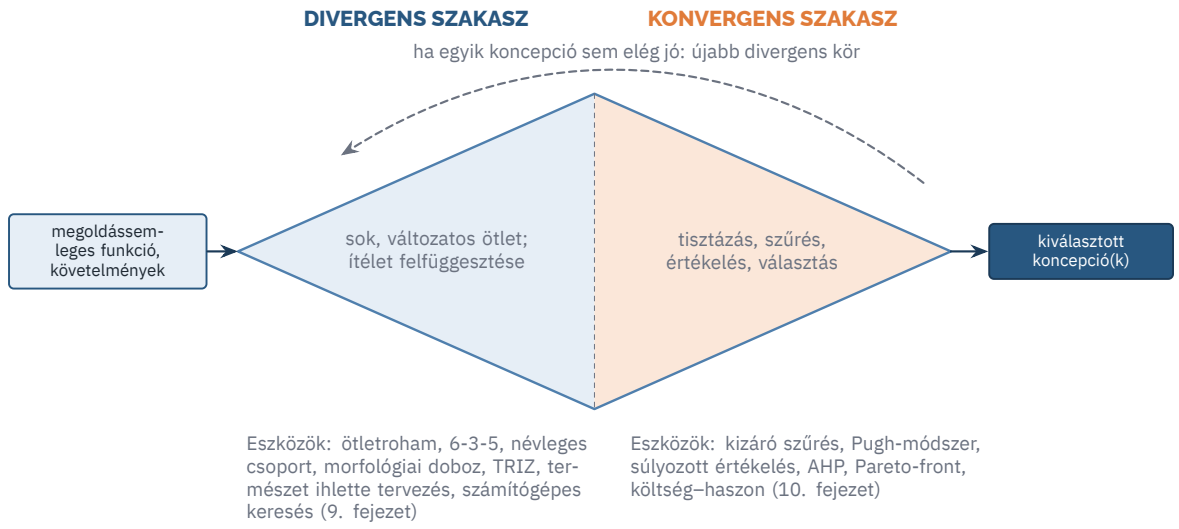
- Hogyan működik a kreatív folyamat, és mi akadályozza?
- Hogyan lehet egy csoport ötleteit a lehető legjobban kiaknázni?
- Hogyan állíthatók elő rendszerezetten koncepciók?
- Milyen segítséget adnak a természet, az algoritmusok és a nyelvi modellek?

9.1. A kreatív folyamat

9.1.1. Divergens és konvergens gondolkodás

A kreativitás rendszeres pszichológiai kutatását Guilford 1950-es elnöki beszéde indította el az Amerikai Pszichológiai Társaságban (Guilford 1950). Guilford intellektusmodelljéből ered a ma is használt megkülönböztetés: a **divergens gondolkodás** egy kiindulópontból sokféle lehetséges választ állít elő, a **konvergens gondolkodás** pedig a lehetőségek közül a legjobbat választja ki. A tervezésben mindkettőre szükség van, de nem egyszerre. Ha egy ötletet azonnal értékelünk, a kritika elfojtja a további ötleteket, és a csoport a legelső elfogadható megoldásnál megáll.

A koncepcióalkotás ezért két jól elkülönülő szakaszból áll (9.1. ábra). A divergens szakaszban az ítéletet felfüggesztjük, és a mennyiségre, a változatosságra törekszünk. A konvergens szakaszban tisztázzuk, szűrjük és értékeljük az ötleteket, majd kiválasztunk egyet vagy néhányat. Ez a szerkezet a problématér feltárásánál már megismert dupla gyémánt második felének felel meg (4.1. szakasz).



9.1. ábra. A koncepcióalkotás divergens és konvergens szakasza. A két szakaszt időben szét kell választani; ha egyik koncepció sem elég jó, újabb divergens kör következik.

9.1.2. A kreatív folyamat szakaszai

Wallas klasszikus modellje szerint a kreatív gondolkodás négy szakaszon megy át: az előkészítésen (a probléma alapos megismerése), a lappangáson (a tudatos foglalkozás szüneteltetése), a megvilágosodáson (az ötlet megjelenése) és az igazoláson (az ötlet tudatos kidolgozása és ellenőrzése) (Wallas 1926). A modell két gyakorlati tanulsággal jár. Az első, hogy az előkészítés nem spórolható meg: a jó ötlet a probléma mély megértéséből születik, ezért kell a koncepcióalkotást az érdekelt felek elemzésére és a követelményekre építeni. A második, hogy a lappangásnak idő kell. Egy kétórás ötletgyűjtő ülés helyett gyakran eredményesebb két rövidebb ülés, köztük néhány nappal.

9.2. A kreativitás gátjai

Egy ismert fejtörő jól szemlélteti a gondolkodási gátakat: kössünk össze kilenc, három sorban elhelyezett pontot négy egyenes vonallal, a ceruza felemelése nélkül. A feladat azért nehéz, mert hallgatólagosan feltételezzük, hogy a vonalak nem léphetnek ki a pontok alkotta négyzetből, pedig ezt senki nem kérte. A tervezésben ugyanilyen ki nem mondott feltevések korlátozzák a megoldásteret.

A legjobban dokumentált gát a **tervezési fixáció** (design fixation). Jansson és Smith kísérleteiben a tervezőknek egy feladat mellé egy példamegoldást is mutattak. A példát látó tervezők megoldásai a példa jellemzőit szignifikánsan gyakrabban tartalmazták, mint a kontrollcsoporté, és ez akkor is így volt, amikor a példa jellemzői kifejezetten hibásak voltak, sőt a feladatleírás figyelmeztetett is a hibákra (Jansson és Smith 1991). A fixáció a tapasztalt mérnököket is érinti, sőt a szakmai tapasztalat a meglévő megoldások erős mintáit is magával hozza.

A tervezési fixáció és a többi gát ellen a következő gyakorlatok segítenek:

- **Megoldássemleges funkció.** A funkciót a megvalósítás említése nélkül fogalmazzuk meg (8.1. szakasz). Amíg a funkció „robotos csavarozás”, nem merül fel más megoldás.
- **Előbb ötletek, aztán példák.** A meglévő megoldásokat (versenytársak, korábbi projektek) csak az első önálló ötletgyűjtés *után* mutassuk meg a csapatnak.
- **A feltevések kimondása.** Írjuk le, mit tételezünk fel hallgatólagosan („a csavarkötés kötelező”, „az operátor áll”), és vizsgáljuk meg, mi történne, ha a feltevés nem igaz.
- **Távoli analógiák.** Keressünk megoldást más szakterületeken és a természetben (9.7. szakasz).
- **Rendszerezett módszerek.** A morfológiai doboz és a TRIZ éppen azért értékes, mert olyan kombinációkat és elveket is elénk tesz, amelyek maguktól nem jutnának eszünkbe.

> **Az első jó ötlet**
>
> A leggyakoribb hiba a koncepcióalkotásban az, hogy a csapat az első elfogadhatónak tűnő ötletnél megáll, és a fennmaradó időt annak kidolgozására fordítja. Az így született koncepció lehet jó, de soha nem derül ki, volt-e jobb. Ezért kell mindig több architektúraváltozatot kidolgozni (8.2. szakasz). Egy egyszerű szabály: a konvergens szakasz csak akkor kezdődhet, ha legalább három, egymástól lényegesen különböző koncepció van az asztalon.

9.3. Csoportos ötletgenerálás

9.3.1. Ötletroham

Az **ötletroham** (brainstorming) Osborn reklámszakember nevéhez fűződik, aki az 1950-es években négy alapszabályt fogalmazott meg: az ítélet felfüggesztése (az ötleteket az ülés alatt nem kritizáljuk), a mennyiségre törekvés, a szokatlan ötletek bátorítása és az ötletek kombinálása, továbbfejlesztése (Osborn 1953). Az ülésen a résztvevők kritikától mentesen, gyorsan állítanak elő ötleteket egy adott problémára. A hangsúly az ötletek mennyiségén van, nem a minőségén, és a csoport minden tagja egyenrangú. A gyakorlatban az ülés három szakaszra tagolódik (Goldberg és mtsai. 1994, 7.7. szakasz):

1. **generálás:** a csoport tagjai ötletlistát állítanak elő;
2. **tisztázás:** a csoport átnézi a listát, hogy mindenki értse az egyes ötleteket;
3. **értékelés:** a csoport kiszűri az ismétlődéseket, a nem odaillő és a kizárt ötleteket.

Fontos ülésekhez érdemes olyan moderátort felkérni, aki ismeri a módszert, segíti az ötletek előállítását, de az eredményben nem érdekelt. Az ötletroham csak javasol megoldást, de nem dönt; a döntéshez a 10. fejezet módszerei kellenek.

Az ötletroham a legismertebb, de nem a leghatékonyabb ötletgeneráló módszer. Diehl és Stroebe kísérletei szerint a szóban, együtt ötletelő csoportok kevesebb ötletet állítanak elő, mint ugyanannyi, egymástól elkülönülten dolgozó ember (úgynevezett névleges csoport). A termelékenységi veszteség fő oka a **produkciós blokkolás**: egyszerre csak egy ember beszélhet, a többiek eközben elfelejtik vagy visszatartják az ötletüket (Diehl és Stroebe 1987). Mullen és szerzőtársai metaanalízise megerősítette, hogy az együtt ötletelő csoportok termelékenységi vesztesége jelentős, és nagyobb csoportokban nagyobb (Mullen, Johnson, és Salas 1991). Ezért alakultak ki az írásos és a csendes változatok.

9.3.2. Ötletírás: a 6-3-5 módszer

A **6-3-5 módszert** Rohrbach írta le (Rohrbach 1969). Hat résztvevő mindegyike egy lapra három ötletet ír öt perc alatt, majd a lapot továbbadja a szomszédjának. A következő körben mindenki a kapott lapon szereplő ötletekre építve ír újabb hármat. Hat kör után, harminc perc alatt legfeljebb $6 \times 3 \times 6 = 108$ ötlet születik. A módszer kiküszöböli a produkciós blokkolást (mindenki egyszerre ír), és a domináns személyiségek hatását is csökkenti, miközben megtartja az egymás ötleteire építés előnyét.

9.3.3. Névleges csoporttechnika

A **névleges csoporttechnikát** (nominal group technique, NGT) Delbecq és Van de Ven dolgozta ki a problémák azonosítására és a programtervezésre (Delbecq és Van de Ven 1971). Az NGT a csendes, egyéni ötletgyűjtést szavazással és rangsorolással ötvöző, strukturált csoportos módszer, ezért akkor hasznos, ha az ötletek mellett prioritást vagy rangsort is fel kell állítani. Kis csoportban működik jól, és minden tagnak részt kell vennie. Az eljárás lépései (Delbecq és Van de Ven 1971; Goldberg és mtsai. 1994, 7.10. szakasz):

1. a moderátor ismerteti a problémát és a szabályokat;
2. a résztvevők 5–15 percig csendben, egyénileg írják le ötleteiket;
3. a moderátor körbejárva, egyesével összegyűjti és felírja az ötleteket;

4. a csoport megbeszéli az ötleteket, de csak tisztázó kérdésekkel, vita nélkül; az ismétlődéseket összevonják;
5. mindenki csendben rangsorolja vagy pontozza az ötleteket;
6. a moderátor összesíti az egyéni rangsorokat.

9.3.4. Delphi-módszer

A **Delphi-módszert** a RAND Corporation kutatói, Dalkey és Helmer dolgozták ki szakértői becslések rendszerezett összegyűjtésére (Dalkey és Helmer 1963). Rowe és Wright áttekintése szerint a módszert négy jellemző határozza meg: a résztvevők névtelensége, a több körös iteráció, a szabályozott visszacsatolás és a csoportvélemény statisztikai összesítése (Rowe és Wright 1999; Goldberg és mtsai. 1994, 7.9. szakasz). A szakértők egymással nem tanácskoznak, hanem véleményüket egy moderátornak küldik, aki összesíti és névtelenül visszaküldi azokat további véleményezésre. A körök addig ismétlődnek, amíg a vélemények nem stabilizálódnak. A névtelenség kiküszöböli a személyes ellentéteket és a domináns személyiségek hatását, és bevonhatók olyan szakértők is, akik egy közös ülésre nem érnének rá. A módszer hátránya, hogy a több kör és a résztvevők összegyűjtése időigényes. A Delphi kevésbé ötletgeneráló, inkább becslő és értékelő módszer: jól használható például egy új technológia érettségének vagy egy koncepció kockázatainak felmérésére.

9.1. táblázat. A csoportos ötletgeneráló és értékelő módszerek összevetése.

Módszer	Fő erőssége	Fő gyengesége	Mikor válaszd?
ötletroham	gyors, egymás ötleteire épít, csapatot épít	produkciós blokkolás, domináns hangok	kis, összeszokott csapat, laza téma
6-3-5 ötletírás	sok ötlet rövid idő alatt, mindenki egyszerre dolgozik	kötött forma, írásban nehezebb a bonyolult ötlet	6 fős csoport, sok ötlet kell gyorsan
névleges csoport	kiegyensúlyozott részvétel, rangsort is ad	erős moderálást igényel	vegyes státuszú csoport, döntés-előkészítés
Delphi	névtelen, a földrajzilag szétszórt szakértőket is bevonja	lassú, több körös	becslés, előrejelzés, szakértői értékelés

ForgácsTech: névleges csoport a késések okairól

A ForgácsTech csapata a késések lehetséges okainak összegyűjtésére először hagyományos ötletrohamot tartott. Az ülésen a termelési vezető és a két legtapasztaltabb műszakvezető beszélt a legtöbbet, a gépkezelők és a beszerző szinte meg sem szólalt. Az ülés tizennégy okot gyűjtött, többségük a gépek állapotához kötődött. A második alkalommal a csapat névleges csoporttechnikát alkalmazott ugyanazzal a tizenegy résztvevővel. Tíz perc csendes írás után 41 ötlet gyűlt össze, az összevonás után 27 különböző ok maradt. A csendes rangsorolás első öt helyére két olyan ok is került (a sürgős rendelések szóbeli „előresorolása” és az alapanyag-beérkezés bizonytalansága), amely az első ülésen fel sem merült. Ez a tapasztalat közvetlenül befolyásolta a MES-koncepciót: az ütemezőmodulnak a sürgősségi átsorolást is naplóznia kell.

9.4. Rendszerezett koncepcióalkotás: a morfológiai doboz

A csoportos módszerek az emberek ötleteit gyűjtik össze. A **morfológiai doboz** (morphological box, morfológiai mátrix) ezzel szemben rendszerezetten állítja elő a lehetséges megoldásokat. A módszert Zwicky svájci asztrofizikus dolgozta ki általános feltaláló és problémamegoldó módszerként (Zwicky 1969). Lényege, hogy a problémát egymástól független részfunkciókra vagy paraméterekre bontjuk, mindegyikhez összegyűjtjük a lehetséges megoldásokat, majd a részmegoldások kombinálásával teljes koncepciókat állítunk elő. Pahl és szerzőtársai rendszeres géptervezési módszertanában a morfológiai doboz a funkcióstruktúrából kiinduló koncepcióalkotás egyik alapeszköze (Pahl és mtsai. 2007).

A morfológiai doboz lépései:

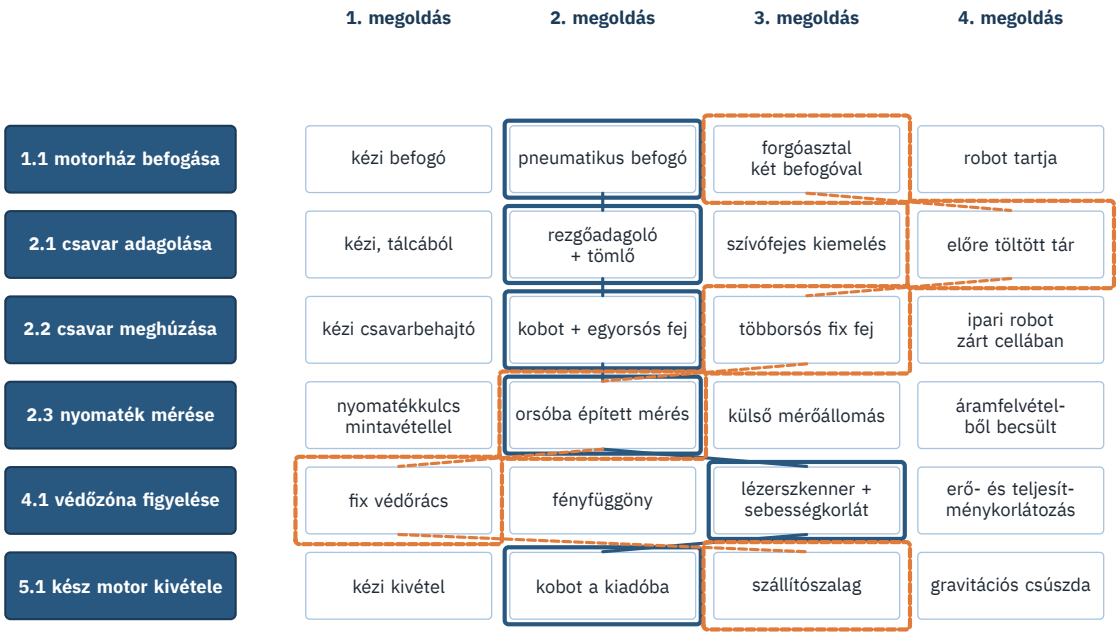
1. a fő funkció lebontása részfunkciókra (a funkciófa második szintje jó kiindulópont, 8.3. szakasz);
2. minden részfunkcióhoz megoldási lehetőségek gyűjtése (ehhez bármelyik csoportos módszer használható);
3. az összeférhetetlen részmegoldás-párok megjelölése;
4. ígéretes kombinációk (koncepciók) összeállítása, mindegyik egy-egy útvonal a dobozon át;
5. a koncepciók rövid leírása és továbbadása a kiválasztáshoz.

A morfológiai doboz legnagyobb erénye és legnagyobb nehézsége ugyanaz: a kombinációk száma. Ha n részfunkció mindegyikéhez m_i megoldás tartozik, az elvi kombinációk száma

$$N = \prod_{i=1}^{n} m_i. \tag{9.1}$$

Hat részfunkció és részfunkciónként négy megoldás esetén ez már $4^6 = 4096$ kombináció. Ennyit kézzel nem lehet értékelni. A kombinatorikus robbanást három lépésben kezeljük: kizárjuk az összeférhetetlen párokat, kizárjuk azokat a részmegoldásokat, amelyek egy kötelező követelményt nem teljesítenek, és a maradékból néhány jellegzetes, egymástól jól különböző koncepciót választunk ki. A kiválasztásnál hasznos, ha a koncepciók egy-egy „vezérelv” köré szerveződnek (például „minden automatikus”, „minimális beruházás”, „maximális rugalmasság”).

> **KoboCella: a morfológiai doboz**
>
> A KoboCella csapata a funkciófa hat legfontosabb részfunkciójához részfunkciónként négy megoldást gyűjtött (9.2. ábra). A $4^6 = 4096$ elvi kombinációból a 17 összeférhetetlen pár (például a kézi csavarbehajtó és a fix védőrács, vagy az ipari robot zárt cellában és a kézi kivétel) kizárása után 1344 maradt. A kötelező követelmények szűrője tovább szűkítette a kört: az R-09 szerint minden csavarkötés nyomatékadatát át kell adni a MES-nek, ezért a mintavételes nyomatékkulcsos ellenőrzés kiesett, és 1008 kombináció maradt.
> A csapat ezekből négy vezérelv köré szervezett koncepciót állított össze: „együttműködő cella” (kobot, lézerszkenner, sebességkorlátozás), „fix állomás” (többorsós fej, forgóasztal, védőrács), „kézi plusz” (fejlesztett kézi munkahely automatikus csavaradagolással és mérő csavarbehajtóval) és „zárt robotcella”. Az ábrán az első kettő útvonala látható. A négy koncepció a 10. fejezetben kerül értékelésre. A morfológiai doboz legfontosabb hozadéka a csapat szerint az volt, hogy a „kézi plusz” koncepció egyáltalán felmerült: a kobotos megoldás iránti lelkesedésben senki nem gondolt rá, holott a beruházási költsége a legkisebb.



Kék, folytonos: „együttműködő cella” koncepció · narancs, szaggatott: „fix állomás” koncepció. A doboz $4^6 = 4096$ elvi kombinációt tartalmaz; a 17 összeférhetetlen pár kizárása után 1344 marad.

9.2. ábra. A KoboCella morfológiai doboza két kiemelt koncepcióval. Minden koncepció egy-egy útvonal a dobozon át.

9.5. TRIZ: az ellentmondások feloldása

A **TRIZ** (orosz rövidítés, magyarul a feltalálói problémamegoldás elmélete) Altsuller szovjet mérnök és feltaláló nevéhez fűződik, aki az 1940-es évektől kollégáival szabadalmak tömegét elemezte azzal a céllal, hogy feltárja a sikeres találmányok mögötti ismétlődő mintázatokat (Altshuller 1984). Altsuller következtetése az volt, hogy a találmányok jelentős része néhány tucat általános elv alkalmazásával jön létre, és hogy az igazán innovatív megoldások egy **ellentmondást** oldanak fel, ahelyett hogy kompromisszumot kötnének (Altshuller 1984).

9.5.1. Az ellentmondás két fajtája

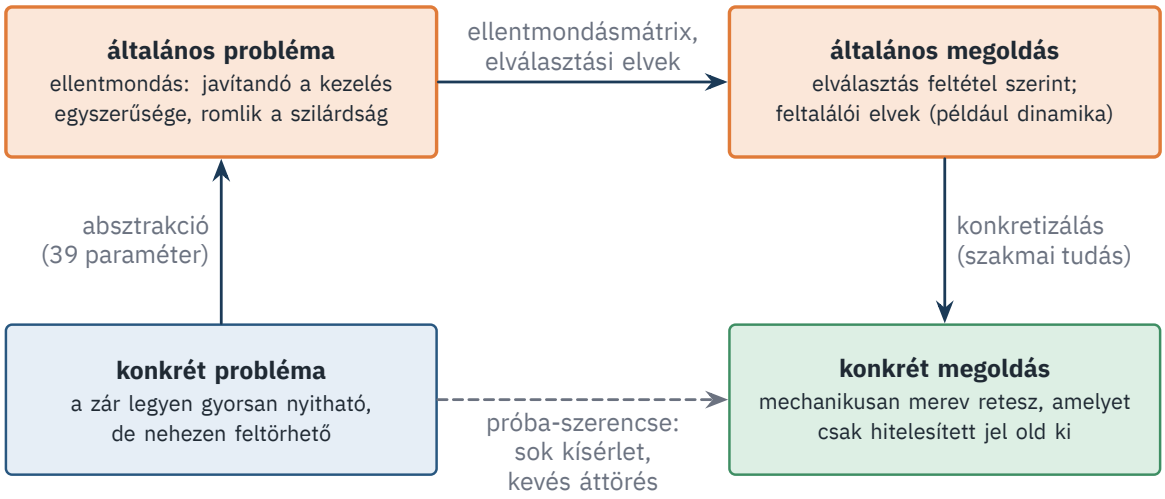
- **Műszaki ellentmondás:** ha a rendszer egyik jellemzőjét javítjuk, egy másik romlik. Például ha erősebbre tervezzük a kerékpárzárat, nehezebb és lassabban nyitható lesz. A hagyományos mérnöki válasz a kompromisszum: valahol középen megállunk. A QFD tetőmátrixának negatív kapcsolatai éppen ilyen ellentmondásokat jeleznek (7.4. szakasz).
- **Fizikai ellentmondás:** ugyanannak a jellemzőnek egyszerre két ellentétes értéket kellene felvennie. Például a zárnak *merevnek* kell lennie (hogy ne lehessen feltörni) és *könnyen mozdulónak* (hogy gyorsan nyíljon).

A fizikai ellentmondásokat a TRIZ **elválasztási elvekkel** oldja fel: az ellentétes követelményeket időben (az egyik értékre az egyik pillanatban, a másikra máskor van szükség), térben (a rendszer egyik részén az egyik, a másikon a másik), feltétel szerint (az egyik helyzetben az egyik, a másikban a másik) vagy a rendszerszintek között (az egész rendelkezik az egyik, a részek a másik tulajdonsággal) választjuk szét.

9.5.2. A 39 paraméter, a 40 elv és az ellentmondásmátrix

A műszaki ellentmondásokhoz Altsuller két listát állított össze (Altshuller 1984). Az egyik **39 általános műszaki paraméter** (például a mozgó tárgy tömege, a sebesség, az erő, a szilárdság, a megbízhatóság, a kezelés egyszerűsége, az automatizáltság foka, a termelékenység), amelyekkel egy konkrét ellentmondás általános alakra hozható. A másik a **40 feltalálói elv** (például a felosztás, az eltávolítás, a helyi minőség, az aszimmetria, az összevonás, a beágyazás, az előzetes cselekvés, a „fordítva”, a dinamika, a visszacsatolás, a közvetítő, az önkiszolgálás, a másolás, a kompozit anyagok). A kettőt az **ellentmondásmátrix** kapcsolja össze: a 39 × 39-es mátrix sorai a javítandó, oszlopai a romló paramétereket tartalmazzák, a cellák pedig azokat az elveket, amelyek a szabadalmak tanúsága szerint az adott ellentmondást leggyakrabban oldották fel.

A TRIZ alapsémája (9.3. ábra) a próba-szerencse módszer helyett egy kerülőutat javasol. A konkrét problémát általános problémává alakítjuk (absztrakció), az általános problémához megkeressük az általános megoldást (ellentmondásmátrix, elválasztási elvek), majd az általános megoldást szakmai tudásunk segítségével konkrét megoldássá alakítjuk.



9.3. ábra. A TRIZ alapsémája a VárosiBringa zárjának példáján. A konkrét problémától az általános problémán és az általános megoldáson át jutunk el a konkrét megoldáshoz.

A TRIZ egy további fontos fogalma az **ideális végeredmény**: az a képzeletbeli rendszer, amely a funkciót ellátja, de nincs költsége, nincs káros hatása, és nem foglal helyet. Az ideális végeredmény megfogalmazása (például „a kerékpár magától marad a helyén, amíg a jogosult felhasználó el nem viszi”) segít elszakadni a meglévő megoldásoktól.

VárosiBringa: a zár ellentmondása

A VárosiBringa minőségházának tetőmátrixában negatív kapcsolat volt a kölcsönzés gyorsasága és a zár feltörési ellenállása között (7.4. szakasz). A csapat először műszaki ellentmondásként fogalmazta meg: javítandó a kezelés egyszerűsége és a gyorsaság, romlik a szilárdság. Ezután rájött, hogy valójában fizikai ellentmondásról van szó: a retesznek egyszerre kell merevnek és könnyen mozdulónak lennie.
Az elválasztás *feltétel szerint* adta a megoldási irányt: a retesz jogosulatlan erőhatással szemben legyen merev, jogosult kérésre pedig könnyen mozduljon. A „dinamika” elv (a rendszer jellemzői változzanak a működési körülményekhez igazodva) és az „előzetes cselekvés” elv (a szükséges műveletet előre végezzük el) két konkrét ötlethez vezetett. Az egyik egy mechanikusan önzáró retesz, amelyet csak a hitelesített jelre működő kis teljesítményű aktuátor

old ki. A másik az, hogy a kioldást már a kölcsönzés jóváhagyásakor, a felhasználó odaérése előtt előkészítik, így a nyitás érzékelt ideje csökken. Az ideális végeredmény megfogalmazása egy harmadik irányt is megnyitott: ha a lopott kerékpár helyzete mindig ismert, a zár szilárdságára kisebb terhelés jut.

9.5.3. A TRIZ korlátai

Ilevbare, Probert és Phaal a TRIZ gyakorlati alkalmazásának előnyeit és nehézségeit felmérő tanulmányukban arra jutottak, hogy a módszer népszerűsége nő, de az elsajátítása és alkalmazása komoly kihívás. A nehézségek a módszertan összetettségétől a szervezeti és kulturális akadályokig terjednek (Ilevbare, Probert, és Phaal 2013). A tanulmány a kezdők számára is ad útmutatást: a felmérésben részt vevők által leggyakrabban használt eszközök (köztük a 40 feltalálói elv és az ellentmondások megfogalmazása) a legjobb kiindulópontok. Fontos tudni azt is, hogy a TRIZ eszköztára ennél jóval bővebb (szabványmegoldások, fejlődési trendek, problémamegoldó algoritmus), és az ellentmondásmátrix csak egy a sok eszköz közül.

9.6. Hat gondolkodó kalap

De Bono **hat gondolkodó kalap** módszere abból indul ki, hogy egy megbeszélésen a résztvevők egyszerre többféle gondolkodásmódot kevernek: tényeket közölnek, érzelmeiket fejezik ki, kockázatokat sorolnak és ötleteket vetnek fel (Bono 1985). Ez zavart és vitát okoz, mert az egyik ember egy ötlet előnyeiről, a másik a hátrányairól beszél, és egyikük sem hallja a másikat. A módszer szerint a csoport egyszerre csak egyféle módon gondolkodik, és ezt egy-egy képzeletbeli, színes kalap jelöli (9.2. táblázat).

9.2. táblázat. A hat gondolkodó kalap és a hozzájuk tartozó kérdések de Bono alapján (Bono 1985).

Kalap	Gondolkodásmód	Jellemző kérdés
fehér	tények, adatok, információ	Mit tudunk biztosan? Milyen adat hiányzik?
piros	érzelmek, megérzések	Mi a zsigeri reakciónk?
fekete	óvatosság, kockázatok, kritika	Milyen kockázatokra kell figyelnünk? Mi mehet félre?
sárga	előnyök, optimizmus	Miért lehetünk bizakodók? Mi az értéke?
zöld	kreativitás, új lehetőségek	Hogyan teremthetünk új lehetőségeket?
kék	a gondolkodási folyamat irányítása	Hol tartunk? Milyen rendszerek és folyamatok kellenek? Mi a következő lépés?

A kék kalapot általában a moderátor viseli: ő nyitja és zárja az ülést, és ő dönti el, mikor melyik kalapot veszi fel a csoport. Egy tipikus sorrend koncepcióértékeléshez: kék (cél), fehér (tények), zöld (ötletek), sárga (előnyök), fekete (kockázatok), piros (megérzések), kék (összegzés, következő lépések). Gyakorlásként érdemes egy egyszerű eseten kipróbálni a módszert, például egy kávéházon, amely egyre több panaszt kap a hosszú várakozás miatt: a csoport a tulajdonos, a vezetés és a fejlesztőcsapat szerepébe képzelve magát, kalapról kalapra haladva vizsgálja a problémát.

A módszer erőssége, hogy a kritikát nem tiltja (mint az ötletroham), hanem kijelöli a helyét. A fekete kalap ideje alatt kifejezetten elvárt a kockázatok felsorolása, a zöld kalap ideje alatt viszont nem. Így a kritikus alkatú résztvevők sem érzik, hogy el kell hallgatniuk, a kreatívabbak pedig nem érzik, hogy minden ötletüket azonnal lelövik.

9.7. Természet ihlette tervezés

A természet ihlette tervezés (biomimetika, biomimikri) az élővilág megoldásait használja fel műszaki problémák megoldására. A tépőzár és a szögesdrót példáját a 8.1. szakasz már bemutatta. Jó kiindulópont az **AskNature** adatbázis, amely az élővilág stratégiáit funkciók szerint rendszerezi: a tervező egy funkcióra (például „rögzítés”, „hűtés”, „energiaelnyelés”) keresve talál biológiai példákat (Biomimicry Institute 2026).

Vincent és szerzőtársai a biológiai és a műszaki problémamegoldás összehasonlítására a TRIZ fogalomrendszerét használták, és egy biológiai változatot (BioTRIZ) dolgoztak ki. Elemzésük szerint a biológia és a technika által alkalmazott megoldási elvek között meglepően kicsi az átfedés. A technika jellemzően az energiafelhasználás módosításával old meg problémákat, a biológia inkább az információ és a szerkezet alakításával (Vincent és mtsai. 2006). Ez a különbség a mérnök számára éppen a lehetőséget jelenti: a természet olyan megoldási irányokat mutathat, amelyek a műszaki gyakorlatban ritkák.

A természet ihlette tervezésnek két iránya van. A **problémavezérelt** megközelítés egy műszaki problémából indul ki, és a megoldáshoz biológiai analógiát keres. A **megoldásvezérelt** megközelítés egy érdekes biológiai jelenségből indul ki (mint a bogáncs horgai), és alkalmazást keres hozzá. A rendszermérnöki munkában az első a gyakoribb, mert a funkció már adott. A legnagyobb kockázat a felszínes analógia: a biológiai megoldás csak akkor használható, ha a működési elvét (és nem csupán a formáját) értjük, és ha a méretarányok és az anyagok közötti különbségek nem teszik érvénytelenné.

9.8. Számítógépes koncepciókeresés

Ha a tervezési tér paraméterekkel leírható, és egy-egy változat teljesítménye számítással vagy szimulációval értékelhető, a koncepciókeresés egy része algoritmusokra bízható. Két, a természetből ihletett keresőmódszer különösen elterjedt.

A **genetikus algoritmus** a darwini evolúció elvét követi (Holland 1975). A lehetséges megoldásokat (fenotípusokat) kódolt formában (genotípusként) ábrázoljuk, és egy populációt hozunk létre belőlük. A populáció egyedeit értékeljük, a jobbakat nagyobb valószínűséggel kiválasztjuk, majd keresztezéssel és mutációval új egyedeket hozunk létre, amelyek a rosszabbak helyére lépnek. A folyamat ismétlődik, amíg a populáció jó megoldásokhoz nem konvergál. A módszer általában gyorsan talál az optimálishoz közeli megoldást, de nem garantálja a globális optimumot.

A **részecskeraj-optimalizálás** (particle swarm optimization, PSO) a madárrajok és halrajok viselkedését utánozza (Kennedy és Eberhart 1995). A lehetséges megoldások részecskékként „repülnek” a problématérben, és információt osztanak meg egymással. Minden részecske emlékszik a saját eddigi legjobb pozíciójára, és ismeri a raj legjobbját is. Mozgási sebességét és irányát e kettő felé módosítja, így a raj fokozatosan a jó megoldások környékére gyűlik.

Mindkét módszer alkalmazásának kulcsa a **kódolás** és az **értékelés**. Egy hőkezelési program optimalizálásánál például a hőmérséklet- és időparamétereket egyetlen számsorba kódolják, és minden jelölt programot dinamikus szimulációval értékelnek. A KoboCellánál ugyanígy kódolható lenne a kobot pályájának néhány köztes pontja, és a ciklusidő a robotszimulációból számítható. Ha több, egymással ütköző célt (például ciklusidő és energiafelhasználás) egyszerre kell figyelembe venni, a többcélú evolúciós algoritmusok, például Deb és szerzőtársai NSGA-II algoritmusa, nem egyetlen megoldást, hanem a Pareto-optimális megoldások halmazát közelítik (Deb és mtsai. 2002). A Pareto-front fogalmával a 10. fejezet foglalkozik.

Fontos látni a számítógépes keresés határait. Az algoritmus csak a megadott kódolás által kifeszített térben keres: ha a kódolás csak a kobotos koncepció paramétereit tartalmazza, a fix állomást soha nem fogja megtalálni. A számítógépes keresés tehát a koncepción *belüli* optimalizálásra való, a koncepciók *közötti* választás továbbra is a csapat feladata.

> **Nagy nyelvi modellek az ötletgenerálásban**
>
> A nagy nyelvi modellek (LLM) gyorsan és nagy mennyiségben képesek ötleteket előállítani, és jól használhatók a csoportos módszerek kiegészítésére: további megoldásokkal tölthetik fel a morfológiai doboz sorait, TRIZ-elvekhez konkrét alkalmazási javaslatokat adhatnak, vagy egy-egy gondolkodó kalap szerepében érvelhetnek. Doshi és Hauser online kísérlete azonban egy fontos kockázatra figyelmeztet. A kísérletben a nyelvi modelltől ötleteket kapó írók történeteit kreatívabbnak és jobban megírtnak értékelték, különösen a kevésbé kreatív írók esetében. A modell segítségével írt történetek viszont jobban hasonlítottak egymásra, mint a csak emberek által írtak (Doshi és Hauser 2024). A szerzők ezt társadalmi dilemmaként írják le: az egyén jobban jár, de együttesen szűkebb lesz az új tartalmak köre. A koncepcióalkotás szempontjából ez azt jelenti, hogy a nyelvi modell ötleteit a csapat saját, önálló ötletgyűjtése *után* érdemes bevonni, különben a modell a tervezési fixáció új forrásává válhat.

9.9. Az ötletgenerálás eredményességének értékelése

Honnan tudjuk, hogy egy ötletgeneráló ülés vagy módszer eredményes volt? Shah, Smith és Vargas-Hernandez négy mérőszámot javasoltak az ötletgenerálás eredményességének mérésére (Shah, Smith, és Vargas-Hernandez 2003):

- **mennyiség:** az előállított ötletek száma;
- **változatosság:** mennyire fedik le az ötletek a megoldástér különböző részeit (például hány különböző működési elvre épülnek);
- **újdonság:** mennyire szokatlanok az ötletek a szokásos megoldásokhoz képest;
- **minőség:** mennyire megvalósíthatók, és mennyire teljesítik a követelményeket.

A négy mérőszám egymást kiegészíti. Egy ülés, amely száz ötletet hoz, de mindegyik ugyanarra a működési elvre épül, mennyiségben kiváló, változatosságban gyenge. A gyakorlatban a változatosság a legfontosabb a koncepcióalkotás korai szakaszában, mert a konvergens szakasz csak abból választhat, amit a divergens szakasz előállított.

A konvergens szakasz első lépése a **kizáró szűrés**: azokat az ötleteket, amelyek egy kötelező követelményt vagy korlátot biztosan nem teljesítenek, indoklással kizárjuk. A KoboCella esetében ilyen volt a mintavételes nyomatékellenőrzés kizárása az R-09 alapján. A szűrésen átjutott koncepciókat ezután összehasonlító értékeléssel rangsoroljuk, amelyről a következő fejezet szól.

Összefoglalás

- A koncepcióalkotás két szakasza a divergens (sok, változatos ötlet, felfüggesztett ítélettel) és a konvergens (tisztázás, szűrés, értékelés, választás). A kettőt időben szét kell választani.
- A legjobban dokumentált gát a tervezési fixáció: a látott példák jellemzői akkor is visszaköszönnek az új tervekben, ha hibásak. Ellene a megoldássemleges funkció, a feltevések kimondása, a távoli analógiák és a rendszerezett módszerek segítenek.
- Az ötletroham négy szabálya: az ítélet felfüggesztése, a mennyiség, a szokatlan ötletek bátorítása, a kombinálás. A szóbeli csoportok termelékenysége a produkciós blokkolás miatt elmarad az egyénileg dolgozókétól; az ötletírás (6-3-5) és a névleges csoporttechnika ezt kiküszöböli. A Delphi névtelen, többkörös szakértői becslő módszer.
- A morfológiai doboz részfunkciónként gyűjtött megoldások kombinálásával állít elő koncepciókat. A kombinációk száma a megoldásszámok szorzata, ezért összeférhetetlenségi és követelményszűrésre, majd vezérelvek szerinti válogatásra van szükség.
- A TRIZ az ellentmondások feloldására épül. A műszaki ellentmondásokhoz a 39 paraméter, a 40 feltalálói elv és az ellentmondásmátrix, a fizikai ellentmondásokhoz az elválasztási elvek (idő, tér, feltétel, rendszerszint) adnak megoldási irányt.
- A hat gondolkodó kalap a gondolkodásmódokat időben választja szét, és a kritikának is kijelöli a helyét. A természet ihlette tervezés a technikában ritka megoldási elvekhez vezethet. A genetikus algoritmus és a részecskeraj-optimalizálás a koncepción belüli paraméterkeresésre való.
- Az ötletgenerálás eredményessége a mennyiséggel, a változatossággal, az újdonsággal és a minőséggel mérhető. A korai szakaszban a változatosság a legfontosabb.

Önellenőrzés

1. Miért kell a divergens és a konvergens szakaszt időben szétválasztani? Mi történik, ha egy ötletet azonnal értékelünk?
2. Mi a tervezési fixáció? Milyen sorrendben mutatnád meg a csapatnak a versenytársak megoldásait egy koncepcióalkotó műhelyben?
3. Hány ötlet születhet legfeljebb egy 6-3-5 ülésen, ha öt résztvevő van, és mindenki minden lapra ír egyszer?
4. Egy morfológiai dobozban öt részfunkció van, a megoldások száma rendre 3, 4, 2, 5, 3. Hány elvi kombináció van? Mennyi marad, ha a második részfunkció egyik megoldását egy kötelező követelmény miatt kizárjuk?
5. Fogalmazd meg fizikai ellentmondásként: „a dokkoló legyen könnyen telepíthető, de ne lehessen elmozdítani”. Melyik elválasztási elv segíthet?
6. Mire használnád a fekete és mire a zöld kalapot egy koncepcióértékelő megbeszélésen? Miért jó, hogy a módszer nem tiltja a kritikát?
7. Miért nem találhatja meg egy genetikus algoritmus a fix állomás koncepciót, ha a kódolás a kobot pályapontjait tartalmazza?

Megoldás.

1. Mert a korai kritika elfojtja a további ötleteket, és a csoport az első elfogadható megoldásnál megáll. Ha egy ötletet azonnal értékelünk, a résztvevők óvatosabbak lesznek, és a szokatlan, de továbbfejleszthető ötletek ki sem mondódnak.
2. A tervezési fixáció az a jelenség, hogy a látott példamegoldások jellemzői akkor is megjelennek az új tervekben, ha hibásak. A versenytársak megoldásait csak az első önálló ötletgyűjtés után érdemes megmutatni.
3. Öt résztvevő, mindenki három ötletet ír minden körben, és öt kör van (minden lap mindenkihez eljut egyszer): $5 \cdot 3 \cdot 5 = 75$ ötlet.
4. $N = 3 \cdot 4 \cdot 2 \cdot 5 \cdot 3 = 360$. Ha a második részfunkcióból egy megoldást kizárunk, $3 \cdot 3 \cdot 2 \cdot 5 \cdot 3 = 270$ marad.
5. A dokkoló talapzatának egyszerre kell könnyen mozdíthatónak (telepítéskor) és mozdíthatatlannak (üzem közben) lennie. Az elválasztás időben segít: telepítéskor könnyű, moduláris elemek, üzemben például ballaszttal vagy talajhoz rögzítő elemmel rögzített szerkezet. Az elválasztás feltétel szerint is működhet: csak célszerszámmal oldható rögzítés.

6. A fekete kalappal a kockázatokat, gyenge pontokat és lehetséges hibákat gyűjtjük, a zöld kalappal új ötleteket és a gyenge pontok javítási lehetőségeit. A kritika kijelölt helyet kap, így a kritikus résztvevők sem hallgatnak el, a kreatív szakasz alatt viszont nem lövik le az ötleteket.
7. Mert az algoritmus csak a kódolás által kifeszített térben keres. A kobot pályapontjai a kobotos koncepció paraméterei; a fix állomás ebben a térben nem ábrázolható, tehát nem is található meg.

Projektlépés: koncepciók előállítása

Saját fejlesztési feladatodhoz:

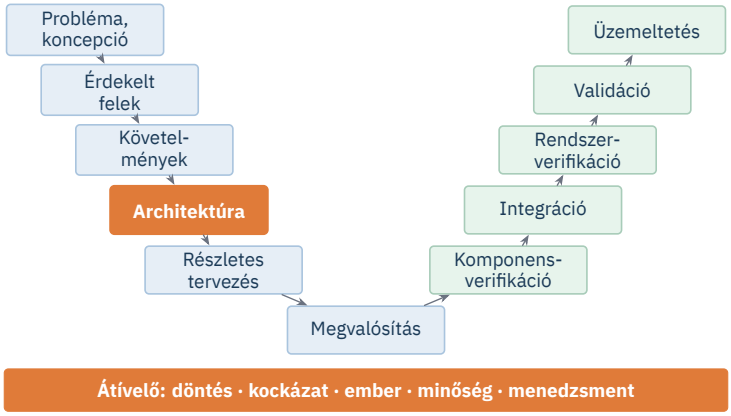
1. írd le a csapat hallgatólagos feltevéseit a megoldásról, és jelöld meg, melyik kérdőjelezhető meg;
2. tarts egy 6-3-5 vagy névleges csoportos ülést a fő funkció megoldási lehetőségeiről, és rögzítsd az ötletek számát és a különböző működési elvek számát;
3. készíts morfológiai dobozt legalább öt részfunkcióval, jelöld az összeférhetetlen párokat, és számítsd ki az elvi és a megmaradó kombinációk számát;
4. keress legalább egy ellentmondást (például a minőségház tetőmátrixából), és fogalmazd meg TRIZ-szel;
5. állíts össze legalább három, egymástól lényegesen különböző koncepciót vezérelvvel és egybekezdéses leírással.

A morfológiai doboz, a TRIZ-elvek, a hat kalap és az ötletroham sablonja és notebookja a Gyakorlati útmutatóban található.

Továbbolvasás

- Osborn (1953) és Diehl és Stroebe (1987): az ötletroham eredeti leírása és a csoportos termelékenységi veszteség klasszikus vizsgálata.
- Delbecq és Van de Ven (1971) és Rowe és Wright (1999): a névleges csoporttechnika eredeti leírása és a Delphi-módszer kritikai áttekintése.
- Zwicky (1969): a morfológiai módszer eredeti leírása.
- Altshuller (1984) és Ilevbare, Probert, és Phaal (2013): a TRIZ alapműve és gyakorlati alkalmazásának kritikai áttekintése.
- Jansson és Smith (1991): a tervezési fixáció klasszikus kísérlete.
- Shah, Smith, és Vargas-Hernandez (2003): az ötletgenerálás eredményességének mérőszámai.
- Goldberg és mtsai. (1994), 7.7., 7.9. és 7.10. szakasz: az ötletroham, a Delphi-módszer és a névleges csoporttechnika eljárásai.

10. Koncepció-kiválasztás és döntéstámogatás



Tanulási eredmények

A fejezet feldolgozása után képes leszel:

- leírni egy választási döntés elemeit és a döntéskezelési folyamat lépéseit, és felismerni, mikor indokolt formális döntéselemzés;
- megfogalmazni jó értékelési kritériumokat, és kizáró szűréssel elkülöníteni a kötelező követelményeket nem teljesítő alternatívákat;
- elvégezni egy Pugh-féle összehasonlítást, és megindokolni, mire alkalmas és mire nem;
- súlyozott értékelést (trade study) készíteni hasznossági függvényekkel, és érzékenységvizsgálattal ellenőrizni a rangsor robusztusságát;
- páros összehasonlítással (AHP) súlyokat és prioritásokat számítani, és ellenőrizni a következetességet;
- dominanciavizsgálattal meghatározni a Pareto-frontot, és költség–haszon elemzést végezni növekményes elemzéssel;
- felismerni a csoportos döntések és az emberi ítéletalkotás csapdáit, és dokumentálni egy döntést.

Az előző fejezet végére a csapatok asztalán több, egymástól lényegesen különböző koncepció van. A KoboCellánál négy, a ForgácsTech-nél három, a VárosiBringánál tucatnyi tervváltozat. Most választani kell. A választás a V-modell bal ágának egyik legkövetkezményesebb lépése: a kiválasztott koncepció határozza meg a további tervezés kereteit, és a rendszerkövetelmény- és rendszertervezési felülvizsgálatokon (2.5. szakasz) ennek a döntésnek az indoklását is be kell mutatni.

A döntés alternatívák közötti tudatos választás: csak ott van mit eldönteni, ahol legalább két valódi lehetőség közül választhatunk. A rendszermérnöki döntések ritkán egyszerűek. Több, egymással ütköző kritérium szerint kell választani, a kritériumok egy része nehezen mérhető, az alternatívák jellemzői bizonytalanok, és a döntésben több érdekelt fél vesz részt, eltérő értékrenddel. Ez a fejezet olyan módszereket mutat be, amelyek ezt

a bonyolultságot átláthatóvá teszik. Egyik módszer sem dönt helyettünk, de mindegyik segít abban, hogy a döntés indokolható, megismételhető és utólag is érthető legyen.

A fejezet négy kérdésre épül:

- Milyen elemekből áll egy döntés, és mikor van szükség formális döntéselemzésre?
- Hogyan hasonlíthatók össze a koncepciók minőségi és mennyiségi kritériumok szerint?
- Hogyan ellenőrizhető, hogy az eredmény nem a bizonytalan becsléseken vagy a súlyokon múlik?
- Milyen csapdák fenyegetik az egyéni és a csoportos döntéseket?

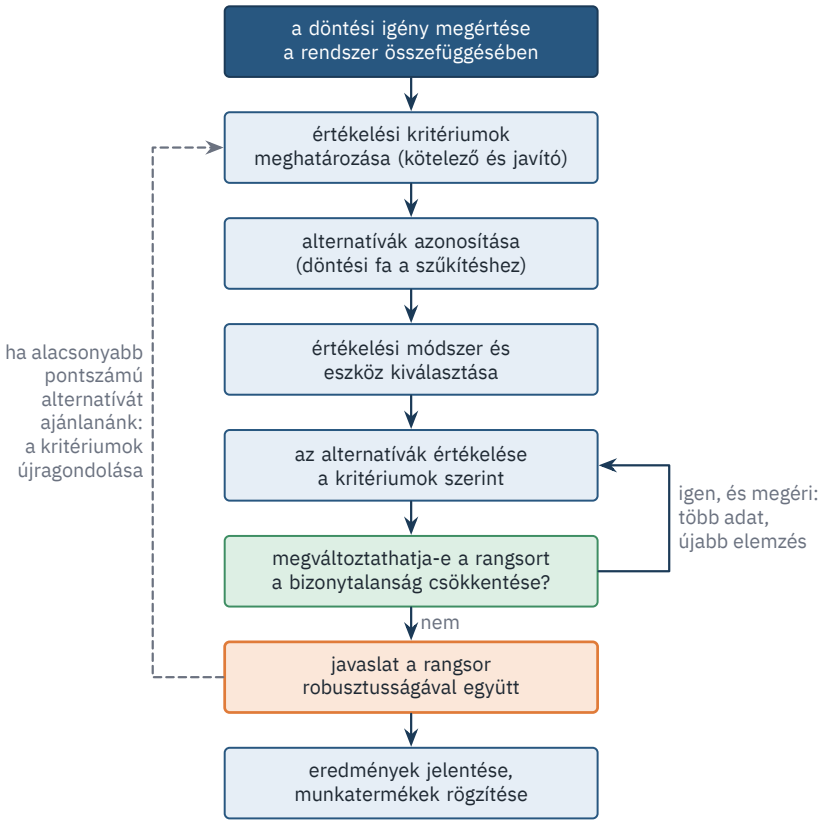
10.1. A döntés elemei és folyamata

Ben-Akiva és Lerman a választási folyamatot négy elemmel írja le: a döntéshozóval, az alternatívákkal, az alternatívák jellemzőivel (attribútumaival) és a döntési szabállyal, amely alapján a döntéshozó a jellemzők ismeretében választ (Ben-Akiva és Lerman 1985). A gyakorlatban ugyanez négy kérdésként jelenik meg: mik az alternatívák, mik a kritériumok, mi az értékítélet, és ki a döntéshozó (egyén vagy csoport). A négy elem közül a leginkább elhanyagolt az értékítélet: az, hogy a kritériumok közül melyik mennyire fontos, nem műszaki, hanem értékválasztási kérdés, amelyet a döntésre jogosultaknak kell meghozniuk.

Az ISO/IEC/IEEE 15288 szabvány döntéskezelési folyamatának célja, hogy rendezett, elemző keretet adjon a döntési alternatívák objektív azonosításához, jellemzéséhez és értékeléséhez, és a legelőnyösebb cselekvési irány kiválasztásához az életciklus bármely pontján (ISO/IEC/IEEE 2023, 6.3.3. szakasz). A döntési modellek a szakértői ítéleteket és a döntéshozói preferenciákat is számszerűsítik, így segítenek abban a nehéz feladatban, hogy hiányos tudás birtokában kell versengő alternatívák közül választani. Ezért kell minden módszer feltevéseit és korlátait megérteni és dokumentálni.

A döntéselemzési folyamat lépéseit a 10.1. ábra mutatja. Nem minden döntés igényel formális elemzést: sok döntés egyszerűen meghozható a felelős szakterületek világos tájékoztatása alapján. Parnell és szerzőtársai szerint a formális döntéselemzés azokban a nehéz döntésekben hoz értéket, amelyekre az alábbi jellemzők közül több is igaz (Parnell és mtsai. 2013; NASA 2016, 6.8. szakasz):

- **összetett alternatívák:** a változatok következményei részletes elemzés nélkül nehezen érthetők;
- **jelentős bizonytalanság:** a legfontosabb bemenetek bizonytalansága az alternatívák rangsorát is bizonytalanná teszi;
- **több, egymással ütköző cél:** minél több a figyelembe veendő szempont, annál nagyobb a formális elemzés haszna;
- **több érdekelt fél:** ha az érdekelt felek értékei, preferenciái és nézőpontjai eltérnek, a célokat külön tisztázni kell;
- **súlyos következmények:** a döntés nagy tétű, és nehezen vagy egyáltalán nem fordítható vissza.



10.1. ábra. A döntéselemzés folyamata az ISO/IEC/IEEE 15288 döntéskezelési folyamata (ISO/IEC/IEEE 2023, 6.3.3. szakasz) és Parnell és szerzőtársai (Parnell és mtsai. 2013) nyomán, saját szerkesztés. A bizonytalanság csak akkor számít, ha csökkentése megváltoztathatja a rangsort.

10.1.1. Az emberi ítéletalkotás korlátai

Az Ebbinghaus-féle optikai csalódásban két egyforma kör közül azt látjuk nagyobbnak, amelyiket kisebb körök vesznek körül. Az ítéletalkotásunk ugyanígy rendszeresen és előre jelezhetően torzul. Tversky és Kahneman klasszikus tanulmánya három heurisztikát írt le, amelyekre bizonytalanság esetén támaszkodunk, és amelyek rendszeres hibákhoz vezetnek: a **reprezentativitást** (annak megítélése, mennyire hasonlít valami egy jellegzetes mintára), a **hozzáférhetőséget** (azt tartjuk valószínűbbnek, ami könnyebben eszünkbe jut) és a **lehorgonyzást** (a becslés a kiinduló értéktől csak kevéssé tér el) (Tversky és Kahneman 1974). Egy koncepcióértékelésnél a lehorgonyzás például úgy jelenik meg, hogy az első elhangzó költségbecslés minden későbbi becslést maga felé húz.

A döntési helyzet kialakítása maga is befolyásolja az eredményt. Johnson és Goldstein európai országok szervadományozási adatait vetették össze. Azokban az országokban, ahol a szervadományozáshoz kifejezett hozzájárulás kell, a hozzájárulók aránya jellemzően harminc százalék alatt maradt. Ahol viszont a hozzájárulás az alapértelmezés, és külön nyilatkozni kell az elutasításról, az arány 85% fölött volt, Magyarországon és Ausztriában gyakorlatilag száz százalék (E. J. Johnson és Goldstein 2003). A rendszermérnök számára ebből két tanulság következik. Az egyik, hogy a döntési folyamat tervezése (milyen sorrendben, milyen alapértelmezéssel, milyen keretezéssel kerülnek elő az alternatívák) nem semleges. A másik, hogy az általunk tervezett rendszerek alapértelmezései a felhasználók döntéseit is erősen befolyásolják (11. fejezet).

10.2. Kritériumok és kizáró szűrés

A döntési kritériumok azok a szempontok, amelyek szerint az alternatívákat egyenként értékeljük. Jellemző kritériumok a költség, az ütemterv, a kockázat, a biztonság, a teljesítmény és a támogathatóság, de a döntéshez kapcsolódó műszaki kritériumokat is figyelembe kell venni. Kepner és Tregoe klasszikus módszere két kritériumfajtát különít el: a **kötelező** („kell”) és a **javító** („kívánatos”) kritériumokat (Kepner és Tregoe 1965). Azt az alternatívát, amely egy kötelező kritériumot nem teljesít, el kell vetni; a javító kritériumok pedig a megmaradt alternatívák rangsorolására szolgálnak.

Keeney és Gregory szerint egy jó mérőszám (attribútum) öt tulajdonsággal rendelkezik (Keeney és Gregory 2005):

- **egyértelmű:** minden értékelő ugyanazt érti rajta, és a következmény alapján egyértelműen megadható az értéke;
- **teljes körű:** a cél teljesülésének minden lehetséges fokát lefedi;
- **közvetlen:** közvetlenül azt méri, amiről a döntéshozónak tudnia kell;
- **működtethető:** elfogadható költséggel és bizonytalansággal mérhető vagy becsülhető;
- **érthető:** a következmények és az átváltások könnyen kommunikálhatók vele.

Ehhez jön egy, a súlyozott összegzésnél különösen fontos követelmény: a kritériumok legyenek egymástól függetlenek, vagyis ne mérjék ugyanazt a tulajdonságot többször (Keeney és Raiffa 1976).

Ez a követelmény gyakran sérül. Ha a „beruházási költség” és a „teljes életciklus-költség” egyaránt szerepel a kritériumok között, a beruházást kétszer számoljuk. Ha a „ciklusidő” és a „termelékenység” is szerepel, ugyanazt a tulajdonságot kétszer jutalmazzuk.

A koncepcióválasztás tölcsérként képzelhető el: a kezdeti ötletekből a szűrés a lehetséges, a megvalósítható, végül az ígéretes koncepciókhoz vezet. Az első szűrő a **kizáró szűrés**: minden alternatívát összevetünk a kötelező követelményekkel és korlátokkal (fizikai törvények, erőforrások, szabványok, hatósági előírások, biztonság), és a nem megfelelőket indoklással kizárjuk. A kizárt alternatívákat is dokumentálni kell. Egy fontos döntés értékelő táblázatát akkor is érdemes elkészíteni, ha csak egyetlen alternatíva bizonyult életképesnek, mert az indoklás így is dokumentálva marad. Azt se felejtsük el, hogy a semmittevés vagy a döntés elhalasztása is lehet alternatíva.

Ha az alternatívák száma nagy, érdemes őket **döntési fába** (trade tree) rendezni. A fa megmutatja, hogyan bomlanak a magas szintű alternatívák egyre alacsonyabb szintű alternatívákra, és segít a változatok számát a részletes elemzés előtt csökkenteni, valamint a szűkítés indoklását az érdekelt felekkel megosztani (Parnell 2017; Goldberg és mtsai. 1994, 2.1.3. szakasz). Egy nagy értékelés így több kisebbre bontható.

10.3. A Pugh-módszer

A koncepció szakaszának legegyszerűbb összehasonlító módszere Pugh skót mérnök nevéhez fűződik (Pugh 1991). A módszer minden alternatívát egy kiválasztott **vonatkoztatási alaphoz** (datum) hasonlít, kritériumonként három fokozattal: jobb (+), azonos (S vagy 0), rosszabb (–). A módszer lépései (Pugh 1991):

1. a kritériumok kiválasztása az alapkövetelmények és a kívánt cél figyelembevételével;
2. az összehasonlítandó alternatívák kiválasztása, amelyeket a lehető legtöbb szempontból azonos részletességgel kell leírni;
3. a pontozás: kiválasztunk egy vonatkoztatási alapot (például a jelenlegi megoldást vagy a legígéretesebbnek vélt koncepciót), és minden alternatívát minden kritérium szerint ehhez hasonlítunk;
4. az összesítés: alternatívánként megszámoljuk a pluszokat és a mínuszokat, és előjelesen összeadjuk. Ez az érték csak tájékoztató jellegű;
5. szükség esetén szélesebb skála (például −3-tól +3-ig) vagy súlyozás is alkalmazható.

A Pugh-módszer ereje, hogy rendezi az eljárást, valódi csapatmunkára ösztönöz, kiszűri a gyengébb ötleteket és továbbviszi az erősebbeket, és segít felismerni a jó kombinációkat. Pugh ezt a fokozatos szűkítést és bővítést irányított konvergenciának nevezte (Pugh 1991). Arra viszont nem alkalmas, hogy automatikusan döntsön: a számok csak tájékoztató jellegűek, nem szabad őket algebrailag összeadva végeredménynek tekinteni.

A módszer legfontosabb tulajdonsága az iteráció. Pugh azt javasolta, hogy az első kör után a csapat vizsgálja meg, hogyan lehetne a legjobb koncepciók gyenge pontjait más koncepciók erősségeivel javítani, majd egy új vonatkoztatási alappal (általában az első kör nyertesével) ismételje meg az összehasonlítást. Így a módszer nemcsak szűr, hanem új, hibrid koncepciókat is létrehoz.

10.1. táblázat. A KoboCella koncepcióinak Pugh-féle összehasonlítása. Vonatkoztatási alap: a jelenlegi kézi szerelés.

Kritérium	A: együttműködő cella	B: fix állomás	C: kézi plusz	D: zárt robotcella
ciklusidő-képesség	+	+	+	+
kötésminőség, utómunka	+	+	+	+

Kritérium	A: együttműködő cella	B: fix állomás	C: kézi plusz	D: zárt robotcella
ergonómia	+	+	S	+
beruházási költség	–	–	S	–
átállási rugalmasság	S	–	S	–
helyigény	S	–	S	–
technikai kockázat	–	S	S	–
bővíthetőség 140 ezer db/évig	+	+	S	+
pluszok / mínuszok	**4 / 2**	**4 / 3**	**2 / 0**	**4 / 4**
előjeles összeg (tájékoztató)	**+2**	**+1**	**+2**	**0**

KoboCella: a Pugh-összehasonlítás tanulságai

A KoboCella csapata a négy koncepciót a jelenlegi kézi szereléshez hasonlította (10.1. táblázat). Az eredmény meglepő volt: a „kézi plusz” koncepció ugyanannyi előjeles pontot kapott, mint az együttműködő cella, és egyetlen mínusza sem volt. A csapat ezért a kizáró szűrés előtt részletesebben megvizsgálta. A munkahely-ergonómiai elemzés szerint a „kézi plusz” koncepcióban a csavarozás alsó pozícióinál a legnagyobb törzshajlítás becsült értéke 25°, miközben az R-05 követelmény legfeljebb 20°-ot enged meg (6. fejezet). A „kézi plusz” így kötelező követelményt sértett, és kiesett. A csapat egy tanulságot mégis megtartott belőle: az automatikus csavaradagolást és a mérő csavarbehajtót a kobotos koncepció tartalék üzemmódjába is beépítette, hogy a kobot meghibásodása esetén a cella csökkentett ütemben kézzel is működtethető legyen (UM-7, leromlott üzem).
A második körben a csapat az együttműködő cellát választotta vonatkoztatási alapnak. A fix állomás és a zárt robotcella rugalmasságban és helyigényben rosszabb, ciklusidőben jobb volt. A Pugh-módszer tehát nem adott egyértelmű nyertest, de világossá tette, hogy a döntés a ciklusidő és a rugalmasság közötti átváltáson múlik. Ennek számszerűsítésére a csapat súlyozott értékelést végzett.

10.4. Súlyozott értékelés és hasznossági függvények

10.4.1. A súlyozott értékelés lépései

A **súlyozott értékelés** akkor használható jól, ha az alternatívák és a követelmények is jól definiáltak. Minden fontos kritériumhoz súlyt rendelünk, az alternatívákat kritériumonként értékeljük, és a döntés a súlyozott értékelés számszerű eredményén alapul. Elméleti alapja a többszempontú hasznosságelmélet (Keeney és Raiffa 1976), egyszerűsített, a gyakorlatban jól bevált változatát Edwards és Barron írta le (Edwards és Barron 1994). Az eljárás jellemző lépései (Parnell és mtsai. 2013; Goldberg és mtsai. 1994, 2.1.3. szakasz):

1. a célok és követelmények meghatározása és lehetőség szerinti rangsorolása;
2. a hiteles alternatívák azonosítása, a követelményeket biztosan nem teljesítők kizárása;
3. döntési fa készítése (nagy értékeléseknél);
4. a kiválasztási kritériumok meghatározása;
5. a kritériumok súlyainak meghatározása, összesen 100 ponttal;
6. szükség esetén a súlyok meghatározása páros összehasonlítással (AHP, 10.5. szakasz);

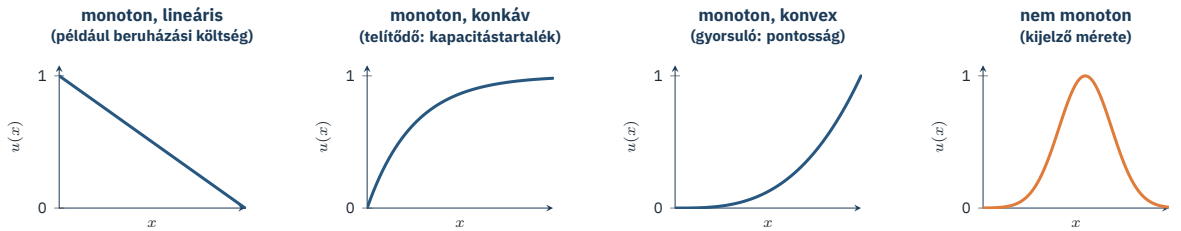
7. hasznossági függvények felállítása a különböző mértékegységű kritériumok közös skálára hozásához;
8. az alternatívák értékelése kritériumonként, a hasznossági pontszámok és a súlyozott pontszámok kiszámítása;
9. az eredmények táblázatba foglalása;
10. érzékenységvizsgálat;
11. a legmagasabb súlyozott összpontszámú alternatíva kiválasztása.

Fontos szervezési szabály, hogy a súlyokat a végső döntési jogkörrel rendelkező személy vagy csoport határozza meg, mégpedig előre, mielőtt az alternatívák pontszámai ismertté válnak. Az elemzők így elfogulatlanul értékelhetik az alternatívákat a kritériumok szerint. Ezzel elválik egymástól az *értékítélet* (mennyire fontos egy kritérium) és a *ténybeli ítélet* (mennyire jó egy alternatíva az adott kritérium szerint).

10.4.2. Hasznossági függvények és normalizálás

A kritériumok különböző mértékegységűek: másodperc, százalék, fok, millió forint, perc, szóbeli kockázati besorolás. Összeadni csak közös skálán lehet őket. A hasznosság (utilitás) a relatív megfelelőség mértéke: a döntéshozó preferenciáit kifejező szubjektív érték, amely a mindenkori döntési céllal változik, ezért nincs általános hasznossági függvény. A hasznosság fogalmának döntéselméleti megalapozása Neumann János és Oskar Morgenstern nevéhez fűződik (Neumann és Morgenstern 1944). A gyakorlatban minden kritériumhoz egy **hasznossági függvényt** rendelünk, amely a természetes mértékegységben kifejezett értéket 0 és 1 (vagy 0 és 10) közötti pontszámra képezi le.

A hasznossági függvény alakja nem mindig lineáris (10.2. ábra). Egy mobiltelefon kijelzőjénél például a nagyobb méret egy határig jobb, azon túl rosszabb: a függvény nem monoton. A kapacitástartaléknál az első néhány százalék nagyon értékes, a további egyre kevésbé: a függvény konkáv, telítődő. A normalizálás módjától függetlenül a legfontosabb a skála **működtethető definíciója**: minden fokozatnak megismételhető, mérhető tartalma legyen (Keeney és Gregory 2005). A „magas” például jelentheti azt, hogy „legalább 67% valószínűség”, az „alacsony” pedig azt, hogy „legfeljebb 33% valószínűség”. A transzformációt minden esetben dokumentálni kell.



10.2. ábra. A hasznossági függvények jellemző alakjai. A lineáris függvény a leggyakoribb, de nem mindig a legjobb közelítés.

Lineáris hasznossági függvénynél, ha az x érték a „rossz” x^- és a „jó” x^+ szint között van, a pontszám

$$u(x) = 10 \cdot \frac{x - x^-}{x^+ - x^-}, \tag{10.1}$$

amely kisebb-a-jobb kritériumnál (költség, idő) is működik, ha $x^{+} < x^{-}$. Az alternatívák összpontszáma a **súlyozott összeg**:

$$U_j = \sum_{i=1}^{m} w_i \, u_i(x_{ij}), \qquad \sum_{i=1}^{m} w_i = 1, \tag{10.2}$$

ahol w_i az i-edik kritérium súlya, $u_i(x_{ij})$ pedig a j-edik alternatíva hasznossági pontszáma az i-edik kritérium szerint.

Mikor szabad összeadni? A függetlenségi feltételek

A súlyozott összeg a többszempontú hasznosságelemzés (multi-attribute utility analysis) legegyszerűbb modellje. Keeney és Raiffa elmélete szerint az additív modell csak akkor érvényes, ha a kritériumok között bizonyos függetlenségi feltételek teljesülnek (Keeney és Raiffa 1976). Keeney és Raiffa három szintet különböztet meg: az additív függetlenség magában foglalja a hasznossági függetlenséget, az pedig a preferenciafüggetlenséget. A legenyhébb, a preferenciafüggetlenség azt jelenti, hogy két kritérium közötti átváltási hajlandóságunk nem függ a többi kritérium szintjétől. Egy szemléletes példa: egy házibuli sikere a sör és a pizza mennyiségétől függ, de a kettő nem cserélhető fel tetszőlegesen: sok sör pizza nélkül nem ugyanannyit ér, mint a kettő kiegyensúlyozott aránya. Ilyenkor az additív modell félrevezet, és összetettebb (például multiplikatív) aggregálásra van szükség. A mérnöki gyakorlatban az additív modell jó közelítés, ha a kritériumok valóban különböző tulajdonságokat mérnek, és egyik sem szükséges feltétele a másik értékének. Ez utóbbiról a kizáró szűrés gondoskodik.

10.4.3. A súlyok meghatározása

A súlyok meghatározásának legegyszerűbb módja a közvetlen pontosztás: a döntéshozók 100 pontot osztanak szét a kritériumok között. Ha a döntéshozók a kritériumok sorrendjében könnyebben egyeznek meg, mint a pontos számokban, Edwards és Barron rangsoralapú súlyozást javasol (Edwards és Barron 1994). A **rangsor-súlypont** (rank order centroid, ROC) módszerben n kritérium esetén az i-edik helyen álló kritérium súlya

$$w_i = \frac{1}{n} \sum_{k=i}^{n} \frac{1}{k}. \tag{10.3}$$

Hat kritérium esetén a ROC-súlyok: 0,408; 0,242; 0,158; 0,103; 0,061; 0,028. A módszer előnye, hogy csak sorrendet kér, hátránya, hogy a sorrend elején álló kritériumokat erősen felülsúlyozza. Ha a döntéshozók szerint két kritérium között nincs nagy különbség, a ROC-súlyokat érdemes közvetlen egyeztetéssel finomítani.

KoboCella: a súlyozott értékelés

A KoboCella csapata a kizáró szűrés után megmaradt három koncepciót hat kritérium szerint értékelte (10.2. táblázat). A súlyokat a gyártásvezető, a minőségügyi vezető és a munkavédelmi felelős közösen, a csapattól függetlenül határozta meg. A ciklusidő és az utómunka-arány 20–20 pontot kapott, a többi kritérium 15–15-öt. A hasznossági függvények lineárisak voltak. A ciklusidőnél például a 90 s jelentette a 0, a 45 s a 10 pontot. A technikai kockázathoz szóbeli skála tartozott operatív definíciókkal (alacsony: 9, közepes: 6, magas: 2 pont).

10.2. táblázat. A KoboCella súlyozott értékelése. A pontszámok 0–10 skálán, a súlyozott összpontszám a súlyokkal képzett átlag.

Kritérium (súly)	Mérték	A: értéke	A: pont	B: értéke	B: pont	D: értéke	D: pont
K1 ciklusidő (20)	s	58,6	7,0	52	8,4	48	9,3
K2 utómunka-arány (20)	%	0,8	9,0	0,7	9,3	0,6	9,7
K3 törzshajlítás (15)	°	15	8,6	15	8,6	10	10,0
K4 beruházás (15)	MFt	78	3,5	95	2,1	110	0,8
K5 átállási idő (15)	perc	5	8,3	30	0,0	20	3,3
K6 technikai kockázat (15)	szóbeli	közepes	6,0	ala-csony	9,0	magas	2,0
súlyozott összpontszám			**7,16**		**6,50**		**6,22**

Az együttműködő cella (A) 7,16 ponttal az első, a fix állomás (B) 6,50 ponttal a második. A különbség 0,66 pont. Mivel a pontszámok minőségi ítéletekből származnak, egy skálalépésnél kisebb különbséget óvatosan kell értelmezni. Itt egy lépés 1 pont, a különbség tehát kisebb egy lépésnél. A csapat ezért érzékenységvizsgálatot végzett.

10.4.4. Érzékenységvizsgálat és a rangsor robusztussága

A döntéshozónak a javaslattal együtt azt is be kell mutatni, mennyire robusztus a rangsor, vagyis megváltoztathatná-e az alternatívák sorrendjét a bizonytalanságok csökkentése (Parnell és mtsai. 2013). Általában egyetlen alternatívát kell javasolni. Ha azonban az alternatívák nem különböznek számottevően, vagy a bizonytalanság csökkentése megváltoztathatná a rangsort, a javaslatnak minden szorosan rangsorolt alternatívát tartalmaznia kell. Ilyenkor érdemes megvizsgálni, megéri-e több adatot gyűjteni; ha ez sem segít, a kritériumokat és a súlyokat kell újragondolni.

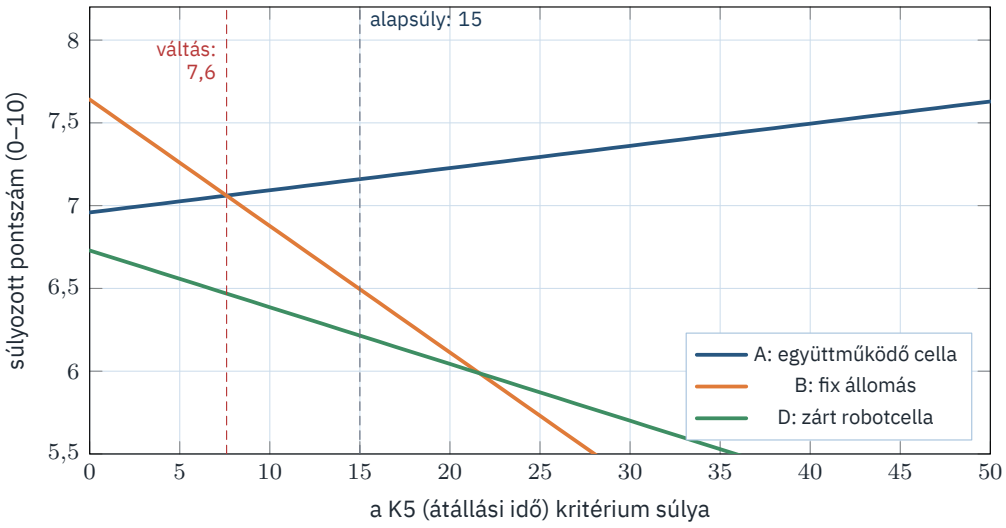
Az érzékenységvizsgálat két leggyakoribb formája:

- **Egytényezős vizsgálat:** egy kritérium súlyát nulla és egy felső határ között változtatjuk, a többi súlyt arányosan módosítjuk, hogy az összeg 100 maradjon, és megkeressük azt a súlyt, amelynél a rangsor megfordul.
- **Monte Carlo-szimuláció:** a súlyokat (és ha szükséges, a pontszámokat is) a bizonytalanságuknak megfelelő valószínűségi eloszlásból sokszor újramintavételezzük, és megszámoljuk, hányszor nyer az egyes alternatíva. Butler, Jia és Dyer módszertani tanulmánya a többszempontú döntési modellek ilyen szimulációs érzékenységvizsgálatának több változatát dolgozta ki (Butler, Jia, és Dyer 1997).

KoboCella: mennyire biztos az első hely?

Az egytényezős vizsgálat (10.3. ábra) megmutatta, hogy az együttműködő cella (A) első helye leginkább az átállási idő (K5) súlyán múlik. Ha ez a súly 15-ről 7,6 alá csökkenne, a fix állomás (B) kerülne az élre. A technikai kockázat (K6) súlyának 30,5 fölé, illetve a ciklusidő (K1) súlyának 43,3 fölé kellene emelkednie ahhoz, hogy a rangsor megforduljon. A csapat ezután Monte Carlo-szimulációt is végzett: 5000 alkalommal mintavételezte a súlyokat egy,

az alapsúlyok körül szóródó eloszlásból. Az A koncepció az esetek 92%-ában lett első, a B 7%-ában.
A K5 súlya tehát a döntés kulcsa. A csapat ezért visszament a gyártásvezetőhöz azzal a kérdéssel, mennyire biztos a változatváltások várható gyakorisága. A válasz az volt, hogy a Kobold két új motorváltozat bevezetését tervezi a következő két évben, így a gyakori átállás valószínű. A csapat az együttműködő cellát javasolta, és a javaslatban rögzítette, hogy a döntés érzékeny az átállási igényre: ha a termékpaletta mégsem bővül, a fix állomás versenyképes alternatíva lett volna.



10.3. ábra. A KoboCella súlyozott pontszámai az átállási idő (K5) súlyának függvényében. A többi súly arányosan változik. A rangsor 7,6-os súlynál fordul meg.

10.5. Páros összehasonlítás: az AHP

Az **analitikus hierarchia eljárás** (analytic hierarchy process, AHP) a súlyozott értékelés egy változata, amelyet Saaty dolgozott ki (Saaty 1980, 1990; Goldberg és mtsai. 1994, 2.1. szakasz). Páros összehasonlításokra épül, ezért akkor hasznos, ha szóbeli kifejezésekkel (egyforma, kissé fontosabb, sokkal fontosabb) könnyebb ítéletet mondani, mint számokkal.

Az AHP a döntési problémát hierarchiába rendezi: a csúcson a cél, alatta a kritériumok (szükség esetén alkritériumok), legalul az alternatívák (10.4. ábra). Minden szinten páronként hasonlítjuk össze az elemeket a felettük álló elem szempontjából. Saaty skáláján az 1 az egyforma fontosságot, a 3 a mérsékelt, az 5 az erős, a 7 a nagyon erős, a 9 a rendkívüli fontosságot jelenti, a páros számok a köztes értékek (Saaty 1990). Ha az i-edik elem a_{ij}-szer fontosabb a j-ediknél, akkor a j-edik $1/a_{ij}$-szer fontosabb az i-ediknél, így a páros összehasonlítási mátrix reciprok, és az átlójában egyesek állnak.

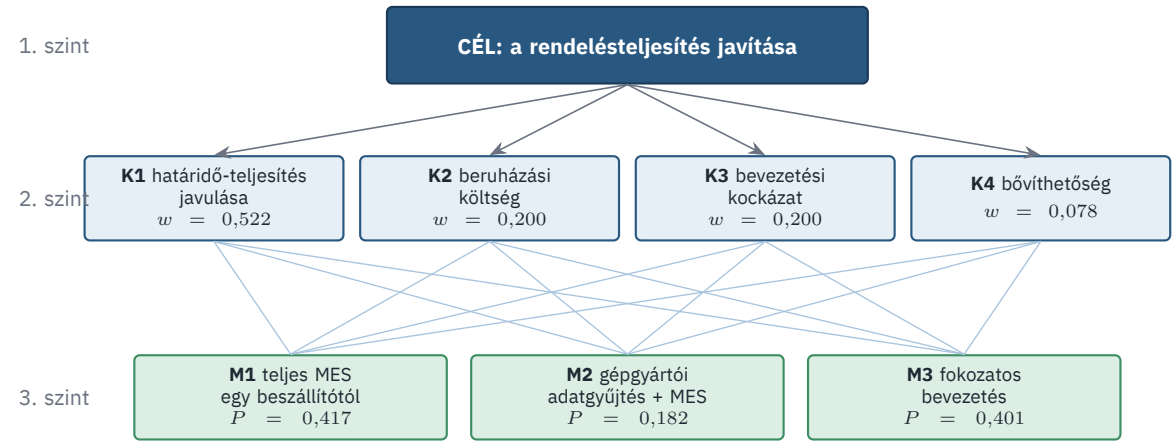
A súlyokat Saaty szerint a mátrix legnagyobb sajátértékéhez tartozó sajátvektor adja, amelyet úgy normálunk, hogy elemeinek összege 1 legyen (Saaty 1990). Gyakorlati közelítésként a sorok mértani közepeinek normált értéke is használható; jól következetes mátrixnál a két eredmény alig tér el. Az AHP nagy előnye, hogy ellenőrizni tudja az ítéletek **következetességét**. Ha az A kétszer fontosabb B-nél, B pedig kétszer fontosabb C-nél, akkor következetes ítéletnél A négyszer fontosabb C-nél. A következetlenség mértéke a következetességi index és a következetességi arány:

$$\mathrm{CI} = \frac{\lambda_{\max} - n}{n - 1}, \qquad \mathrm{CR} = \frac{\mathrm{CI}}{\mathrm{RI}_n}, \tag{10.4}$$

ahol $\lambda_{\max}$ a mátrix legnagyobb sajátértéke, n a mátrix mérete, RI_n pedig a véletlen mátrixok átlagos következetességi indexe. Saaty értékei: $n = 3$ esetén 0,58, $n = 4$ esetén 0,90, $n = 5$ esetén 1,12, $n = 6$ esetén 1,24, $n = 7$ esetén 1,32 (Saaty 1980). Saaty szerint a 0,10 alatti következetességi arány elfogadható; e fölött az ítéleteket felül kell vizsgálni (Saaty 1990).

ForgácsTech: a MES-koncepció AHP-val

A ForgácsTech csapata három MES-koncepciót hasonlított össze: M1 a teljes MES egy beszállítótól, M2 a gépgyártói adatgyűjtés és egy erre épülő MES, M3 a fokozatos bevezetés (első ütemben gépi adatgyűjtés, OEE-monitoring és ütemezőmodul, később bővítés). A négy kritériumot a vezetőség páronként hasonlította össze. A határidő-teljesítés javulását (K1) a beruházási költséghez (K2) és a bevezetési kockázathoz (K3) képest mérsékelten fontosabbnak (3), a bővíthetőséghez (K4) képest erősen fontosabbnak (5) ítélte. A költséget és a kockázatot egyformán fontosnak (1), mindkettőt a bővíthetőségnél mérsékelten fontosabbnak (3) tartotta.
A sajátvektor-módszer szerint a súlyok: K1 0,522, K2 0,200, K3 0,200, K4 0,078. A legnagyobb sajátérték 4,043, így $\mathrm{CI} = 0{,}014$ és $\mathrm{CR} = 0{,}016$, ami jóval a 0,10-es határ alatt van. Az alternatívák kritériumonkénti páros összehasonlítása után a globális prioritások: M1 0,417, M2 0,182, M3 0,401 (10.4. ábra). Az M1 és az M3 közötti különbség alig 4%, ami az ítéletek bizonytalanságához képest nem tekinthető érdemi különbségnek. Az AHP tehát az M2-t egyértelműen kiszűrte, az M1 és az M3 között azonban nem döntött. A csapat ezért költség–haszon elemzést is végzett (10.7. szakasz).



10.4. ábra. A ForgácsTech MES-döntésének AHP-hierarchiája a kritériumsúlyokkal és az alternatívák globális prioritásaival.

Az AHP-t széles körben használják, de kritikusai is vannak. Belton és Gear mutatott rá elsőként, hogy egy új, az eddigiek egyikével azonos alternatíva hozzáadása megfordíthatja a korábbi alternatívák rangsorát (rangsorfordulás) (Belton és Gear 1983). A jelenség oka a prioritások normálása. A gyakorlatban ezért érdemes az alternatívák listáját az AHP előtt véglegesíteni, és az eredményt más módszerrel is ellenőrizni.

10.6. Dominancia és Pareto-front

A súlyozott értékelés és az AHP egyetlen számmá aggregálja a kritériumokat. Ehhez súlyokat kell meghatározni, ami értékítélet. A **dominanciavizsgálat** súlyok nélkül is szűkíti az alternatívák körét. Maximalizálandó kritériumok esetén egy alternatíva akkor dominál egy másikat, ha minden jellemzőben legalább olyan jó, és legalább egyben jobb. A dominált alternatívát nyugodtan elvethetjük, mert van nála minden szempontból legalább olyan jó lehetőség.

A dominancia páros összehasonlítással vizsgálható. Az eredmény a **dominanciamátrix**: a sorok és oszlopok az alternatívák, a j-edik sor k-adik eleme 1, ha a j-edik alternatíva dominálja a k-adikat. A j-edik sor összege azt mutatja meg, hány másik alternatívát dominál a j-edik, a k-adik oszlop összege pedig azt, hány másik alternatíva dominálja a k-adikat. Azok az alternatívák, amelyek oszlopösszege nulla, a **nem dominált** megoldások.

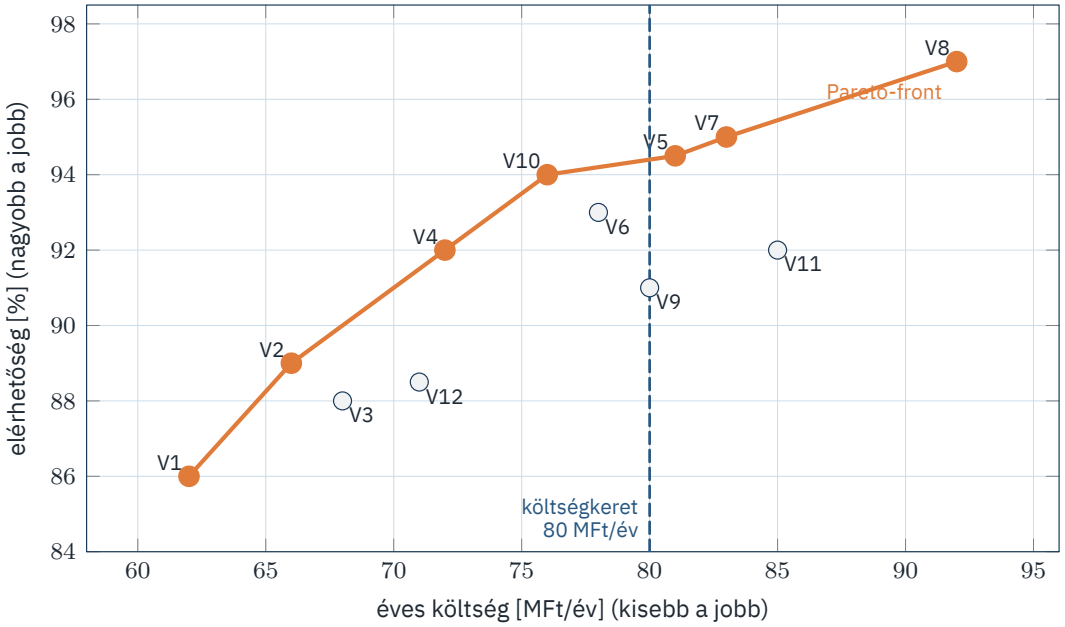
A nem dominált megoldások halmaza a **Pareto-front**. Egy fontos finomság: minden Pareto-optimális pont nem dominált, de nem minden nem dominált pont Pareto-optimális. Ha csak néhány tervváltozatot vizsgálunk, a közöttük nem dominált változatok csak *közelítik* a valódi Pareto-frontot, amely a teljes tervezési tér nem dominált pontjaiból áll. A teljes front közelítésére a többcélú evolúciós algoritmusok alkalmasak (9.8. szakasz).

A Pareto-front a kompromisszumokat teszi láthatóvá. Egy bank telefonos ügyfélszolgálata például a gyors kiszolgálás és a munkatermelékenység között választ: nagyon rövid várakozási időt csak gyakori üresjárattal lehet elérni, magas kihasználtságot csak hosszabb várakozással. Az iparági front megmutatja, mely szolgáltatók működnek hatékonyan, és melyek vannak a front mögött, vagyis melyek javíthatnának mindkét szempontból egyszerre. A rendszermérnök feladata kettős: a front mögötti hatékonytalanságok megszüntetése, és új koncepciókkal magának a frontnak az eltolása.

> **VárosiBringa: kerékpárok, dokkolók, furgonok**
>
> A VárosiBringa csapata tizenkét tervváltozatot vizsgált, amelyek a kerékpárok számában (250, 300, 350), a dokkolóállomások számában (25, 30, 35) és az áthelyezést végző furgonok számában (1 vagy 2) különböztek. Minden változatra szimulációval becsülte az éves teljes költséget és az elérhetőséget, vagyis azt, hogy a kölcsönzési kérések hány százalékában van kerékpár 300 méteren belül (10.5. ábra).
> A dominanciavizsgálat öt változatot kiszűrt. A legtanulságosabb a V6 (350 kerékpár, 30 dokkoló, 1 furgon) esete volt: ezt a V10 (300 kerékpár, 35 dokkoló, 1 furgon) dominálta, amely olcsóbb (76 MFt/év a 78 helyett) és jobb elérhetőségű (94% a 93 helyett). Vagyis a kerékpárok számának növelése helyett az állomássűrűség növelése a hatékonyabb út, ami egybevág a minőségház eredményével, ahol az állomássűrűség holtversenyben az első helyen végzett, és a Gyakorlati útmutató QFD-notebookjának érzékenységvizsgálata szerint ez a jellemző tartotta meg legstabilabban az élmezőnyben elfoglalt helyét (7.3. szakasz). A Pareto-frontra hét változat került. A 80 MFt/éves költségkeret alatt a V10 kínálta a legjobb elérhetőséget. Az eredeti terv (V4) a fronton van, de a V10 évi 4 MFt többletköltségért 2 százalékpontnyi elérhetőség-javulást ad. A választás a két változat között már értékítélet, amelyet a város döntéshozóinak kell meghozniuk.

10.7. Költség–haszon elemzés

A **költség–haszon elemzés** (benefit–cost analysis) az alternatívákat a várható jövőbeli hasznok és költségek arányával hasonlítja össze. A hasznokat és a költségeket is jelenér-



10.5. ábra. A VárosiBringa tervváltozatai a költség–elérhetőség síkon. A narancs pontok a nem dominált változatok, a vonal a közelítő Pareto-front.

téken kell kifejezni, hogy a különböző időpontokban esedékes összegek összevethetők legyenek. Boardman és szerzőtársai tankönyve az elemzést a következő lépésekre bontja (Boardman és mtsai. 2018; vö. Goldberg és mtsai. 1994, 2.2. szakasz):

1. az egymást kizáró alternatívák meghatározása;
2. annak eldöntése, kinek a hasznai és költségei számítanak (például csak a vállalaté, vagy a társadalomé is);
3. a hatások azonosítása és mérőszámaik kiválasztása (beruházás, üzemeltetés, karbantartás, maradványérték, hasznok és hátrányok);
4. a hatások mennyiségi előrejelzése a rendszer teljes élettartamára;
5. a hatások pénzben kifejezése;
6. a jövőbeli összegek diszkontálása jelenértékre;
7. a nettó jelenérték (és szükség esetén a haszon–költség arány) kiszámítása;
8. érzékenységvizsgálat;
9. javaslat.

Egy t-edik évben esedékes X_t összeg jelenértéke r kamatláb mellett $X_t/(1+r)^t$. Ha egy összeg T éven át minden évben esedékes, jelenértéke

$$\mathrm{PV} = X \sum_{t=1}^{T} \frac{1}{(1+r)^t}. \tag{10.5}$$

A bevezetés kockázata és a műszaki érettség egy f kockázati szorzóval vehető figyelembe a hasznok oldalán. Mivel a szorzó szubjektív becslés, érdemes kevés, durva fokozatot használni (például 0,5; 0,75; 1), mert a finomabb megkülönböztetés csak látszólagos pontosságot adna.

10.7.1. A növekményes elemzés

Ha sem a költségkeret, sem a kívánt haszon nincs rögzítve, a legnagyobb B/C arányú alternatíva nem feltétlenül a legjobb választás. Ilyenkor **növekményes elemzést** kell végezni (Sullivan, Wicks, és Koelling 2019): először kizárjuk az 1-nél kisebb B/C arányú alternatívákat, a maradékot növekvő költség szerint sorba rendezzük, majd minden szomszédos párra kiszámítjuk a többlethaszon és a többletköltség arányát ($\Delta B/\Delta C$). Egy költségnövekmény akkor éri meg, ha $\Delta B/\Delta C > 1$. A preferált alternatíva a sorrendben az utolsó olyan, amelynek növekménye még megéri. A logika egyszerű: a drágább alternatíva csak akkor jobb, ha a ráfordított *többlet* pénz is legalább annyi többlethasznot hoz.

ForgácsTech: költség–haszon elemzés

A ForgácsTech csapata az AHP után költség–haszon elemzést végzett a három MES-koncepcióra, ötéves időtávra és 8%-os kamatlábbal. A hasznokat a késedelmi kötbérek, a sürgős átállások és a két legnagyobb vevő elvesztésének kockázatából becsülte, a bevezetési kockázatot kockázati szorzóval vette figyelembe. Az ötéves annuitástényező 3,993.

10.3. táblázat. A ForgácsTech MES-koncepcióinak költség–haszon elemzése (5 év, 8%).

	M1: teljes MES	M2: gépgyártói + MES	M3: fokozatos
beruházás [MFt]	42	30	14
éves működtetés [MFt/év]	4	6	3
éves haszon [MFt/év]	28	22	15
kockázati szorzó (f)	0,75	0,5	1,0
költségek jelenértéke, C [MFt]	58,0	54,0	26,0
hasznok jelenértéke, B [MFt]	83,8	43,9	59,9
B/C	1,45	0,81	2,31
nettó jelenérték, B − C [MFt]	25,9	−10,0	33,9

Az M2 B/C aránya 1 alatt van, ezért kiesett (10.3. táblázat). Az M3-ról az M1-re való átlépés növekménye $\Delta B = 24{,}0$ MFt, $\Delta C = 32{,}0$ MFt, így $\Delta B/\Delta C = 0{,}75 < 1$: a többletberuházás nem éri meg. A költség–haszon elemzés tehát az M3-at, a fokozatos bevezetést preferálta, amelynek a nettó jelenértéke is a legnagyobb.
A két módszer eredményét összevetve a csapat arra jutott, hogy az M3 gyengesége a bővíthetőség (az AHP-ban ebben volt a leggyengébb), erőssége pedig a kis kockázat és a gyors megtérülés. A végső javaslat ezért az M3 volt, egy kiegészítő feltétellel: az első ütem eszközeit olyan beszállítótól kell beszerezni, amelynek rendszere a teljes MES-funkcionalitásig bővíthető. Ezzel a csapat megőrizte az egyetlen interfészgazda előnyét, amelyet az architektúraelemzés feltárt (8.7. szakasz), és a beruházás első üteme jóval a 45 millió forintos keret alatt maradt.

A költség–haszon elemzés legnagyobb előnye, hogy a teljes életciklust figyelembe veszi, és dokumentált. Korlátai a „forintosíthatóság” problémájában foglalhatók össze: az elemzés félrevezető, ha az alternatívák listája nem teljes, ha a költségbecslések pontatlanok, és ha a gyakran nehezen megfogható hasznokat nem lehet hitelesen pénzben kifejezni (Boardman és mtsai. 2018). A biztonság, az ergonómia vagy a környezeti hatás pénzbeli értékelése különösen vitatott. Ezeket gyakran célszerűbb kötelező követelményként vagy külön kritériumként kezelni.

10.8. Csoportos döntés

A rendszermérnöki döntéseket ritkán hozza egyetlen ember. A csoportos döntésnek azonban van egy meglepő buktatója: a csoport tagjai egyenként következetesek és racionálisak lehetnek, a csoport összesített preferenciája mégsem az.

Tegyük fel, hogy a KoboCella három csapattagja a három megmaradt koncepciót így rangsorolja:

- első tag: A > B > D;
- második tag: B > D > A;
- harmadik tag: D > A > B.

Többségi szavazással az A-t a B-vel szemben ketten preferálják (az első és a harmadik), a B-t a D-vel szemben szintén ketten (az első és a második), a D-t az A-val szemben szintén ketten (a második és a harmadik). A csoport preferenciája tehát körbeér: A > B > D > A. Ezt a jelenséget Condorcet márki írta le a 18. században. Arrow később általánosan bizonyította, hogy nincs olyan, az egyéni rangsorokat csoportrangsorrá összesítő szabály, amely egyszerre teljesítene néhány, ésszerűnek tűnő követelményt (többek között a Pareto-elvet, a független alternatívák függetlenségét és a diktátor hiányát) (Arrow 1951).

A rendszermérnöki gyakorlat számára ebből nem az következik, hogy a csoportos döntés reménytelen, hanem az, hogy a rangsorok egyszerű összeszavazása helyett rendszerezett eljárás kell. A bemutatott módszerek éppen ezt teszik: szétválasztják a ténybeli ítéleteket (az alternatívák pontszámait, amelyeket a szakértők adnak) és az értékítéleteket (a kritériumok súlyait, amelyeket a döntéshozók adnak), és mindkettőt átláthatóvá teszik. Ha a vita a súlyokról szól, az érzékenységvizsgálat megmutatja, hogy a vitának van-e egyáltalán jelentősége a döntés szempontjából. A döntéshozó ugyanakkor mindig szabadon választhat bármelyik alternatívát, vagy kérheti további alternatívák értékelését: az elemzés a döntést támogatja, nem helyettesíti.

10.9. A módszerek összevetése és a döntés dokumentálása

A módszerek a fejlesztés menetében egymásra épülnek: a koncepcionális tervezés közben a Pugh-módszer és, ha elérhetők, egyszerű matematikai modellek; az előzetes tervezés során a hasznossági vizsgálat és a dominanciavizsgálat, minden koncepcióhoz több tervváltozattal. 10.4. táblázat a fejezetben tárgyalt módszereket veti össze.

10.4. táblázat. A koncepció-kiválasztás módszereinek összevetése.

Módszer	Bemenet	Kimenet	Mikor használd?	Fő korlát
kizáró szűrés	kötelező követelmények	megfelelő/-nem megfelelő	mindig, elsőként	csak kizár, nem rangsorol
Pugh-módszer	minőségi összehasonlítás egy alaphoz	+/S/− mintázat, hibrid ötletek	korai koncepció szakasz, sok koncepció	a számok csak tájékoztatók
súlyozott értékelés	súlyok, hasznossági függvények, becslések	összpont-szám, rangsor	jól definiált alternatívák, mérhető kritériumok	a súlyok és a függvények szubjektívek

Módszer	Bemenet	Kimenet	Mikor használd?	Fő korlát
AHP	páros szóbeli ítéletek	súlyok, prioritások, CR	szubjektív kritériumok, kevés elem	sok összehasonlítás, rangsorfordulás
Pareto-front	két-három mennyiségi cél	nem dominált változatok	sok tervváltozat, nincs megállapodás a súlyokról	nem választ a fronton belül
költség–haszon	pénzben kifejezett költségek, hasznok	B/C, nettó jelenérték, növekmény	beruházási döntések, életciklus-szemlélet	a hasznok forintosítása

A döntést dokumentálni kell; az ISO/IEC/IEEE 15288 is előírja, hogy a döntést, annak indoklását és feltevéseit rögzíteni kell (ISO/IEC/IEEE 2023, 6.3.3. szakasz; NASA 2016, 6.8. szakasz). Egy jó döntési jelentés tartalmazza a döntés rendszerszintű összefüggéseit, a szükséges döntést és a kívánt eredményt, a döntési kritériumokat, az azonosított alternatívákat, az alkalmazott értékelési módszereket és eszközöket, a feltevéseket, bizonytalanságokat és érzékenységeket, az összes alternatíva értékelésének eredményét, a javaslatokat, a meghozott végső döntést az indoklással együtt, valamint a tanulságokat. A dokumentált döntés évekkel később is érthetővé teszi, miért alakult így a rendszer, és lehetővé teszi, hogy ha a feltevések megváltoznak, a döntést tudatosan felül lehessen vizsgálni.

A súlyok utólagos hangolása

Gyakori, de súlyos hiba, hogy a csapat már a kiértékelés előtt tudja, melyik koncepciót szeretné, és addig hangolja a súlyokat vagy a pontszámokat, amíg a kívánt eredmény ki nem jön. Az így készült értékelés hamis tudományosságot kölcsönöz egy előre meghozott döntésnek. Ha nem a legmagasabb pontszámú alternatívát javasoljuk, meg kell indokolni, miért. Ha a kisebb pontszámú alternatíva előnyei mégis meghaladják a nagyobbét, akkor az értékelés valamit nem ragadott meg: ilyenkor nem a súlyokat kell csendben módosítani, hanem a kritériumokat és azok definícióit kell a döntéshozó egyetértésével újragondolni. A súlyokat ezért a pontozás előtt kell rögzíteni, lehetőleg a csapattól függetlenül.

Összefoglalás

- A döntés elemei az alternatívák, a kritériumok, az értékítélet és a döntéshozók. Formális döntéselemzés akkor indokolt, ha a döntés összetett, bizonytalan, sok szempontú, és sokféle érdekelt felet érint.
- Az emberi ítéletalkotás rendszeresen torzul (reprezentativitás, hozzáférhetőség, lehorgonyzás), és a döntési helyzet kialakítása, például az alapértelmezés, maga is befolyásolja az eredményt.
- A kritériumok legyenek mérhetők, megkülönböztetők, függetlenek és a követelményekhez kötöttek. A kötelező kritériumot nem teljesítő alternatívát kizáró szűréssel el kell vetni.
- A Pugh-módszer egy vonatkoztatási alaphoz hasonlít (+/S/–), szűr és hibrid koncepciókat hoz létre, de nem dönt helyettünk.
- A súlyozott értékelés hasznossági függvényekkel közös skálára hozza a kritériumokat, és súlyozott összeget képez. Az additív modell függetlenségi feltételeket feltételez. A súlyokat a döntésre jogosultak rögzítik, a pontozás előtt.
- Az érzékenységvizsgálat (egytényezős vagy Monte Carlo) megmutatja, mennyire robusztus a rangsor. Szoros eredménynél minden közeli alternatívát be kell mutatni a

döntéshozónak.

- Az AHP páros szóbeli ítéletekből számít súlyokat és prioritásokat, és ellenőrzi a következetességet (CR $< 0{,}10$). A dominanciavizsgálat súlyok nélkül szűkít; a nem dominált változatok a Pareto-frontot közelítik. A költség–haszon elemzés jelenértéken veti össze a hasznokat és a költségeket; ha a keret nem rögzített, növekményes elemzés kell.
- A csoport preferenciája akkor is lehet következetlen, ha a tagoké nem az (Condorcet-paradoxon, Arrow-tétel). Ezért kell a ténybeli ítéleteket és az értékítéleteket szétválasztani, és a döntést dokumentálni.

Önellenőrzés

1. Mi a különbség a kötelező és a javító kritérium között? Mi a teendő azzal az alternatívával, amely egy kötelező kritériumot nem teljesít?
2. Miért nem szabad a Pugh-összehasonlítás előjeles összegét végeredménynek tekinteni? Mire használnád inkább?
3. Egy kritérium „rossz” szintje 120 MFt, „jó” szintje 0 MFt. Mekkora a 78 MFt-os alternatíva lineáris hasznossági pontszáma 0–10 skálán?
4. Számítsd ki a ROC-súlyokat négy kritérium esetén!
5. Egy 3 × 3-as páros összehasonlítási mátrix legnagyobb sajátértéke 3,08. Elfogadható-e a következetesség?
6. Négy változat (költség, teljesítmény) értékei: P (10, 70), Q (12, 80), R (14, 75), S (15, 90). A költség minimalizálandó, a teljesítmény maximalizálandó. Melyik változat dominált, és melyik melyiket dominálja?
7. Két alternatíva: X költsége 40, haszna 60; Y költsége 70, haszna 85 (jelenértéken). Melyiket választanád növekményes elemzéssel, ha a keret nincs rögzítve?
8. Mutasd meg egy példán, hogy három, egyenként következetes rangsor többségi összesítése körbeérő csoportpreferenciát adhat!

Megoldás.

1. A kötelező kritérium („kell”) nem teljesítése kizáró ok, a javító kritérium az alternatívák rangsorolását szolgálja. A kötelező kritériumot nem teljesítő alternatívát indoklással el kell vetni, de a kizárást dokumentálni kell.
2. Mert a pluszok és mínuszok nem azonos súlyúak és nem azonos nagyságú különbségeket jeleznek, így összegük nem mérhető mennyiség. A Pugh-összehasonlítás a gyenge koncepciók kiszűrésére, az erősségek és gyengeségek mintázatának feltárására és hibrid koncepciók ötletének előállítására való.
3. $u = 10 \cdot (78 - 120)/(0 - 120) = 10 \cdot 0{,}35 = 3{,}5$.
4. $w_1 = (1 + 1/2 + 1/3 + 1/4)/4 = 0{,}521$; $w_2 = (1/2 + 1/3 + 1/4)/4 = 0{,}271$; $w_3 = (1/3 + 1/4)/4 = 0{,}146$; $w_4 = (1/4)/4 = 0{,}063$.
5. CI $= (3{,}08 - 3)/2 = 0{,}04$; CR $= 0{,}04/0{,}58 \approx 0{,}069 < 0{,}10$, tehát elfogadható.
6. Az R dominált: a Q olcsóbb (12 < 14) és jobb teljesítményű (80 > 75). A P, a Q és az S nem dominált: a P a legolcsóbb, az S a legjobb teljesítményű, a Q-nál pedig nincs olyan változat, amely olcsóbb és jobb is lenne.
7. Mindkettő B/C aránya 1 fölött van (1,5 és 1,21). A növekmény: $\Delta B/\Delta C = (85 - 60)/(70 - 40) = 0{,}83 < 1$, tehát a többletköltség nem éri meg: az X a preferált.
8. Például: első tag A > B > C, második B > C > A, harmadik C > A > B. Az A-t a B-vel szemben az első és a harmadik, a B-t a C-vel szemben az első és a második, a C-t az A-val szemben a második és a harmadik preferálja. A többségi preferencia: A > B > C > A.

Projektlépés: a koncepció kiválasztása

Saját fejlesztési feladatodhoz:

1. sorold fel a kötelező kritériumokat, és végezz kizáró szűrést a koncepcióidon, a kizárások indoklásával;
2. készíts Pugh-összehasonlítást legalább három koncepcióval, majd egy második kört új vonatkoztatási alappal;
3. készíts súlyozott értékelést legalább öt kritériummal; a súlyokat egy csapattársad vagy

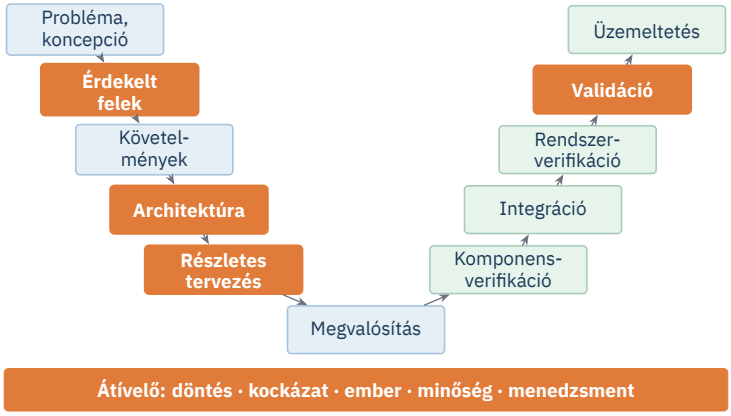
a megrendelő határozza meg a pontozás előtt;
4. végezz egytényezős érzékenységvizsgálatot a két legfontosabb kritérium súlyára, és határozd meg a rangsorfordulás küszöbét;
5. ha van legalább két mennyiségi célod és legalább hat tervváltozatod, határozd meg a nem dominált változatokat;
6. írj egyoldalas döntési jelentést a 10.9. szakaszban leírt szerkezetben.

A Pugh-módszer, a súlyozott értékelés, a MAUT, az AHP, a Pareto-front és a költség–haszon elemzés sablonja és notebookja a Gyakorlati útmutatóban található.

Továbbolvasás

- Parnell és mtsai. (2013): a döntéselemzés gyakorlati kézikönyve a célok, a kritériumok és a bizonytalanság kezeléséről.
- Boardman és mtsai. (2018): a költség–haszon elemzés részletes, alkalmazott tankönyve.
- Keeney és Raiffa (1976): a többszempontú hasznosságelmélet alapműve.
- Saaty (1990): az AHP rövid, olvasmányos bemutatása a szerzőtől.
- Pugh (1991): a Pugh-módszer eredeti leírása a teljes terméktervezési folyamat részeként.
- Tversky és Kahneman (1974): az ítéletalkotási heurisztikák és torzítások klasszikus tanulmánya.
- NASA (2016), 6.8. szakasz: a döntéselemzési folyamat, a kritériumok, a robusztusság és a dokumentálás.
- Goldberg és mtsai. (1994), 2.1–2.2. szakasz: a súlyozott értékelés, az AHP és a költség–haszon elemzés részletes eljárása kidolgozott példákkal.

11. Emberközpontú tervezés és emberi rendszerintegráció (HSI)



Tanulási eredmények

A fejezet feldolgozása után képes leszel:

- megindokolni, miért az ember a rendszer egyik eleme, és mit jelent az emberi rendszerintegráció (HSI);
- leírni a HSI-terv tartalmát és a HSI-tevékenységek helyét a V-modellben;
- alkalmazni az ISO 9241-210 emberközpontú tervezési ciklusát, és meghatározni a használhatóságot az ISO 9241-11 szerint;
- figyelembe venni az ember képességeit és korlátait: az észlelést, a figyelmet, a helyzettudatosságot, a testméreteket és a testtartást;
- hierarchikus feladatelemzést (HTA) készíteni;
- megtervezni a használhatóság és a terhelés mérését (SUS, NASA-TLX).

Kezdjük egy kellemetlen tapasztalattal. A hagyományos rendszermérnöki folyamatban a működési követelményeket műszaki teljesítménymértékekben rögzítjük, és bár ezek közül sok az emberre is vonatkozik, az emberi szempontok gyakran alacsony prioritást kapnak. A következmény ismerős. A gép jól működik, de a kezelője nem tudja jól működtetni; a szoftver minden funkciót tud, de a felhasználó nem találja; a cella teljesíti a ciklusidőt, de a kezelő a műszak végére elfárad, és hibázik.

Ez a fejezet arról szól, hogyan kerül az ember a rendszertervezés középpontjába. A következő két fejezet erre épül: a 12. fejezet azt tárgyalja, mit végezzen az ember és mit a gép, a 13. fejezet pedig azt, hogyan tervezzünk az emberi hibák ellen.

A fejezet négy kérdésre épül:

- Miért része az ember a rendszernek, és mit jelent ez a rendszermérnöki munkában?
- Milyen folyamattal érhető el, hogy a rendszer a felhasználói számára is jól használható legyen?
- Milyen emberi képességekkel és korlátokkal kell tervezni?

- Hogyan elemezzük a feladatokat, és hogyan mérjük a használhatóságot és a terhelést?

11.1. Az ember mint rendszerelem

A rendszerszemléletet a rendszer minden elemére alkalmazni kell, az emberre is. Az ISO/IEC/IEEE 15288 szabvány rendszerdefiníciója ezt egyértelművé teszi: a rendszer elemei között a hardver, a szoftver, az adatok, a folyamatok, az eljárások és a létesítmények mellett az **ember** is szerepel (ISO/IEC/IEEE 2023). A rendszermérnöki megközelítésnek ezért a hardvert, a szoftvert és az embert egyenrangúan kell kezelnie és integrálnia: a rendszer a szükséges képességet a három elem együttműködésével hozza létre.

Az emberre fordított figyelem gazdasági kérdés is. A 2.1. szakaszban láttuk, hogy a koncepció és a tervezés döntései az életciklusköltség nagy részét már elkötelezik (INCOSE 2015). Az üzemeltetés költségeinek jelentős része pedig emberi (Booher 2003): a kezelők, karbantartók, oktatók munkája, a betanítás, a hibák következményei. Ha a tervezés nem számol az emberrel, ezek a költségek az életciklus későbbi szakaszában, a legdrágább helyen jelentkeznek.

Az Európai Bizottság Ipar 5.0-koncepciója az emberközpontúságot az ipari átalakulás egyik alappillérévé tette: a technológia az embert szolgálja, és nem fordítva (Breque, De Nul, és Petridis 2021). A rendszermérnök számára ez nem csak érték, hanem tervezési feladat. Az ember képességeit, korlátait és igényeit ugyanolyan gondossággal kell modellezni, követelményekké fordítani és verifikálni, mint a gép jellemzőit.

11.2. Emberi rendszerintegráció (HSI)

Az INCOSE meghatározása szerint az **emberi rendszerintegráció** (human systems integration, HSI) interdiszciplináris műszaki és vezetési folyamat, amely az emberi szempontokat a rendszer minden elemével és minden elemén át integrálja, és a rendszermérnöki gyakorlat nélkülözhetetlen része (INCOSE 2023b; NASA 2015). Booher összefoglalása szerint a HSI célja hármas (Booher 2003):

- a rendszer emberi elemei jól illeszkedjenek a hardver- és szoftverelemekhez, és a rendszer egésze a lehető legjobb teljesítményt nyújtsa;
- a fejlesztéshez és az üzemeltetéshez szükséges emberi erőforrás megjelenjen az életciklusköltség tervezésében, és a költség a lehető legkisebb legyen;
- a rendszer illeszkedjen azoknak a felhasználóknak a jellemzőihez, akik üzemeltetik, karbantartják és támogatják.

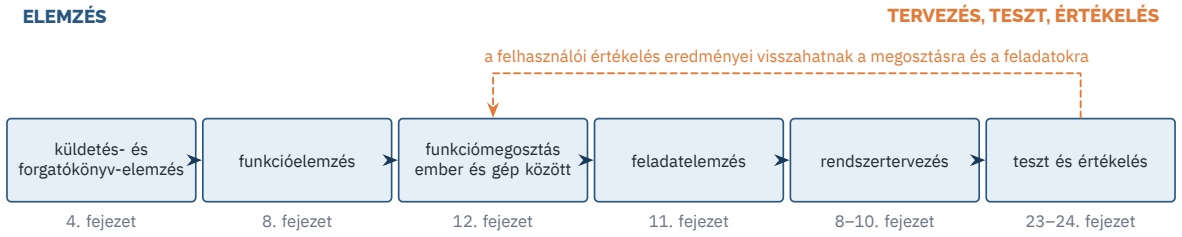
A HSI-t az emberrel kapcsolatos szakterületek (tartományok) összehangolásaként szokás leírni. A HSI kézikönyvei a tartományokat eltérő bontásban sorolják fel, de jellemzően ide tartozik a létszám, a szükséges szakképzettség, a képzés, az emberi tényezők mérnöki tervezése (ergonómia, kezelőfelületek), a munkabiztonság és egészségvédelem, valamint a munkakörnyezet (Booher 2003). Jó kiindulópont a korábbi, hasonló rendszerek tapasztalatainak összegyűjtése: azoknak az eseteknek a feltárása, amikor a létszám, a szakképzettség, a képzés, a kezelői idő, a karbantartás vagy a biztonsági kockázatok meghaladták a várakozásokat.

11.2.1. A HSI-terv

Nagyobb projektekben az emberi rendszerintegráció stratégiáját és tervezett megvalósítását külön HSI-tervben érdemes rögzíteni. A terv tartalmazza a HSI-tartományok közötti integráció szerepeit és felelősségeit, a HSI céljait és munkatermékeit életciklus-szakaszonként, a felülvizsgálatok belépési és kilépési feltételeit mérőszámokkal, a tervezett módszereket és eszközöket, a HSI-kockázatok azonosításának és kezelésének módját, valamint az összhangot a rendszermérnöki tervvel. A HSI nagyrészt a rendszermérnökség meglévő eszközeire épül: a működési koncepcióra, a funkciók egyértelmű elosztására a hardver, a szoftver és az ember között, és az életciklus során követett teljesítménymérő-számokra. A terv célja tehát nem a többi terv megismétlése, hanem az, hogy az emberi elem ugyanakkora figyelmet kapjon, mint a hardver és a szoftver.

11.2.2. A HSI a V-modellben

Az emberi tényezők mérnöki tervezésének klasszikus folyamata, amelyet az amerikai védelmi szabványosítás is rögzített, hat lépésben foglalható össze (11.1. ábra) (U.S. Department of Defense 2011): küldetés- és forgatókönyv-elemzés, funkcióelemzés, funkciómegosztás, feladatelemzés, rendszertervezés, valamint teszt és értékelés. A lépések nem új, különálló folyamatot alkotnak, hanem a V-modell már ismert lépéseinek emberi nézőpontját adják. A forgatókönyvek a működési koncepcióból jönnek (4.4. szakasz), a funkcióelemzés az architektúra része (8.3. szakasz), a funkciómegosztás a 12. fejezet tárgya, a teszt és az értékelés pedig a verifikáció és a validáció része.



11.1. ábra. Az emberi tényezők mérnöki tervezésének egyszerűsített folyamata a MIL-STD-46855A (U.S. Department of Defense 2011) nyomán, a könyv fejezeteihez rendelve.

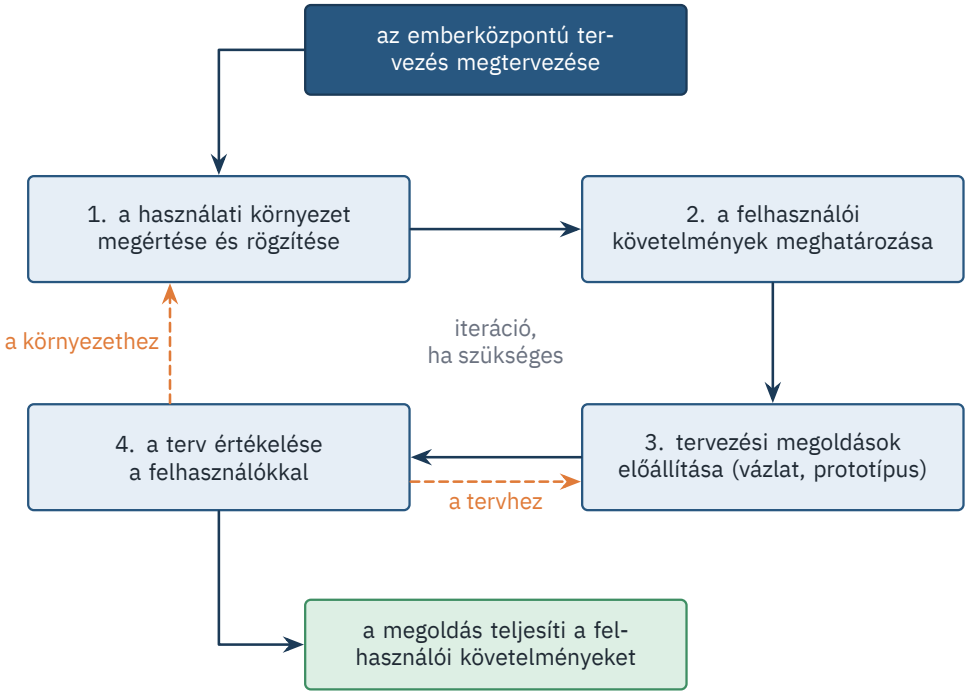
Részletesebben is érdemes végiggondolni, milyen emberközpontú tevékenységek kísérik a fejlesztés egyes szakaszait. Az előzetes tervezésben a szükséglet- és feladatelemzés, a forgatókönyvek és a látványtervek (storyboard), a teljesítmény- és használhatósági követelmények, valamint a létszám- és költségtervezés. A részletes tervezésben a felhasználói felület tervei és prototípusai, a munkamenet-elemzés, a súgó és a dokumentáció. A megvalósítás és az integráció során a prototípusok használhatósági értékelése és a heurisztikus felülvizsgálat. Az átadáskor a képzés, a felhasználói támogatás és a helyszíni értékelés, amelynek tapasztalatai a következő változat koncepciójába táplálódnak vissza.

11.3. Az emberközpontú tervezés folyamata

Az interaktív rendszerek emberközpontú tervezésének nemzetközi szabványa az ISO 9241-210 (ISO 2019). A szabvány hat alapelvet fogalmaz meg:

1. a tervezés a felhasználók, a feladatok és a környezet kifejezett megértésén alapul;
2. a felhasználók a tervezés és a fejlesztés során végig részt vesznek;
3. a tervezést felhasználóközpontú értékelés vezérli és finomítja;
4. a folyamat iteratív;
5. a tervezés a teljes felhasználói élményt figyelembe veszi;
6. a tervezőcsapat multidiszciplináris készségeket és nézőpontokat egyesít.

A szabvány négy, egymásra épülő tevékenységet ír elő (11.2. ábra): a használati környezet megértését és rögzítését, a felhasználói követelmények meghatározását, a tervezési megoldások előállítását és a terv értékelését. Ha az értékelés szerint a megoldás nem teljesíti a követelményeket, a ciklus megismétlődik: vissza kell lépni a környezet megértéséhez, a követelményekhez vagy a tervhez.



11.2. ábra. Az emberközpontú tervezés tevékenységei és iterációja az ISO 9241-210 alapján.

A ciklus kulcsfogalma a **használati környezet** (context of use): kik a felhasználók, milyen céljaik és feladataik vannak, milyen erőforrásokkal, milyen fizikai, társadalmi és szervezeti környezetben dolgoznak. Ugyanaz a termék más környezetben másképp használható: egy kerékpárkölcsönző alkalmazás a lakásban, nyugodt körülmények között könnyen kezelhető, esőben, kesztyűben, egy forgalmas buszmegállóban viszont már egészen más feltételek között kell működnie.

Az ISO 9241-11 szerint a **használhatóság** az a mérték, amennyire egy rendszert, terméket vagy szolgáltatást meghatározott felhasználók meghatározott céljaik elérésére eredményesen, hatékonyan és megelégedéssel használhatnak egy meghatározott használati környezetben (ISO 2018c). A definíció három mérhető összetevője:

- **eredményesség:** a felhasználók elérik-e a céljukat, és milyen pontossággal, teljességgel (például a feladatok hány százalékát fejezik be hibátlanul);
- **hatékonyság:** milyen ráfordítással (idő, lépésszám, erőfeszítés) érik el;
- **megelégedettség:** hogyan élik meg a használatot (például kérdőívvel mérve).

A definíció legfontosabb üzenete a „meghatározott” szó háromszori ismétlése. A használhatóság nem a termék önálló tulajdonsága, hanem a termék, a felhasználó, a cél és a környezet viszonya. Ezért egy használhatósági követelmény csak akkor verifikálható, ha mind a négyet rögzíti.

> **VárosiBringa: a használati környezet**
>
> A VárosiBringa csapata a kölcsönzés használati esetét (4.4. szakasz) eredetileg egyetlen felhasználói csoportra írta meg: okostelefont rutinosan használó egyetemistákra. A használatikörnyezet-elemzés négy további csoportot azonosított: az idősebb, ritkán kölcsönző lakókat, a városba látogató turistákat (más nyelv, nincs regisztráció), a gyengénlátó felhasználókat és a kesztyűt viselő téli használókat. A követelmények ennek megfelelően bővültek. Például: „a kölcsönzésnek egy korábban nem regisztrált felhasználó számára is legfeljebb 3 percen belül befejezhetőnek kell lennie, a felhasználók legalább 90%-ánál, segítség nélkül”. Ez a követelmény meghatározza a felhasználót (nem regisztrált), a célt (kölcsönzés), az eredményességet (90%), a hatékonyságot (3 perc) és a környezetet (segítség nélkül).

11.4. Emberi képességek és korlátok

11.4.1. Észlelés, figyelem, emlékezet

Az ember információfeldolgozó képessége korlátos. A rövid távú emlékezet kapacitásáról a 8.3. szakaszban már volt szó: a klasszikus „hét plusz-mínusz kettő” (Miller 1956) helyett a mai kutatások inkább három-öt egységgel számolnak (Cowan 2001). Egy kezelőfelület tervezésénél ez azt jelenti, hogy a kezelőnek ne kelljen egyszerre sok információt fejben tartania: amit egy lépés elvégzéséhez tudnia kell, az legyen a képernyőn, és ne az emlékezetére hagyatkozzon.

A figyelem szintén szűk erőforrás. Fitts klasszikus összevetése szerint (Fitts 1951) az ember elsősorban egycsatornás információfeldolgozó, és teljesítményét a tanulás és a kifáradás is befolyásolja (12. fejezet). Egy kezelő, akinek egyszerre kell figyelnie a gépre, a képernyőre és a kollégájára, könnyen elmulaszt egy jelzést.

11.4.2. Helyzettudatosság

Összetett, dinamikus rendszereknél (műhelyirányítás, forgalomirányítás, folyamatirányítás) a kezelő teljesítményének kulcsa a **helyzettudatosság** (situation awareness). Endsley meghatározása szerint ez a környezet elemeinek észlelése egy adott idő- és térbeli tartományban, jelentésük megértése és közeli jövőbeli állapotuk előrevetítése (Endsley 1995). A három szint egymásra épül:

1. **észlelés:** a kezelő látja-e a releváns adatokat (például hogy egy gép leállt);
2. **megértés:** érti-e, mit jelentenek (például hogy a leállás egy sürgős rendelést érint);
3. **előrevetítés:** meg tudja-e becsülni, mi fog történni (például hogy a rendelés két nap késést szenved, ha nem sorolják át).

A helyzettudatosság a kezelőfelületek tervezésének egyik legfontosabb mércéje. Egy jó műhelyirányítási felület nem csak adatokat mutat (első szint), hanem azok jelentését (második szint) és várható következményeit (harmadik szint) is.

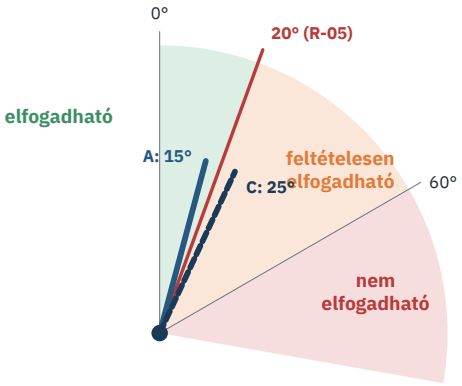
ForgácsTech: a műszakvezető helyzettudatossága

A ForgácsTech műszakvezetője a jelenlegi működésben a gépállapotot a műhelyben körbejárva, a rendelések állapotát egy táblázatból, a sürgős kéréseket telefonon kapta. Az információ három helyen volt, és mindhárom csak részben naprakész. A MES-koncepció felülettervét a csapat Endsley három szintje szerint értékelte. Az első változat a gépek állapotát mutatta (első szint). A második változat színnel jelölte azokat a gépeket, amelyek leállása határidős rendelést érint (második szint). A harmadik változat azt is kiírta, hány óra késést okoz a leállás a következő rendelésekben, ha nincs átsorolás (harmadik szint). A műszakvezetők a harmadik változatot választották, mert az „nem csak azt mondja, mi történt, hanem azt is, mit kell tennem”.

11.4.3. Testméretek és testtartás

A fizikai munkakörnyezet tervezésének alapja az **antropometria**: a felhasználói népesség testméreteinek ismerete. A szokásos tervezési gyakorlat szerint a munkahelyet úgy kell kialakítani, hogy a felhasználók nagy többségének (jellemzően az 5. és a 95. percentilis közé eső testméretűeknek) megfeleljen. Ahol ez egyetlen méretezéssel nem érhető el, ott állíthatóság kell: állítható magasságú munkaasztal, ülés, kijelző.

A testtartás értékelésére szabványos módszerek állnak rendelkezésre. Az ISO 11226 a statikus munkatesttartások elfogadhatóságát a testszögek és a tartási idő alapján értékeli (ISO 2000). A törzs előrehajlásánál a 0–20° tartomány elfogadható, a 20–60° tartomány elfogadhatósága a tartási időtől és a megtámasztástól függ, a 60° fölötti előrehajlás nem elfogadható (11.3. ábra). Gyors, megfigyelésen alapuló értékelésre a RULA (a felső végtag terhelésére) és a REBA (a teljes testre) módszer terjedt el (McAtamney és Corlett 1993; Hignett és McAtamney 2000).



A törzs előrehajlásának értékelési zónái statikus testtartásnál (ISO 11226 alapján; a középső zóna elfogadhatósága a tartási időtől függ). A KoboCella R-05 követelménye a 20°-os határt rögzíti. Kék: az együttműködő cella becsült legnagyobb törzshajlítása; szaggatott: a kézi plusz koncepcióé.

11.3. ábra. A törzs előrehajlásának értékelési zónái és a KoboCella két koncepciójának becsült legnagyobb törzshajlítása.

KoboCella: honnan jön a 20°?

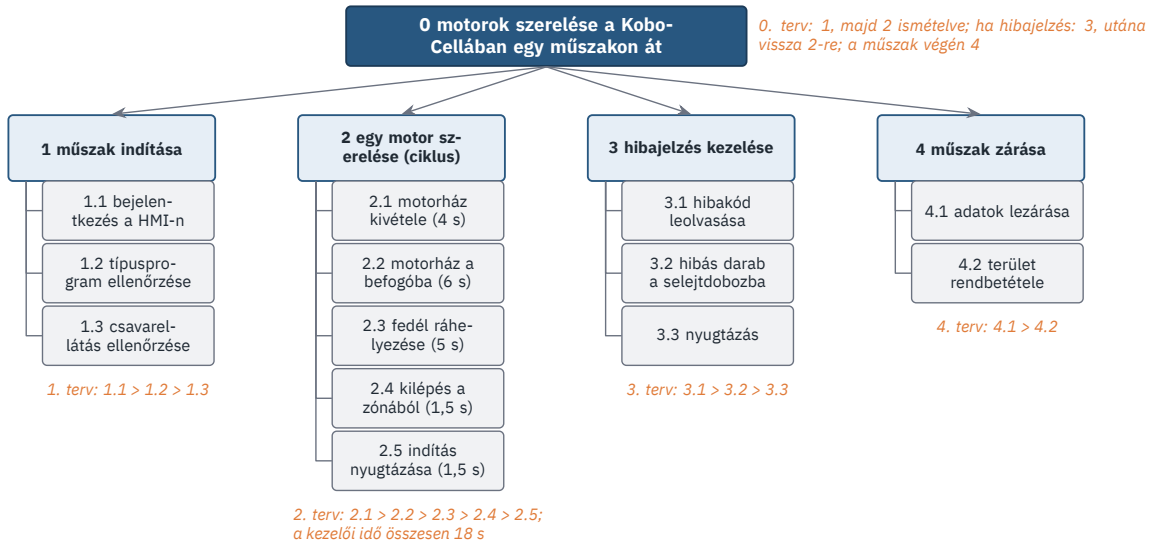
A KoboCella R-05 követelménye szerint a munkaasztalon végzett minden műveletnek legfeljebb 20°-os törzshajlítással elvégezhetőnek kell lennie (6. fejezet). A 20° nem önkényes szám: az ISO 11226 elfogadható zónájának felső határa. A követelmény így ellenőrizhető és szakmailag megalapozott. A csapat a koncepciók értékelésénél (10.3. szakasz) videós mozgáselemzéssel becsülte meg a legnagyobb törzshajlítást: az együttműködő cellában 15°, a kézi plusz koncepcióban a csavarozás alsó pozícióinál 25° adódott. Ez utóbbi a feltételesen elfogadható zónába esik, de sérti a követelményt, ezért a koncepció kiesett. A verifikáció tervében (23. fejezet) a csapat rögzítette, hogy az R-05-öt az 5. és a 95. percentilis testma-

gasságú operátorokkal kell ellenőrizni, mert az állítható asztal ellenére a legmagasabb és a legalacsonyabb kezelők testtartása eltérhet.

11.5. Feladatelemzés

A funkcióelemzés azt írja le, mit kell a rendszernek tennie (8.3. szakasz). A **feladatelemzés** azt, hogy az embernek mit kell tennie ahhoz, hogy a rá bízott funkciókat ellássa. A leggyakrabban használt módszer a **hierarchikus feladatelemzés** (hierarchical task analysis, HTA), amelyet Annett és Duncan dolgozott ki az 1960-as években, eredetileg a képzés tervezésére (Annett és Duncan 1967). Stanton áttekintése szerint a HTA azóta az ergonómia egyik legszélesebb körben alkalmazott módszerévé vált, a képzéstől a kezelőfelület-tervezésen és a létszámtervezésen át az emberi hibák elemzéséig (Stanton 2006).

A HTA a feladatot célok és részcélok hierarchiájaként írja le. Minden szinthez tartozik egy **terv** (plan), amely megmondja, milyen sorrendben és milyen feltételek mellett kell a részcélokat végrehajtani: egymás után, választás szerint, ismételve, vagy egy feltétel teljesüléséig (Stanton 2006). A lebontás ott áll meg, ahol a további bontás már nem ad hasznos információt az elemzés céljához. Képzési célú elemzésnél például ott, ahol a művelet már biztosan nem okoz nehézséget a kezelőnek.



11.4. ábra. A KoboCella kezelői feladatainak HTA-ja. A tervek a végrehajtás sorrendjét és feltételeit adják meg.

KoboCella: a kezelő feladatai

A KoboCella csapata a kezelő munkáját egy műszakra elemezte (11.4. ábra). A legfontosabb részcél a 2-es, egy motor szerelése, amely ciklusonként ismétlődik. Az öt művelet becsült ideje összesen 18 s, ami megegyezik a ciklusidő-költségvetés kezelői részével (6.5. táblázat). Az elemzés három tervezési következtetéshez vezetett. Először: a 2.4-es lépés (kilépés a zónából) nélkül a kobot nem indulhat, ezért a biztonsági alrendszernek a zóna elhagyását kell érzékelnie, nem csak a nyugtázást. Másodszor: a 2.2-es és a 2.3-as lépés a hibás behelyezés legvalószínűbb helye, ezért ezekre hibabiztos kialakítás kell (13. fejezet). Harmadszor: a 3-as részcél (hibakezelés) ritka, ezért a kezelő nem gyakorolja rendszeresen; a HMI-nek lépésről lépésre végig kell vezetnie rajta.

11.6. A használhatóság és a terhelés mérése

A használhatóság három összetevője közül az eredményességet és a hatékonyságot megfigyeléssel mérjük (befejezett feladatok aránya, hibák száma, feladatidő), a megelégedettséget kérdőívvel. A két legelterjedtebb szabványosított kérdőív a rendszer használhatósági skálája (SUS) és a NASA terhelési indexe (NASA-TLX).

A **SUS** (System Usability Scale) Brooke tíz kérdésből álló, „gyors és egyszerű” használhatósági skálája. A páratlan kérdések pozitív, a párosak negatív állítások, mindegyik ötfokú skálán értékelve. Az eredmény egy 0 és 100 közötti pontszám (Brooke 1996). A pontszám nem százalék, ezért értelmezéséhez viszonyítási alap kell. Bangor, Kortum és Miller több mint kétezer kérdőív elemzése alapján a 70 fölötti pontszámot elfogadhatónak, az 50 alattit elfogadhatatlannak találták (Bangor, Kortum, és Miller 2008). Egy későbbi vizsgálatukban a SUS-pontszámokat melléknévi minősítésekhez kötötték: az „OK” minősítés átlagos pontszáma 50,9, a „jó” minősítésé 71,4, a „kiváló” minősítésé 85,5 volt (Bangor, Kortum, és Miller 2009). Sauro és Lewis nagy adatbázisa szerint az átlagos SUS-pontszám 68 (Sauro és Lewis 2016).

A **NASA-TLX** (Task Load Index) Hart és Staveland módszere a szubjektív munkaterhelés mérésére (Hart és Staveland 1988). Hat alskálán (szellemi igénybevétel, fizikai igénybevétel, időbeli nyomás, teljesítmény, erőfeszítés, frusztráció) a résztvevő 0 és 100 közötti értéket ad. Az eredeti eljárásban az alskálák súlyát 15 páros összehasonlítás adja; Hart húszéves áttekintése szerint a gyakorlatban széles körben elterjedt a súlyozás nélküli, egyszerű átlagolás is (nyers TLX) (Hart 2006).

Mindkét kérdőív egyszerű, gyors és jól összehasonlítható eredményt ad, de egyik sem helyettesíti a megfigyelést. A SUS azt méri, hogyan *élik meg* a felhasználók a használatot, nem azt, hogy *mennyire jól* használják. Előfordul, hogy egy felhasználó elégedett, közben hibázik, vagy fordítva. Ezért a használhatósági tesztben mindig mérni kell a feladatok eredményességét és hatékonyságát is (24. fejezet).

VárosiBringa: két prototípus, két SUS-pontszám

A VárosiBringa csapata a kölcsönző alkalmazás két prototípusát tesztelte tizenkét résztvevővel, akik között mind a négy felhasználói csoport képviselve volt. Az első prototípus átlagos SUS-pontszáma 64 lett, ami az átlag (68) alatt van. A megfigyelés szerint a nem regisztrált felhasználók fele elakadt a fizetési lépésnél, és a gyengénlátó résztvevők nem találták a térképen a dokkolók szabad helyeinek jelzését. A javított prototípusban a csapat egyszerűsítette a vendégfizetést, és a szabad helyek számát a térkép mellett szövegesen is megjelenítette. A második teszt átlagos SUS-pontszáma 79 lett, és a feladatok befejezési aránya 75%-ról 96%-ra nőtt. A részletes számítást a Gyakorlati útmutató NASA-TLX és SUS eszközlapja mutatja be.

Akadálymentesség

Az emberközpontú tervezés a fogyatékossággal élő felhasználókat is magában foglalja. A digitális szolgáltatásoknál ezt az akadálymentességi irányelvek és jogszabályok követelményeként is előírják. A rendszermérnöki gyakorlatban az akadálymentesség akkor valósul meg a legolcsóbban, ha már a használati környezet elemzésekor megjelenik, és nem a kész termék utólagos átalakításaként. Ugyanaz a megoldás gyakran mindenkinek segít: a szöveges kiírás a gyengénlátó felhasználón kívül annak is, aki tűző napon nézi a telefonját.

A felhasználó mint átlagember

Gyakori hiba, hogy a csapat egyetlen „átlagos felhasználóra” tervez, aki legtöbbször maguk a tervezők. Átlagos felhasználó azonban nem létezik: egy munkaasztal, amely az átlagos testmagasságú kezelőnek kényelmes, a legmagasabbaknak és a legalacsonyabbaknak rossz lehet. A használati környezet elemzésében a felhasználói csoportokat és a szélső eseteket (méret, tapasztalat, nyelv, képességek) kifejezetten fel kell sorolni, és mindegyikre ellenőrizni kell a követelményeket.

Összefoglalás

- A rendszer a hardver, a szoftver és az ember együttese; a rendszermérnöki megközelítésnek a hármat egyenrangúan kell kezelnie. A tervezés rögzíti az életciklus-költségek nagy részét, amelynek jelentős része emberi.
- Az emberi rendszerintegráció (HSI) az emberi szempontokat a fejlesztés egészébe építi be. A HSI-terv rögzíti a felelősségeket, a célokat, a módszereket, a mérőszámokat és a kockázatkezelést.
- Az ISO 9241-210 emberközpontú tervezési ciklusa négy tevékenységből áll: a használati környezet megértése, a felhasználói követelmények meghatározása, a tervezési megoldások előállítása és az értékelés, iteratívan.
- Az ISO 9241-11 szerint a használhatóság eredményesség, hatékonyság és megelégedettség meghatározott felhasználók, célok és használati környezet mellett.
- Az ember információfeldolgozó képessége korlátos. Dinamikus rendszereknél a helyzettudatosság (észlelés, megértés, előrevetítés) a kezelőfelületek kulcsmércéje. A fizikai munkahelyet antropometriai adatokra és szabványos testtartás-értékelésre (ISO 11226, RULA, REBA) kell tervezni.
- A hierarchikus feladatelemzés célok, részcélok és tervek hierarchiájaként írja le, mit kell az embernek tennie. A használhatóság megelégedettségi összetevőjét a SUS, a szubjektív terhelést a NASA-TLX méri; a megfigyelést egyik sem helyettesíti.

Önellenőrzés

1. Miért kell a rendszermérnökségnek a hardvert, a szoftvert és az embert egyenrangúan kezelnie? Mi a gazdasági érv mellette?
2. Sorold fel a HSI-terv legalább öt tartalmi elemét!
3. Fogalmazz meg egy verifikálható használhatósági követelményt a ForgácsTech műszakvezetői felületére az ISO 9241-11 definíciójának minden elemével!
4. Mi a különbség a helyzettudatosság második és harmadik szintje között? Adj példát a KoboCella kezelőfelületéről!
5. Egy munkafolyamatban a törzs előrehajlása rövid ideig 35°. Hogyan értékelnéd az ISO 11226 zónái szerint, és mit tennél?
6. Mi a HTA-ban a terv szerepe? Írd fel a „kávé főzése kávéfőzővel” feladat kétszintű HTA-ját tervekkel!
7. Egy alkalmazás SUS-pontszáma 72. Mit mondhatunk róla, és mit nem?

Megoldás.

1. Mert a rendszer a szükséges képességet a három elem együttműködésével hozza létre; ha az ember kimarad, a rendszer nem fogja a kívánt képességet nyújtani. A gazdasági érv: a tervezés rögzíti az életciklus-költségek nagy részét (a 2.1. szakasz szerint a tervezés végére mintegy 85%-át), és az üzemeltetési költségek jelentős része emberi (létszám, képzés, hibák).
2. Szerepek és felelősségek a HSI-tartományok integrálására; HSI-célok és munkatermékek életciklus-szakaszonként; felülvizsgálati belépési és kilépési feltételek mérőszámokkal; tervezett módszerek, eszközök, szabványok; a HSI-kockázatok azonosítása és kezelése; összhang a rendszermérnöki tervvel.
3. Például: „A műszakvezetőknek (felhasználó) a felület bevezetése után egy hét gyakorlattal (környezet) egy gépleállás által érintett rendelések azonosítását (cél) a próbák legalább 95%-ában hibátlanul (eredményesség), legfeljebb 30 s alatt (hatékonyság) kell elvégezniük, és a felület SUS-pontszáma legalább 70 legyen (megelégedettség).”

4. A második szint a helyzet megértése: például a felület jelzi, hogy a nyomatékhiba a 3. csavarnál ismétlődik. A harmadik szint az előrevetítés: például jelzi, hogy ha a hiba így folytatódik, a bitbetét 40 ciklus múlva cserére szorul, vagy a műszak végéig 12 selejt várható.
5. A 35° a feltételesen elfogadható zónába esik; az elfogadhatóság a tartási időtől és a megtámasztástól függ. Rövid ideig, megtámasztással elfogadható lehet, de érdemes a munkahelyet úgy átalakítani (magasság, alkatrész-elhelyezés), hogy a hajlítás 20° alá kerüljön.
6. A terv a részcélok végrehajtásának sorrendjét és feltételeit adja meg. Példa: 0 kávé főzése; 0. terv: 1 > 2 > 3 > 4. 1 víz betöltése; 2 kávé adagolása (2.1 szűrő behelyezése, 2.2 őrölt kávé a szűrőbe; 2. terv: 2.1 > 2.2); 3 bekapcsolás; 4 kiszolgálás, ha lefőtt (4. terv: vár, amíg a jelzőfény kialszik, majd tölt).
7. A 72 az átlag (68) fölött van, és Bangor és szerzőtársai melléknévi skáláján a „jó” minősítés környékére esik. Azt nem mondja meg, hogy a felhasználók eredményesen és hatékonyan használják-e az alkalmazást, sem azt, hogy melyik funkcióval van gond: ehhez megfigyelés és feladatmérés kell.

Projektlépés: az ember a rendszerben

Saját fejlesztési feladatodhoz:

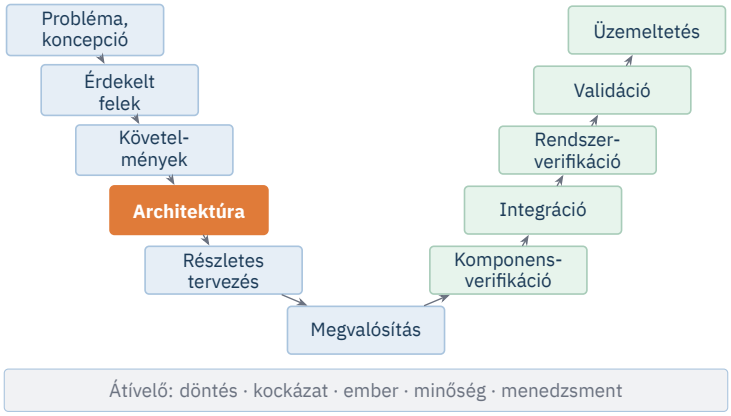
1. sorold fel a felhasználói csoportokat (kezelők, karbantartók, végfelhasználók, szélső esetek), és írd le a használati környezetüket;
2. fogalmazz meg legalább két verifikálható használhatósági vagy ergonómiai követelményt az ISO 9241-11 elemeivel;
3. készíts HTA-t a legfontosabb kezelői vagy felhasználói feladatra tervekkel és időbecsléssel;
4. tervezd meg, hogyan méred a használhatóságot (SUS, feladatmérés) és a terhelést (NASA-TLX) a validáció során.

A HTA és a NASA-TLX/SUS sablonja és notebookja a Gyakorlati útmutatóban található.

Továbbolvasás

- U.S. Department of Defense (2011): az emberi tényezők mérnöki tervezésének folyamata és eljárásai.
- ISO (2019) és ISO (2018c): az emberközpontú tervezés és a használhatóság nemzetközi szabványai.
- Booher (2003): az emberi rendszerintegráció kézikönyve.
- Endsley (1995): a helyzettudatosság elméletének alapműve.
- Stanton (2006): a hierarchikus feladatelemzés fejlődésének és alkalmazásainak áttekintése.
- Hart (2006) és Bangor, Kortum, és Miller (2009): a NASA-TLX és a SUS használata és értelmezése.
- NASA (2016), 2.6. szakasz és R függelék: az emberi rendszerintegráció helye a rendszermérnökségben és a HSI-terv tartalma.
- NASA (2015): az emberi rendszerintegráció gyakorlati útmutatója.

12. Funkciómegosztás és automatizálás



Tanulási eredmények

A fejezet feldolgozása után képes leszel:

- leírni a funkciómegosztás lépéseit, és megindokolni, miért „inkább művészet, mint tudomány”;
- alkalmazni a Fitts-listát és a megosztási kritériumokat, és megnevezni a módszer korlátait;
- az automatizálás szintjét Sheridan és Verplank tízfokú skáláján, az automatizálás típusát Parasuraman, Sheridan és Wickens négy szakasza szerint meghatározni;
- felismerni az automatizálás ironiáit és buktatóit (a helyzettudatosság elvesztése, a túlzott és a hiányos bizalom);
- ismertetni az ember–robot együttműködés módjait és biztonsági alapelveit;
- statikus és dinamikus (adaptív) funkciómegosztást tervezni, a leromlott üzemmódokat is beleértve.

Az architektúra kialakításakor a funkciókat a forma elemeihez rendeltük (8.5. szakasz). Ezek között az elemek között ott van az ember is: a kezelő, a karbantartó, a felhasználó. Az emberi rendszerintegráció egyik alapeszköze éppen a funkciók egyértelmű elosztása a rendszer elemei, vagyis a hardver, a szoftver és az ember között (11.2. szakasz). Melyik funkciót végezze az ember, melyiket a gép, és melyiket együtt? Ez a kérdés minden automatizálási projekt középpontjában áll, a KoboCella kobotjától a ForgácsTech ütemezőjéig.

A kérdésre csábító egyszerű választ adni: „amit lehet, automatizáljunk”. A tapasztalat azonban azt mutatja, hogy az automatizálás nem egyszerűen átveszi az ember feladatait, hanem meg is változtatja azokat. Új feladatokat (felügyelet, beavatkozás, hibaelhárítás) hoz létre, és ezek gyakran nehezebbek, mint a régiek voltak. Ez a fejezet a funkciómegosztás módszereit és az automatizálás emberi következményeit tárgyalja.

A fejezet négy kérdésre épül:

- Milyen elvek alapján osszuk meg a funkciókat ember és gép között?

- Hogyan írható le az automatizálás szintje és típusa?
- Milyen buktatói vannak az automatizálásnak?
- Hogyan működhet együtt biztonságosan az ember és a robot?

12.1. A funkciómegosztás feladata

A funkciómegosztásra nincs mechanikus eljárás, a tervezői ítéletnek itt különösen nagy a szerepe. A munka lépései:

1. a funkciók azonosítása (a funkcióelemzésből, a használati esetekből, interjúkból és a vevői elvárásokból);
2. annak azonosítása, ki a legmegfelelőbb az adott funkcióra: ember, automatika vagy a kettő együtt;
3. annak eldöntése, hogy a megosztás statikus (a tervezéskor rögzített) vagy dinamikus, adaptív (a terheléstől és a helyzettől függően változó) legyen.

A funkciómegosztás a funkciófára épül (8.3. szakasz), és eredménye az allokációs táblázatban jelenik meg (8.5. szakasz). A különbség az, hogy itt kifejezetten azt vizsgáljuk, mit jelent egy-egy döntés az emberre nézve: milyen feladatai maradnak, mennyire terhelt, érti-e, mit csinál a gép, és be tud-e avatkozni, ha kell.

12.1.1. A Fitts-lista

A funkciómegosztás klasszikus eszköze a **Fitts-lista**, amely Fitts 1951-es, a légiforgalmi irányítás emberi tényezőiről szóló jelentéséből származik (Fitts 1951). A lista összeveti, miben jobb az ember és miben a gép. A 12.1. táblázat a lista szellemében, mai példákkal kiegészítve foglalja össze az összevetést.

12.1. táblázat. Az ember és a gép erősségei a Fitts-lista (Fitts 1951) szellemében, saját összeállítás.

Tulajdonság	Gép	Ember
sebesség	kiváló	viszonylag lassú
kimenő teljesítmény	kiváló, állandó	viszonylag gyenge
következetesség	ideális, ismétlődő eseményeknél	megbízhatatlan; a tanulás és a kifáradás befolyásolja
információs kapacitás	többcsatornás	elsősorban egycsatornás
emlékezet	pontos, szó szerinti felidézés, korlátozott hozzáférés	elvek és stratégiák jobb felidézése, sokoldalú hozzáférés, innováció
logikai számítás	deduktív, bonyolult program, gyors és alapos, gyenge hibajavítás	induktív, egyszerűbb program, lassú, pontos, jó hibajavítás
érzékelés	jó mennyiségi értékelés, gyenge mintafelismerés	széles hatókör, sokoldalúság, ítélőképesség
észlelés	a változatossággal nehezen birkózik, zajérzékeny	a változásokkal jobban megbirkózik, zajérzékeny

A Fitts-listát sokan kritizálták. A leggyakoribb ellenvetés, hogy a lista az embert és a gépet egymás helyettesítőiként kezeli, holott a jó rendszerekben kiegészítik egymást, és hogy a gépek képességei azóta jelentősen változtak. De Winter és Dodou mégis amellett érvelnek, hogy a lista azért maradt fenn hét évtizeden át, mert lényegi megállapításai, például hogy a gép a gyorsaságban, a pontosságban és az ismétlődő feladatokban, az ember az ítéletalkotásban, az alkalmazkodásban és a váratlan helyzetek kezelésében erősebb, ma is többnyire érvényesek, és a lista jó kiindulópontot ad a tervezői gondolkodáshoz (Winter és Dodou 2014).

12.1.2. Megosztási kritériumok

Price az ember és a gép relatív képességei alapján hat esetet különít el (Price 1985):

1. nincs különbség az emberi és a gépi képességek között;
2. az ember teljesítménye jobb;
3. a gép teljesítménye jobb;
4. a gép teljesítménye olyan gyenge, hogy a funkciót az embernek kell kiosztani;
5. az ember teljesítménye olyan gyenge, hogy a funkciót a gépnek kell kiosztani;
6. sem az ember, sem a gép teljesítménye nem elfogadható.

A 4. és az 5. eset egyértelmű. A 6. esetben a funkciót vagy a rendszert újra kell gondolni (a funkció megoldássemleges megfogalmazásától kezdve, 8.1. szakasz). Az 1–3. esetben a döntés nem csak a képességeken múlik. Price ezért további megosztási szempontokat is megnevez (Price 1985): az **érték-egyensúlyt** (mi a jó az emberi munka minősége szempontjából), a **haszonelvű és költségalapú** megosztást (mi a gazdaságos), és az érzelmi vagy kognitív támogatást szolgáló megosztást (mit fogadnak el és mit értenek az érintettek). A gyakorlatban mindhárom számít: egy technikailag és gazdaságilag indokolt automatizálás is megbukhat, ha az érintettek nem fogadják el, vagy ha az ember számára értelmetlen, monoton maradékfeladatot hagy.

> **KoboCella: ki mit csinál?**
>
> A KoboCella csapata a funkciófa tizenhárom részfunkcióján végigment a Fitts-lista és a megosztási kritériumok szerint. A csavar meghúzása és a nyomaték mérése egyértelműen gépi funkció lett (3. és 5. eset: következetesség, pontosság, naplózhatóság). A motorház befogása az emberé maradt: a különböző változatok kezelése, a sérült alkatrész felismerése ítélőképességet igényel, és a gépi megoldás aránytalanul drága lett volna (2. eset, költségalapú szempont). A hibás darab jelölése megosztott funkció lett: a gép jelzi a tűrésen kívüli nyomatékot, az ember dönt a további teendőről (javítás vagy selejt). A csapat külön megvizsgálta az érzelmi szempontot is: az operátorokkal folytatott interjúk szerint a kezelők attól tartottak, hogy „a robot mellett csak adogatók lesznek”. Ezért a cella kezelőjének feladatai közé bekerült a minőségi ellenőrzés és a változatváltás is, a HMI pedig minden ciklusban visszajelzést ad a kezelőnek a kötés minőségéről.

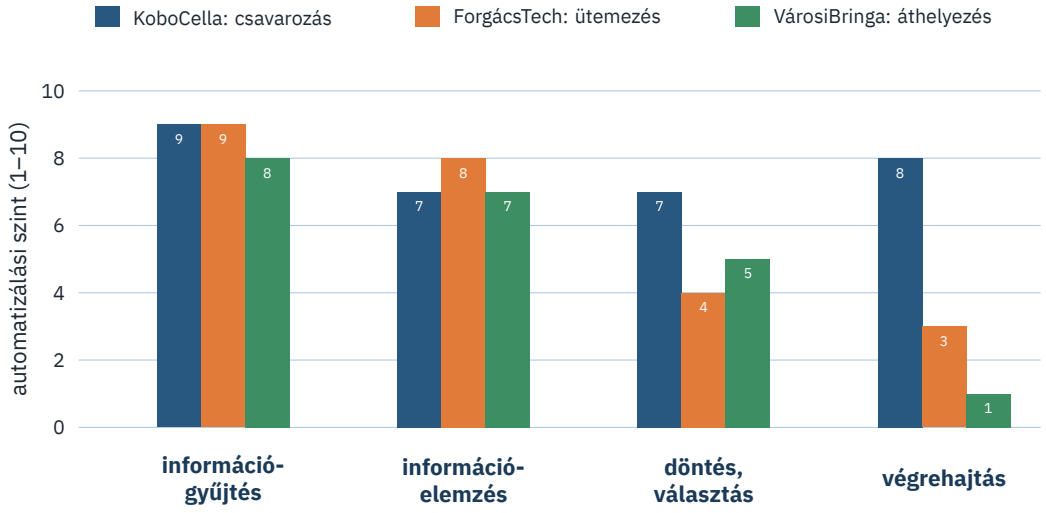
12.2. Az automatizálás szintjei és típusai

Az automatizálás nem kétállású kapcsoló (ember vagy gép), hanem fokozatos. Sheridan és Verplank tízfokú skálája a döntés és a cselekvés kiválasztásának automatizálását írja le (12.2. táblázat) (Sheridan és Verplank 1978).

12.2. táblázat. Az automatizálás tíz szintje Sheridan és Verplank alapján (Sheridan és Verplank 1978), saját fordítás.

Szint	Az automatika szerepe
1	a számítógép nem ad segítséget: mindenről az ember dönt, és ő cselekszik
2	a számítógép a döntési vagy cselekvési alternatívák teljes körét felkínálja, vagy
3	leszűkíti a választékot néhány lehetőségre, vagy
4	egy alternatívát javasol, és
5	végrehajtja a javaslatot, ha az ember jóváhagyja, vagy
6	korlátozott ideig lehetőséget ad az embernek a vétóra, mielőtt automatikusan végrehajtja, vagy
7	automatikusan végrehajtja, majd szükségszerűen tájékoztatja az embert, és
8	csak akkor tájékoztatja az embert, ha az kérdezi, vagy
9	csak akkor tájékoztatja az embert, ha az automatika úgy dönt,
10	a számítógép mindenről dönt, önállóan cselekszik, és figyelmen kívül hagyja az embert

Parasuraman, Sheridan és Wickens ezt a gondolatot továbbfejlesztették. Szerintük az automatizálás nem egyetlen szinttel, hanem négy információfeldolgozási szakasz mentén írható le: az **információgyűjtés** (érzékelés, adatgyűjtés), az **információelemzés** (feldolgozás, következtetés, előrejelzés), a **döntés és választás**, valamint a **végrehajtás**. A négy szakaszban az automatizálás szintje egymástól függetlenül változhat (Parasuraman, Sheridan, és Wickens 2000). Egy rendszer így jellemezhető egy négyelemű **automatizálási profillal**. A szerzők azt is javasolják, hogy a megfelelő szintet szakaszonként az emberi teljesítményre gyakorolt hatás (terhelés, helyzettudatosság, beidegződés, készségek leépülése) és a megbízhatóság, valamint a döntési hibák költsége alapján válasszuk meg (Parasuraman, Sheridan, és Wickens 2000).



12.1. ábra. A három futó eset egy-egy kulcsfunkciójának automatizálási profilja a négy információfeldolgozási szakaszban.

A 12.1. ábra a három eset egy-egy funkciójának profilját mutatja. A KoboCella csavarozása mind a négy szakaszban magas szintű: a gép gyűjti az adatokat, értékeli a nyomatékot, dönt a megfelelőségről és végre is hajtja a műveletet (a kezelőt tájékoztatja). A ForgácsTech ütemezése az információgyűjtésben és -elemzésben magas, a döntésben viszont szándé-

kosan alacsonyabb szintű (4: egy javaslatot ad), a végrehajtásban pedig a műszakvezetőé a szó. A VárosiBringa áthelyezése hasonló: az áthelyezés-tervező javaslatot tesz a következő útvonalra, de a fizikai végrehajtás teljes egészében emberi.

> **ForgácsTech: miért nem dönt az ütemező?**
>
> A ForgácsTech MES-ütemezőmodulja technikailag képes lenne önállóan átsorolni a rendeléseket egy gépleállás után (7-es szint). A csapat mégis a 4-es szint mellett döntött: az ütemező egy átsorolási javaslatot ad a várható késésekkel, és a műszakvezető hagyja jóvá vagy módosítja. Az indoklás három pontból állt. Először: a műszakvezető olyan információkat is ismer, amelyek a rendszerben nincsenek meg (például hogy egy vevő elfogad-e részszállítást). Másodszor: a döntési hibák költsége magas, egy rossz átsorolás a két legnagyobb vevő egyikét is érintheti. Harmadszor: a névleges csoportos elemzés (9.3. szakasz) szerint a késések első számú oka a szóbeli előresorolás volt; ha az ütemező teljesen automatikusan dönt, a műszakvezetők a rendszert megkerülve folytatnák a régi gyakorlatot. A jóváhagyási lépés az átsorolást láthatóvá és naplózottá teszi. A csapat rögzítette, hogy egy év üzemi tapasztalat után a szint emelését újra megvizsgálja.

12.3. Az automatizálás iróniái és buktatói

Bainbridge nagy hatású tanulmánya az automatizálás **iróniáit** írta le (Bainbridge 1983). Az első irónia, hogy a tervezők, akik az embert megbízhatatlannak tartják, és ezért automatizálnak, maguk is emberek, és a tervezési hibáik az automatikában jelennek meg. A második, hogy az automatizálás után az embernek azok a feladatok maradnak, amelyeket a tervezők nem tudtak automatizálni: a ritka, váratlan, nehéz helyzetek kezelése. Ehhez azonban gyakorlat és naprakész helyzetismeret kellene, amelyet az ember éppen az automatizálás miatt veszít el. Aki csak felügyeli a gépet, annak kézi készségei leépülnek, és a beavatkozás pillanatában nem érti pontosan, mi történik.

Parasuraman és Riley az automatizálás négy problémás használati módját különítették el (Parasuraman és Riley 1997):

- **helyes használat** (use): az ember akkor támaszkodik az automatikára, amikor kell;
- **túlzott támaszkodás** (misuse): az ember akkor is az automatikára hagyatkozik, amikor nem kellene, például nem veszi észre a hibáit (automatizálási torzítás, elkényelmesedés);
- **mellőzés** (disuse): az ember nem használja az automatikát, például a túl gyakori téves riasztások miatt figyelmen kívül hagyja a jelzéseit;
- **visszaélés** (abuse): a tervezők vagy a vezetők anélkül automatizálnak, hogy figyelembe vennék a következményeket az ember teljesítményére.

A középpontban a **bizalom** áll. Lee és See áttekintése szerint a cél nem a bizalom maximalizálása, hanem a **kalibrált bizalom**: az ember bizalma igazodjon az automatika tényleges képességeihez. A túlzott bizalom túlzott támaszkodáshoz, a hiányos bizalom mellőzéshez vezet. A kalibrált bizalmat az segíti, ha az automatika átlátható: a kezelő látja, mit csinál, miért, és mennyire biztos benne (Lee és See 2004).

A buktatók elleni tervezési elvek:

- **tartsd a hurokban az embert:** a felügyeleti feladat mellett maradjanak aktív feladatai, és a kijelzők a helyzettudatosság mindhárom szintjét támogassák (11.4. szakasz);
- **tedd átláthatóvá az automatikát:** a kezelő lássa az automatika állapotát, szándékát és bizonytalanságát;

- **kezeld a téves riasztásokat:** a túl sok téves riasztás mellőzéshez vezet; a riasztási küszöböket a téves és az elmulasztott riasztások költségének egyensúlya alapján kell beállítani;
- **gyakoroltasd a kézi üzemet:** ha az automatika kieshet, az embernek gyakorlatban kell maradnia a kézi működtetésben.

> **KoboCella: a nyugtázás csapdája**
>
> A KoboCella első HMI-tervében a kezelőnek minden tűrésen kívüli nyomaték után egy gombnyomással kellett nyugtáznia a hibajelzést. A próbaüzemben kiderült, hogy a kezelők a második-harmadik nap után rutinszerűen, a jelzés elolvasása nélkül nyomták meg a gombot: a nyugtázás a mozdulatsor része lett (túlzott támaszkodás, Parasuraman és Riley kategóriájában). A csapat két változtatást vezetett be. A hibás darab nem kerülhet a kiadóba, amíg a kezelő nem helyezte át a selejtdobozba, amit érzékelő ellenőriz (a nyugtázás helyett fizikai cselekvés). A HMI pedig a hiba helyét is megmutatja (melyik csavar), így a kezelő ténylegesen ellenőrizni tudja a darabot. A leromlott üzemmódra (UM-7) a csapat negyedéves kézi gyakorlást írt elő, hogy a kézi szerelési készség ne vesszen el.

12.4. Ember–robot együttműködés

Az együttműködő robotok (kobotok) a funkciómegosztás egy különleges formáját teszik lehetővé: az ember és a robot közös munkaterületen, védőkerítés nélkül dolgozik. Az ipari robotok biztonsági követelményeit az ISO 10218 szabványsorozat rögzíti: az 1. rész a robotokra, a 2. rész a robotalkalmazásokra és robotcellákra vonatkozik (ISO 2025a, 2025b). A 2025-ös kiadás beépítette az együttműködő alkalmazások korábban külön műszaki specifikációban (ISO/TS 15066) rögzített biztonsági követelményeit is.

Az együttműködő működésnek négy alapvető módja van:

- **biztonsági felügyelt megállás:** a robot megáll, amikor az ember belép a közös munkaterületre, és csak akkor folytatja, amikor kilépett;
- **kézi vezetés:** az ember közvetlenül, kézzel vezeti a robotot (például betanításkor);
- **sebesség- és távolságfelügyelet:** a robot sebességét az ember távolságától függően csökkentik, és a robot megáll, mielőtt a távolság egy kritikus érték alá csökkenne;
- **teljesítmény- és erőkorlátozás:** a robot tervezése és vezérlése korlátozza az érintkezéskor fellépő erőket és nyomásokat, így az érintkezés nem okoz sérülést.

A módok kiválasztása a kockázatértékelés eredménye (17. fejezet), és egy cellán belül többféle mód is alkalmazható.

> **KoboCella: a választott együttműködési mód**
>
> A KoboCella koncepciójában a védőzónát lézerszkenner figyeli, sebességkorlátozással (9.4. szakasz). Ez a sebesség- és távolságfelügyelet módja: amikor a kezelő belép a közös zónába, a kobot csökkentett sebességre vált (R-11: legfeljebb 250 mm/s), és ha a kezelő a kritikus távolságon belülre kerül, megáll. A csapat azért nem a teljesítmény- és erőkorlátozást választotta egyedüli védelemként, mert a csavarozó fej éles bitbetétje a kezelő keze közelében mozog, és ennél a szerszámnál az érintkezés még korlátozott erő mellett is sérülést okozhatna. A választást a kockázatértékelés indokolja, amelyet a 17. fejezet mutat be.

12.5. Statikus és dinamikus megosztás

A **statikus megosztás** a tervezéskor rögzíti, ki mit csinál. Egyszerű, kiszámítható és könnyen betanítható, de nem alkalmazkodik a változó helyzetekhez. A **dinamikus** vagy **adaptív megosztás** a helyzettől függően változtatja a feladatok elosztását: nagy terhelésnél több feladatot vesz át az automatika, kis terhelésnél vagy az automatika meghibásodásakor több marad az emberé.

A dinamikus megosztás legfontosabb gyakorlati formája a **leromlott üzemmód**: ha egy automatikus funkció kiesik, a rendszer csökkentett képességgel, nagyobb emberi részvétellel tovább működik. Ezt a működési koncepcióban kell megtervezni (4.4. szakasz), mert minden leromlott üzemmód új követelményeket szül: az embernek képesnek kell lennie átvenni a funkciót, a kezelőfelületnek támogatnia kell az átvételt, és az átvételt gyakorolni kell.

A dinamikus megosztás kockázata a **módzavar**: a kezelő nem tudja pontosan, melyik üzemmódban van a rendszer, és ennek megfelelően rosszul cselekszik. Ezért minden üzemmódváltásnak egyértelműen jelzettnek kell lennie, és az átmeneteket a forgatókönyvekben végig kell gondolni.

VárosiBringa: amikor az áthelyezés-tervező téved

A VárosiBringa áthelyezés-tervezője a kölcsönzési adatokból előrejelzi, melyik állomás fog kiürülni vagy megtelni, és javaslatot tesz a furgonos áthelyezés útvonalára. A csapat a forgatókönyvek elemzésekor felismerte, hogy rendezvények, útlezárások vagy hirtelen időjárásváltozás esetén az előrejelzés megbízhatatlan. Ezért két üzemmódot terveztek: normál üzemben a sofőr a javasolt útvonalat követi (5-ös szint: a rendszer javasol, az ember jóváhagyja), rendkívüli helyzetben pedig a diszpécser kézi üzemmódra válthat, amelyben a rendszer csak az állomások aktuális állapotát mutatja, és az útvonalat a diszpécser állítja össze (2–3-as szint). Az üzemmód a sofőr kijelzőjén mindig látható, és minden váltást naplóznak, hogy utólag elemezni lehessen, mikor és miért tért el az ember a javaslattól.

Mesterséges intelligencia a döntési szakaszban

A gépi tanulási és nyelvi modelleken alapuló döntéstámogató rendszerek egyre gyakrabban az információelemzés és a döntés szakaszában jelennek meg: hibát diagnosztizálnak, ütemezést javasolnak, karbantartást ütemeznek. A Parasuraman–Sheridan–Wickens-modell erre is jól alkalmazható. Különösen fontos a kalibrált bizalom: a modellek hibái gyakran nem szembetűnők, és a felhasználó a meggyőzően megfogalmazott javaslatot hajlamos ellenőrzés nélkül elfogadni. A tervezési elvek ugyanazok: átláthatóság (miért ezt javasolja, mennyire biztos benne), a döntési jogkör egyértelmű kijelölése, és annak megtervezése, mi történik, ha a javaslat téves.

Automatizáljunk mindent, amit lehet

A „technológiai imperatívusz” az a hozzáállás, hogy mindent automatizálni kell, ami technikailag automatizálható, és az embernek a maradék jut. Ez Bainbridge iróniájához vezet: az ember feladatai töredezettek, monotonok és ritkán gyakorolt vészhelyzeti beavatkozásokból állnak. A funkciómegosztást az ember szempontjából is meg kell tervezni: a megmaradó feladatoknak együtt értelmes, kezelhető és gyakorolható munkát kell alkotniuk.

Összefoglalás

- A funkciómegosztás eldönti, mit végezzen az ember, mit a gép és mit együtt. Lépései a funkciók azonosítása, a legalkalmasabb végrehajtó kiválasztása és a statikus vagy dinamikus megosztás eldöntése.
- A Fitts-lista az ember és a gép relatív erősségeit veti össze. Kritikusai szerint a kettőt helyettesítőként kezeli, mégis máig hasznos kiindulópont. A képességek mellett az érték-egyensúly, a költség és az elfogadottság is megosztási szempont.
- Sheridan és Verplank tíz szintje a döntés és a cselekvés automatizálásának fokát írja le. Parasuraman, Sheridan és Wickens szerint az automatizálás négy szakaszban (információgyűjtés, -elemzés, döntés, végrehajtás) eltérő szintű lehet.
- Az automatizálás iróniája, hogy az embernek a legnehezebb, legritkább feladatok maradnak, miközben készségei és helyzettudatossága leépül. A túlzott támaszkodás, a mellőzés és a visszaélés elkerülésének kulcsa a kalibrált bizalom és az átlátható automatika.
- Az együttműködő robotok biztonságát az ISO 10218 szabványsorozat szabályozza; a négy együttműködési mód a felügyelt megállás, a kézi vezetés, a sebesség- és távolságfelügyelet, valamint a teljesítmény- és erőkorlátozás.
- A dinamikus megosztás legfontosabb formája a leromlott üzemmód, amelyet a működési koncepcióban kell megtervezni. Az üzemmódokat egyértelműen jelezni kell a módzavar elkerülésére.

Önellenőrzés

1. Miért mondják a funkciómegosztásra, hogy „inkább művészet, mint tudomány”?
2. Hogyan osztanád meg a Fitts-lista alapján egy raktári komissiózás két funkcióját: (a) a megfelelő tétel kiválasztása egy polcról, (b) a kiszedett mennyiség ellenőrzése?
3. Melyik Sheridan–Verplank-szinten működik egy navigációs alkalmazás, amely útvonalat javasol, és csak akkor indul, ha a felhasználó jóváhagyja? És egy autó vészfékasszisztense?
4. Írd fel egy laboratóriumi mintaelemző automata automatizálási profilját a négy szakaszban!
5. Mi Bainbridge második iróniája? Mivel csökkenthető a hatása?
6. Mi a különbség a túlzott támaszkodás és a mellőzés között? Mi okozhatja az utóbbit egy riasztórendszernél?
7. Miért nem elég a KoboCellánál a teljesítmény- és erőkorlátozás?

Megoldás.

1. Mert a döntés sok, egymással nehezen összemérhető szemponton múlik: a képességeken, a költségen, az érintettek elfogadásán, a munka minőségén és a ritka helyzetek kezelésén, és ezekre nincs egyetlen, mechanikusan alkalmazható szabály.
2. (a) A változatos tételek felismerése és megfogása mintafelismerést és ügyességet igényel; ez hagyományosan az ember erőssége (bár a gépi látás fejlődik). (b) A mennyiség ellenőrzése pontosságot és következetességet igényel; ez a gépé (mérleg vagy vonalkód-olvasás).
3. A navigációs alkalmazás az 5-ös szinten működik (végrehajtja, ha az ember jóváhagyja). A vészfékasszisztens a 7-es szint körül (automatikusan végrehajtja, majd tájékoztat), mert a reakcióidő nem teszi lehetővé az emberi jóváhagyást.
4. Például: információgyűjtés 9 (a minták automatikus azonosítása és mérése), elemzés 8 (az eredmények automatikus értékelése a referenciatartományokhoz), döntés 4 (a kóros eredményt a rendszer jelzi, a szakember dönt), végrehajtás 7 (a jóváhagyott eredmény automatikusan továbbítódik, a rendszer tájékoztat).
5. Az automatizálás után az embernek a ritka, váratlan, nehéz helyzetek kezelése marad, miközben az ehhez szükséges gyakorlata és helyzetismerete éppen az automatizálás miatt leépül. Csökkenthető rendszeres gyakorlással (szimuláció, kézi üzem), aktív feladatokkal a felügyelet mellett és átlátható kijelzőkkel.
6. Túlzott támaszkodásnál az ember akkor is az automatikára hagyatkozik, amikor nem kellene (nem veszi észre a hibáit); mellőzésnél nem használja, amikor kellene. Riasztórendszernél a mellőzés gyakori oka a sok téves riasztás: a kezelő megtanulja figyelmen kívül hagyni a jelzéseket.

7. Mert a csavarozó fej éles bitbetétje a kezelő keze közelében mozog, és ennél a szerszámnál az érintkezés korlátozott erő mellett is sérülést okozhat. Ezért sebesség- és távolságfelügyelet kell, amely megakadályozza az érintkezést.

Projektlépés: funkciómegosztás

Saját fejlesztési feladatodhoz:

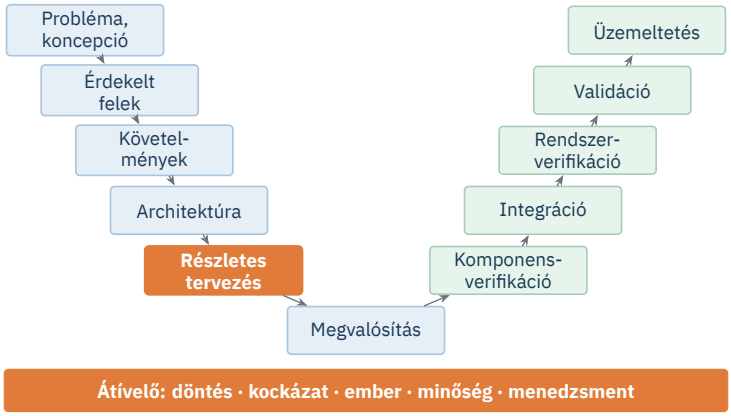
1. válaszd ki a funkciófa legalább hat funkcióját, és osszd meg őket ember, gép vagy együttes végrehajtás között a Fitts-lista és a megosztási kritériumok alapján, indoklással;
2. a két legfontosabb automatizált funkcióhoz határozd meg az automatizálási profilt a négy szakaszban;
3. írd le, milyen feladatai maradnak az embernek, és értelmes, gyakorolható munkát alkotnak-e;
4. tervezz legalább egy leromlott üzemmódot, és írd le, milyen követelményeket szül.

A funkciómegosztás sablonja és notebookja a Gyakorlati útmutatóban található.

Továbbolvasás

- Parasuraman, Sheridan, és Wickens (2000): az automatizálás típusainak és szintjeinek modellje; a funkciómegosztás egyik legtöbbet hivatkozott tanulmánya.
- Bainbridge (1983): az automatizálás iróniái; rövid, ma is aktuális klasszikus.
- Parasuraman és Riley (1997) és Lee és See (2004): az automatizálás használata, a visszaélések és a kalibrált bizalom.
- Winter és Dodou (2014): a Fitts-lista kritikája és tartós érvényességének magyarázata.
- ISO (2025b): az együttműködő robotalkalmazások biztonsági követelményei.
- NASA (2015): az emberi rendszerintegráció gyakorlati útmutatója, a funkciómegosztás és a feladatelemzés helyével a fejlesztésben.

13. Emberi teljesítmény és emberi hiba



Tanulási eredmények

A fejezet feldolgozása után képes leszel:

- megkülönböztetni az emberi hibát mint okot és mint tünetet, valamint a személyközpontú és a rendszerszemléletű megközelítést;
- besorolni az emberi hibákat (megcsúszás, kihagyás, tévedés, szabályszegés) és összekapcsolni őket Rasmussen viselkedési szintjeivel;
- felsorolni a teljesítményt befolyásoló tényezőket, és ismertetni az emberi megbízhatóság-elemzés (HRA) célját és korlátait;
- egy egyszerű HEART-számítással összehasonlítani tervezési változatokat;
- hibabiztos (poka-yoke) megoldásokat és kényszerítő funkciókat tervezni;
- az emberi hibák elemzését beépíteni a kockázatelemzésbe és a szervezeti tanulásba.

Az emberközpontú tervezés (11. fejezet) és a funkciómegosztás (12. fejezet) után is marad egy kérdés: mi történik, ha az ember hibázik? Az ember hibázni fog: ez nem jellemhiba, hanem az emberi működés természetes velejárója. A rendszermérnök feladata nem a hibátlan ember feltételezése, hanem olyan rendszer tervezése, amelyben a hibák ritkák, időben észrevehetők, és nem vezetnek súlyos következményhez.

A fejezet négy kérdésre épül:

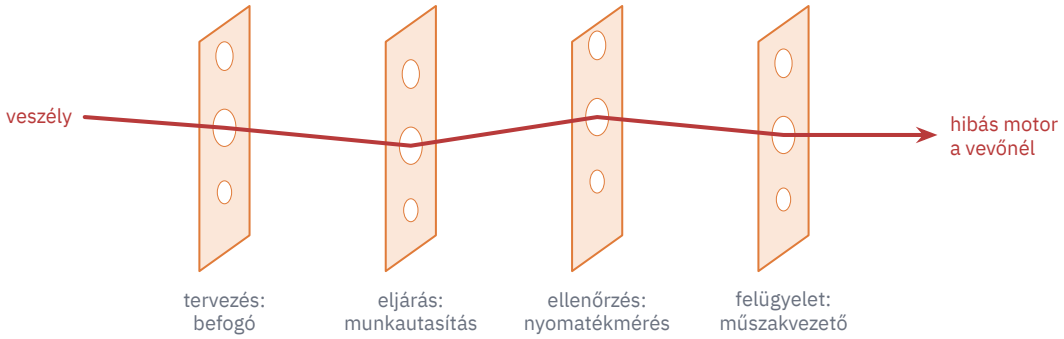
- Hogyan gondolkodjunk az emberi hibáról: ok vagy tünet?
- Milyen típusú hibákat követünk el, és miért?
- Hogyan becsülhető az emberi hiba valószínűsége, és mire jó ez a becslés?
- Hogyan tervezhetünk a hibák ellen?

13.1. Az emberi hiba két szemlélete

Reason két megközelítést különít el az emberi hiba kezelésében (Reason 2000). A **személyközpontú megközelítés** a hibát az egyén figyelmetlenségével, hanyagságával, moti-

válatlanságával magyarázza, és a válasza a figyelmeztetés, a fegyelmezés, az újabb utasítás. A **rendszerszemléletű megközelítés** abból indul ki, hogy az ember esendő, és a hibák a legjobb szervezetekben is előfordulnak. A hibák ezért inkább következmények, mint okok: a munkahely és a szervezet rejtett hiányosságaiból fakadnak. A válasz itt nem az ember megváltoztatása, hanem azoké a feltételeké, amelyek között dolgozik.

Reason híres **svájcisajt-modellje** szerint a rendszer veszélyekkel szembeni védelme több rétegből áll (műszaki megoldások, eljárások, képzés, felügyelet), és minden rétegnek vannak „lyukai” (13.1. ábra). A lyukak egy része **aktív hiba**: az első vonalban dolgozó ember tévedése, kihagyása. Más része **lappangó feltétel**: a tervezés, az eljárások, a szervezet hiányossága, amely hosszú ideig rejtve maradhat. Baj akkor lesz, ha a lyukak pillanatnyilag egy vonalba kerülnek (Reason 2000).



Reason modelljében minden védelmi réteg „lyukas”: a lyukak a tervezés, az eljárások és a szervezet rejtett hiányosságai. Baj akkor történik, ha a lyukak egy vonalba kerülnek. A megelőzés a rétegek javításáról szól, nem az utolsó ember hibáztatásáról.

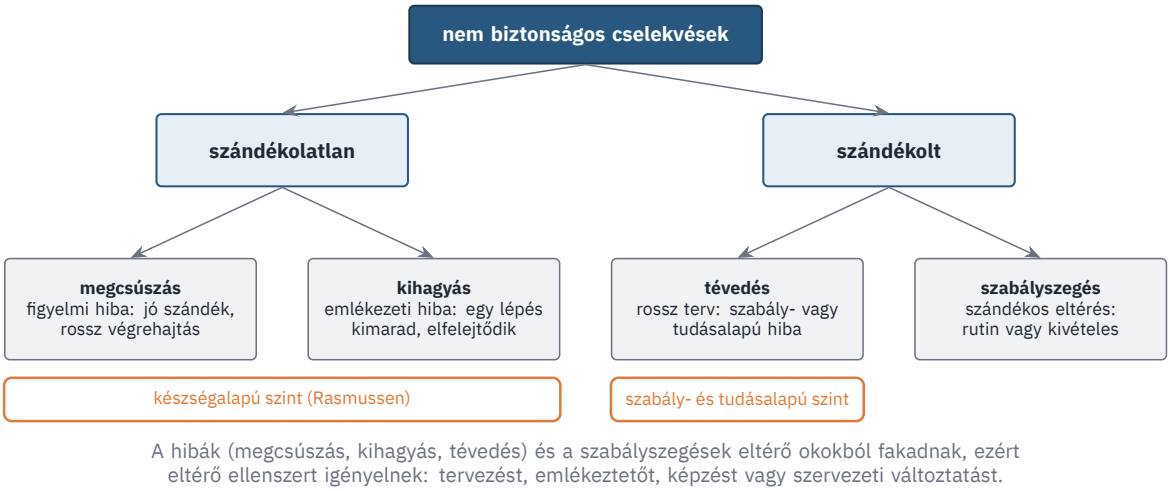
13.1. ábra. A svájcisajt-modell a KoboCella egyik lehetséges hibalánca példáján.

Dekker gyakorlati útmutatója ugyanezt a szemléletet képviseli: az „emberi hiba” nem magyarázat, hanem kiindulópont a vizsgálathoz. Azt kell megérteni, miért tűnt ésszerűnek az adott embernek abban a helyzetben azt tenni, amit tett (Dekker 2014). A rendszermérnök számára ez döntő különbség. Ha egy balesetvizsgálat „emberi hiba” megállapításánál megáll, a rendszer nem változik, és a következő ember ugyanabban a helyzetben ugyanazt a hibát fogja elkövetni.

13.2. A hibák típusai

Reason osztályozása a nem biztonságos cselekvéseket aszerint csoportosítja, hogy szándékosak voltak-e, és hol csúszott el a folyamat (Reason 1990) (13.2. ábra):

- **megcsúszás** (slip): a szándék jó, de a végrehajtás téves, jellemzően figyelmi hiba miatt (például rossz gombot nyom meg, mert ugyanott van, mint a megszokott);
- **kihagyás** (lapse): egy lépés kimarad vagy elfelejtődik, emlékezeti hiba miatt (például félbeszakítás után nem folytatja ott, ahol abbahagyta);
- **tévedés** (mistake): maga a terv hibás. Szabályalapú tévedésnél egy jó szabályt rossz helyzetben, vagy egy rossz szabályt alkalmaz; tudásalapú tévedésnél új helyzetben hiányos tudás vagy rossz helyzetértékelés alapján dönt;
- **szabályszegés** (violation): szándékos eltérés az előírt eljárástól, jellemzően nem ártó szándékkal, hanem mert az előírás betartása nehézkes, időigényes vagy a helyzethez nem illő. Lehet rutinszerű (mindenki így csinálja) vagy kivételes (egy rendkívüli helyzetben).



13.2. ábra. Reason osztályozása és kapcsolata Rasmussen viselkedési szintjeivel.

Rasmussen modellje az emberi viselkedés három szintjét különíti el (Rasmussen 1983). A **készségalapú** viselkedés begyakorolt, tudatos figyelem nélkül zajló mozdulatsor (például egy alkatrész behelyezése). A **szabályalapú** viselkedés ismert helyzetben tárolt szabályokat alkalmaz („ha a jelzőfény piros, akkor…"). A **tudásalapú** viselkedés új, ismeretlen helyzetben tudatos elemzést és problémamegoldást igényel. A megcsúszások és a kihagyások a készségalapú szinten, a tévedések a szabály- és tudásalapú szinten jellemzők.

A besorolásnak gyakorlati jelentősége van, mert a különböző típusú hibák ellen különböző eszközök hatásosak. A megcsúszások ellen a tervezés (egyértelmű elrendezés, egymástól megkülönböztethető kezelőszervek, hibabiztos kialakítás), a kihagyások ellen az emlékeztetők és az ellenőrzőlisták, a tévedések ellen a képzés, a jobb információ és a döntéstámogatás, a szabályszegések ellen az eljárások észszerűsítése és a szervezeti kultúra.

13.3. A teljesítményt befolyásoló tényezők

Az emberi teljesítményt a helyzet számos jellemzője befolyásolja: az időnyomás, a fáradtság, a zaj, a megvilágítás, a visszajelzés minősége, a feladat ismerőssége, a képzettség, az eljárások minősége, a munkahelyi légkör. Az emberi megbízhatóság-elemzés módszerei ezeket teljesítményt befolyásoló tényezőként vagy hibát előidéző feltételként veszik figyelembe.

Ezek közül a tervező közvetlenül befolyásolja a legfontosabbakat. Norman a tervezés néhány alapelvével írja le, hogyan segíthet a környezet a helyes cselekvésben (Norman 2013):

- **jelzések** (signifiers): a tárgy vagy felület mutassa meg, mit lehet vele tenni és hogyan;
- **leképezés** (mapping): a kezelőszervek elrendezése feleljen meg a vezérelt dolgok elrendezésének;
- **visszajelzés** (feedback): a cselekvés eredménye azonnal és érthetően látsszon;
- **korlátok** (constraints): a fizikai, logikai vagy kulturális korlátok szűkítsék a lehetséges (és ezzel a hibás) cselekvéseket;

- **kényszerítő funkciók** (forcing functions): a helytelen cselekvés fizikailag ne legyen végrehajtható.

13.4. Emberi megbízhatóság-elemzés

Az **emberi megbízhatóság-elemzés** (human reliability analysis, HRA) célja az emberi hibák azonosítása, valószínűségük becslése és csökkentésük módjainak megtalálása. Kirwan gyakorlati kézikönyve szerint a HRA három fő lépésből áll: a lehetséges emberi hibák azonosításából, mennyiségi becsléséből és a csökkentésükre irányuló javaslatokból (Kirwan 1994). A mennyiségi eredmény bekerülhet a megbízhatósági és kockázatelemzésbe, például egy hibafa eseményeként (16. fejezet).

A legismertebb módszerek közül a THERP Swain és Guttmann nukleáris erőművi alkalmazásokra kidolgozott, részletes feladatlebontáson alapuló eljárása (Swain és Guttmann 1983). A **HEART** Williams egyszerűbb, gyorsan alkalmazható módszere (J. C. Williams 1988). A HEART három lépésben becsül:

1. a feladatot besoroljuk egy **általános feladattípusba**, amelyhez névleges hibavalószínűség tartozik. Például egy teljesen ismerős, jól megtervezett, gyakorlott, óránként többször ismétlődő rutinfeladaté 0,0004; egy rendszer előírt eljárással, némi ellenőrzéssel végzett visszaállításáé 0,003;
2. azonosítjuk a **hibát előidéző feltételeket**, mindegyikhez egy legnagyobb hatást jelentő szorzóval. Például az időhiány legfeljebb 11-szeresére, a gyenge, félreérthető vagy rosszul illeszkedő visszajelzés legfeljebb négyszeresére növelheti a hibavalószínűséget;
3. minden feltételhez megbecsüljük, mekkora hányadában érvényesül a helyzetben (0–1), és a hibavalószínűséget így számoljuk:

$$\mathrm{HEP} = \mathrm{GTT} \cdot \prod_i \left[(\mathrm{EPC}_i - 1) \cdot \mathrm{APOA}_i + 1 \right], \tag{13.1}$$

ahol GTT a feladattípus névleges valószínűsége, EPC_i az i-edik feltétel szorzója, APOA_i pedig a becsült érvényesülési hányada.

A HRA mennyiségi eredményeit óvatosan kell kezelni. A becslések nagymértékben az elemző ítéletén múlnak, és az abszolút értékek bizonytalansága egy nagyságrend is lehet. A módszer erőssége inkább az **összehasonlítás**: megmutatja, melyik tervezési változat vagy beavatkozás csökkenti leginkább a hibavalószínűséget, és melyik feltétel járul hozzá a legjobban (Kirwan 1994).

> **KoboCella: a fordítva behelyezett fedél**
>
> A KoboCella HTA-ja (11.5. szakasz) szerint a fedél ráhelyezése (2.3) a hibás behelyezés egyik legvalószínűbb helye. A fedél két irányban is ráhelyezhető, és fordított helyzetben a csavarok egy része meghúzható, de a tömítés nem zár. A csapat HEART-tal hasonlított össze három változatot.
> **Alapváltozat.** A feladat gyakorlott, óránként sokszor ismétlődő rutinfeladat (GTT = 0,0004). Két hibát előidéző feltétel érvényesül: az időnyomás (szorzó 11, a csapat becslése szerint 10%-ban érvényesül: $10 \cdot 0{,}1 + 1 = 2{,}0$) és a gyenge visszajelzés, mert a fordított fedél ránézésre alig különbözik a helyestől (szorzó 4, 50%-ban: $3 \cdot 0{,}5 + 1 = 2{,}5$). A becsült hibavalószínűség $0{,}0004 \cdot 2{,}0 \cdot 2{,}5 = 0{,}002$. Napi mintegy 610 motorral ez átlagosan 1,2 fordított fedelet jelent naponta.

Visszajelzéssel. Egy optikai érzékelő ellenőrzi a fedél helyzetét, és a HMI azonnal jelez. A gyenge visszajelzés feltétele megszűnik: $0{,}0004 \cdot 2{,}0 = 0{,}0008$, napi 0,5 eset, de ezeket a kezelő még a csavarozás előtt kijavítja.
Hibabiztos kialakítás. A befogóba aszimmetrikus illesztőcsapok kerülnek, így a fedél fordítva fizikailag nem helyezhető be. Ez a hibamód megszűnik.
A csapat a hibabiztos kialakítás mellett döntött, mert ez a hibát nem csak észleli, hanem lehetetlenné teszi, és kevesebbe került, mint az érzékelő. A számítás abszolút értékei bizonytalanok, de a sorrend egyértelmű volt, és a döntéshez ennyi elég.

13.5. Tervezés a hibák ellen

A hibák elleni védekezésnek több szintje van, és a hatékonyságuk eltérő:

1. **a hiba lehetetlenné tétele:** kényszerítő funkciók és hibabiztos kialakítás. A helytelen cselekvés fizikailag nem hajtható végre;
2. **a hiba azonnali észlelése:** visszajelzés, érzékelők, automatikus ellenőrzés, mielőtt a hiba következménye kialakulna;
3. **a következmény korlátozása:** a hiba visszafordíthatósága (visszavonás, megerősítés), tűrőképes kialakítás;
4. **az ember felkészítése:** képzés, eljárások, ellenőrzőlisták.

A sorrend fontos: a lista elején álló megoldások a rendszerbe épülnek, és nem függenek attól, hogy az ember éppen figyel-e. A lista végén álló megoldások szükségesek, de egyedül nem elegendők.

13.5.1. Hibabiztos kialakítás

A **poka-yoke** (japánul nagyjából „a figyelmetlenségi hiba elkerülése”) Shingo nevéhez fűződik, aki a nulla hibás gyártás egyik alapeszközévé tette (Shingo 1986). A poka-yoke megoldások vagy megakadályozzák a hibát (például csak egy irányban illeszthető alkatrész), vagy azonnal jelzik (például ha a szerelés után egy alkatrész a tálcán marad, a művelet nem fejeződhet be). Shingo az ellenőrzés helyét is hangsúlyozta: a hibát a forrásánál, a keletkezése pillanatában kell észlelni, nem a végellenőrzésen (Shingo 1986).

Norman a **kényszerítő funkciók** három típusát írja le (Norman 2013):

- **reteszelés** (interlock): a műveletek csak a helyes sorrendben hajthatók végre (például a mikrohullámú sütő nem indul nyitott ajtóval);
- **benntartás** (lock-in): a rendszer nem engedi idő előtt befejezni a műveletet (például a szoftver mentés nélküli kilépéskor megerősítést kér);
- **kizárás** (lockout): a rendszer megakadályozza a veszélyes helyre vagy állapotba jutást (például a karbantartás idejére lelakatolt energiaforrás).

ForgácsTech: kihagyások a műszakátadásban

A ForgácsTech késési okainak elemzésében (9.3. szakasz) az ötödik helyre a hiányos műszakátadás került. A szóbeli átadásnál a sürgős tudnivalók egy része rendszeresen elveszett: tipikus kihagyási hiba, amelyet a műszakvégi fáradtság és a félbeszakítások erősítenek. A csapat nem újabb utasítást adott ki („figyelmesebben kell átadni”), hanem a MES-be beépített egy rövid, strukturált átadólapot. A lapon az ütemező automatikusan megjeleníti a nyitott eltéréseket és a következő műszak határidős rendeléseit, a leköszönő műszakvezetőnek

pedig csak a megjegyzéseit kell hozzáírnia. A műszak lezárása az átadólap kitöltése nélkül nem lehetséges (benntartás), a következő műszakvezető pedig a lap elolvasását nyugtázza.

VárosiBringa: amikor a zár nem zárt

A VárosiBringa első próbaüzemében a felhasználók egy része úgy hagyta ott a kerékpárt, hogy az nem rögzült a dokkolóban: a kerékpárt nem tolták be ütközésig. A kölcsönzés nyitva maradt, a felhasználó díjat fizetett, a kerékpár pedig őrizetlenül állt. Ez megcsúszás: a felhasználó szándéka helyes volt, a végrehajtás pontatlan, és a hiba oka a gyenge visszajelzés volt. A javítás a visszajelzés erősítése lett: a dokkoló a sikeres rögzítéskor zöld fénnyel és hangjelzéssel, a sikertelennél villogó sárga fénnyel jelez, az alkalmazás pedig értesítést küld, ha a kerékpár öt másodpercen belül nem rögzült. A probléma a következő hónapban a töredékére csökkent.

13.6. Az emberi hiba a kockázatelemzésben és a szervezetben

Az emberi hibákat ugyanúgy be kell építeni a kockázatelemzésbe, mint a műszaki meghibásodásokat. Az FMEA-ban (15. fejezet) a HTA műveletei a folyamatlépések, a hibamódok pedig a lehetséges emberi hibák (kihagyás, rossz sorrend, rossz végrehajtás, késés). A hibafákban (16. fejezet) az emberi hibák alapeseményként szerepelhetnek. A rendszerelméleti biztonsági elemzés (17. fejezet) pedig kifejezetten azt vizsgálja, milyen irányítási döntések vezethetnek veszélyes állapothoz, akár emberi, akár gépi az irányító.

A szervezet tanulni tud a hibákból, ha a hibákról és a „majdnem bekövetkezett” eseményekről (near miss) értesül. Ehhez az kell, hogy a jelentés ne vonjon maga után automatikus büntetést. Reason szerint a rendszerszemléletű szervezetek egyik jellemzője az, hogy a jelentési kultúrát tudatosan építik, és megkülönböztetik a büntetendő, szándékos szabályszegést a büntetést nem érdemlő emberi hibától (Reason 2000). Ezt a megkülönböztetést szokás méltányos kultúrának (just culture) nevezni.

Az emberi teljesítmény mint erőforrás

A biztonságtudomány újabb irányzatai arra is felhívják a figyelmet, hogy az ember nem csak hibaforrás, hanem a rendszer rugalmasságának forrása is. A legtöbb nap azért telik el baj nélkül, mert az emberek folyamatosan észlelik és korrigálják a rendszer apró hiányosságait, alkalmazkodnak a váratlan helyzetekhez. Ha egy fejlesztés csak az emberi hibák kiküszöbölésére törekszik, és közben elveszi az emberi beavatkozás lehetőségét, ezt a rugalmasságot is elveszítheti. A tervezési cél tehát kettős: a hibák elleni védelem, és annak megőrzése, ami az emberben a rendszer javára működik.

„Az operátor hibázott"

A balesetvizsgálati jelentések egyik leggyakoribb, és legkevésbé hasznos megállapítása. Ha a vizsgálat itt megáll, és a válasz az operátor figyelmeztetése vagy egy újabb utasítás, a rendszer nem változik. A helyes kérdés: miért tűnt észszerűnek a cselekvés abban a helyzetben, és milyen lappangó feltételek (tervezés, eljárás, időnyomás, képzés) tették lehetővé, hogy a hiba következménnyel járjon?

Összefoglalás

- A rendszerszemléletű megközelítés szerint az emberi hiba inkább következmény, mint ok: a munkahely és a szervezet rejtett hiányosságaiból fakad. A svájcisajt-modellben a baj akkor következik be, ha a védelmi rétegek lyukai egy vonalba kerülnek.

- Reason osztályozása szerint a nem biztonságos cselekvés lehet megcsúszás, kihagyás, tévedés vagy szabályszegés. Rasmussen szerint a viselkedés készség-, szabály- vagy tudásalapú. Az eltérő típusok ellen eltérő eszközök hatásosak.
- A teljesítményt az időnyomás, a visszajelzés, a fáradtság, a feladat ismerőssége és sok más tényező befolyásolja. Norman tervezési elvei (jelzések, leképezés, visszajelzés, korlátok, kényszerítő funkciók) a helyes cselekvést segítik.
- Az emberi megbízhatóság-elemzés azonosítja, számszerűsíti és csökkenti az emberi hibákat. A HEART a feladattípus névleges valószínűségét a hibát előidéző feltételekkel módosítja. Az abszolút értékek bizonytalanok; a módszer a változatok összehasonlítására való.
- A hibák elleni védekezés hatékonysági sorrendje: lehetetlenné tétel (poka-yoke, kényszerítő funkciók), azonnali észlelés, a következmény korlátozása, felkészítés. Az emberi hibák az FMEA, a hibafa és a biztonsági elemzés részei; a szervezeti tanuláshoz méltányos jelentési kultúra kell.

Önellenőrzés

1. Mi a különbség a személyközpontú és a rendszerszemléletű megközelítés között? Melyik vezet tartós javuláshoz, és miért?
2. Sorold be a következő eseteket Reason kategóriái szerint: (a) a kezelő a szokott helyen nyomja meg a gombot, de az új felületen ott a „törlés” van; (b) egy telefonhívás után a karbantartó elfelejti visszaszerelni a védőburkolatot; (c) a műszakvezető egy szokatlan hibánál a megszokott újraindítást alkalmazza, de az itt nem segít; (d) a dolgozók rendszeresen kikerülik a lassú biztonsági ellenőrzést.
3. Egy feladat névleges hibavalószínűsége 0,003. Két hibát előidéző feltétel érvényesül: 11-es szorzó 20%-ban és 3-as szorzó 50%-ban. Mennyi a becsült hibavalószínűség?
4. Miért értékes a HEART akkor is, ha az abszolút értékei bizonytalanok?
5. Adj példát a reteszelésre, a benntartásra és a kizárásra egy ipari környezetből!
6. Miért jobb egy hibabiztos kialakítás, mint egy figyelmeztető felirat?
7. Hogyan építenéd be a KoboCella HTA-jának lépéseit egy folyamat-FMEA-ba?

Megoldás.

1. A személyközpontú megközelítés az egyént hibáztatja, és az egyént próbálja megváltoztatni (figyelmeztetés, fegyelmezés); a rendszerszemléletű a hibát a feltételek következményének tekinti, és a feltételeket változtatja meg. Az utóbbi vezet tartós javuláshoz, mert a következő ember ugyanabban a helyzetben már nem kerül ugyanabba a csapdába.
2. (a) megcsúszás (figyelmi, a megszokott mozdulat rossz helyen); (b) kihagyás (emlékezeti, félbeszakítás után); (c) szabályalapú tévedés (jó szabály rossz helyzetben); (d) rutinszerű szabályszegés.
3. $(11-1)\cdot 0{,}2+1=3{,}0$; $(3-1)\cdot 0{,}5+1=2{,}0$; $\mathrm{HEP}=0{,}003\cdot 3{,}0\cdot 2{,}0=0{,}018$.
4. Mert a változatok összehasonlításához a relatív különbségek elegendők, és ezek kevésbé érzékenyek a becslési bizonytalanságra. A módszer azt is megmutatja, melyik feltétel járul hozzá leginkább a hibához, vagyis hol érdemes beavatkozni.
5. Reteszelés: a présgép csak akkor indul, ha a védőajtó zárva van. Benntartás: a gépvezérlő nem engedi leállítani a szerszámcserét, amíg minden rögzítőcsavar nincs meghúzva. Kizárás: a karbantartás idejére lelakatolt villamos főkapcsoló.
6. Mert a felirat az ember figyelmén múlik, amely fáradtság, időnyomás és rutin mellett kihagy; a hibabiztos kialakítás a helytelen cselekvést fizikailag lehetetlenné teszi, függetlenül attól, hogy az ember figyel-e.
7. A HTA minden művelete (például 2.2, 2.3) egy folyamatlépés lesz. Minden lépéshez felsoroljuk a lehetséges emberi hibamódokat (kihagyás, rossz sorrend, rossz végrehajtás, például fordított fedél), azok hatását (tömítetlenség), okát (gyenge visszajelzés, időnyomás) és a meglévő ellenőrzést, majd értékeljük a súlyosságot, a gyakoriságot és az észlelhetőséget.

Projektlépés: emberi hibák

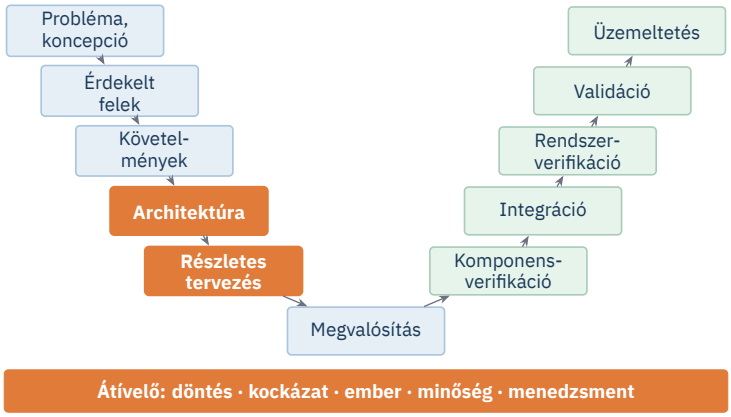
Saját fejlesztési feladatodhoz:

1. a HTA legalább öt műveletéhez sorold fel a lehetséges emberi hibákat Reason kategóriái szerint;
2. a két legsúlyosabb következményűhöz azonosítsd a hibát előidéző feltételeket, és becsüld meg HEART-tal a hibavalószínűséget legalább két tervezési változatban;
3. javasolj legalább egy hibabiztos (poka-yoke) megoldást vagy kényszerítő funkciót;
4. vidd át az emberi hibamódokat a projekt FMEA-jába (15. fejezet).

Továbbolvasás

- Reason (2000): rövid, olvasmányos összefoglaló a személyközpontú és a rendszerszemléletű megközelítésről és a svájcisajt-modellről.
- Reason (1990): az emberi hiba osztályozásának és elméletének alapműve.
- Norman (2013): a tervezés hatása az emberi cselekvésre és hibákra; mérnököknek is kötelező olvasmány.
- Kirwan (1994): az emberi megbízhatóság-elemzés gyakorlati kézikönyve.
- Shingo (1986): a poka-yoke és a forrásnál végzett ellenőrzés eredeti leírása.
- Dekker (2014): gyakorlati útmutató az emberi hiba rendszerszemléletű vizsgálatához.

14. Kockázatkezelés



Tanulási eredmények

A fejezet feldolgozása után képes leszel:

- megkülönböztetni a veszélyt, a veszélyes eseményt, a kockázatot, a maradó kockázatot és a biztonságot;
- leírni a kockázatkezelés folyamatát az ISO 31000 és az ISO/IEC/IEEE 15288 alapján;
- kockázati mátrixot szerkeszteni és kalibrálni, és alkalmazni az ALARP-elvet;
- előzetes veszélyelemzést (PHA) és energia–gát elemzést végezni;
- a kockázatcsökkentő intézkedéseket hatékonyságuk szerint rangsorolni;
- kockázati nyilvántartást vezetni, és a kockázatokat a fejlesztés során folyamatosan követni.

Minden fejlesztés bizonytalanságban zajlik. Nem tudjuk biztosan, hogy a kiválasztott koncepció teljesíti-e a követelményeket, hogy a beszállító időben szállít-e, hogy a felhasználók elfogadják-e az új rendszert, és azt sem, hogy a rendszer üzem közben okozhat-e sérülést vagy kárt. A kockázatkezelés nem a bizonytalanság megszüntetése, hanem annak rendszeres feltárása, értékelése és tudatos kezelése. Az ISO/IEC/IEEE 15288 a technikai irányítás átívelő folyamatai között tárgyalja: a kockázatkezelés a fejlesztés teljes életciklusán végigkíséri a döntéseket (ISO/IEC/IEEE 2023, 6.3.4. szakasz; NASA 2016, 6.4. szakasz).

A fejezet négy kérdésre épül:

- Mit nevezünk kockázatnak, és mikor biztonságos egy rendszer?
- Hogyan épül fel a kockázatkezelés folyamata?
- Hogyan értékeljük és rangsoroljuk a kockázatokat?
- Hogyan azonosítjuk a veszélyeket, és mivel csökkentjük a kockázatot?

14.1. Veszély, kockázat, biztonság

A biztonsági szabványok közös fogalomtára szerint a **kár** (harm) a személyek sérülése, egészségkárosodása, illetve a vagyon vagy a környezet károsodása; a **veszély** (hazard) a kár lehetséges forrása; a **veszélyes esemény** az az esemény, amely kárt okozhat; a **kockázat** (risk) pedig a kár bekövetkezési valószínűségének és súlyosságának kombinációja (ISO 2014). A műszaki rendszerek tipikus veszélyhelyzete az, hogy egy eszköz, anyag vagy berendezés a környezeténél magasabb energiaszinten van, és a tervezett üzemállapottól eltérés lép fel (14.4. szakasz).

A kockázatot gyakran egyetlen szorzattal fejezik ki:

$$R = P \cdot S, \tag{14.1}$$

ahol P a veszélyes esemény valószínűsége vagy gyakorisága egy adott expozíciós időre, S a következmény súlyossága. A valószínűség mindig egy expozíciós időintervallumhoz tartozik: egy „évi 0,01” és egy „élettartam alatt 0,01” valószínűség teljesen más kockázatot jelent (Ericson 2015).

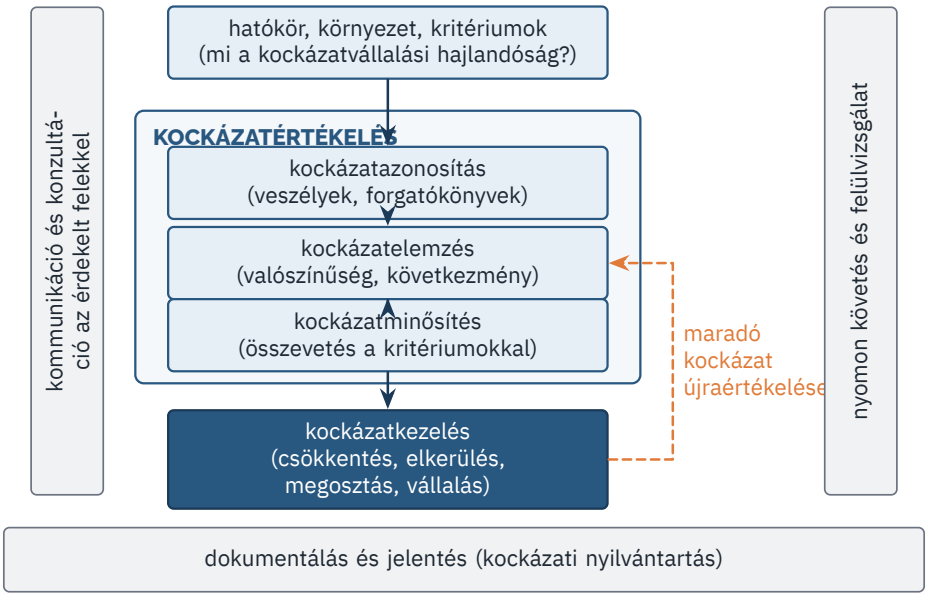
A projektmenedzsment tágabb kockázatfogalmat használ: a kockázat annak lehetősége, hogy a projekt nem teljesíti kitűzött céljait a biztonság, a műszaki teljesítmény, a költség vagy az ütemterv terén. Kaplan és Garrick klasszikus meghatározása szerint a kockázatot hármasok halmaza írja le: a nem kívánt állapothoz vezető **forgatókönyv**, ennek **valószínűsége** és a **következmény** súlyossága (Kaplan és Garrick 1981). A meghatározás azért hasznos, mert arra kényszerít, hogy minden kockázatnál megkérdezzük: mi történhet, mennyire valószínű, és mi lenne a következménye? Az ISO 31000 még általánosabb: számára a kockázat a bizonytalanság hatása a célokra, amely lehet kedvező is (ISO 2018b). A fejezetben elsősorban a kedvezőtlen kimenetekkel foglalkozunk, két területen: a rendszer üzemeltetésének **biztonsági kockázataival** és a fejlesztési projekt **technikai, költség- és ütemezési kockázataival**.

A kockázatok kezelésük szempontjából is csoportosíthatók. Az **azonosított kockázatot** valamilyen elemzéssel feltártuk; a **nem azonosított** kockázatról nem tudunk. Az azonosított kockázat lehet **elfogadható**, amely további csökkentés nélkül is vállalható, vagy **elfogadhatatlan**, amelyet csökkenteni kell. A **maradó kockázat** a kockázatcsökkentés után megmaradó kockázat. A szabványok szerint a **biztonság** az elfogadhatatlan kockázattól való mentesség (ISO 2014). Ez fontos különbség: a biztonság nem a kockázat teljes hiánya, hanem az, hogy a megmaradt kockázat a társadalom, a jogszabályok és az érintettek által elfogadott határon belül van.

14.2. A kockázatkezelés folyamata

Az ISO 31000 szerint a kockázatkezelés folyamata a hatókör, a környezet és a kockázati kritériumok meghatározásával kezdődik. Ezt követi a **kockázatértékelés** három lépése (azonosítás, elemzés, minősítés), majd a **kockázatkezelés**, vagyis az intézkedések kiválasztása és végrehajtása. A folyamatot végigkíséri a kommunikáció és konzultáció az érdekelt felekkel, a nyomon követés és felülvizsgálat, valamint a dokumentálás (ISO 2018b) (14.1. ábra).

Az ISO/IEC/IEEE 15288 kockázatkezelési folyamata öt tevékenységből áll: a kockázatkezelés megtervezése, a kockázati profil kezelése, a kockázatok elemzése, a kockázatok



14.1. ábra. A kockázatkezelés folyamata az ISO 31000 szerkezetét követve.

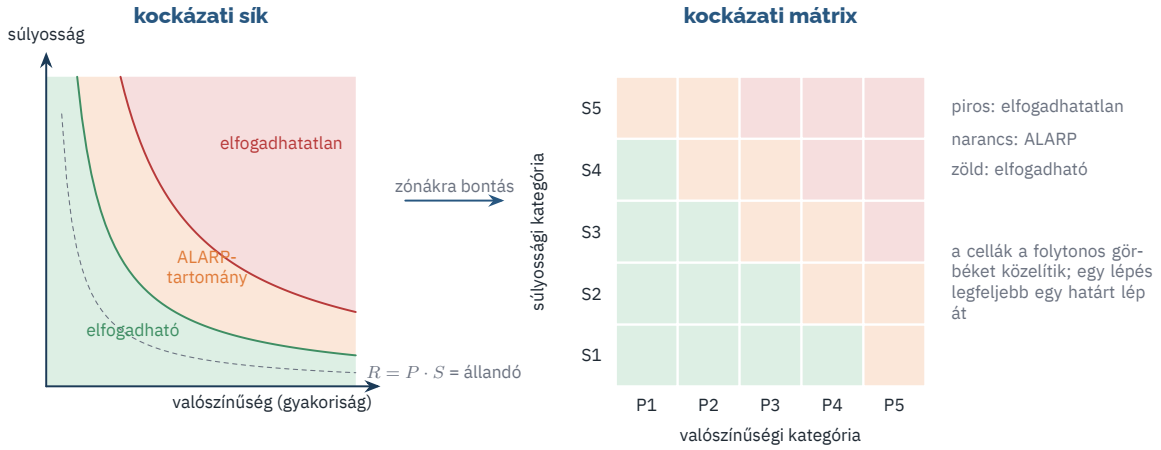
kezelése és a kockázatok figyelése (ISO/IEC/IEEE 2023, 6.3.4. szakasz). A gyakorlatban két, egymást kiegészítő szemlélet fonódik össze. A **kockázatinformált döntéshozatal** (risk-informed decision making, RIDM) a nagy, irányt szabó tervezési döntéseket támogatja: az alternatívák kiválasztását a kockázati és bizonytalansági információk alapján (14.6. szakasz). A **folyamatos kockázatkezelés** (continuous risk management, CRM) ezután a fejlesztés és a megvalósítás során kezeli az egyes kockázatokat. A Carnegie Mellon Egyetem szoftvermérnöki intézetének (SEI) útmutatója szerint a CRM ciklusa az azonosítás, az elemzés, a tervezés, a nyomon követés és a beavatkozás, és mindegyiket a kommunikáció kíséri (Dorofee és mtsai. 1996).

A kockázatokat a projektben **kockázati nyilvántartás** (kockázati regiszter) rögzíti. Egy jó bejegyzés a kockázatot feltétel–következmény formában fogalmazza meg (Dorofee és mtsai. 1996): „mivel [feltétel, amely ma igaz], fennáll a lehetősége, hogy [esemény], ami [következmény a célokra]”. Ez a megfogalmazás elválasztja a tényt (a feltételt) a bizonytalan eseménytől, és megmutatja, hol lehet beavatkozni. A bejegyzéshez tartozik a valószínűség és a következmény becslése, a felelős, a kezelési mód, a beavatkozást kiváltó küszöb és az állapot. A kockázatok állapotáról rendszeresen, a küszöbhöz közeledő kockázatokról sűrűbben kell jelenteni, és a kockázatok állapotát minden életciklus-felülvizsgálaton (2.5. szakasz) érdemes áttekinteni.

14.3. A kockázat minősítése: kockázati mátrix és ALARP

A 14.1. egyenlet szerint az azonos kockázatú pontok a valószínűség–súlyosság síkon hiperbolákon, úgynevezett **izokockázati görbéken** helyezkednek el. A görbék a síkot tartományokra osztják: elfogadható, feltételesen elfogadható és elfogadhatatlan kockázatra (14.2. ábra). Mivel a valószínűség és a súlyosság ritkán ismert pontosan, a gyakorlatban a síkot kategóriákra bontjuk. Így jön létre a **kockázati mátrix**, amelynek cellái a folytonos görbéket közelítik (Cox 2008).

Cox elemzése szerint a kockázati mátrixok könnyen félrevezetnek: kis felbontásuk miatt



14.2. ábra. A kockázati sík izokockázati görbékkel és a belőle képzett 5 × 5-ös kockázati mátrix.

a valóban eltérő kockázatokat azonos cellába sorolhatják, vagy a sorrendjüket meg is fordíthatják (Cox 2008). Cox elemzése és a rendszerbiztonsági gyakorlat (U.S. Department of Defense 2012; Goldberg és mtsai. 1994, 3.1. szakasz) alapján a mátrix szerkesztésének gyakorlati szabályai:

- a szomszédos valószínűségi kategóriák nagyságrendben (tízszeres szorzóval) különbözzenek;
- ne legyen túl sok cella, mert a szubjektív értékelés finomabb felbontása csak zavart okoz;
- a zónák legyenek folytonosak: egy lépés a mátrixban legfeljebb egy zónahatárt lépjen át;
- csak annyi zóna legyen, ahány különböző kezelési mód létezik (például elfogadhatatlan, felmentéssel elfogadható, rutinszerűen elfogadható);
- a mátrixot egy ismert, mindenki által értett forgatókönyvvel kell **kalibrálni**;
- a zónahatárokat nem az elemző, hanem a vezetés állapítja meg és hagyja jóvá.

A kockázatot a **legrosszabb hihető** (worst credible), nem a legrosszabb elképzelhető következményre kell értékelni (Ericson 2015). A mátrixok kategóriái gyakran az amerikai védelmi szabvány rendszerbiztonsági gyakorlatából származnak, amely négy súlyossági kategóriát (katasztrofális, kritikus, mérsékelt, elhanyagolható) és hat valószínűségi szintet (gyakori, valószínű, alkalmi, ritka, valószínűtlen, kizárt) használ (U.S. Department of Defense 2012). A könyv eseteiben egy 5 × 5-ös változatot használunk (14.1. táblázat).

14.1. táblázat. A könyv eseteiben használt súlyossági és valószínűségi skála. A zónák: $S + P \geq 8$ elfogadhatatlan, $6 \leq S + P \leq 7$ ALARP, $S + P \leq 5$ elfogadható.

Szint	Súlyosság (személy)	Valószínűség (egy cellára)
5	halál vagy maradandó egészségkárosodás	gyakori: legalább havonta (> 10/év)
4	súlyos sérülés, kórházi ellátás, tartós munkaképtelenség	valószínű: évente néhányszor (1–10/év)
3	orvosi ellátást igénylő sérülés	alkalmi: néhány évente (0,1–1/év)
2	elsősegéllyel ellátható sérülés	ritka: az élettartam alatt előfordulhat (0,01–0,1/év)
1	sérülés nélküli esemény	valószínűtlen: nem várható (< 0,01/év)

A feltételesen elfogadható tartomány értelmezésére a brit munkavédelmi hatóság **ALARP-elve** (as low as reasonably practicable, olyan alacsony, amennyire észszerűen megvalósítható) terjedt el (14.3. ábra). A felső, elfogadhatatlan tartományban a kockázat a haszontól függetlenül nem vállalható. A középső, tűrhető tartományban a kockázatot csökkenteni kell mindaddig, amíg a további csökkentés költsége nem válik aránytalanul nagyra a hasznához képest. Az alsó tartományban a kockázat általánosan elfogadható, de fenn kell tartani ezt a szintet (Health and Safety Executive 2001). Az ALARP tehát nem azt jelenti, hogy „elég jó", hanem fordított bizonyítási terhet: a tervezőnek kell igazolnia, hogy a további csökkentés már nem észszerű.



14.3. ábra. Az ALARP-elv három tartománya.

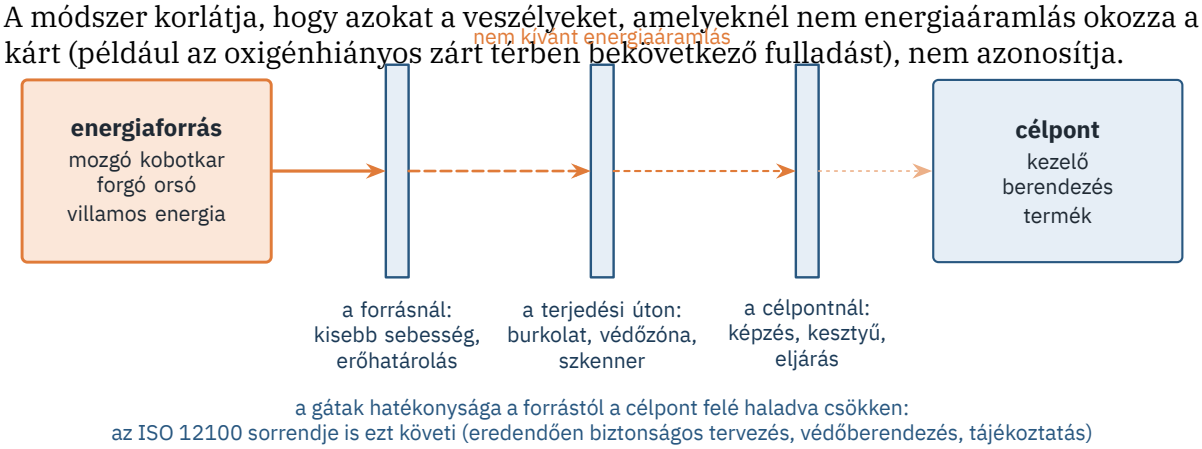
A kockázati mátrix egyszerű és szemléletes, de korlátai is vannak. Cox matematikai elemzése szerint a mátrixok felbontása gyenge: a véletlenszerűen választott kockázatpároknak csak kis hányadát tudják egyértelműen sorba rendezni, mennyiségileg nagyon különböző kockázatokat ugyanabba a cellába sorolhatnak, és előfordulhat, hogy a kisebb kockázat kap magasabb minősítést (Cox 2008). A mátrix ezért a kockázatok **előszűrésére** és a kommunikációra jó eszköz. Ha két intézkedés vagy két tervváltozat között kell dönteni, és vannak adatok, a mennyiségi elemzés (megbízhatósági modell, hibafa, eseményfa, 16. fejezet) megbízhatóbb.

14.4. Veszélyazonosítás: PHA és energia–gát elemzés

A kockázat csak akkor értékelhető, ha a veszélyt azonosítottuk. Az **előzetes veszélyelemzés** (preliminary hazard analysis, PHA) a fejlesztés korai szakaszában, a koncepció ismeretében készül. Célja a veszélyek leltárba vétele, a célpontok (személy, berendezés, termék, környezet) meghatározása, a kockázat első értékelése és az ellenintézkedések kijelölése. A PHA erőssége, hogy már a koncepció szakaszában leltárt ad a veszélyekről és az ellenintézkedésekről; korlátja, hogy nem foglalkozik az egyidejűleg fennálló hibák kombinációival, ezért a részletesebb elemzések (FMEA, hibafa) nem maradhatnak el (Ericson 2015; Goldberg és mtsai. 1994, 3.2. szakasz).

A veszélyek azonosításának hatékony vezérfonala az **energia**. Haddon klasszikus megfigyelése szerint a sérülések nagy része energiaátadásból ered: a mozgási, helyzeti, hő-, villamos, kémiai és sugárzási energia olyan mértékben vagy ütemben jut el a célpontig, amelyet az nem tud elviselni. Haddon tíz stratégiát írt le a kár csökkentésére, a veszély létrehozásának elkerülésétől és mennyiségének csökkentésétől a forrás és a célpont térbeli vagy időbeli elválasztásán és a gátakon át a célpont ellenállóbbá tételéig és a kár enyhítéséig (Haddon 1973). Az **energia–gát elemzés** ezt a gondolatot követi: számba veszi a rendszer energiaforrásait, a célpontokat és a közöttük lévő gátakat, és megvizsgálja, hogy

a gátak elegendőek-e (Ericson 2015; Goldberg és mtsai. 1994, 3.3. szakasz) (14.4. ábra). A módszer korlátja, hogy azokat a veszélyeket, amelyeknél nem energiaáramlás okozza a kárt (például az oxigénhiányos zárt térben bekövetkező fulladást), nem azonosítja.



14.4. ábra. Az energia–gát szemlélet: a nem kívánt energiaáramlást a forrásnál, a terjedési úton és a célpontnál elhelyezett gátak állítják meg.

KoboCella: előzetes veszélyelemzés

A KoboCella csapata a koncepció kiválasztása után PHA-t készített. Az energiaforrások listája (a kobot mozgó karja, a csavarozó orsó, a befogó szorítóereje, a villamos energia, a motorház helyzeti energiája, a vezérlőszoftver) és a kezelő HTA-ja (11. fejezet) adta a kiindulópontot. A 14.2. táblázat a PHA egy részletét mutatja.

14.2. táblázat. A KoboCella előzetes veszélyelemzésének részlete. S: súlyosság, P: valószínűség a 14.1. táblázat skáláin.

Veszély	Kiinduló S/P	Ellenintézkedés	Maradó S/P
a kobotkar ütközése a kezelővel	3/4 (ALARP)	erő- és teljesítményhatárolás, sebesség- és távolságfelügyelet lézerszkennerrel, legfeljebb 250 mm/s a közös térben (R-11)	2/3 (elfogadható)
kéz becsípődése a befogó és a fedél között	4/3 (ALARP)	a befogó csak zónaelhagyás után zár; korlátozott szorítóerő	3/2 (elfogadható)
forgó orsó, ruha vagy kesztyű behúzása	3/3 (ALARP)	burkolat; az orsó csak zónaelhagyás után indul	3/1 (elfogadható)
áramütés a vezérlőszekrényben karbantartáskor	5/2 (ALARP)	érintésvédelem, kizárási eljárás, csak képzett villamos szakember	5/1 (ALARP, indokolva)
ismétlődő törzshajlítás (ergonómia)	3/5 (elfogadhatatlan)	munkamagasság a kezelőhöz állítható, legfeljebb 20° törzshajlítás (R-05)	2/3 (elfogadható)
váratlan indulás szoftverhiba miatt	4/2 (ALARP)	a zónafelügyelet független biztonsági vezérlésen fut	4/1 (elfogadható)

Az elemzés két fontos tanulsággal járt. Először: a legnagyobb kiinduló kockázat nem a robottól, hanem a jelenlegi kézi szerelés ergonómiai terheléséből származott; a cella ezt csökkenti. Másodszor: az áramütés kockázata a csökkentés után is az ALARP-tartományban maradt, mert a súlyosság nem csökkenthető. A csapat ezért írásban indokolta, miért nem észszerű a további csökkentés (a karbantartási munkák ritkák, a villamos szakemberek képzettek, a kizárási eljárás bevett), és a vezetés ezzel a maradó kockázatot elfogadta.

14.5. Kockázatcsökkentés

A gépek biztonságának alapszabványa háromlépéses kockázatcsökkentést ír elő, rögzített sorrendben. Az első lépés az **eredendően biztonságos tervezés**: a veszély megszüntetése vagy csökkentése magával a kialakítással (kisebb energia, kisebb erő, a veszélyes zóna elkerülése). A második a **műszaki védőintézkedések**: védőburkolatok és védőberendezések (reteszelés, fényfüggöny, lézerszkenner). A harmadik a **használati információ**: figyelmeztetések, jelzések, utasítások, képzés (ISO 2010). A sorrend nem véletlen: az eredendően biztonságos megoldás az ember figyelmétől és a védőberendezés működőképességétől függetlenül hat, a tájékoztatás viszont csak akkor, ha az ember tudja, emlékszik rá és be is tartja. Ugyanezt a logikát követi az energia–gát szemlélet: a forrásnál alkalmazott gát hatékonyabb, mint a célpontnál alkalmazott (14.4. ábra), és az emberi hibák elleni védekezés hatékonysági sorrendje is (13. fejezet).

Az ISO 31000 általánosabban írja le a kockázatkezelési lehetőségeket: a kockázat elkerülése (a tevékenység elhagyásával), a kockázati forrás megszüntetése, a valószínűség vagy a következmény megváltoztatása, a kockázat megosztása (például szerződéssel vagy biztosítással), valamint a kockázat tudatos vállalása; kedvező kimenetnél akár a kockázat növelése is, egy lehetőség kihasználására (ISO 2018b). A projektkockázatoknál mindegyik lehetőség szóba jöhet, a biztonsági kockázatoknál a megosztás nem csökkenti a személyeket érintő kárt, ezért ott a háromlépéses sorrend az irányadó.

Minden kockázatcsökkentő intézkedés után újra kell értékelni a kockázatot (14.1. ábra), mert az intézkedés új veszélyt is behozhat. A KoboCellában például a lézerszkenner miatt a kezelő a zóna elhagyása nélkül nem indíthatja a ciklust; ha ez lassítja a munkát, a kezelők megkerülhetik a védelmet. Ezt a lehetőséget a csapat a HTA-ban szabályszegésként azonosította, és a megoldást is a tervezésben kereste: a teljes sebesség csak zónaelhagyás után engedélyezett, így a megkerülés nem gyorsítja a munkát.

VárosiBringa: az akkumulátor mint energiaforrás

A VárosiBringa dokkolóállomásain az elektromos kerékpárok akkumulátorai töltődnek, köztéren, felügyelet nélkül. Az energia–gát elemzés legfontosabb energiaforrása a lítiumion-akkumulátor kémiai és villamos energiája lett, célpontként a járókelők, a dokkoló és a szomszédos kerékpárok. A csapat a gátakat Haddon stratégiái szerint rendezte: a töltés áramának korlátozása (a forrásnál), az akkumulátorkezelő rendszer hőmérséklet-figyelése és a töltés leállítása rendellenesség esetén (a felszabadulás megakadályozása), fémházas akkumulátortartó és távolság a dokkolóhelyek között (elválasztás), valamint automatikus riasztás az üzemeltetőnek (a kár enyhítése). A súlyosság itt nem csökkenthető a tájékoztatással, ezért a csapat a gátakat a forrás közelében sűrítette.

14.6. Kockázatinformált döntés és folyamatos kockázatkezelés

A kockázat nemcsak a kész rendszer biztonsági tulajdonsága, hanem a fejlesztési döntések egyik szempontja is. A kockázatinformált döntéshozatalban a tervváltozatokat nemcsak a várható teljesítményük, hanem a teljesítmény bizonytalansága és a kudarc valószínűsége szerint is összevetjük (Parnell és mtsai. 2013). A 10. fejezet Monte Carlo-elemzései (a súlyozott értékelés és a költség–haszon elemzés bizonytalansága) éppen ezt teszik: egy változat, amely átlagosan jobb, de jelentős valószínűséggel kudarcot vall, kevésbé vonzó lehet, mint egy kissé gyengébb, de biztosabb változat.

A kiválasztás után a folyamatos kockázatkezelés veszi át a szerepet. A kockázati nyilvántartás élő dokumentum: új kockázatok kerülnek bele, a régiek pontszáma az intézkedések hatására csökken, a bekövetkezett kockázatokból problémák lesznek, a megszűntek lezárulnak. A kockázatok alakulását sok projekt idősoros ábrán követi: a tervezett csökkentési lépések és a tényleges állapot összevetése megmutatja, ha egy kockázat nem a tervek szerint csökken.

ForgácsTech: a MES-bevezetés kockázatai

A ForgácsTech csapata a fokozatos MES-bevezetés (M3, 10. fejezet) első ütemének indításakor kockázati nyilvántartást nyitott. Három bejegyzés:

- „Mivel a régi gépek egy részének nincs digitális interfésze, fennáll a lehetősége, hogy a gépi adatgyűjtés csak kiegészítő hardverrel oldható meg, ami az első ütemet 2–3 hónappal késlelteti.” Valószínűség 4, következmény 3; kezelés: a gépek felmérése a szerződéskötés előtt, és a kiegészítő hardver beárazása tartalékként.
- „Mivel a sürgős rendelések előresorolása eddig szóban történt, fennáll a lehetősége, hogy a műszakvezetők megkerülik az ütemezőt, így a határidő-teljesítés nem javul.” Valószínűség 3, következmény 4; kezelés: a műszakvezetők bevonása az ütemező szabályainak kialakításába, és az átsorolás naplózása (12. fejezet).
- „Mivel a platformot egyetlen beszállító fejleszti, fennáll a lehetősége, hogy a későbbi ütemek ára vagy ütemezése a beszállító helyzetétől függ.” Valószínűség 2, következmény 4; kezelés: nyílt adatexport és a második ütem árának rögzítése a keretszerződésben (kockázatmegosztás).

A nyilvántartást a csapat havonta frissítette. A második kockázat a próbaüzem első hónapjában a vártnál lassabban csökkent: a naplók szerint az átsorolások harmada a rendszeren kívül történt. A csapat ezért a beavatkozási küszöb elérésekor a tervezett tartalékintézkedést indította el: a napi műszakindító megbeszélés napirendjére került az előző nap átsorolásainak áttekintése.

Kockázat és ellenálló képesség

A klasszikus kockázatkezelés előre azonosított veszélyekből és forgatókönyvekből indul. Az összetett, sok szereplős rendszerekben azonban mindig lesznek előre nem látott helyzetek. Ezért a kockázatkezelést egyre gyakrabban egészíti ki az **ellenálló képesség** (reziliencia) szemlélete: a rendszer képessége, hogy a váratlan zavarokat észlelje, elviselje, alkalmazkodjon hozzájuk és helyreálljon. A rendszermérnök számára ez olyan tervezési jellemzőket jelent, mint a tartalékok, a leromlott üzemmódok (a KoboCella UM-7 üzemmódja), a moduláris felépítés és a gyors helyreállítást támogató megfigyelhetőség. A rendszerelméleti biztonságelemzés (17. fejezet) ezt a gondolatot viszi tovább.

A kockázati mátrix mint döntési automata

Gyakori hiba, hogy a mátrix színe „dönt”: ami zöld, azzal nem foglalkozunk, ami piros, azt bármi áron csökkentjük. A mátrix kategóriái durvák, a besorolás szubjektív, és a cellahatárok

közelében kis becslési eltérés is más szint ad (Cox 2008). A mátrix arra való, hogy a figyelmet a fontos kockázatokra irányítsa és a kommunikációt egységessé tegye. A döntéshez az ALARP-elv szerinti indoklás, és ahol lehet, mennyiségi elemzés kell.

Összefoglalás

- A veszély a kár lehetséges forrása, a kockázat a kár valószínűségének és súlyosságának kombinációja. A biztonság nem a kockázat hiánya, hanem az elfogadhatatlan kockázattól való mentesség; a maradó kockázatot tudatosan kell elfogadni.
- A kockázatkezelés folyamata: hatókör és kritériumok, azonosítás, elemzés, minősítés, kezelés, a kommunikáció és a felülvizsgálat folyamatos kíséretében. A kockázatinformált döntéshozatal (RIDM) a nagy tervezési döntéseket, a folyamatos kockázatkezelés (CRM) az egyes kockázatok követését szolgálja.
- A kockázati mátrix a kockázati sík izokockázati görbéit közelíti. Szerkesztésének szabályai: nagyságrendi valószínűségi lépések, kevés cella és zóna, folytonos zónahatárok, kalibrálás; a határokat a vezetés hagyja jóvá. Az ALARP-elv szerint a tűrhető tartományban a kockázatot addig kell csökkenteni, amíg a csökkentés költsége nem aránytalan.
- A PHA a veszélyek korai leltára ellenintézkedésekkel; az energia–gát elemzés az energiaforrások, a célpontok és a gátak vizsgálatával azonosítja a veszélyeket.
- A kockázatcsökkentés sorrendje: eredendően biztonságos tervezés, műszaki védőintézkedések, használati információ. Minden intézkedés után újra kell értékelni a kockázatot.

Önellenőrzés

1. Mi a különbség a veszély és a kockázat között? Adj példát mindkettőre a saját környezetedből!
2. Miért kell a valószínűséghez expozíciós időt rendelni?
3. Egy kockázati mátrixban a valószínűségi kategóriák „ritka: évente egyszer” és „alkalmi: évente kétszer”. Mi a probléma ezzel a skálával?
4. Mit jelent az ALARP-elv, és kit terhel a bizonyítás?
5. Miért korlátozott egy kockázati mátrix felbontása? Mikor van szükség mennyiségi elemzésre?
6. Végezz rövid energia–gát elemzést egy háztartási kenyérpirítóra!
7. Miért a használati információ az utolsó a kockázatcsökkentés sorrendjében?
8. Fogalmazz meg feltétel–következmény formájú kockázati állítást egy félévi projektfeladat egyik kockázatára!

Megoldás.

1. A veszély a kár lehetséges forrása (például a forgó köszörűkorong), a kockázat a kár valószínűségének és súlyosságának kombinációja (például annak esélye, hogy egy évben a korong szilánkja szemsérülést okoz, és ennek súlyossága).
2. Mert ugyanaz az esemény egy óra alatt, egy év alatt vagy a teljes élettartam alatt nagyon különböző valószínűséggel következik be; expozíciós idő nélkül a valószínűségek nem vethetők össze, és a kockázat nem értelmezhető.
3. A két kategória között csak kétszeres különbség van, így a besorolás a becslési bizonytalanságon belül esik; a szomszédos kategóriák nagyságrendben különbözzenek.
4. A tűrhető tartományban a kockázatot addig kell csökkenteni, amíg a további csökkentés költsége (pénz, idő, erőfeszítés) nem válik aránytalanul nagyra a kockázatcsökkentés hasznához képest. A bizonyítás a rendszer tervezőjét és üzemeltetőjét terheli: neki kell igazolnia, hogy a további csökkentés nem észszerű.
5. Mert kevés kategóriája van, a besorolás szubjektív, ezért mennyiségileg nagyon különböző kockázatok kerülhetnek egy cellába, és a sorrend is hibás lehet. Mennyiségi elemzés kell, ha változatok között kell dönteni, ha a kockázat a zónahatár közelében van, vagy ha a súlyosság katasztrofális.
6. Energiaforrások: villamos (hálózati feszültség), hő (izzó fűtőszálak). Célpontok: a felhasználó (áramütés, égés), a konyha (tűz). Gátak: kettős szigetelés és földelés (villamos), hőszigetelt ház,

automatikus kikapcsolás időzítővel és hőbiztosítékkal, morzsatálca (tűz), használati utasítás (fém eszközt ne dugjunk bele).
7. Mert hatása az ember tudásán, emlékezetén és szabálykövetésén múlik, amely fáradtság, időnyomás és rutin mellett megbízhatatlan; a tervezési és a műszaki intézkedés ettől függetlenül hat.
8. Például: „Mivel a csapat két tagja a vizsgaidőszakban gyakornokoskodik, fennáll a lehetősége, hogy a validációs próba elmarad, ami miatt a zárójelentés a felhasználói visszajelzések nélkül készül el."

Projektlépés: kockázatok

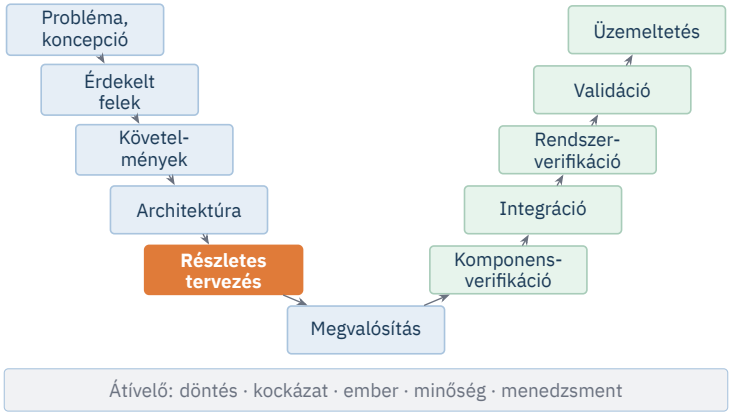
Saját fejlesztési feladatodhoz:

1. készíts kockázati mátrixot a projekt jellegéhez igazított súlyossági és valószínűségi skálával, és kalibráld egy ismert forgatókönyvvel;
2. végezz előzetes veszélyelemzést az energiaforrások és a felhasználói feladatok alapján, legalább hat veszéllyel, ellenintézkedésekkel és maradó kockázattal;
3. nyiss kockázati nyilvántartást legalább öt projektkockázattal, feltétel–következmény formában, felelőssel és beavatkozási küszöbbel;
4. a módszerekhez használd a Gyakorlati útmutató kockázati mátrix és PHA eszközlapját, sablonját és notebookját.

Továbbolvasás

- Dorofee és mtsai. (1996): a folyamatos kockázatkezelés eredeti, gyakorlati útmutatója.
- Ericson (2015): a veszélyelemzési módszerek (PHA, gátelemzés és mások) részletes kézikönyve.
- Cox (2008): a kockázati mátrixok korlátainak kritikai elemzése.
- ISO (2018b) és ISO (2010): a kockázatkezelés általános keretrendszere és a gépek kockázatcsökkentésének háromlépéses módszere.
- Health and Safety Executive (2001): az ALARP-elv és a tűrhető kockázat eredeti, közérthető leírása.
- Cox (2008): a kockázati mátrixok matematikai korlátainak elemzése.
- Haddon (1973): az energiaátadás és a tíz ellenstratégia klasszikus cikke.
- NASA (2016), 6.4. szakasz: a technikai kockázatkezelés folyamata, a RIDM és a CRM viszonya.
- Goldberg és mtsai. (1994), 3.1–3.3. szakasz: a kockázati mátrix szerkesztése, a PHA és az energia–gát elemzés.

15. Hibamód- és hatáselemzés (FMEA)



Tanulási eredmények

A fejezet feldolgozása után képes leszel:

- elmagyarázni az FMEA célját, típusait és helyét a fejlesztésben;
- végigvezetni egy FMEA-t az AIAG–VDA kézikönyv hét lépésén;
- felírni a hibaláncot (hibahatás, hibamód, hibaok) a rendszer szintjein;
- a súlyosságot, az előfordulást és az észlelhetőséget következetesen értékelni;
- ismertetni a kockázati prioritásszám (RPN) korlátait és a cselekvési prioritás elvét;
- kritikusságot számítani az FMECA módszerével, és az FMEA-t élő dokumentumként kezelni.

A kockázatkezelés (14. fejezet) a veszélyekből és a projekt bizonytalanságaiból indul ki, felülről lefelé. A hibamód- és hatáselemzés (failure mode and effects analysis, FMEA) ezt alulról felfelé egészíti ki: a rendszer elemeit egyenként veszi sorra, és mindegyiknél megkérdezi, hogyan hibásodhat meg, mi lesz ennek a következménye, és mi okozhatja. Az FMEA a megelőző minőségügy módszere: a meglévő hibák elhárítása helyett a lehetséges hibák elkerülésére törekszik, ezért a vizsgálat szempontjából mindegy, hogy a hiba előfordult-e már, vagy csak lehetőség (IEC 2018).

A fejezet négy kérdésre épül:

- Mire való az FMEA, és milyen fajtái vannak?
- Milyen lépésekből áll, és hogyan írjuk fel a hibaláncokat?
- Hogyan értékeljük és rangsoroljuk a hibamódokat?
- Hogyan lesz az FMEA-ból intézkedés és élő dokumentum?

15.1. Az FMEA célja és típusai

Az FMEA szabványos leírását a nemzetközi szabvány (IEC 2018), a kritikusságelemzéssel kiegészített változatot (FMECA) az amerikai védelmi szabvány (U.S. Department of Defense

1980), az autóipari gyakorlatot az AIAG és a VDA közös kézikönyve (AIAG és VDA 2019) adja meg. Az FMEA alapos módszer az egypontos hibák és következményeik azonosítására, a kritikusságelemzés pedig ezek kockázatát is értékeli. Korlátja, hogy munkaigényes, és egyszerre csak egy hibamódot vizsgál, a hibák kombinációit nem; erre a hibafa-elemzés való (16. fejezet) (IEC 2018; Goldberg és mtsai. 1994, 3.4. szakasz).

A gyakorlatban két fő típust különítenek el (AIAG és VDA 2019):

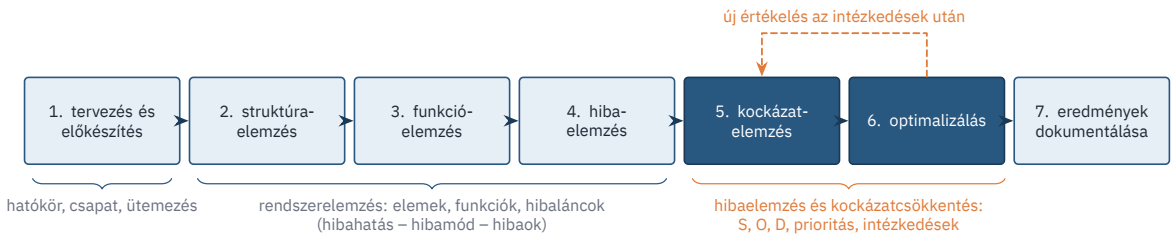
- A **tervezési (termék-) FMEA** a termék konstrukciós hibáit, gyenge pontjait tárja fel. A tervezéssel párhuzamosan fut, a tervdokumentáció jóváhagyásáig befejeződik, és a konstrukciós szervezet felelőssége. Szintje szerint lehet rendszer-, alrendszer- és alkatrész-FMEA.
- A **folyamat-FMEA** a gyártási, szerelési, ellenőrzési és logisztikai folyamat hibáit elemzi. A tervezési FMEA-ra épül, a sorozatgyártás engedélyezéséig befejeződik, és a gyártás-előkészítés felelőssége.

Az autóipari kézikönyv szerint az FMEA három helyzetben szükséges: új termék, technológia vagy folyamat esetén (teljes körűen); meglévő termék vagy folyamat módosításakor (a módosításra és hatásaira összpontosítva); és ha egy meglévő termék vagy folyamat új környezetbe, helyszínre vagy alkalmazásba kerül (AIAG és VDA 2019). A módszer legeredményesebben csoportmunkában végezhető: egy felelős vezeti, de a tervezés, a gyártás, a minőségügy, a karbantartás és lehetőség szerint a vevő és a beszállító is részt vesz.

A rendszermérnöki fejlesztésben az FMEA helye a V bal ágán van: a koncepció és az architektúra kialakítása után a rendszer- és alrendszerszintű FMEA a tervezési döntéseket, a részletes tervezésben az alkatrész- és folyamat-FMEA a konstrukciót és a gyártást támogatja. A funkcióelemzésben a funkciókat érdemes a hibamódjaikkal és a hibák következményeivel együtt leírni (8.3. szakasz), így a funkcióelemzés közvetlenül táplálja az FMEA-t.

15.2. Az FMEA lépései

Az AIAG–VDA kézikönyv az FMEA-t hét lépésre bontja (15.1. ábra) (AIAG és VDA 2019):



15.1. ábra. Az FMEA hét lépése az AIAG–VDA kézikönyv szerkezetében.

1. **Tervezés és előkészítés:** a hatókör, a csapat, az ütemezés és az alkalmazott eszközök rögzítése; mit elemzünk, és mit nem.
2. **Struktúraelemzés:** a vizsgált rendszer felbontása elemekre (rendszer, alrendszer, komponens; vagy folyamatnál folyamatlépés, munkaelem). Itt használható a fizikai architektúra és a HTA (11. fejezet).
3. **Funkcióelemzés:** az elemek funkcióinak és követelményeinek hozzárendelése a struktúrához, mérhető formában, a funkciófa (8. fejezet) alapján.

4. **Hibaelemzés:** minden funkcióhoz a hibamódok, ezek hatásai és okai, vagyis a **hibaláncok** felírása.
5. **Kockázatelemzés:** a meglévő megelőző és észlelő intézkedések rögzítése, és a súlyosság, az előfordulás és az észlelhetőség értékelése.
6. **Optimalizálás:** intézkedések a kockázat csökkentésére, felelőssel és határidővel, majd az új értékelés.
7. **Az eredmények dokumentálása** és közlése.

A hibaelemzés kulcsfogalmai a **hibamód**, a **hibahatás** és a **hibaok**. A hibamód az, ahogyan a vizsgált elem nem teljesíti a funkcióját (például „a nyomaték alacsony”). A hibahatás az, ahogyan a következmény a fölötte lévő szinten, végső soron a vevőnél megjelenik (a vevő lehet belső vevő is). A hibaok az, amiért a hibamód létrejön. A hibamódot fizikai, műszaki fogalmakkal kell leírni (repedés, meglazulás, szivárgás, nincs nyomatékátvitel, időszakos jelzés), nem pedig a vevő által tapasztalt tünettel. A hibaláncok a rendszer szintjei között összekapcsolódnak: egy szint hibamódja a fölötte lévő szint hibaoka és az alatta lévő szint hibahatása (15.2. ábra).



15.2. ábra. A hibalánc a KoboCella szintjein: a csavarozó egység FMEA-jában a hibamód „alacsony a nyomaték”, a hibahatás a vevőhöz kerülő tömítetlen motor, a hibaok a kopott bitbetét.

15.3. Súlyosság, előfordulás, észlelhetőség

A kockázatelemzés három tényezőt értékel, mindegyiket 1-től 10-ig terjedő skálán:

- **Súlyosság** (S): a hibahatás jelentősége a vevő számára. A 10 a biztonságot vagy az egészséget, a 9 a jogszabályi megfelelést érintő hatásoké (AIAG és VDA 2019), a 7–8 a funkció elvesztéséé vagy jelentős romlásáé, a 4–6 a kényelmi funkciók zavaráé, a 2–3 az alig észrevehető hibáké, az 1 a hatás nélküli hibáé. A súlyosság csak a konstrukció megváltoztatásával csökkenthető.
- **Előfordulás** (O): annak valószínűsége, hogy a hibaok a termék élete során (vagy a folyamatban) bekövetkezik, a meglévő megelőző intézkedések mellett. A skála gyakorisági adatokhoz is kalibrálható, például az ezer egységre jutó hibák számához: a 10 pont a nagyon gyakori hibaokot (ezer egységből legalább száz hibás), az 1 pont a megelőző intézkedéssel gyakorlatilag kizárt hibaokot jelenti. Az AIAG–VDA kézikönyv az előfordulást már elsősorban a megelőző intézkedések hatékonysága alapján pontoztatja (AIAG és VDA 2019). Az előfordulás megelőző intézkedéssel (konstrukciós vagy folyamatbeli változtatással) csökkenthető.
- **Észlelhetőség** (D): annak esélye, hogy a meglévő észlelő intézkedések felfedezik a hibát (vagy okát), mielőtt a vevőhöz ér. Itt a nagy szám a rossz: 1 pont, ha a hiba nyilvánvaló vagy a 100%-os automatikus ellenőrzés biztosan kiszűri; 9–10 pont, ha a hiba rejtett, vagy nincs ellenőrzés.

Két gyakorlati szabályt érdemes betartani. Először: a skálák relatívak, egy FMEA-n belül érvényesek, ezért a csapatnak előre meg kell egyeznie a kiértékelési szempontokban, és azokat következetesen kell alkalmaznia. Másodszor: a 9-es és 10-es súlyossági fokozat a biztonsághoz és a jogszabályokhoz kötődik, ezért ezek kritériumait a csapat nem lazíthatja fel.

KoboCella: folyamat-FMEA a kezelői feladatokból

A KoboCella csapata a folyamat-FMEA struktúráját a kezelő HTA-jából (11. fejezet) és a cella gépi lépéseiből építette fel. A 15.1. táblázat hat hibamódot mutat a kezdeti értékeléssel és a tervezett intézkedésekkel.

15.1. táblázat. A KoboCella folyamat-FMEA-jának részlete. O', D', RPN’: a tervezett intézkedés utáni értékelés; a súlyosság egyik esetben sem változik.

Folyamatlépés	Hibamód	Hibahatás	S	O	D	RPN	Intézkedés	O'	D'	RPN’
2.3 fedél ráhelyezése	fordított fedél	tömítetlen motor	8	5	7	280	aszimmetrikus illesztőcsapok	1	1	8
3.2 hibás darab kiválogatása	a darab a kiadóba kerül	hibás motor a vevőnél	8	3	8	192	a kiadó reteszelése a selejtdoboz jelzéséig	3	2	48
1.2 típusprogram	rossz program marad	rossz nyomaték egy sorozaton	7	3	5	105	a HMI a rendelés alapján tölti be a programot	1	5	35
csavaradagolás	hiányzó csavar	laza fedél, tömítetlenség	8	5	2	80	töltöttségérzékelő az adagolóban	3	2	48
csavarmeghúzás	alacsony nyomaték	laza kötés a vevőnél	8	4	2	64	bitcsere a ciklusszám alapján	2	2	32
2.2 motorház behelyezése	nem ül fel teljesen	a cella nem indul, időveszteség	4	6	2	48	–	–	–	48

Az elemzés legfontosabb tanulsága a fordított fedél: a nagy előfordulás és a gyenge észlelhetőség együtt adta a legmagasabb RPN-t. A csapat itt nem az észlelést javította (érzékelő), hanem az előfordulást szüntette meg hibabiztos kialakítással, összhangban a HEART-összehasonlítással (13. fejezet). Mivel a fordított fedél fizikailag nem helyezhető be, az észlelhetőség is a legjobb értéket kapta: a hiba nem jöhet létre. A táblázat utolsó két sora a következő szakasz problémáját is mutatja: a hiányzó csavar intézkedés utáni RPN-je (48) azonos a motorház behelyezéséével (48), holott az előbbi a vevőnél tömítetlen motort, az utóbbi csak időveszteséget okoz.

15.4. A kockázati prioritásszám és korlátai

A hagyományos FMEA a három tényező szorzatával, a **kockázati prioritásszámmal** rangsorol:

$$\mathrm{RPN} = S \cdot O \cdot D, \qquad 1 \le \mathrm{RPN} \le 1000. \tag{15.1}$$

Sok szervezet tájékoztató RPN-határokat is használ (például 100 fölött intézkedés szükséges), az autóipari kézikönyvek azonban kifejezetten nem javasolják, hogy az intézkedés szükségességét egyetlen RPN-küszöb döntse el (AIAG és VDA 2019). Az RPN egyszerű és elterjedt, de módszertani hibái jól dokumentáltak. Bowles elemzése szerint (Bowles 2003):

- az RPN **sorrendi skálákat** kezel arányskálaként: a 8-as súlyosság nem „kétszer olyan súlyos”, mint a 4-es, így a szorzatnak nincs mennyiségi értelme;
- a skála **lyukas**: az 1 és 1000 közötti egész számok közül csak 120 állhat elő három 1 és 10 közötti szám szorzataként, és a lehetséges értékek többsége a skála alsó részén sűrűsödik;
- **azonos RPN mögött nagyon különböző kockázat** állhat: $10 \cdot 2 \cdot 4 = 80$ (biztonságkritikus, ritka, jól észlelhető) és $4 \cdot 5 \cdot 4 = 80$ (kényelmi, gyakoribb) ugyanazt a számot adja;
- az RPN **érzékenysége egyenetlen**: egy pont változás az egyik tényezőben a többi tényező értékétől függően kicsi vagy nagy változást okoz.

Liu és munkatársai 75 tanulmányt tekintettek át, amelyek az RPN hiányosságait különféle módszerekkel (fuzzy logika, többszempontú döntési módszerek, súlyozás) próbálják kiküszöbölni (Liu, Liu, és Liu 2013). Az AIAG–VDA kézikönyv a gyakorlat számára egyszerűbb utat választott: az RPN helyett **cselekvési prioritást** (action priority, AP) vezetett be, amely a három tényező minden kombinációjához magas, közepes vagy alacsony prioritást rendel egy táblázat alapján, és elsősorban a súlyosságra, azután az előfordulásra, végül az észlelhetőségre helyezi a hangsúlyt (AIAG és VDA 2019). Az elv a lényeg: a súlyos hatású hibamódok akkor is figyelmet kapnak, ha az előfordulásuk vagy az észlelhetőségük kedvező. A könyv sablonjai ezért az RPN mellett egy egyszerűsített szabályt is alkalmaznak: a 9 és 10 súlyosságú hibamódoknál minden esetben intézkedést kell mérlegelni, az RPN-től függetlenül.

15.5. FMECA és kritikusságelemzés

Az FMECA (failure modes, effects and criticality analysis) az FMEA-t mennyiségi kritikusságelemzéssel egészíti ki. Az amerikai védelmi szabvány szerint egy hibamód kritikussági száma (U.S. Department of Defense 1980):

$$C_m = \beta \cdot \alpha \cdot \lambda_p \cdot t, \tag{15.2}$$

ahol λ_p az alkatrész meghibásodási rátája, α a hibamód aránya (az alkatrész hibáinak hányad része ilyen módú), β annak feltételes valószínűsége, hogy a hibamód a vizsgált súlyosságú hatást okozza, és t az üzemidő. Az alkatrész kritikussági száma a hibamódjaié összegezve. A hibamódokat súlyossági kategória és kritikusság szerint **kritikussági mátrixba** rendezik, amely a kockázati mátrix (14. fejezet) mennyiségi változata.

Tegyük fel például, hogy a KoboCella csavarozó egységének meghibásodási rátája $\lambda_p = 20$ hiba millió üzemóránként, a hibák 30%-a az alacsony nyomaték módja ($\alpha = 0{,}3$), és ennek a fele vezet tömítetlen motorhoz a vevőnél ($\beta = 0{,}5$). Évi 3450 üzemórával $C_m = 0{,}5 \cdot 0{,}3 \cdot 20 \cdot 10^{-6} \cdot 3450 \approx 0{,}01$, vagyis évente átlagosan 0,01 ilyen esemény várható. Az FMECA tehát nem pontszámot, hanem a megbízhatósági adatokból levezetett gyakoriságot ad; ehhez meghibásodási ráták kellenek (16. fejezet).

15.6. Az FMEA a gyakorlatban

A tervezési FMEA **élő dokumentum** (AIAG és VDA 2019): elkészítését a tervezési koncepció idején kell elkezdeni, folyamatosan naprakészen kell tartani minden változáskor és minden új információ birtokában, és teljesen készen kell lennie, mielőtt a gyártási dokumentációt kiadják a szerszámok elkészítéséhez. Az FMEA értéke nem a kitöltött táblázat, hanem a végrehajtott intézkedések: minden magas prioritású hibamódhoz felelős, határidő és az intézkedés utáni új értékelés tartozik.

Néhány bevált gyakorlat:

- **A funkcióból induljunk, ne az alkatrészből.** A „hogyan nem teljesíti a funkcióját?” kérdés teljesebb hibamódlistát ad, mint az „mi romolhat el rajta?”.
- **Használjuk a korábbi tapasztalatokat.** A garanciális adatok, a reklamációk, a hasonló termékek FMEA-i és a karbantartási naplók a hibamódok legjobb forrásai; ezeket ötletroham egészíti ki.
- **Az emberi hibamódok is hibamódok.** A folyamat-FMEA-ban a HTA minden lépéséhez meg kell vizsgálni a kihagyást, a rossz sorrendet, a rossz végrehajtást (13. fejezet).
- **A súlyosságot csak a konstrukció csökkenti.** Ha egy biztonságkritikus hibamódnál a csapat csak az észlelést javítja, a súlyosság változatlan marad, és a hibamód a következő változásnál ismét előkerül.
- **Az FMEA nem helyettesíti a kombinációk elemzését.** Az egyidejű hibákat és a közös okú meghibásodásokat a hibafa (16. fejezet), a vezérlési kölcsönhatásokat a rendszerelméleti elemzés (17. fejezet) vizsgálja.

VárosiBringa: a dokkoló zárjának tervezési FMEA-ja

A VárosiBringa csapata a dokkoló tervezési FMEA-jában a „kerékpár rögzítése” funkcióból indult. A „nem reteszel” hibamódhoz három okot talált: a kerékpár nincs ütközésig betolva (felhasználói megcsúszás, 13. fejezet), a reteszcsap kopása és a zárvezérlő szoftverhibája. A hibahatás mindhárom esetben ugyanaz: a kerékpár őrizetlenül áll, a kölcsönzés nem zárul le. Az első ok előfordulása messze a legnagyobb volt, ezért a csapat két irányban lépett. Az előfordulást a dokkoló bevezető formájának módosításával csökkentette (a kerékpár „beugrik” a végállásba), az észlelést a rögzítésérzékelővel, a fény- és hangjelzéssel és az alkalmazás értesítésével javította. A reteszcsap kopására a karbantartási terv kapott ellenőrzési pontot, a szoftverhibára pedig a verifikációs terv egy tesztesetet.

FMEA és digitális modellek

A modellalapú rendszermérnökségben (3. fejezet) a hibamódok a rendszermodell elemeihez köthetők, így az FMEA struktúrája és funkcióelemzése a modellből automatikusan előállítható, és a modell változásakor jelezhető, mely sorokat kell felülvizsgálni. Az üzemi adatok (gépi naplók, karbantartási rendszerek) pedig az előfordulás becslését teszik tényszerűbbé. Az elemzés lényegi része, a hibaláncok végiggondolása és az intézkedések kiválasztása azonban

továbbra is szakértői csapatmunka.

Az FMEA mint papírmunka

Gyakori, hogy az FMEA-t az auditra készítik el, a tervezés után, egy ember a meglévő dokumentációból. Az ilyen FMEA a már meghozott döntéseket igazolja, nem alakítja. Jelei: a hibamódok az alkatrészlistát követik, nincsenek hibaláncok a szintek között, minden RPN 100 alatt van, és az intézkedések oszlopa üres vagy „oktatás”. Az FMEA akkor hasznos, ha a tervezéssel párhuzamosan, csapatban készül, és a magas prioritású sorok konstrukciós változáshoz vezetnek.

Összefoglalás

- Az FMEA alulról felfelé, elemenként vizsgálja a lehetséges hibamódokat, hatásaikat és okaikat. Fő típusai a tervezési (termék-) és a folyamat-FMEA; a kritikusságelemzéssel kiegészített változat az FMECA.
- Az AIAG–VDA kézikönyv hét lépése: tervezés, struktúraelemzés, funkcióelemzés, hibaelemzés, kockázatelemzés, optimalizálás, dokumentálás. A hibaláncok a szintek között összekapcsolódnak: egy szint hibamódja a fölötte lévő szint hibaoka.
- A kockázatot a súlyosság, az előfordulás és az észlelhetőség 1–10-es skáláján értékeljük. A súlyosság csak konstrukciós változtatással, az előfordulás megelőző, az észlelhetőség észlelő intézkedéssel javítható.
- Az RPN sorrendi skálákat szoroz, csak 120 különböző értéket vehet fel, és azonos érték mögött nagyon különböző kockázat állhat. Az AIAG–VDA cselekvési prioritása a súlyosságot helyezi előtérbe.
- Az FMECA a hibamódok kritikusságát meghibásodási rátákból számítja. Az FMEA élő dokumentum: értéke a végrehajtott intézkedésekben van.

Önellenőrzés

1. Mi a különbség a tervezési és a folyamat-FMEA között? Ki a felelőse az egyiknek és a másiknak?
2. Írd fel egy kerékpárfék hibaláncát a rendszer (kerékpár), az alrendszer (fék) és a komponens (fékbetét) szintjén!
3. Miért kell a hibamódot fizikai fogalmakkal, és nem a vevő által tapasztalt tünettel leírni?
4. Melyik tényező csökkenthető megelőző és melyik észlelő intézkedéssel? Melyik csak konstrukciós változtatással?
5. Két hibamód: $S = 9, O = 2, D = 3$ és $S = 3, O = 6, D = 3$. Számítsd ki az RPN-t! Mit mutat az eredmény?
6. Hány különböző RPN-érték létezik? Miért probléma ez?
7. Egy alkatrész meghibásodási rátája 50 hiba millió óránként, egy hibamód aránya 0,2, a súlyos hatás feltételes valószínűsége 0,4. Mennyi a hibamód kritikussági száma 2000 üzemórára?
8. Miért nevezzük az FMEA-t élő dokumentumnak?

Megoldás.

1. A tervezési FMEA a termék konstrukciós hibáit vizsgálja, a tervezéssel párhuzamosan fut, és a konstrukciós szervezet felelőssége; a folyamat-FMEA a gyártási, szerelési és ellenőrzési folyamat hibáit vizsgálja, a tervezési FMEA-ra épül, és a gyártás-előkészítés felelőssége.
2. Komponens: a fékbetét elkopott (hibamód), oka a karbantartás elmaradása. Alrendszer: a fékerő elégtelen (hibamód), oka a kopott betét. Rendszer: a kerékpár nem áll meg időben (hatás), a felhasználó balesetet szenvedhet.
3. Mert a tünet több hibamód következménye is lehet, ezért nem vezet el az okhoz és az intézkedéshez; a fizikai hibamód (például „a csavar meglazul”) egyértelműen azonosítja, mit kell megelőzni vagy észlelni.
4. Az előfordulás megelőző intézkedéssel (például konstrukciós vagy folyamatbeli változtatással),

az észlelhetőség észlelő intézkedéssel (ellenőrzés, teszt, érzékelő) javítható; a súlyosság csak a konstrukció megváltoztatásával.

5. $9 \cdot 2 \cdot 3 = 54$ és $3 \cdot 6 \cdot 3 = 54$. Az RPN azonos, holott az első egy biztonsági hatású, ritka hibamód, a második egy enyhe, gyakoribb; az RPN nem különbözteti meg őket, ezért a súlyosságot külön is figyelni kell.
6. 120. Probléma, mert a skála lyukas és egyenetlen: a számok közötti különbségek nem arányosak a kockázatok közötti különbségekkel, és a küszöbértékek (például 100) körül kis változás nagy ugrást okozhat.
7. $C_m = 0{,}4 \cdot 0{,}2 \cdot 50 \cdot 10^{-6} \cdot 2000 = 0{,}008$.
8. Mert a tervezés korai szakaszában kezdődik, minden változáskor és új információ (próba, reklamáció, üzemi adat) birtokában frissíteni kell, és értéke a végrehajtott intézkedésekben és az utána következő újraértékelésben van, nem az egyszeri kitöltésben.

Projektlépés: FMEA

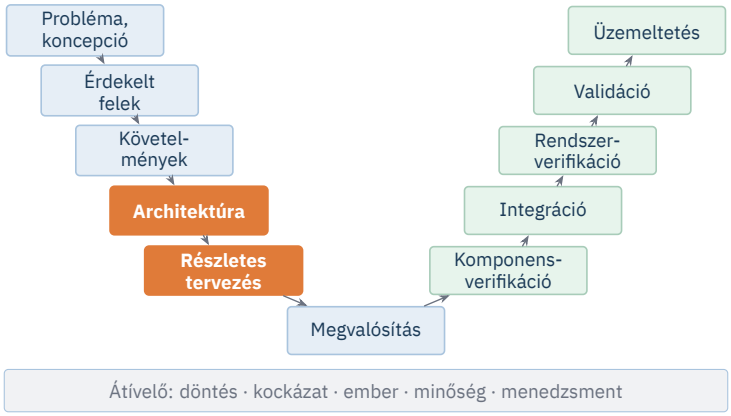
Saját fejlesztési feladatodhoz:

1. válaszd ki a legkockázatosabb alrendszert vagy folyamatot (a PHA és a kockázati nyilvántartás alapján);
2. készíts FMEA-t legalább nyolc hibamóddal, mindegyikhez hibahatással, hibaokkal, meglévő megelőző és észlelő intézkedéssel;
3. értékeld a súlyosságot, az előfordulást és az észlelhetőséget, számíts RPN-t, és jelöld külön a 9–10 súlyosságú sorokat;
4. a legmagasabb prioritású sorokhoz javasolj intézkedést, és add meg a várható új értékelést;
5. használd a Gyakorlati útmutató FMEA eszközlapját, sablonját és notebookját.

Továbbolvasás

- AIAG és VDA (2019): az autóipari FMEA-gyakorlat hét lépése és a cselekvési prioritás.
- IEC (2018): az FMEA és az FMECA nemzetközi szabványa.
- U.S. Department of Defense (1980): a kritikusságelemzés klasszikus leírása.
- Bowles (2003) és Liu, Liu, és Liu (2013): az RPN korlátai és a javasolt alternatívák áttekintése.
- Goldberg és mtsai. (1994), 3.4. szakasz: az FMEA és az FMECA eljárásának gyakorlati leírása.

16. Megbízhatóság és rendelkezésre állás



Tanulási eredmények

A fejezet feldolgozása után képes leszel:

- definiálni a megbízhatósági függvényt, a meghibásodási rátát és a kádgörbe szakaszait;
- számolni az exponenciális eloszlással, és értelmezni a Weibull-eloszlás alakparaméterét;
- megkülönböztetni az MTTF, MTBF és MTTR mutatókat, és rendelkezésre állást számolni;
- soros, párhuzamos és k-ból-n rendszerek megbízhatóságát meghatározni blokkdiagrammal;
- hibafát és eseményfát szerkeszteni, minimális vágathalmazokat keresni, és valószínűséget számolni;
- a modelleket egymásba alakítani, és felismerni a függetlenségi feltevés csapdáit.

A kockázatelemzés (14. fejezet) és az FMEA (15. fejezet) azonosítja, mi romolhat el. A megbízhatósági elemzés azt kérdezi, *milyen valószínűséggel* és *milyen gyakran*. Erre a követelményeknek is szükségük van: egy „a cella rendelkezésre állása legalább 98%” vagy „a biztonsági funkció veszélyes hibájának valószínűsége évente legfeljebb 10^{-4}” követelmény csak akkor tervezhető és verifikálható, ha a rendszert az elemei megbízhatóságából modellezni tudjuk.

A fejezet négy kérdésre épül:

- Hogyan írjuk le egy elem megbízhatóságát az időben?
- Mit jelent a rendelkezésre állás, és hogyan számoljuk?
- Hogyan következik a rendszer megbízhatósága az elemekéből?
- Hogyan modellezzük a hibák kombinációit és a következmények láncolatát?

16.1. A megbízhatóság alapfüggvényei

A **megbízhatóság** annak valószínűsége, hogy egy rendszer adott időtartamon át, adott feltételek mellett a követelményeknek megfelelően működik. Ha T a meghibásodásig eltelő (véletlen) idő, akkor a megbízhatósági függvény és a hibavalószínűség:

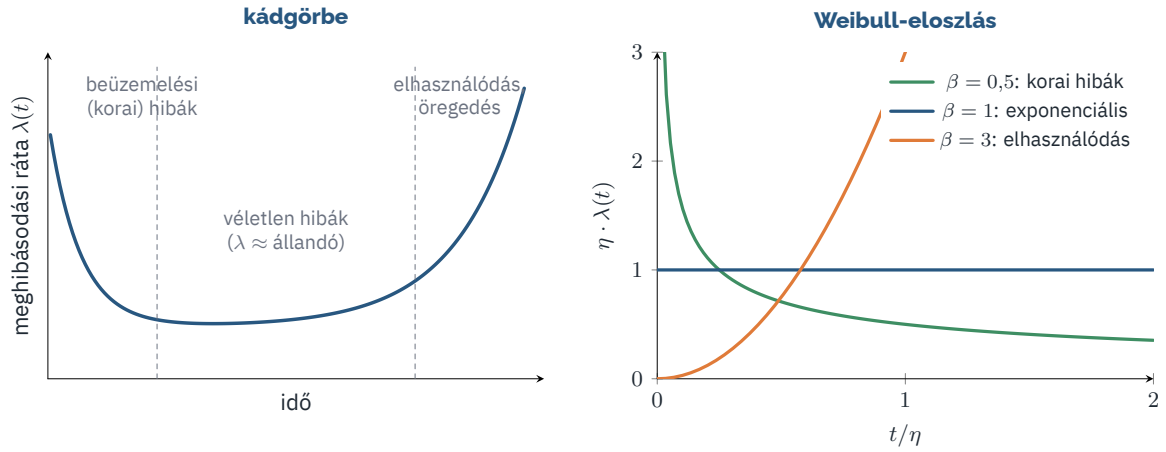
$$R(t) = P(T > t), \qquad F(t) = P(T \le t) = 1 - R(t). \tag{16.1}$$

A **meghibásodási ráta** $\lambda(t)$ annak a valószínűségnek az időegységre eső értéke, hogy a t időpontig hibátlanul működő rendszer a következő pillanatban meghibásodik. A feltételes valószínűség definíciójából vezethető le (O'Connor és Kleyner 2012):

$$\lambda(t) = \frac{f(t)}{R(t)}, \qquad R(t) = \exp\left(-\int_0^t \lambda(u)\,\mathrm{d}u\right), \tag{16.2}$$

ahol $f(t) = \mathrm{d}F/\mathrm{d}t$ a meghibásodási idő sűrűségfüggvénye (O'Connor és Kleyner 2012).

Sok műszaki rendszer meghibásodási rátája a **kádgörbét** követi (16.1. ábra): a beüzemelési szakaszban a rejtett gyártási hibák miatt a ráta magas és csökkenő, az üzemszerű szakaszban közel állandó (véletlen hibák), az öregedési szakaszban pedig a visszafordíthatatlan fizikai-kémiai folyamatok (kopás, kifáradás, korrózió) miatt növekvő.



16.1. ábra. A kádgörbe és a Weibull-eloszlás meghibásodási rátája három alakparaméterre.

Állandó meghibásodási ráta esetén a meghibásodásig eltelő idő **exponenciális eloszlású**. Ez az elektronikus rendszerekre a hasznos élettartam alatt jellemzően jó közelítés (O'Connor és Kleyner 2012):

$$R(t) = e^{-\lambda t}, \qquad \mathrm{MTTF} = \frac{1}{\lambda}, \qquad t_{50\%} = \frac{\ln 2}{\lambda}. \tag{16.3}$$

Ha például egy gép átlagos működési ideje a meghibásodásig $\mathrm{MTTF} = 1000$ óra, akkor annak valószínűsége, hogy 500 óránál tovább üzemel, $R(500) = e^{-0{,}5} = 0{,}607$; annak, hogy addig meghibásodik, $F(500) = 0{,}393$. Az exponenciális eloszlás „emlékezet nélküli": egy már 400 órája működő gép a következő 500 órát ugyanakkora valószínűséggel éli túl, mint egy új. Ez az öregedő alkatrészekre nyilvánvalóan nem igaz.

A **Weibull-eloszlás** mindhárom szakaszt leírja (Weibull 1951):

$$R(t) = \exp\left[-\left(\frac{t}{\eta}\right)^{\beta}\right], \qquad \lambda(t) = \frac{\beta}{\eta}\left(\frac{t}{\eta}\right)^{\beta-1}, \tag{16.4}$$

ahol η a skálaparaméter (karakterisztikus élettartam), β az alakparaméter. Ha $\beta < 1$, a ráta csökkenő (korai hibák), ha $\beta = 1$, állandó (exponenciális eset), ha $\beta > 1$, növekvő (elhasználódás). A gyakorlatban a Weibull-eloszlást meghibásodási adatokra illesztik, és β értéke maga is diagnosztikai információ: megmutatja, hogy a hibákat gyártási, véletlen vagy kopási folyamatok okozzák-e, és így azt is, hogy a megelőző csere segít-e (O'Connor és Kleyner 2012).

16.2. Javítható rendszerek: MTTF, MTBF, MTTR, rendelkezésre állás

A legtöbb ipari rendszert meghibásodás után megjavítják. A javítható rendszerekre a következő átlagos időket szokás megkülönböztetni (O'Connor és Kleyner 2012):

- **MTTFF** (mean time to first failure): az első üzembe helyezéstől az első meghibásodásig eltelő átlagos idő;
- **MTTF** (mean time to failure): egy javítás végétől a következő meghibásodásig eltelő átlagos idő;
- **MTTR** (mean time to repair): a meghibásodástól a javítás végéig, az újbóli üzembe állásig eltelő átlagos idő;
- **MTBF** (mean time between failures): két meghibásodás között eltelő átlagos idő, $\mathrm{MTBF} = \mathrm{MTTF} + \mathrm{MTTR}$.

A szakirodalom és a gyártói adatlapok a betűszavakat nem egységesen használják: sok helyen az MTBF a két hiba közötti átlagos *működési* időt jelenti. Egy adatlap értékét ezért mindig a definíciójával együtt kell átvenni.

Ha a meghibásodás és a javítás is exponenciális eloszlású (állandó λ meghibásodási és μ javítási rátával), a rendszer hosszú távon a ki nem esett idő meghatározott hányadában üzemképes. Ez a **belső (inherens) rendelkezésre állás**:

$$A = \frac{\mu}{\lambda + \mu} = \frac{\mathrm{MTTF}}{\mathrm{MTTF} + \mathrm{MTTR}}. \tag{16.5}$$

Ha egy gép átlagosan 100 óránként hibásodik meg, és a javítási ráta $\mu = 0{,}05$ 1/óra (MTTR $= 20$ óra), akkor $A = 100/120 = 0{,}833$. A belső rendelkezésre állás csak a javítás tényleges idejét veszi figyelembe; a gyakorlati (operatív) rendelkezésre állás a logisztikai és adminisztratív várakozást (alkatrészre, szerelőre várás) is tartalmazza, és ezért kisebb. A gyártásban a rendelkezésre állás az OEE egyik tényezője is (19. fejezet).

16.3. Soros, párhuzamos és vegyes rendszerek

A **megbízhatósági blokkdiagram** (reliability block diagram, RBD) a rendszer sikeres működésének logikai modellje: a rendszer akkor működik, ha a bemenettől a kimenetig vezet működő elemeken át út (IEC 2016a). Független elemek esetén:

$$R_{\text{soros}} = \prod_{i=1}^{n} R_i, \qquad R_{\text{párhuzamos}} = 1 - \prod_{i=1}^{n} (1 - R_i). \tag{16.6}$$

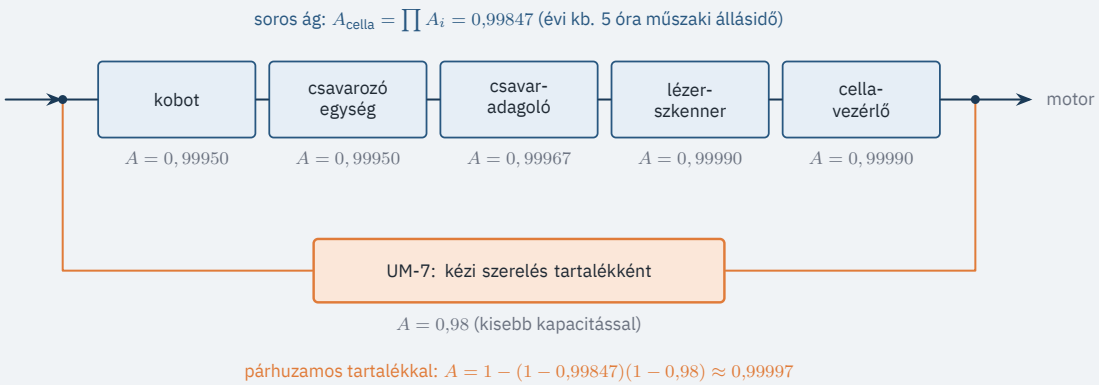
A soros rendszer csak akkor működik, ha minden eleme működik, ezért megbízhatósága kisebb a leggyengébb eleménél; a párhuzamos (redundáns) rendszer akkor működik, ha legalább egy eleme működik. Egy egyszerű számpélda ezt látványosan mutatja: ugyanabból az eszközből egyre többet sorba kötve a megbízhatósági függvény egyre gyorsabban csökken, párhuzamosan kötve egyre lassabban. A **k-ból-n rendszer** akkor működik, ha n eleméből legalább k működik; azonos, független elemekre a binomiális eloszlással számolható:

$$R_{k/n} = \sum_{j=k}^{n} \binom{n}{j} R^j (1 - R)^{n-j}. \tag{16.7}$$

A rendelkezésre állásra ugyanezek a képletek érvényesek, ha az elemek egymástól függetlenül hibásodnak meg és javíthatók. Az összetett szerkezeteket alapkapcsolású részrendszerekre bontjuk, és ezek eredő értékeiből számolunk tovább.

KoboCella: a cella rendelkezésre állása

A KoboCella termelési funkciójához öt elem soros működése kell. A beszállítói adatlapok és a csapat becslése szerint az MTBF és az MTTR: kobot 8000 és 4 óra, csavarozó egység 2000 és 1 óra, csavaradagoló 1500 és 0,5 óra, lézerszkenner 20 000 és 2 óra, cellavezérlő 30 000 és 3 óra. A soros ág rendelkezésre állása 0,99847, ami évi 3450 üzemórával mintegy 5 óra műszaki állásidőt jelent (16.2. ábra). A cella eredő meghibásodási rátája az elemek rátáinak összege, $\lambda = 1{,}375 \cdot 10^{-3}$ 1/óra, vagyis a cella átlagosan 727 óránként, évente mintegy 4,7-szer áll meg műszaki hiba miatt; egy 7,5 órás műszakot 0,99 valószínűséggel hiba nélkül teljesít.



16.2. ábra. A KoboCella termelési funkciójának megbízhatósági blokkdiagramja a leromlott üzemmód (UM-7) kézi tartalékával.

A leromlott üzemmód (UM-7) párhuzamos ágként működik: ha a cella áll, a kézi szerelés folytatható. Ezzel a termelési funkció rendelkezésre állása 0,99997-re nő, de a kézi tartalék

kisebb kapacitással dolgozik, ezért a ciklusidő-követelményt (R-01) leromlott üzemben nem teljesíti. A blokkdiagram tehát a funkció elérhetőségét mutatja, a teljesítményét nem. A legnagyobb hozzájárulás az állásidőhöz a kobottól és a csavarozó egységtől jön; a csapat a csavarozó egységhez tartalék orsót rendelt, amellyel az MTTR rövidebb lesz.

A párhuzamos redundancia számított előnye a valóságban gyakran kisebb. A blokkdiagrammal számított rendszer-megbízhatóság ezért irreálisan magas lehet (O'Connor és Kleyner 2012; Goldberg és mtsai. 1994, 3.5. szakasz). Ennek leggyakoribb oka a **közös okú meghibásodás**: ha a két redundáns elemet ugyanaz a tápellátás, ugyanaz a szoftver vagy ugyanaz a karbantartási hiba érinti, nem függetlenek, és a $1-(1-R)^2$ képlet túlbecsül. A redundáns ágakat ezért eltérő elvű, fizikailag elkülönített megoldásokkal érdemes kialakítani (diverzitás), ami a biztonsági rendszerek tervezésének egyik alapelve (17. fejezet).

VárosiBringa: hány dokkoló kell egy állomáson?

Egy VárosiBringa-állomás tíz dokkolóhelyből áll, és a csapat úgy ítélte, hogy az állomás akkor teljesíti a funkcióját, ha legalább nyolc dokkoló működik. Egy dokkoló rendelkezésre állása 0,98. A 16.7. egyenlet szerint az állomás 0,999 valószínűséggel működik; ha mind a tíz dokkolóra szükség lenne, csak 0,82 valószínűséggel. A k-ból-n szemlélet így a követelmény megfogalmazására is hat: a „minden dokkoló működik” követelmény a gyakorlatban teljesíthetetlen, a „legalább nyolc dokkoló működik” viszont jól mérhető, és illeszkedik a felhasználók igényéhez.

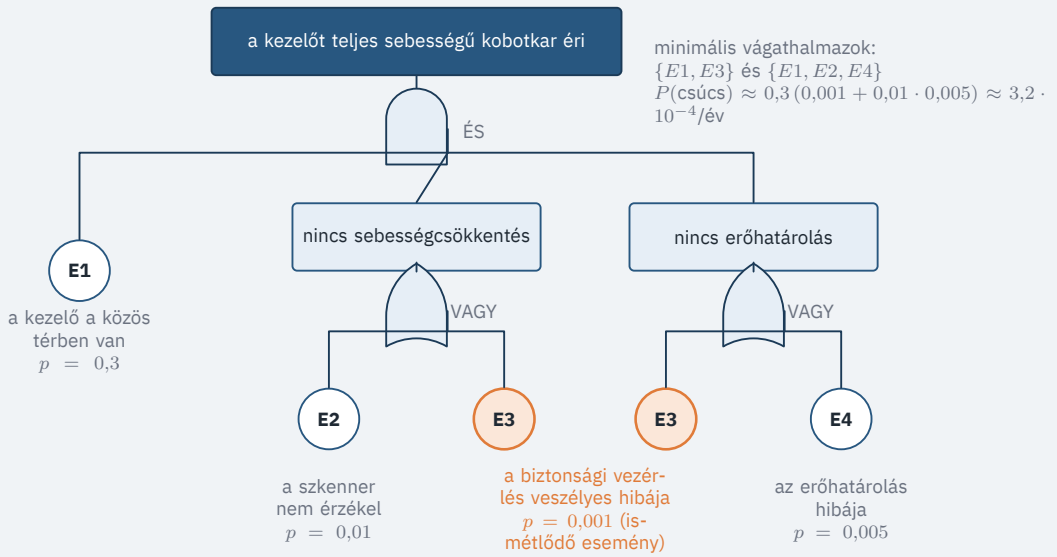
16.4. Hibafa-elemzés

A **hibafa-elemzés** (fault tree analysis, FTA) egy kiválasztott nem kívánt eseményből, a **csúcseseményből** indul, és felülről lefelé, logikai kapukkal bontja le azokra az eseménykombinációkra, amelyek előidézhetik (Vesely és mtsai. 1981; IEC 2006). Az ÉS-kapu kimenete akkor következik be, ha minden bemenete bekövetkezik, a VAGY-kapué akkor, ha legalább egy. A legalsó szint **alapeseményei** az alkatrészhibák, emberi hibák és külső események, amelyekhez valószínűséget rendelünk. A hibafa legfőbb erőssége, hogy az egyidejűleg fennálló hibák valószínűségét is értékelni tudja; korlátja, hogy egyetlen, az elemző által előre megnevezett csúcseseményt vizsgál, és a teljes fák nagyon nagyok lehetnek (Vesely és mtsai. 1981; Ericson 2015; Goldberg és mtsai. 1994, 3.6. szakasz).

A hibafa minőségi eredménye a **minimális vágathalmazok** listája. Egy vágathalmaz olyan alapesemény-kombináció, amely együtt előidézi a csúcseseményt; minimális, ha egyetlen eleme sem hagyható el. Az egyelemű minimális vágathalmazok az **egypontos hibák**, amelyekre a tervezésnek különös figyelmet kell fordítania. Független alapesemények esetén a mennyiségi kiértékelés: az ÉS-kapu valószínűsége a bemenetek szorzata, a VAGY-kapué $1-\prod(1-p_i)$, kis valószínűségeknél jó közelítéssel az összeg (ritka esemény közelítés).

KoboCella: hibafa egy biztonsági csúcseseményre

A PHA (14. fejezet) szerint a KoboCella legfontosabb biztonsági veszélye a kobotkar ütközése a kezelővel. A csapat hibafát szerkesztett arra a csúcseseményre, hogy a kezelőt teljes sebességű kobotkar éri (16.3. ábra). Ehhez három feltételnek kell együtt teljesülnie: a kezelő a közös térben van (E1), elmarad a sebességcsökkentés (a szkenner nem érzékel, E2, vagy a biztonsági vezérlés hibás, E3), és nem működik az erőhatárolás (E4 vagy ugyanaz az E3).



16.3. ábra. A KoboCella hibafája. Az E3 esemény két ágban is szerepel: ez határozza meg a fa viselkedését.

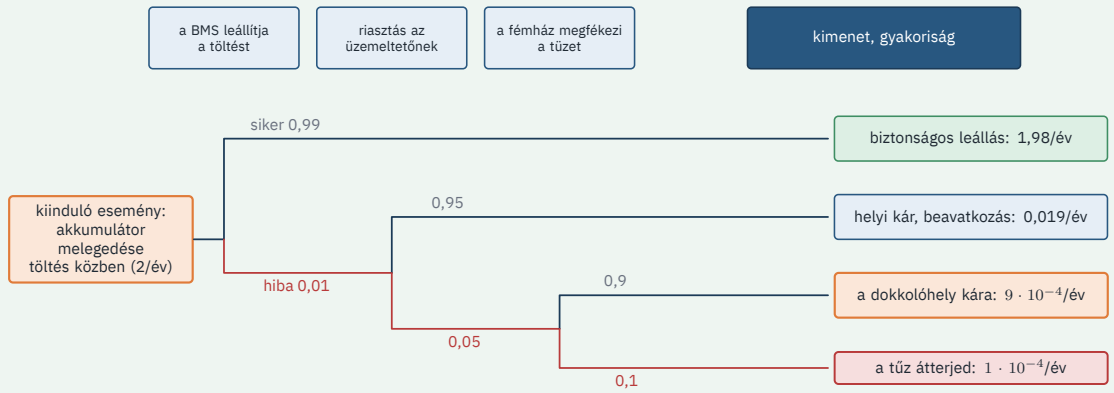
A minimális vágathalmazok: $\{E1, E3\}$ és $\{E1, E2, E4\}$. Az évi valószínűség (a szkenner és az erőhatárolás veszélyes, fel nem fedezett hibáira vonatkozó becslésekkel) $P \approx 0{,}3 \cdot (0{,}001 + 0{,}01 \cdot 0{,}005) \approx 3{,}2 \cdot 10^{-4}$, és ennek 95%-a az első vágathalmazból jön. Ha valaki a fát a kapuk szintjén, a független események képleteivel számolná, $2{,}0 \cdot 10^{-5}$ adódna, tizenhatszor kevesebb: az E3 ismétlődő esemény, ezért a két VAGY-kapu kimenete nem független. A tanulság a tervezésnek is szól: a biztonsági vezérlés közös pont a két védelmi rétegben, ezért a csapat a sebességfelügyeletet és az erőhatárolást független csatornára tervezte (17. fejezet).

16.5. Eseményfa-elemzés

Az **eseményfa-elemzés** (event tree analysis, ETA) a hibafával ellentétes irányban halad: egy **kiinduló eseményből** indul, és előre követi, hogyan alakul a folyamat attól függően, hogy az egymás után következő védelmi rétegek (gátak, biztonsági funkciók, emberi beavatkozások) működnek-e (IEC 2010b). Minden réteg egy elágazás siker és hiba ágra, a végpontok a lehetséges kimenetek, gyakoriságuk a kiinduló esemény gyakoriságának és az ágak valószínűségének szorzata. Az eseményfa egyszerre dolgozik a siker és a hiba tartományában, és a végkimeneteket nem kell előre ismerni; korlátja, hogy egyetlen kiinduló eseményt vizsgál, és a részleges sikert nehezen tudja megkülönböztetni (Ericson 2015).

VárosiBringa: az akkumulátor túlmelegedésének eseményfája

A kockázatelemzésben (14. fejezet) azonosított akkumulátorveszélyre a VárosiBringa csapata eseményfát készített (16.4. ábra). A kiinduló esemény egy akkumulátor rendellenes melegedése töltés közben; a teljes flottára évi két ilyen eseménnyel számoltak. Három védelmi réteg követi egymást: az akkumulátorkezelő rendszer (BMS) leállítja a töltést, a hőmérséklet-riasztás az üzemeltetőhöz fut be, és a fémház megfékezi a tüzet.

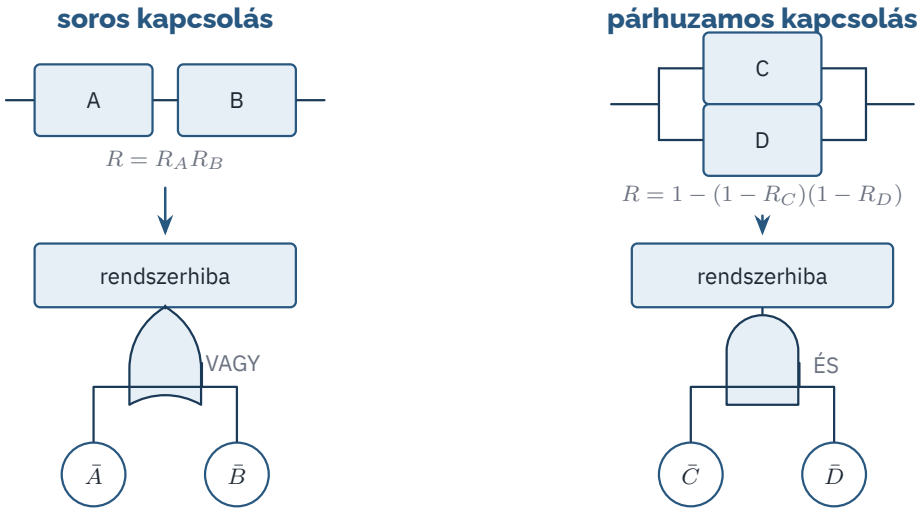


16.4. ábra. A VárosiBringa akkumulátor-túlmelegedésének eseményfája.

A legsúlyosabb kimenet, a tűz átterjedése a szomszédos kerékpárokra, évente $2 \cdot 0{,}01 \cdot 0{,}05 \cdot 0{,}1 = 10^{-4}$ gyakoriságú. Az eseményfa azt is megmutatta, hol a leghatékonyabb a beavatkozás: a BMS 0,99-es sikere a legerősebb réteg, de a riasztás és a fémház együtt további kétszázszoros csökkentést ad. Az egyes rétegek sikertelenségének valószínűségét a csapat külön hibafákkal becsülte meg, így a két módszer egymást egészítette ki.

16.6. A modellek kapcsolata

A blokkdiagram, a hibafa és az eseményfa ugyanannak a rendszernek különböző nézetei, és egymásba alakíthatók (Vesely és mtsai. 1981; IEC 2016a; Goldberg és mtsai. 1994, 3.9. szakasz). A blokkdiagram a sikert, a hibafa a hibát modellezi: a soros blokkokból VAGY-kapu, a párhuzamos blokkokból ÉS-kapu lesz a hibák között (16.5. ábra). Az eseményfa egyes ágainak hibavalószínűsége hibafával számolható, a hibafa egy-egy alapeseménye pedig blokkdiagramból. Az átalakítás új információt nem ad, de segít abban, hogy az egyik módszer gyengeségét a másik erősségével ellensúlyozzuk.



a blokkdiagram a sikert, a hibafa a hibát modellezi: a soros ágból VAGY-kapu, a párhuzamos ágból ÉS-kapu lesz

16.5. ábra. A soros és a párhuzamos blokkdiagram megfelelője a hibafában.

A módszerek kiválasztásában a kérdés a döntő (16.1. táblázat).

16.1. táblázat. A megbízhatósági és biztonsági modellek alkalmazási területe.

Kérdés	Módszer	Irány
Mekkora a rendszer megbízhatósága, rendelkezésre állása?	blokkdiagram	a sikerből, alulról felfelé
Milyen kombinációk vezetnek egy nem kívánt eseményhez, és mekkora valószínűséggel?	hibafa	a csúcseseményből, felülről lefelé
Mi következhet egy kiinduló eseményből a védelmi rétegektől függően?	eseményfa	a kiinduló eseményből, előre
Milyen hibamódok vannak, és mi a hatásuk egyenként?	FMEA (15. fejezet)	az elemekből, alulról felfelé

Adatok és állapotfüggő megbízhatóság

A klasszikus megbízhatósági számítás alkatrész-adatbázisok és gyártói adatlapok rátáiból dolgozik (például a katonai elektronikai alkatrészekre kidolgozott meghibásodásiráta-modellekből). Az üzemelő rendszerekben ma egyre több a tényleges üzemi adat: a gépi naplók, a karbantartási rendszerek és az érzékelők alapján a Weibull-paraméterek folyamatosan frissíthetők, és a megbízhatóság az alkatrész tényleges állapotához köthető. Ez vezet az állapotalapú és előrejelző karbantartáshoz (25. fejezet), amelyben a cserét nem a naptár, hanem a romlás mértéke vezérli.

Függetlenség ott, ahol nincs

A soros és párhuzamos képletek, a hibafa kapuképletei és az eseményfa szorzatai mind függetlenséget feltételeznek. Ha egy esemény több ágban is szerepel (ismétlődő esemény), vagy a redundáns elemeknek közös okuk van (tápellátás, szoftver, karbantartó), a kapuszintű számítás nagyságrendekkel is tévedhet, és mindig a túl optimista irányba. A minimális vágathalmazokon alapuló számítás és a közös okú hibák külön vizsgálata a megoldás.

Összefoglalás

- A megbízhatóság annak valószínűsége, hogy a rendszer adott ideig megfelelően működik. A meghibásodási ráta időbeli alakja a kádgörbe: korai hibák, véletlen hibák, elhasználódás.
- Állandó rátánál az élettartam exponenciális eloszlású ($R = e^{-\lambda t}$, $\mathrm{MTTF} = 1/\lambda$). A Weibull-eloszlás alakparamétere megmutatja, melyik szakaszban vagyunk.
- Javítható rendszereknél az MTTF, az MTTR és az MTBF jellemzi a működést; a belső rendelkezésre állás $A = \mathrm{MTTF}/(\mathrm{MTTF} + \mathrm{MTTR})$.
- A blokkdiagramon a soros elemek megbízhatósága szorzódik, a párhuzamos elemek hibavalószínűsége szorzódik; a k-ból-n rendszer binomiálisan számolható. A redundancia hasznát a közös okú hibák csökkentik.
- A hibafa a csúcseseményből lefelé keresi a hibakombinációkat; eredménye a minimális vágathalmazok listája és a csúcsesemény valószínűsége. Az eseményfa a kiinduló eseményből előre követi a védelmi rétegek sikerét és kudarcát. A modellek egymásba alakíthatók.

Önellenőrzés

1. Egy alkatrész meghibásodási rátája állandó, $\lambda = 2 \cdot 10^{-4}$ 1/óra. Mennyi az MTTF, és mennyi a valószínűsége, hogy 1000 órát hiba nélkül működik?
2. Három gép meghibásodásainak időpontja (óra): 1. gép 500, 1300, 2600, 4300, 6000; 2. gép 800, 1500, 3300, 5000, 7100; 3. gép 400, 1200, 2500, 4000, 5500. A javítási időt

elhanyagolva mennyi az egyes gépek MTBF-je?
3. Egy gép meghibásodásai között átlagosan 100 óra telik el, a javítási ráta 0,05 1/óra. Mennyi a rendelkezésre állása?
4. Két 0,9 megbízhatóságú elem: mennyi a rendszer megbízhatósága sorosan és párhuzamosan?
5. Mit jelent a Weibull-eloszlás $\beta = 3$ alakparamétere, és milyen karbantartási következménye van?
6. Miért adhat túl optimista eredményt a párhuzamos redundancia képlete?
7. Mi a minimális vágathalmaz? Mit jelent, ha egy minimális vágathalmaz egyelemű?
8. Miért nem szabad egy ismétlődő eseményt tartalmazó hibafát a kapuk szintjén, a független események képleteivel kiértékelni?

Megoldás.

1. MTTF $= 1/\lambda = 5000$ óra; $R(1000) = e^{-0{,}2} = 0{,}819$.
2. Az utolsó meghibásodás időpontja osztva a hibák számával: 1. gép $6000/5 = 1200$ óra, 2. gép $7100/5 = 1420$ óra, 3. gép $5500/5 = 1100$ óra.
3. MTTR $= 1/0{,}05 = 20$ óra, $A = 100/(100 + 20) = 0{,}833$.
4. Sorosan $0{,}9 \cdot 0{,}9 = 0{,}81$, párhuzamosan $1 - 0{,}1 \cdot 0{,}1 = 0{,}99$.
5. A meghibásodási ráta az idővel növekszik (elhasználódás, kopás, kifáradás). Ilyenkor a megelőző csere egy alkalmas üzemidő után csökkenti a hibák számát; állandó rátánál ($\beta = 1$) a megelőző csere nem segítene.
6. Mert a képlet a redundáns elemek független meghibásodását feltételezi, a valóságban viszont közös okok (közös táp, szoftver, környezet, karbantartási hiba) egyszerre több ágat is kiejthetnek.
7. Olyan alapesemény-kombináció, amely együtt előidézi a csúcseseményt, és egyik eleme sem hagyható el. Az egyelemű minimális vágathalmaz egypontos hiba: egyetlen esemény önmagában előidézi a csúcseseményt.
8. Mert az ismétlődő eseményt tartalmazó ágak nem függetlenek; a kapuszintű képletek a közös eseményt többször, egymástól függetlenként veszik figyelembe, és ezért (ÉS-kapuknál) nagyságrendekkel alábecsülhetik a valószínűséget. A helyes számítás a minimális vágathalmazokból indul.

Projektlépés: megbízhatóság

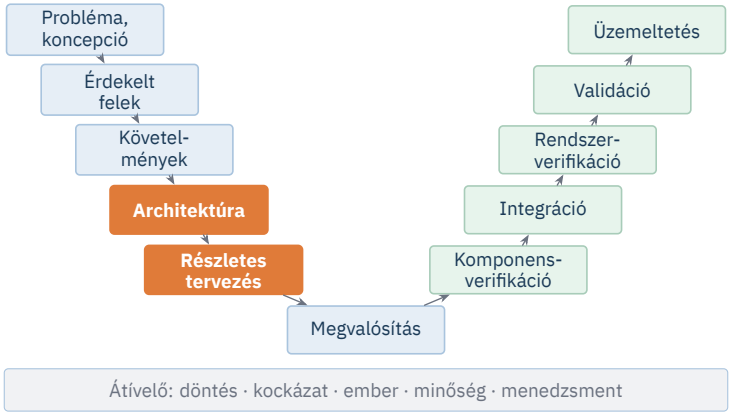
Saját fejlesztési feladatodhoz:

1. készíts megbízhatósági blokkdiagramot a fő funkcióhoz, és becsüld meg a rendelkezésre állást adatlapok vagy becslések alapján (jelöld az adatok forrását);
2. azonosítsd az egypontos hibákat, és vizsgáld meg, hol érdemes redundanciát vagy leromlott üzemmódot tervezni;
3. a PHA legsúlyosabb veszélyére készíts hibafát, határozd meg a minimális vágathalmazokat, és értékeld a csúcsesemény valószínűségét;
4. egy kiinduló eseményre készíts eseményfát a védelmi rétegekkel;
5. használd a Gyakorlati útmutató blokkdiagram-, hibafa- és eseményfa-eszközlapját.

Továbbolvasás

- O'Connor és Kleyner (2012): a megbízhatósági mérnökség gyakorlatias tankönyve.
- Vesely és mtsai. (1981): a hibafa-elemzés klasszikus kézikönyve, szabadon elérhető.
- IEC (2016a), IEC (2006) és IEC (2010b): a blokkdiagram, a hibafa és az eseményfa nemzetközi szabványai.
- Weibull (1951): a Weibull-eloszlás eredeti cikke.
- Goldberg és mtsai. (1994), 3.5–3.9. szakasz: a blokkdiagram, a hibafa, az eseményfa és átalakításaik gyakorlati leírása.

17. Biztonságkritikus rendszerek



Tanulási eredmények

A fejezet feldolgozása után képes leszel:

- elmagyarázni a funkcionális biztonság fogalmát és a hibaok, a hiba és a hibajelenség láncát;
- megkülönböztetni a véletlen és a szisztematikus hibaokokat, és a megfelelő védekezési módot hozzájuk rendelni;
- értelmezni a biztonsági integritási szinteket (SIL, PL, ASIL), és kapcsolatukat a kockázatcsökkentéssel;
- független védelmi rétegeket tervezni és értékelni;
- rendszerelméleti biztonságelemzést (STPA) végezni egy egyszerű irányítási struktúrán;
- az ember–robot együttműködés biztonsági szempontjait a kockázatértékelésbe illeszteni.

A megbízhatósági elemzés (16. fejezet) azt kérdezi, működik-e a rendszer. A biztonsági elemzés azt, hogy ha nem működik, vagy éppen ha működik, de rossz helyzetben, okoz-e kárt. A két kérdés nem azonos. Egy megbízható rendszer lehet veszélyes, ha jól teljesíti a rossz parancsot; és egy kevésbé megbízható rendszer is lehet biztonságos, ha minden hiba után biztonságos állapotba kerül. A biztonság emergens tulajdonság (1. fejezet): az elemek kölcsönhatásából születik, ezért nem elég az elemeket egyenként vizsgálni.

Knight meghatározása szerint biztonságkritikus az a rendszer, amelynek hibája emberi életet vagy egészséget veszélyeztethet, illetve jelentős anyagi vagy környezeti kárt okozhat (Knight 2002). Az ilyen rendszer bizonyos jellemzőire ezért szigorú követelményeket rögzítünk, mert ezek nem teljesülése veszélyhelyzethez vezet.

A fejezet négy kérdésre épül:

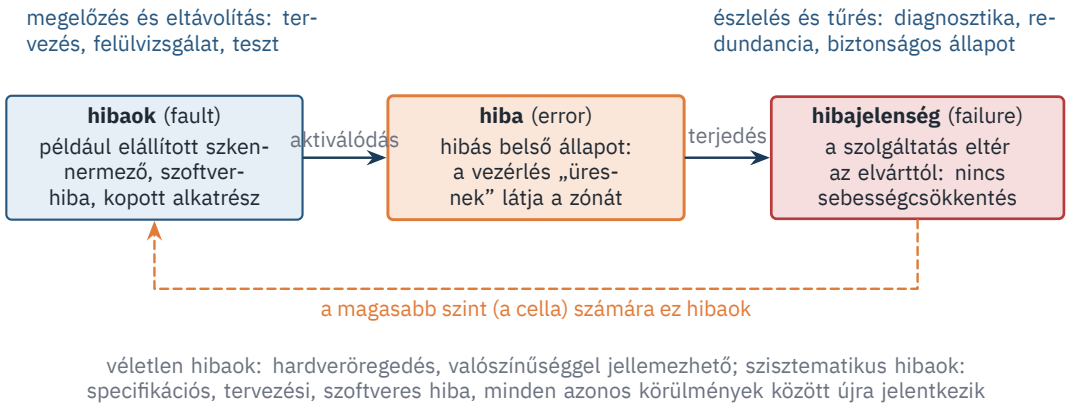
- Hogyan vezet egy hibaok veszélyes hibajelenséghez, és hogyan védekezünk ellene?
- Mennyire kell megbízhatónak lennie egy biztonsági funkciónak?
- Hogyan épül fel a rétegezett védelem, és mikor független egy réteg?

- Hogyan elemezzük a biztonságot, ha a baj nem alkatrészhibából, hanem az irányítás hiányosságaiból fakad?

17.1. Funkcionális biztonság és a hibamodell

Az elektromos, elektronikus és programozható elektronikus rendszerek biztonságának alapszabványa, az IEC 61508 szerint a **funkcionális biztonság** a teljes biztonságnak az a része, amely a biztonsággal kapcsolatos rendszerek helyes működésén múlik (IEC 2010a). A **biztonsági funkció** egy meghatározott veszélyes eseménnyel szemben biztonságos állapotot ér el vagy tart fenn: például a robot lelassítása, amikor ember lép a közös térbe. A szabvány szemléletét három gondolat foglalja össze: kockázatmentes állapot soha nem érhető el; a biztonságra a tervezés kezdetétől törekedni kell; és a nem tűrhető kockázatot csökkenteni kell (IEC 2010a). A szabvány szerint az irányított berendezés (equipment under control, EUC) alapvezérlésének el kell különülnie a biztonsági (védelmi) rendszertől; ma az adatátvitel közös hálózaton is történhet, de a két rendszer érzékelőinek és irányítóinak elkülönítése továbbra is követelmény.

A hibák terjedését háromszintű modell írja le (17.1. ábra). A **hibaok** (fault) a hiba feltételezett oka: elállított érzékelő, tervezési hiba, kopott alkatrész. Ha aktiválódik, **hibát** (error), hibás belső állapotot okoz. Ha a hiba a rendszer szolgáltatásáig terjed, **hibajelenség** (failure) lép fel: a szolgáltatás eltér az elvárttól. Egy alrendszer hibajelensége a befoglaló rendszer számára hibaok, így a lánc szintről szintre folytatódik (Avižienis és mtsai. 2004). Ez ugyanaz a szerkezet, mint az FMEA hibalánca (15. fejezet).



17.1. ábra. A hibaok, a hiba és a hibajelenség lánca a KoboCella egy lehetséges biztonsági eseményén.

A hibaokok két nagy csoportja eltérő védekezést kíván. A **véletlen hibaok** (a hardver öregedése, alkatrészhiba) valószínűséggel jellemezhető, és megbízhatósági modellel kezelhető (16. fejezet): redundanciával, diagnosztikával, rendszeres próbával. A **szisztematikus hibaok** (specifikációs, tervezési vagy szoftverhiba) azonos körülmények között mindig újra jelentkezik, ezért nem valószínűséggel, hanem a fejlesztési folyamat minőségével kezeljük: felülvizsgálattal, verifikációval, szigorú konfigurációkezeléssel. Avižienis és munkatársai rendszerezése szerint a megbízható rendszer négy eszközcsoportja a hibaokok megelőzése, tűrése, eltávolítása és előrejelzése (Avižienis és mtsai. 2004). Ugyanez a fejlesztés és az üzem szerint is felbontható: a hibaok elkerülése (a fejlesztés során), eltávolítása (tesztelés az üzembe helyezés előtt), a hiba észlelése (üzem közben) és elhárítása (beavatkozás, hogy hatása mielőbb mérséklődjön).

17.2. Biztonsági integritási szintek

Teljes funkcionális biztonság nem létezik: egyetlen biztonsági rendszer sem zárja ki teljesen sem a véletlen, sem a szisztematikus hibát. A **biztonsági integritás** annak valószínűsége, hogy egy biztonsági rendszer az előírt biztonsági funkciókat adott időszakban, meghatározott körülmények között megfelelően végrehajtja. Az IEC 61508 ezt négy **biztonsági integritási szintbe** (safety integrity level, SIL) sorolja; a szint határozza meg a megengedett hibavalószínűséget és az alkalmazandó tervezési, fejlesztési és üzemeltetési módszerek szigorúságát (IEC 2010a) (17.1. táblázat). A folyamatiparban az IEC 61511 alkalmazza ugyanezt a rendszert (IEC 2016b).

17.1. táblázat. A biztonsági integritási szintek az IEC 61508 szerint.

SIL	Alacsony igényű üzem: átlagos hibavalószínűség igénykor (PFD)	Magas igényű vagy folyamatos üzem: veszélyes hiba óránkénti valószínűsége (PFH)	Kockázatcsökkentés (alacsony igény)
4	10^{-5}–10^{-4}	10^{-9}–10^{-8}	10 000–100 000
3	10^{-4}–10^{-3}	10^{-8}–10^{-7}	1000–10 000
2	10^{-3}–10^{-2}	10^{-7}–10^{-6}	100–1000
1	10^{-2}–10^{-1}	10^{-6}–10^{-5}	10–100

Az **alacsony igényű üzemmódban** a biztonsági funkciót ritkán kell működtetni: a szabvány jelenlegi kiadása szerint az igény legfeljebb évente egyszer jelentkezik (a szabvány első, 1998-as kiadása ezen felül azt is kikötötte, hogy az igény ne legyen gyakoribb a bizonyító próbák gyakoriságának kétszeresénél). Itt az a kérdés, hogy a funkció működik-e, amikor szükség van rá; erről a rejtett hibák felderítésére végzett **bizonyító próbák** (proof test) gondoskodnak, amelyek a rendszert az újszerű vagy a gyakorlatilag elérhető legbiztonságosabb állapotba hozzák vissza. A **magas igényű vagy folyamatos üzemmódban** a funkció gyakran vagy állandóan működik, ezért a veszélyes hibák óránkénti gyakorisága a mérték.

Más ágazatok ugyanezt a gondolatot saját skálával fejezik ki. A gépek biztonsággal kapcsolatos vezérlésrészeire az ISO 13849-1 öt **teljesítményszintet** (performance level, PL a–e) határoz meg; a szükséges szintet kockázati gráf adja a sérülés súlyossága, a veszélynek való kitettség gyakorisága és a veszély elkerülésének lehetősége alapján (ISO 2023). A közúti járművekre az ISO 26262 négy **autóipari biztonsági integritási szintet** (ASIL A–D) használ; a szintet a súlyosság (S), a kitettség (E) és az irányíthatóság (C) kombinációja adja, a D a legszigorúbb, a legalacsonyabb kockázatú esetek pedig a szokásos minőségirányítás (QM) körébe tartoznak (ISO 2018a). A három rendszer rendezési elve azonos: a magasabb szint ritkább veszélyes hibát és szigorúbb fejlesztési folyamatot jelent.

17.3. Védelmi rétegek

A biztonságot ritkán egyetlen intézkedés adja. A rétegezett védelem gondolata a folyamatipari kockázatelemzésből ismert (17.2. ábra): a folyamat eredendően biztonságos tervezését az alapvezérlés, a riasztások és a kezelői beavatkozás, a független biztonsági funkció, a fizikai védelem és végül a vészhelyzeti reagálás veszi körül. A **védelmi rétegek elemzése** (layer of protection analysis, LOPA) egy kiinduló esemény gyakoriságát

a független védelmi rétegek hibavalószínűségével szorozza, és az eredményt a tűrhető gyakorisággal veti össze (Center for Chemical Process Safety 2001). Ez az eseményfa (16. fejezet) egyszerűsített, szabványosított változata.



17.2. ábra. A védelmi rétegek a folyamatipari gyakorlat szerint.

Egy réteg csak akkor számít **független védelmi rétegnek**, ha hatékony (önmagában képes megakadályozni a következményt), független a kiinduló eseménytől és a többi rétegtől, és működése ellenőrizhető (Center for Chemical Process Safety 2001). A függetlenség a leggyakoribb gyenge pont. Reason sajtmodellje szerint a rétegek lyukai „vándorolnak" (Reason 2000): a tervezésből kimaradt kölcsönhatás, a riasztást figyelmen kívül hagyó kezelő, a karbantartás alatt álló védelmi elem egy időben is előfordulhat (13. fejezet). A KoboCella hibafája (16. fejezet) ugyanezt mutatta mennyiségileg: amíg a sebességfelügyelet és az erőhatárolás ugyanazon a biztonsági vezérlésen futott, a két réteg nem volt független, és a kockázat 95%-a ebből a közös pontból eredt.

17.4. Rendszerelméleti biztonságelemzés (STPA)

Az eddigi módszerek (FMEA, hibafa, eseményfa) a baleseteket hibalánckén értelmezik: egy elem meghibásodik, és a hiba végigterjed. Leveson szerint a mai, szoftverrel irányított, sok szereplős rendszerekben a balesetek jelentős része nem alkatrészhibából fakad, hanem abból, hogy az irányítás nem tartja be a biztonsági korlátokat: minden elem a specifikációja szerint működik, de kölcsönhatásuk veszélyes állapotot hoz létre (Leveson 2011). A biztonság ebben a szemléletben irányítási probléma. Erre épül a **rendszerelméleti folyamatelemzés** (system-theoretic process analysis, STPA), amelynek négy lépése (Leveson és Thomas 2018):

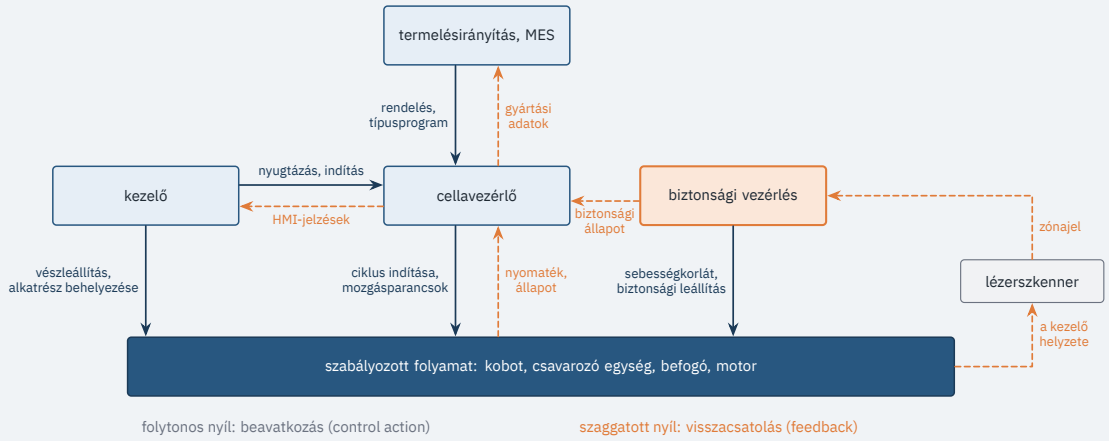
1. **A cél meghatározása:** a veszteségek (mit nem engedhetünk meg), a rendszerszintű veszélyek (a rendszer olyan állapotai, amelyek bizonyos körülmények között veszteséghez vezetnek) és a rendszerszintű biztonsági korlátok.
2. **Az irányítási struktúra modellezése:** az irányítók (ember, szoftver, szervezet), a beavatkozások (control actions) és a visszacsatolások.
3. **A nem biztonságos beavatkozások azonosítása:** minden beavatkozásra négy kérdés: veszélyes-e, ha *nem adják ki*; ha *kiadják*; ha *túl korán, túl későn vagy rossz sorrendben* adják ki; ha *túl sokáig tart vagy túl korán áll le*?

4. **A veszteséghez vezető forgatókönyvek azonosítása:** miért adhatja ki az irányító a nem biztonságos beavatkozást (hibás folyamatmodell, hiányzó visszacsatolás, rossz algoritmus), és miért nem hajtódhat végre helyesen egy helyes beavatkozás?

Az STPA ereje, hogy a szoftver, az ember és a szervezet hibáit ugyanabban a keretben vizsgálja, mint a hardverét. Az irányító **folyamatmodellje** (amit a szabályozott folyamatról „hisz”) központi fogalom: a nem biztonságos beavatkozások nagy része abból fakad, hogy az irányító modellje eltér a valóságtól. Ez az emberre is igaz; a helyzettudatosság (11. fejezet) ennek az emberi oldala.

KoboCella: STPA a cella irányítására

A KoboCella csapata a PHA két, vezérlésen múló veszélyére (ütközés, váratlan indulás) STPA-t végzett. A veszteségek: L1 a kezelő megsérül; L2 hibás motor jut a vevőhöz; L3 a termelés kiesik. A rendszerszintű veszélyek: H1 a kobot a kezelő közelében a biztonságos határt meghaladó sebességgel vagy erővel mozog (L1); H2 hibás kötésű motor hagyja el a cellát (L2); H3 a cella ok nélkül áll, vagy indokolatlanul leromlott üzemben marad (L3). Az irányítási struktúrát a 17.3. ábra mutatja.



17.3. ábra. A KoboCella irányítási struktúrája az STPA második lépésében.

A „teljes sebesség engedélyezése” beavatkozásra (biztonsági vezérlés → kobot) a négy kérdés:

17.2. táblázat. Nem biztonságos beavatkozások a „teljes sebesség engedélyezése” beavatkozásra.

Típus	Nem biztonságos beavatkozás	Veszély
nem adja ki	– (a cella lassan dolgozik, ez nem veszélyes)	H3
kiadja	a teljes sebességet engedélyezi, miközben a kezelő a közös térben van	H1
túl korán	a teljes sebességet azelőtt engedélyezi, hogy a kezelő teljesen elhagyta a zónát	H1
túl sokáig	a kezelő belépése után is fenntartja a teljes sebességet	H1

A forgatókönyvek közül három vezetett tervezési változáshoz. Először: a szkenner védőmezejét karbantartás után rosszul állítják vissza, így a biztonsági vezérlés folyamatmodellje szerint a zóna üres, holott nem az; erre a mezőkonfiguráció ellenőrzése és a napi indítási próba lett a válasz (az FMEA F7 sora, 15. fejezet). Másodszor: a kezelő egy olyan irányból is beléphet, amelyet a tálcaállvány takar a szkenner elől; az elrendezést módosították. Harmadszor: a kezelő „nyugtázás" beavatkozása veszélyes, ha a hibás darab még nincs a selejtdobozban (H2). Ezt a forgatókönyvet a csapat a próbaüzemben meg is figyelte (12. fejezet), és fizikai cselekvéshez kötötte. Az STPA tehát olyan okokat is feltárt, amelyek egyetlen alkatrész meghibásodásával sem írhatók le.

17.5. Ember–robot együttműködés biztonsága

Az ipari robotok biztonsági követelményeit az ISO 10218 szabvány két része rögzíti: az 1. rész a robotot, a 2. rész a robotalkalmazást és a robotcellát. A 2025-ös kiadás az együttműködő alkalmazásokra vonatkozó, korábban az ISO/TS 15066 műszaki specifikációban szereplő követelményeket is beépítette, és a „kobot" helyett az **együttműködő alkalmazás** fogalmát használja (ISO 2025a, 2025b). A szóváltás lényeges: nem a robot együttműködő, hanem az alkalmazás, és a biztonságot az adott alkalmazás kockázatértékelése alapozza meg. A szerszám, a munkadarab, az elrendezés és a kezelő feladatai legalább annyit számítanak, mint maga a robot.

Az együttműködés négy alapvető módja (biztonsági felügyelt megállás, kézi vezetés, sebesség- és távolságfelügyelet, teljesítmény- és erőkorlátozás) a 12. fejezetben szerepelt. A biztonsági elemzés szempontjából két mód kíván különös figyelmet:

- **Sebesség- és távolságfelügyeletnél** a robotnak meg kell állnia, mielőtt az ember elérné. A szükséges védőtávolság az ember közeledési sebességétől, a biztonsági vezérlés reakcióidejétől, a robot fékútjától és a mérési bizonytalanságoktól függ; ezért a szkenner, a vezérlés és a robot fékezése együtt alkotja a biztonsági funkciót, és együtt kell verifikálni.
- **Teljesítmény- és erőkorlátozásnál** az érintkezés megengedett, de az erő és a nyomás nem haladhatja meg a testtájanként eltérő biomechanikai határértékeket, amelyeket az ISO/TS 15066 közölt először (ISO 2016). Az éles, hegyes szerszámok és a becsípődési helyek ezt a módot kizárhatják, mert ott kis erő is sérülést okoz.

A biztonsági funkciókhoz szükséges teljesítményszintet (PL) a kockázatértékelés alapján kell meghatározni és a megvalósítást verifikálni (23. fejezet).

KoboCella: miért nem elég az erőkorlátozás?

A 12. fejezet szerint a KoboCella csapata a sebesség- és távolságfelügyeletet választotta elsődleges védelemnek. A kockázatértékelés ezt három érvvel támasztotta alá. A csavarozó fej bitbetétje éles és hegyes, ezért a kéz és a fej érintkezésénél a biomechanikai határérték már kis erőnél is túlléphető. A befogó és a fedél között becsípődési hely van, ahol az erő a kezet csapdába ejtheti. A műszakmegfigyelés szerint a kezelő a ciklusidő mintegy 30%-ában a közös térben van (a HTA szerint 18 s a 58,6 s-ból), ezért az érintkezés nem kivételes, hanem rendszeres esemény lenne. Az erő- és teljesítménykorlátozás ezért második rétegként maradt meg, és a hibafa tanulsága nyomán (16. fejezet) a csapat a két funkciót független csatornán valósította meg. A biztonsági funkciók szükséges teljesítményszintjét a csapat az ISO 13849-1 kockázati gráfjával határozta meg (súlyos sérülés lehetősége, gyakori kitettség, a csökkentett sebesség miatt korlátozottan elkerülhető veszély), és a verifikációs tervbe az ütközési erő mérését és a reakcióidő-mérést is felvette.

Biztonság és kiberbiztonság

A hálózatra kötött gépekben a biztonsági funkciókat szándékos támadás is veszélyeztetheti: egy manipulált konfiguráció vagy szoftverfrissítés ugyanúgy hamis folyamatmodellt adhat a biztonsági vezérlésnek, mint egy véletlen hiba. A 2025-ös robotbiztonsági szabványok ezért már kiberbiztonsági szempontokat is tartalmaznak, és a rendszerelméleti szemlélet a szándékos fenyegetéseket is az irányítási struktúra keretében, hamis visszacsatolásként vagy jogosulatlan beavatkozásként vizsgálhatja. A ForgácsTech kockázati nyilvántartásának kibertámadási kockázata (14. fejezet) ugyanennek a kérdésnek a projektszintű megjelenése.

A biztonság mint utólagos ráépítés

Gyakori, hogy a rendszert először a termelési funkciókra tervezik meg, és a biztonsági funkciókat a végén, a meglévő vezérlésre „ráépítve” egészítik ki. Ilyenkor a biztonsági funkció közös pontokat örököl az alapvezérléssel (közös érzékelő, közös szoftver), a függetlenség nem teljesül, és a verifikáció nehézkes. Az IEC 61508 alapelve szerint a biztonságra a tervezés kezdetétől törekedni kell: a veszélyelemzés a koncepcióval együtt indul, és a biztonsági követelmények a rendszerkövetelmények részei.

Összefoglalás

- A funkcionális biztonság a biztonságnak az a része, amely a biztonsági rendszerek helyes működésén múlik. A hibaok aktiválódva hibát, a hiba terjedve hibajelenséget okoz; egy alrendszer hibajelensége a befoglaló rendszer hibaoka.
- A véletlen hibaokok valószínűséggel kezelhetők (redundancia, diagnosztika, próba), a szisztematikusak a fejlesztési folyamat minőségével (felülvizsgálat, verifikáció, konfigurációkezelés).
- A SIL (IEC 61508), a PL (ISO 13849-1) és az ASIL (ISO 26262) a biztonsági funkciók megengedett hibavalószínűségét és a fejlesztés szigorúságát rögzíti; a magasabb szint ritkább veszélyes hibát jelent.
- A védelmi rétegek akkor csökkentik a kockázatot, ha hatékonyak, függetlenek és ellenőrizhetők; a LOPA ezt egyszerűsített eseményfaként számszerűsíti.
- Az STPA a biztonságot irányítási problémaként kezeli: veszteségek, veszélyek, irányítási struktúra, nem biztonságos beavatkozások négy típusa, veszteséghez vezető forgatókönyvek. Feltárja a hibátlan elemek veszélyes kölcsönhatásait is.
- Az együttműködő robotalkalmazás biztonságát az alkalmazás egészének kockázatértékelése alapozza meg; a szerszám, a becsípődési helyek és a kezelő feladatai döntik el, mely együttműködési mód elfogadható.

Önellenőrzés

1. Mi a különbség a megbízhatóság és a biztonság között? Adj példát megbízható, de veszélyes rendszerre!
2. Írd fel a hibaok–hiba–hibajelenség láncot egy lift ajtóérzékelőjére!
3. Miért nem kezelhető a szoftverhiba redundanciával ugyanúgy, mint a hardverhiba?
4. Egy biztonsági funkció átlagos hibavalószínűsége igénykor 0,004. Melyik SIL-szintet teljesíti alacsony igényű üzemben, és mekkora kockázatcsökkentést ad?
5. Mitől lesz egy védelmi réteg független? Miért nem független két réteg, amely ugyanazt az érzékelőt használja?
6. Sorold fel a nem biztonságos beavatkozások négy típusát, és adj mindegyikre példát egy közlekedési lámpa vezérlőjéből!
7. Miért mondja a 2025-ös robotbiztonsági szabvány, hogy nem a robot, hanem az alkalmazás együttműködő?
8. Miért nem elegendő a teljesítmény- és erőkorlátozás a KoboCella csavarozó fejénél?

Megoldás.

1. A megbízhatóság a követelményeknek megfelelő működés valószínűsége, a biztonság az elfo-

gadhatatlan kockázattól való mentesség. Megbízható, de veszélyes: egy pontosan és hibátlanul működő szállítószalag, amely védőburkolat nélkül behúzhatja a kezelő ruháját.
2. Hibaok: az ajtó fényérzékelőjének lencséje szennyezett. Hiba: a vezérlés úgy „látja”, hogy az ajtónyílás szabad. Hibajelenség: az ajtó záródik, miközben valaki a nyílásban áll.
3. Mert a szoftverhiba szisztematikus: két azonos szoftverpéldány azonos bemenetre ugyanúgy hibázik, ezért a redundáns példányok egyszerre esnek ki. Ellene a fejlesztési folyamat minősége (felülvizsgálat, teszt) vagy eltérő elvű (diverz) megoldás véd.
4. A 0,004 a 10^{-3}–10^{-2} tartományba esik: SIL 2; a kockázatcsökkentés $1/0{,}004 = 250$-szeres.
5. Ha a kiinduló eseménytől és a többi rétegtől független (nincs közös érzékelő, táp, szoftver, kezelő), önmagában hatékony, és működése ellenőrizhető. Közös érzékelőnél az érzékelő egyetlen hibája mindkét réteget kiejti, tehát valójában egy rétegről van szó.
6. Nem adja ki: nem vált pirosra, amikor a gyalogos átkelést kér, és jön egy autó. Kiadja: zöldet ad mindkét keresztező iránynak. Túl korán vagy későn: a pirosra váltás előtt nem ad sárgát, vagy túl későn vált. Túl sokáig tart vagy túl korán áll le: a gyalogos zöld túl rövid az átkeléshez.
7. Mert a biztonság nem csak a robottól függ: ugyanaz a robot egy puha munkadarabbal és tompa szerszámmal biztonságosan együttműködhet, éles szerszámmal vagy becsípődési helyekkel viszont nem. A kockázatot az alkalmazás egészére kell értékelni.
8. Mert a hegyes, éles bitbetét és a befogó becsípődési helyei miatt a biomechanikai határértékek már kis erőnél is túlléphetők, és a kezelő keze a ciklus jelentős részében a közös térben van, így az érintkezés rendszeres esemény lenne.

Projektlépés: biztonság

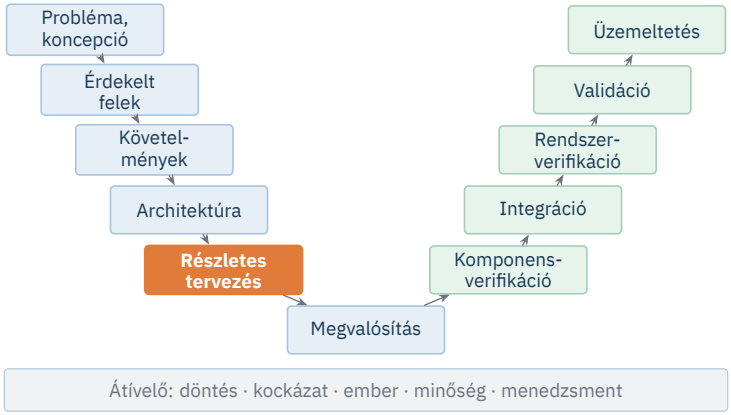
Saját fejlesztési feladatodhoz:

1. azonosítsd a rendszer biztonsági funkcióit (ha vannak), és határozd meg, milyen veszélyes esemény ellen védenek;
2. sorold fel a védelmi rétegeket egy kiinduló eseményre, és ellenőrizd a függetlenségüket;
3. végezz STPA-t a rendszer egy irányítási hurkára: veszteségek, veszélyek, irányítási struktúra, legalább két beavatkozás nem biztonságos változatai a négy típus szerint, és legalább három forgatókönyv;
4. használd a Gyakorlati útmutató STPA-eszközlapját, sablonját és notebookját.

Továbbolvasás

- Leveson (2011): a rendszerelméleti biztonságszemlélet alapműve, szabadon elérhető.
- Leveson és Thomas (2018): az STPA gyakorlati kézikönyve példákkal.
- IEC (2010a): a funkcionális biztonság alapszabványa; ISO (2023) és ISO (2018a) az ágazati megfelelők.
- Avižienis és mtsai. (2004): a megbízhatóság és a hibák fogalomrendszerének klasszikus összefoglalása.
- Center for Chemical Process Safety (2001): a védelmi rétegek elemzésének (LOPA) módszertana.
- ISO (2025b): az ipari robotalkalmazások és az együttműködő alkalmazások biztonsági követelményei.

18. Robusztus tervezés és tűrések



Tanulási eredmények

A fejezet feldolgozása után képes leszel:

- érzékenységvizsgálattal azonosítani a rendszer teljesítményét leginkább befolyásoló paramétereket;
- tűrésláncot számítani legrosszabb eset, négyzetes összegzés (RSS) és Monte Carlo-szimuláció alapján, és megítélni a módszerek feltevéseit;
- elmagyarázni a Taguchi-féle minőségi veszteségfüggvényt, és kiszámítani a várható veszteséget;
- teljes faktoros kísérletet tervezni és kiértékelni, főhatásokat és kölcsönhatásokat értelmezni;
- megkülönböztetni a tervezési és a zajtényezőket, és kölcsönhatásuk kihasználásával robusztus beállítást választani;
- terhelés–szilárdság interferenciával megbízhatóságot becsülni, és összevetni a biztonsági tényező szemléletével.

A korábbi fejezetek a meghibásodással foglalkoztak: mi romolhat el, milyen valószínűséggel, és mi a következménye. A minőségi problémák nagy része azonban nem meghibásodásból, hanem **szórásból** fakad: az alkatrészek méretei, az anyagjellemzők, a környezeti feltételek és a használat módja ingadozik, és a rendszer teljesítménye ezzel együtt ingadozik. A **robusztus tervezés** célja olyan rendszer, amelynek teljesítménye a lehető legkevésbé érzékeny ezekre az ingadozásokra. A fejezet a robusztus tervezés legfontosabb eszközeit mutatja be: az érzékenységvizsgálatot, a méretezést és a tűrésláncot, a minőségi veszteségfüggvényt, a kísérlettervezést és a valószínűségi tervezést (Phadke 1989; Taguchi, Chowdhury, és Wu 2005; Goldberg és mtsai. 1994, 4. fejezet).

A fejezet négy kérdésre épül:

- Melyik paraméter számít igazán?
- Hogyan halmozódnak az alkatrészek tűrései az összeállításban?
- Mennyibe kerül az eltérés a célértéktől, és hogyan csökkenthető a szórás érzékenysége?

- Hogyan becsülhető a tönkremenetel valószínűsége a biztonsági tényező helyett?

18.1. Érzékenységvizsgálat

Az **érzékenységvizsgálat** (paraméteres elemzés) azt vizsgálja, hogyan változik a rendszer egy jellemzője, ha egy tervezési paraméter megváltozik (Saltelli és mtsai. 2008). Legegyszerűbb formája az egyparaméteres változtatás: a többi paramétert névleges értékén tartva egyet változtatunk, és figyeljük a kimenetet. A paraméterek összevetéséhez a **relatív érzékenység** (rugalmasság) a hasznos:

$$E_x = \frac{\partial y}{\partial x} \cdot \frac{x}{y}, \tag{18.1}$$

amely megmutatja, hogy a paraméter 1%-os változása hány százalékkal változtatja a kimenetet. A KoboCella soros ciklusideje ($T = 58{,}6$ s, 6. fejezet) például a nyolc csavar meghúzási idejétől $25{,}6/58{,}6 = 0{,}44$, a kezelő idejétől $18/58{,}6 = 0{,}31$ rugalmassággal függ: a csavarozás 10%-os gyorsítása 4,4%-kal rövidíti a ciklust, a kezelő 10%-kal gyorsabb munkája csak 3,1%-kal. Az érzékenységvizsgálat ezzel megmutatja, hova érdemes a fejlesztési erőforrást összpontosítani; a döntéselemzésben a súlyok érzékenységét ugyanígy vizsgáltuk (10. fejezet).

Az egyparaméteres vizsgálat nem mutatja meg a paraméterek kölcsönhatását: két paraméter együttes változásának hatása eltérhet az egyenkénti hatások összegétől. Ezt a kísérlettervezés és a Monte Carlo-szimuláció tárja fel (lásd alább).

18.2. Méretezés és tűréslánc

Minden gyártott méret ingadozik, ezért a rajzon a névleges érték mellett a megengedett eltérést, a **tűrést** is meg kell adni. A méretezés és a tűrésmegadás szabályait (beleértve a geometriai tűréseket: alak, helyzet, irány) szabvány rögzíti (ASME 2018). Összeállításban az alkatrészek tűrései **tűrésláncot** alkotnak: egy funkcionálisan fontos méret (rés, átfedés, összenyomódás) több alkatrész méretének összegeként vagy különbségeként adódik. A tűrésláncot két klasszikus módon számolják (Chase és Parkinson 1991):

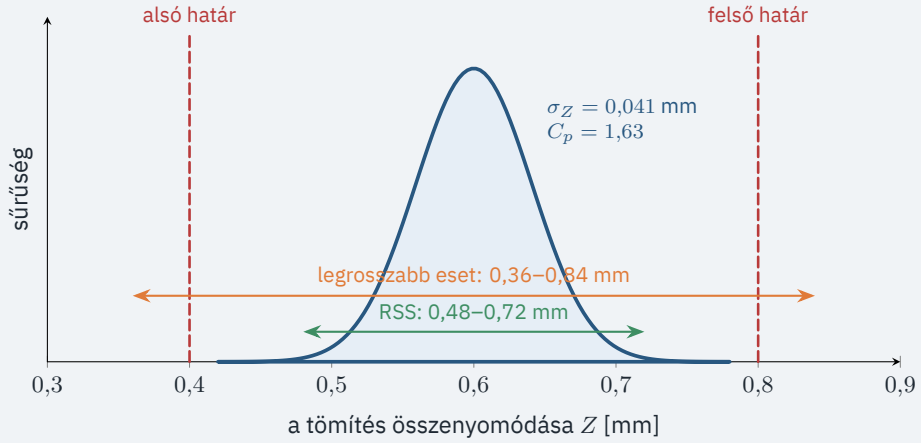
- **Legrosszabb eset:** az eredő tűrés a tagok tűréseinek összege, $T_Z = \sum |a_i| T_i$, ahol a_i a tag érzékenysége (lineáris láncnál ± 1). Garantálja, hogy minden összeállítás a határon belül marad, de sok tagnál nagyon szoros, drága alkatrésztűréseket követel.
- **Statisztikus (négyzetes) összegzés (RSS):** $T_Z = \sqrt{\sum a_i^2 T_i^2}$. Feltételezi, hogy a tagok egymástól függetlenek, normális eloszlásúak, a tűrésmező közepére szórnak, és a tűrés például $\pm 3\sigma$-nak felel meg. Ekkor a szélső értékek együttes előfordulása nagyon valószínűtlen, és lazább alkatrésztűrések is elegendők.

Ha a feltevések nem teljesülnek (eltolt középérték, nem normális eloszlás, összefüggő tagok, nemlineáris lánc), a **Monte Carlo-szimuláció** ad megbízható eredményt: a tagokat a tényleges eloszlásukból mintázzuk, és az eredő méret eloszlását közvetlenül számoljuk (Chase és Parkinson 1991).

KoboCella: a fedéltömítés összenyomódása

A fordított fedél hibamódja után (15. fejezet) a csapat a helyesen felhelyezett fedél tömítését is megvizsgálta. A tömítés akkor zár megbízhatóan, ha összenyomódása 0,40 és 0,80 mm között van. Az összenyomódás négy méretből adódik: a tömítés vastagsága ($2{,}50 \pm 0{,}08$ mm, +), a fedél horonymélysége ($2{,}20 \pm 0{,}05$ mm, –), a motorház tömítőperemének magassága ($0{,}30 \pm 0{,}06$ mm, +) és a fedél síklapúsági eltérése ($0 \pm 0{,}05$ mm, –). A névleges összenyomódás 0,60 mm.

A legrosszabb eset szerint az eredő tűrés $\pm 0{,}24$ mm, így az összenyomódás 0,36 és 0,84 mm között lehet: a követelmény nem teljesül. Az RSS szerint az eredő tűrés $\pm 0{,}12$ mm (0,48–0,72 mm), és ha a tűrések $\pm 3\sigma$-nak felelnek meg, az összenyomódás szórása 0,041 mm, a folyamatképesség $C_p = 1{,}63$ (18.1. ábra). A csapat a statisztikus méretezést fogadta el, mert napi mintegy 610 motornál a nagy darabszám indokolja, és a beszállítók folyamatképessége mérésekkel igazolható. Kikötötte viszont, hogy a tömítés vastagságát és a horonymélységet a beszállítók statisztikai folyamatszabályozással felügyeljék, mert a középérték eltolódása a statisztikus számítás legfontosabb feltevését rontaná. A Gyakorlati útmutató notebookja Monte Carlo-szimulációval azt is megmutatja, mekkora eltolódás fér bele a követelménybe.



18.1. ábra. A tömítés összenyomódásának eloszlása az RSS-feltevések mellett, a legrosszabb eset és az RSS-tartomány a követelmény határaival.

18.3. A minőségi veszteségfüggvény

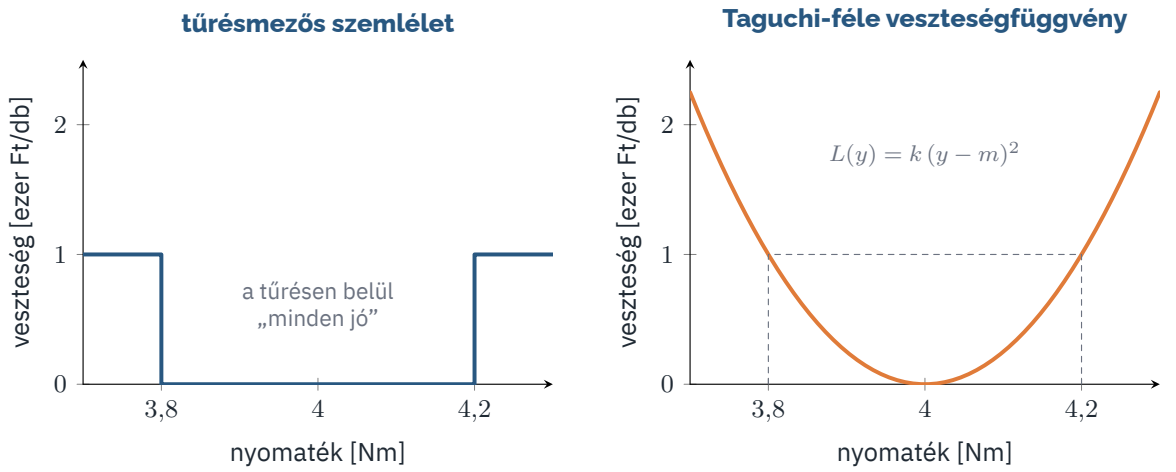
A hagyományos minőségszemlélet szerint a tűrésmezőn belüli termék jó, a kívüli rossz („kapufa-szemlélet”). Taguchi szerint azonban a célértéktől való minden eltérés veszteséget okoz a társadalomnak: a vevőnek, a gyártónak és a környezetnek, és ez a veszteség a célértéktől távolodva négyzetesen nő (Taguchi, Chowdhury, és Wu 2005) (18.2. ábra):

$$L(y) = k\,(y - m)^2, \qquad k = \frac{A_0}{\Delta_0^2}, \tag{18.2}$$

ahol m a célérték, Δ_0 az a funkcionális eltérés, amelynél a termék már nem teljesíti a funkcióját, A_0 pedig az ekkor fellépő veszteség (javítás, csere, reklamáció). Egy μ középértékű, σ szórású folyamat darabonkénti várható vesztesége:

$$\mathrm{E}[L] = k\left[\sigma^2 + (\mu - m)^2\right]. \tag{18.3}$$

A képlet a robusztus tervezés két irányát mutatja: a középértéket a célértékre kell állítani, és a szórást kell csökkenteni. Mindkettő pénzben kifejezhető, így az intézkedések költsége összevethető a veszteségcsökkentéssel.



18.2. ábra. A tűrésmezős szemlélet és a Taguchi-féle négyzetes veszteségfüggvény a csavarkötés nyomatékára.

Ha például a KoboCella csavarkötésének nyomatéka $4{,}0 \pm 0{,}2$ Nm, és a tűréshatáron egy motor javítása 1000 Ft-ba kerül, akkor $k = 1000/0{,}2^2 = 25\,000$ Ft/Nm². A $\mu = 4{,}02$ Nm középértékű, $\sigma = 0{,}05$ Nm szórású folyamat várható vesztesége $25\,000 \cdot (0{,}0025 + 0{,}0004) = 72{,}5$ Ft darabonként, évi 140 000 motornál mintegy 10 MFt. Ha a szórás 0,03 Nm-re csökken, és a középérték a célértékre kerül, a veszteség 22,5 Ft-ra, évente 3,2 MFt-ra csökken, pedig a tűrésmezős szemlélet szerint mindkét folyamat „hibátlan” termékeket gyárt.

18.4. Kísérlettervezés

A kísérlettervezés (design of experiments, DoE) arra szolgál, hogy a lehető legkevesebb kísérlettel megtudjuk, mely tényezők és milyen kölcsönhatásaik befolyásolják a vizsgált jellemzőt (Montgomery 2017). A „egyszerre egy tényező” módszerrel szemben a faktoros kísérlet minden tényezőt egyszerre változtat, így kevesebb kísérletből több információt ad, és a kölcsönhatásokat is feltárja.

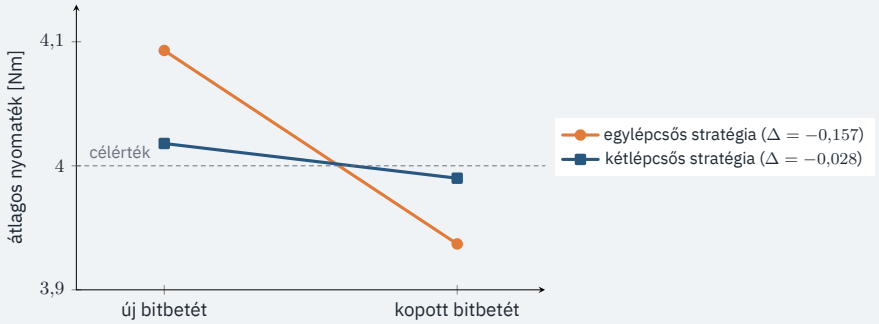
A 2^k **teljes faktoros kísérletben** k tényező mindegyikét két szinten (kódolva -1 és $+1$) vizsgáljuk, minden kombinációban. Egy tényező **főhatása** a válasz átlaga a felső szinten mínusz az átlag az alsó szinten; két tényező **kölcsönhatása** azt mutatja, mennyiben függ az egyik hatása a másik szintjétől. Montgomery a kísérletezés három alapelvét emeli ki: a **véletlen sorrendet** (hogy az időben változó zavaró hatások ne torzítsák az eredményt), az **ismétlést** (a kísérleti hiba becslésére) és a **blokkosítást** (az ismert zavaró tényezők kiszűrésére) (Montgomery 2017). Sok tényezőnél a **részfaktoros** tervek a kombinációk egy jól választott részhalmazával dolgoznak, a magasabb rendű kölcsönhatások összemosása árán.

Taguchi **paramétertervezése** a kísérletezést a robusztusság szolgálatába állítja. Megkülönbözteti a **tervezési tényezőket**, amelyeket a tervező szabadon beállíthat, és a **zajtényezőket**, amelyek a gyártásban vagy a használatban ingadoznak, és nem, vagy csak

drágán szabályozhatók (környezeti hőmérséklet, anyagingadozás, kopás). A cél olyan tervezési beállítás, amelynél a zajtényezők hatása a legkisebb, és ehhez éppen a tervezési és a zajtényezők közötti kölcsönhatást használja ki (Phadke 1989; Taguchi, Chowdhury, és Wu 2005). Taguchi a robusztusságot jel–zaj viszonnyal méri; Montgomery szerint ugyanez a cél általában hatékonyabban elérhető, ha a választ a tervezési és a zajtényezők függvényeként modellezzük, és a modellből választjuk ki a robusztus beállítást (Montgomery 2017).

KoboCella: robusztus csavarozási stratégia

A QFD-ből tudjuk (7. fejezet), hogy a nyomaték pontosságát a bitbetét kopása erősen befolyásolja. A csapat 2^3 teljes faktoros kísérletet végzett két ismétléssel (16 csavarozás véletlen sorrendben). Tényezők: A a végső fordulatszám (alacsony, magas), B a csavarozási stratégia (egylépcsős, kétlépcsős: gyors előhúzás, majd lassú meghúzás), C a bitbetét állapota (új, kopott). Az A és a B tervezési tényező, a C zajtényező: a kopás a gyártásban elkerülhetetlen. A főhatások: A $+0{,}086$ Nm, C $-0{,}092$ Nm, B elhanyagolható; a legnagyobb kölcsönhatás a B×C ($+0{,}065$ Nm). A kölcsönhatás a robusztusság kulcsa (18.3. ábra): egylépcsős stratégiánál a kopott bitbetét $0{,}157$ Nm-rel csökkenti a nyomatékot, kétlépcsősnél csak $0{,}028$ Nm-rel. A kétlépcsős stratégia tehát a zajtényezőre majdnem érzéketlen. A csapat ezt választotta, a fordulatszámmal pedig a középértéket állította a célértékre. A bitcsere-tervet (15. fejezet) megtartotta, de a cseréket ritkíthatta.



18.3. ábra. A bitbetét állapotának hatása a nyomatékra a két csavarozási stratégiánál (B×C kölcsönhatás).

18.5. Valószínűségi tervezés

A hagyományos méretezés **biztonsági tényezővel** dolgozik: az alkatrész szilárdsága legyen a várható terhelés valahányszorosa. A biztonsági tényező azonban nem mondja meg, mekkora a tönkremenetel valószínűsége, mert sem a terhelés, sem a szilárdság szórását nem veszi figyelembe. A **valószínűségi tervezés** mindkettőt valószínűségi változóként kezeli, és a tönkremenetel valószínűségét az eloszlásaik átfedéséből (**terhelés–szilárdság interferencia**) számítja (O'Connor és Kleyner 2012; Goldberg és mtsai. 1994, 3.14. szakasz). Normális eloszlású, független terhelés (L) és szilárdság (S) esetén:

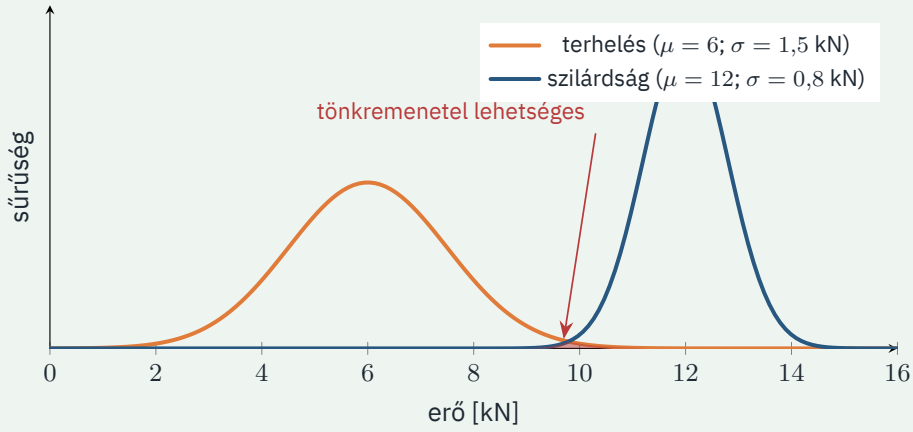
$$R = P(S > L) = \Phi\left(\frac{\mu_S - \mu_L}{\sqrt{\sigma_S^2 + \sigma_L^2}}\right), \tag{18.4}$$

ahol Φ a standard normális eloszlásfüggvény. A módszer a biztonsági tényezők gyakorlati alternatívája, de jelentős statisztikai tapasztalatot kíván. Eredménye erősen függ

az eloszlások szélétől, ezért csak akkor megbízható, ha a felhasznált adatok a tervezett populációra érvényesek (O'Connor és Kleyner 2012).

VárosiBringa: elég erős-e a reteszcsap?

A VárosiBringa dokkolójának reteszcsapja nyírószilárdságának várható értéke 12 kN, szórása 0,8 kN. A csapat a próbaüzem rángatási és vandalizmusadataiból a nagy terheléses események erejét 6 kN várható értékűnek és 1,5 kN szórásúnak becsülte. A biztonsági tényező 2, ami első ránézésre bőségesnek tűnik. Az interferenciaszámítás szerint (18.4. ábra) egy ilyen esemény $3{,}53$ szórásnyi távolságra van, a tönkremenetel valószínűsége eseményenként $2{,}1 \cdot 10^{-4}$. Ha a 300 dokkolóhely mindegyikét évente 20 ilyen esemény éri, az évente átlagosan 1,25 tönkrement reteszcsapot jelent. A csapat a csap anyagának erősítése helyett a terhelés szórását csökkentette: a dokkoló bevezető íve a kerékpárt rángatáskor megtámasztja, így a csapra jutó erő csúcsai kisebbek lettek.



18.4. ábra. A terhelés és a szilárdság eloszlása; az átfedés a tönkremenetel lehetőségét mutatja.

Bizonytalanság a teljes rendszerben

A tűrés, a veszteségfüggvény, a kísérlettervezés és az interferencia ugyanannak a gondolatnak a változatai: a tervezési döntéseket a bizonytalanság ismeretében kell meghozni, nem a névleges értékek alapján. Ugyanez a gondolat jelent meg a döntéselemzés Monte Carlo-vizsgálataiban (10. fejezet) és a megbízhatósági szimulációban (16. fejezet). A modellalapú fejlesztésben (3. fejezet) a rendszermodell paramétereihez eloszlás rendelhető, és a bizonytalanság a teljes modellen végigterjeszthető; a szimulációval kapott eloszlás mutatja meg, mekkora tartalékkal teljesülnek a követelmények.

Névleges értékre tervezni

Gyakori hiba, hogy a tervezés a névleges méretekkel és anyagjellemzőkkel számol, és a tűréseket a végén, a rajzkészítéskor, szokásértékekkel adja meg. Ilyenkor az összeállítás egy része nem működik, vagy a tűrések feleslegesen szigorúak és drágák. A funkcionálisan fontos méretláncokat már a részletes tervezés elején azonosítani kell, és a tűréseket a lánc elemzése alapján kell elosztani.

Összefoglalás

- Az érzékenységvizsgálat megmutatja, mely paraméterek befolyásolják leginkább a teljesítményt; a relatív érzékenység (rugalmasság) a paraméterek összevetésére alkalmas.
- A tűréslánc legrosszabb esetre számolva mindig biztonságos, de drága; az RSS-

számítás függetlenséget, normális eloszlást és központosított folyamatot feltételez. A feltevések sérülésekor Monte Carlo-szimuláció kell.

- Taguchi veszteségfüggvénye szerint a célértéktől való minden eltérés veszteség; a várható veszteség a szórás és a középérték-eltérés négyzetével arányos.
- A faktoros kísérlet egyszerre változtatja a tényezőket, így a főhatások mellett a kölcsönhatásokat is feltárja; alapelvei a véletlen sorrend, az ismétlés és a blokkosítás.
- A robusztus tervezés a tervezési és a zajtényezők kölcsönhatását használja ki: olyan beállítást keres, amelynél a zaj hatása a legkisebb.
- A valószínűségi tervezés a terhelés és a szilárdság szórásából számítja a tönkremenetel valószínűségét; azonos biztonsági tényező mellett is nagyon különböző lehet a kockázat.

Önellenőrzés

1. Egy rendszer kimenete $y = x_1 x_2^2$. Mekkora a relatív érzékenység x_1 és x_2 szerint?
2. Egy háromtagú lineáris lánc tűrései $\pm 0{,}1$, $\pm 0{,}2$ és $\pm 0{,}2$ mm. Mennyi az eredő tűrés legrosszabb esetre és RSS szerint?
3. Milyen feltevésekre épül az RSS-számítás? Mi történik, ha egy tag középértéke a tűrésmező szélére tolódik?
4. Egy jellemző célértéke 10, a funkcionális határ ± 1, a határon a veszteség 500 Ft. Mennyi a darabonkénti várható veszteség, ha $\mu = 10{,}2$ és $\sigma = 0{,}3$?
5. Mi a különbség a főhatás és a kölcsönhatás között? Miért nem tárja fel a kölcsönhatást az egyszerre egy tényezőt változtató kísérlet?
6. Mi a tervezési és a zajtényező? Adj példát mindkettőre egy háztartási gépnél!
7. Egy alkatrész szilárdsága $\mu_S = 100$, $\sigma_S = 5$, a terhelés $\mu_L = 70$, $\sigma_L = 10$ (azonos egységben). Mennyi a biztonsági tényező, és mennyi a tönkremenetel valószínűsége?
8. Miért mondhatjuk, hogy a biztonsági tényező önmagában nem jellemzi a kockázatot?

Megoldás.

1. $E_{x_1} = 1$, $E_{x_2} = 2$: x_2 1%-os változása kétszer akkora relatív változást okoz.
2. Legrosszabb eset: $0{,}1 + 0{,}2 + 0{,}2 = \pm 0{,}5$ mm; RSS: $\sqrt{0{,}01 + 0{,}04 + 0{,}04} = \pm 0{,}3$ mm.
3. A tagok függetlenek, normális eloszlásúak, a tűrésmező közepére szórnak, és a tűrés ismert szórásnak (például 3σ) felel meg. Ha egy tag középértéke eltolódik, az eredő méret középértéke is eltolódik, és a határon kívül eső összeállítások aránya jelentősen nőhet; ilyenkor Monte Carlo-szimuláció vagy az eltolódást figyelembe vevő számítás kell.
4. $k = 500/1^2 = 500$; $\mathrm{E}[L] = 500 \cdot (0{,}09 + 0{,}04) = 65$ Ft/db.
5. A főhatás egy tényező átlagos hatása; a kölcsönhatás azt mutatja, hogy egy tényező hatása függ egy másik tényező szintjétől. Az egyszerre egy tényezőt változtató kísérlet a többi tényezőt egy rögzített szinten tartja, így nem látja, hogyan változna a hatás más szinteken.
6. A tervezési tényezőt a tervező beállítja (például egy mosógép dobfordulatszáma vagy a programidő), a zajtényező ingadozik, és nem szabályozható (például a víz keménysége, a ruha mennyisége és anyaga, a hálózati feszültség).
7. A biztonsági tényező $100/70 = 1{,}43$; $z = 30/\sqrt{25 + 100} = 2{,}68$; a tönkremenetel valószínűsége $\Phi(-2{,}68) \approx 0{,}0037$.
8. Mert ugyanaz a biztonsági tényező kis szórásnál elhanyagolható, nagy szórásnál jelentős tönkremeneteli valószínűséget jelenthet; a kockázatot a középértékek különbsége és a szórások együtt határozzák meg.

Projektlépés: robusztusság

Saját fejlesztési feladatodhoz:

1. végezz érzékenységvizsgálatot a legfontosabb teljesítményjellemzőre (MOP vagy TPM), és rangsorold a paramétereket relatív érzékenységük szerint;
2. ha a rendszerben van funkcionálisan fontos méretlánc (rés, illesztés, összenyomódás), számítsd ki legrosszabb esetre, RSS szerint és Monte Carlo-szimulációval;
3. tervezz egy legalább három tényezős faktoros kísérletet (tényezők, szintek, válasz, ismétlések, sorrend), és jelöld a tervezési és a zajtényezőket;

4. használd a Gyakorlati útmutató tűréslánc- és kísérlettervezési eszközlapját.

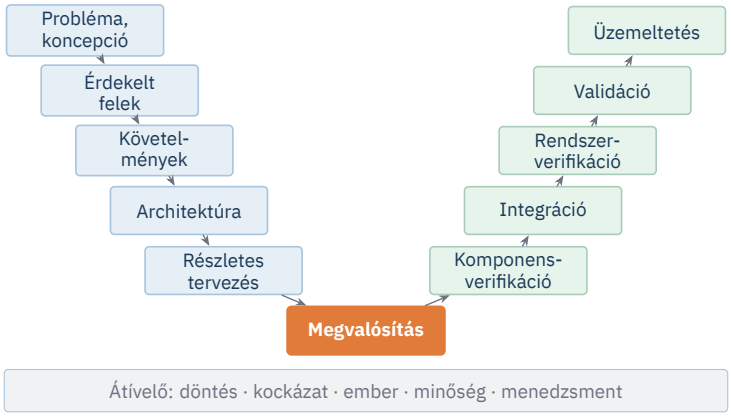
Továbbolvasás

- Montgomery (2017): a kísérlettervezés alapvető tankönyve, a robusztus tervezés válaszfelület-alapú megközelítésével.
- Taguchi, Chowdhury, és Wu (2005) és Phadke (1989): a Taguchi-módszer, a veszteségfüggvény és a paramétertervezés.
- Chase és Parkinson (1991): a tűrésanalízis módszereinek áttekintése.
- Saltelli és mtsai. (2008): az érzékenységvizsgálat módszereinek bevezető kézikönyve.
- Goldberg és mtsai. (1994), 3.14., 4.1–4.3., 7.5. és 7.13. szakasz: a valószínűségi tervezés, az érzékenységvizsgálat, a tűrések, a kísérlettervezés és a veszteségfüggvény gyakorlati leírása.

III. rész – A V csúcsa: megvalósítás és folyamattervezés

Pannon Egyetem
University of Pannonia

19. Folyamatelemzés és hatékonyság



Tanulási eredmények

A fejezet feldolgozása után képes leszel:

- leírni egy folyamatot az áram, a folyamatidő és a készlet jellemzőivel, és alkalmazni a Little-törvényt;
- meghatározni az erőforrások kapacitását, a szűk keresztmetszetet, a folyamat áramát és az erőforrások kihasználtságát;
- kiszámítani a munkatartalmat, a ciklusidőt és az átlagos munkaerő-kihasználtságot;
- kiszámítani az ütemidőt, Yamazumi-diagrammal értékelni egy szerelősor kiegyenlítettségét, és javaslatot tenni az átrendezésre;
- kiszámítani és értelmezni az OEE-t és az OLE-t, felismerni a hat nagy veszteséget és a mutatók korlátait;
- értékáram-térképpel feltárni a veszteségeket, és a SMED módszerével csökkenteni a beállítási időt.

A V-modell csúcsán a tervekből valóság lesz. Ha a fejlesztés tárgya maga egy gyártó- vagy szolgáltatófolyamat (egy szerelősor, egy javítóműhely, egy rendeléstelje sítési lánc), a megvalósítás magja a folyamat megtervezése és méretezése. Az ISO/IEC/IEEE 15288 megvalósítási folyamata a rendszerelemek legyártását, beszerzését vagy újrafelhasználását és a gyártás előkészítését foglalja magában (ISO/IEC/IEEE 2023, 6.4.7. szakasz; NASA 2016, 5.1. szakasz). A folyamat megtervezésének és javításának alapeszköze a folyamatábra: láthatóvá teszi a folyamat lépéseit és azok kapcsolatát, és segít kiszűrni a nem értékteremtő lépéseket (Rother és Shook 1999; Goldberg és mtsai. 1994, 7.15–16. szakasz).

A fejezet a műveletmenedzsment alapfogalmaira épül (Cachon és Terwiesch 2013; Hopp és Spearman 2008; Lakatos, é. n.b), és öt kérdésre keres választ:

- Mennyi egység halad át a folyamaton, mennyi ideig tart, és mennyi van belőle egyszerre a rendszerben?
- Melyik erőforrás korlátozza a teljesítményt, és mennyire használjuk ki a többit?
- Elég gyors-e a folyamat a vevői igényhez, és egyenletesen oszlik-e meg a munka?
- Mennyi veszteség keletkezik a berendezéseken és a munkában?

- Hol várakozik a termék, és hogyan rövidíthető az átfutás?

19.1. A folyamat alapjellemzői és a Little-törvény

Egy folyamatot három alapjellemzővel írunk le (Cachon és Terwiesch 2013, 2. fejezet):

- **áram** (átbocsátás, R): az egységnyi idő alatt a folyamaton áthaladó egységek száma;
- **folyamatidő** (átfutási idő, T): az az átlagos idő, amelyet egy egység a belépéstől a kilépésig a folyamatban tölt;
- **készlet** (I): az adott időpontban a folyamatban lévő egységek száma.

Az áramló egység a vizsgálat céljától függ: lehet munkadarab, rendelés, kerékpár, vevő vagy egy ügyirat. A három jellemzőt a **Little-törvény** kapcsolja össze (19.1. ábra):

$$I = R \cdot T. \tag{19.1}$$



19.1. ábra. A folyamat három alapjellemzője és a Little-törvény.

Little 1961-ben bizonyította, hogy az összefüggés stabil rendszerben a hosszú távú átlagokra érvényes, és nem függ az érkezések és a kiszolgálások eloszlásától, sem a kiszolgálás sorrendjétől (Little 1961). Ötven évvel később összegezte a törvény gyakorlati alkalmazásait: a műveletmenedzsmentben azért használják ilyen széles körben, mert a három mennyiség közül kettő mérésével a harmadik számítható (Little 2011). A törvény a rendszer határától függ: ugyanaz az összefüggés érvényes egy gépre, egy műhelyre vagy a teljes rendeléstelje sítési láncra, ha a három mennyiséget ugyanarra a határra mérjük. Az egységeknek is egyezniük kell: ha az áram rendelés/nap, a folyamatidő napban adódik.

> **ForgácsTech: miért tart tíz napig egy rendelés?**
>
> A ForgácsTech üzemében egy átlagos hétköznapon 60 rendelés van folyamatban (a gyártáselőkészítéstől a kiszállításig), és naponta átlagosan 6 rendelés készül el. A Little-törvény szerint egy rendelés átlagosan $60/6 = 10$ munkanapot tölt az üzemben. A vevő által elfogadható átfutás 5 munkanap. Ha az áram nem változik, ehhez a folyamatban lévő rendelések számát 30-ra kell csökkenteni. Ez nem gyorsabb gépeket igényel, hanem azt, hogy kevesebb rendelés várakozzon egyszerre: a rendelések kiadását a kapacitáshoz kell igazítani (21. fejezet). A számítás azt is megmutatja, miért keletkezik a késések fele a beállításnál és az anyagbeszerzésnél (4. fejezet): a rendelések az idejük nagy részét nem megmunkálással, hanem várakozással töltik (19.6. szakasz).

19.2. Kapacitás, szűk keresztmetszet, kihasználtság

Egy erőforrás **kapacitása** az az áram, amelyet akkor teljesítene, ha folyamatosan dolgozna. Ha az erőforráson m_i párhuzamos egység (dolgozó vagy gép) dolgozik, és egy egység feldolgozása p_i ideig tart, a kapacitás $K_i = m_i / p_i$. A folyamat kapacitását a legkisebb kapacitású erőforrás, a **szűk keresztmetszet** határozza meg. A folyamat tényleges árama a kereslet és a folyamatkapacitás közül a kisebb, egy erőforrás **kihasználtsága** pedig a tényleges áram és a kapacitás hányadosa (Cachon és Terwiesch 2013, 3. fejezet):

$$K_i = \frac{m_i}{p_i}, \qquad R = \min\left\{D,\ \min_i K_i\right\}, \qquad u_i = \frac{R}{K_i}. \tag{19.2}$$

Ha a kereslet meghaladja a folyamatkapacitást, a folyamat **kapacitáskorlátos**, és a szűk keresztmetszet kihasználtsága 100%. Ha a kereslet kisebb, a folyamat **keresletkorlátos**, és minden erőforrásnak van szabad kapacitása. A szűk keresztmetszet szemléletét Goldratt tette ismertté: a rendszer teljesítményét a korlátozó erőforrás szabja meg, ezért a szűk keresztmetszeten elvesztett idő az egész rendszer számára elveszett, a többi erőforráson megtakarított idő viszont nem növeli a kibocsátást (Goldratt és Cox 1984). A fejlesztést tehát a szűk keresztmetszeten kell kezdeni, és minden javítás után újra meg kell vizsgálni, hol van a korlát.

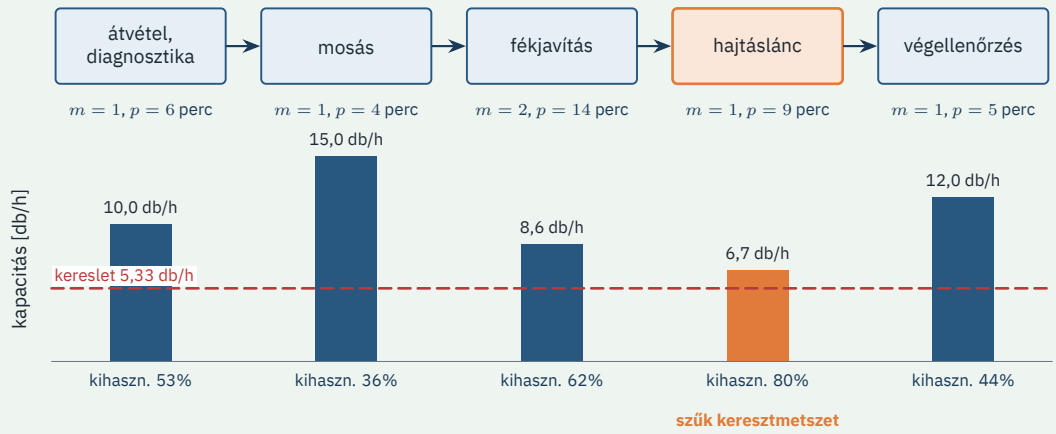
Összetett folyamatokban, ahol több terméktípus különböző útvonalon halad, az elemzés lépései a következők (Cachon és Terwiesch 2013, 3. fejezet):

1. a folyamatábra elkészítése: hogyan áramlik az egység az erőforrásokon át;
2. minden erőforrás kapacitásának meghatározása;
3. az egységek munkaigényének táblázatos összesítése erőforrásonként és terméktípusonként;
4. az igények összesítése erőforrásonként;
5. a származtatott kihasználtság számítása: az összesített igény és a kapacitás hányadosa; a legnagyobb értékű erőforrás a szűk keresztmetszet.

VárosiBringa: a javítóműhely kapacitása

A VárosiBringa javítóműhelyébe naponta átlagosan 40 kerékpár érkezik egy 7,5 órás műszakban, azaz a kereslet $D = 5{,}33$ db/h. Minden kerékpár öt lépésen halad át (19.2. ábra). A fékjavításon két szerelő dolgozik párhuzamosan, a többi lépésen egy-egy. A kapacitások: átvétel és diagnosztika $60/6 = 10$ db/h, mosás $60/4 = 15$ db/h, fékjavítás $2 \cdot 60/14 = 8{,}6$ db/h, hajtáslánc $60/9 = 6{,}7$ db/h, végellenőrzés $60/5 = 12$ db/h. A szűk keresztmetszet a hajtáslánc-javítás, a folyamatkapacitás 6,7 db/h, azaz műszakonként 50 kerékpár. A folyamat keresletkorlátos: az áram 5,33 db/h, a hajtásláncon dolgozó szerelő kihasználtsága 80%, a mosóé csak 36%.

A műhelyvezető úgy tervezte, hogy a nyári csúcsban, napi 55 kerékpárnál (7,33 db/h) egy második mosóberendezést állít be. A számítás szerint ez semmit nem javítana: a csúcsban a hajtáslánc-javítás kapacitása (6,7 db/h) lesz kevés. A második mosó helyett a hajtáslánc-javítás egy részét (a lánc tisztítását és kenését, 3 perc) a mosóhoz csoportosították át. Ekkor a hajtáslánc ideje 6 percre, kapacitása 10 db/h-ra nő, a mosásé 7 percre nő, kapacitása 8,6 db/h-ra csökken, és az új szűk keresztmetszet a fékjavítás és a mosás 8,6 db/h-val: ez a csúcsigényt is fedezi.



19.2. ábra. A VárosiBringa javítóműhelyének kapacitásai és a kereslet. A szűk keresztmetszet a hajtáslánc-javítás.

A kihasználtság nem lehet tartósan 100%. Ha az érkezések és a feldolgozási idők ingadoznak, a várakozási idő a kihasználtság növekedésével egyre gyorsabban nő, és a teljes kihasználtság közelében korlátlanul megnő (Hopp és Spearman 2008, 8. fejezet). Ezt a sorbanállási modellek segítségével a 21. fejezet vizsgálja.

19.3. Munkaerő-termelékenység

A gépek mellett a munka hatékonyságát is mérni kell. A **munkatartalom** egy egység elkészítéséhez szükséges összes kézi munka ideje, a **ciklusidő** két egymást követő egység elkészülése között eltelt idő, azaz az áram reciproka ($CT = 1/R$). Ha egy erőforráson m_i dolgozó dolgozik, és az erőforrás munkaigénye egységenként p_i, egy ciklus alatt a dolgozóknak $m_i \cdot CT - p_i$ kihasználatlan ideje marad. Az **átlagos munkaerő-kihasználtság** a munkatartalom és a ciklusonként fizetett teljes munkaidő hányadosa (Cachon és Terwiesch 2013, 4. fejezet):

$$\bar{u} = \frac{\sum_i p_i}{\sum_i p_i + \sum_i (m_i \cdot CT - p_i)} = \frac{\sum_i p_i}{CT \cdot \sum_i m_i}. \tag{19.3}$$

A VárosiBringa műhelyében a munkatartalom $6 + 4 + 14 + 9 + 5 = 38$ perc, a ciklusidő $1/5{,}33 = 11{,}25$ perc, a dolgozók száma 6. Az átlagos munkaerő-kihasználtság $38/(11{,}25 \cdot 6) = 56{,}3\%$, azaz a fizetett munkaidő 44%-a kihasználatlan. A közvetlen munkaerőköltség egy kerékpárra a ciklusidő, a létszám és az órabér szorzata, és nem a munkatartalomé: a kihasználatlan időt is ki kell fizetni. A munkaerőköltség egy termék teljes költségében első ránézésre kicsi, de a beszállítói láncon végigösszegezve meghatározó lehet, ezért a munkaerő-termelékenység javítása a beszállítói lánc minden szintjén fontos.

19.4. Ütemidő és sorkiegyenlítés

Ha a folyamatnak a vevői igényt kell követnie, a tervezés kiindulópontja az **ütemidő** (takt time): a rendelkezésre álló termelési idő és a kereslet hányadosa, vagyis az az idő, amely egy termék előállítására a vevői igény szerint jut (Rother és Shook 1999):

$$TT = \frac{\text{rendelkezésre álló termelési idő}}{\text{kereslet}}. \tag{19.4}$$

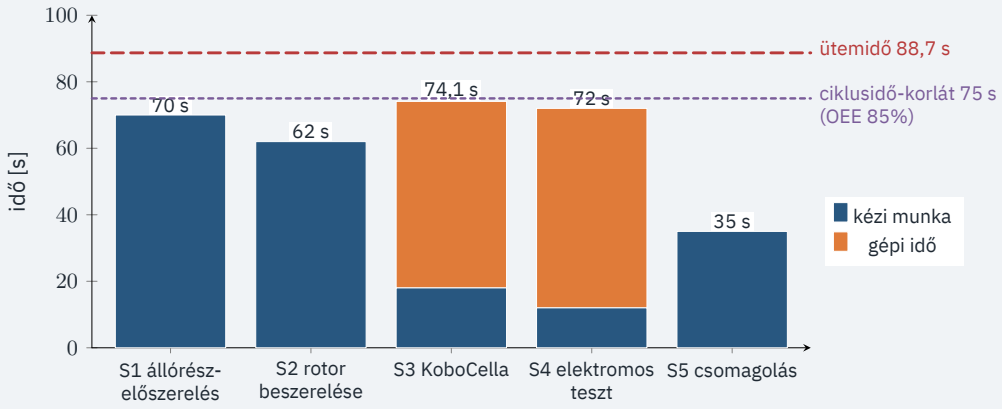
Az ütemidőt a **ciklusidővel** vetjük össze, amely egy művelet vagy egy állomás tényleges időigénye. Ha egy állomás ciklusideje nagyobb az ütemidőnél, a sor nem tudja kiszolgálni az igényt; ha jóval kisebb, az állomáson várakozás és kihasználatlan munkaidő keletkezik. A **sorkiegyenlítés** (line balancing) célja, hogy a műveleteket úgy osszuk el az állomások között, hogy egyik állomás ideje se haladja meg az ütemidőt, és a munka a lehető legegyenletesebben oszoljon meg. Az elméletileg szükséges legkisebb állomásszám és a kiegyenlítés hatásfoka:

$$N_{\min} = \left\lceil \frac{\sum_j t_j}{TT} \right\rceil, \qquad \eta = \frac{\sum_j t_j}{N \cdot c}, \tag{19.5}$$

ahol t_j a műveletek ideje, N az állomások száma és c a legnagyobb állomásidő. A műveletek sorrendi kötöttségei (előbb kell a rotort beszerelni, mint tesztelni) miatt az elméleti minimum nem mindig érhető el. A feladat két alapváltozata a legkevesebb állomás keresése adott ütemidőre és a legkisebb ciklusidő keresése adott állomásszámra. Mindkettő kombinatorikus optimalizálási feladat, amelyre pontos és heurisztikus eljárások egyaránt léteznek (Scholl és Becker 2006).

A **Yamazumi-diagram** (japánul: halmozott) a sorkiegyenlítés szemléletes eszköze: állomásonként egy halmozott oszlopot rajzol, amelyben az egyes műveletek vagy időfajták (kézi munka, gépi idő, gyaloglás, várakozás) egymásra rakódnak, és vízszintes vonal jelöli az ütemidőt. Egy pillantással látszik, melyik állomás lépi túl az ütemidőt, és hol van tartalék.

> **KoboCella: a szerelősor kiegyenlítése**
>
> A KoboCella egy ötállomásos motorszerelő sor harmadik állomása. A sor két műszakban, évi 3450 órában dolgozik, és évi 140 ezer motort kell gyártania, így az ütemidő 88,7 s. A tervezett 85%-os berendezéshatékonyság miatt egyik állomás ciklusideje sem lehet nagyobb 75 s-nál (6. fejezet). A 19.3. ábra az állomások idejét mutatja; a KoboCella oszlopa a helyszíni átvételen mért 74,1 s, amelyből 18 s a kezelő kézi munkája.



19.3. ábra. A KoboCella-sor Yamazumi-diagramja. Az ütemidő 88,7 s, a 85%-os berendezéshatékonyságból adódó ciklusidő-korlát 75 s.

A munkatartalom $70+62+74{,}1+72+35 = 313{,}1$ s, az elméleti minimum $\lceil 313{,}1/88{,}7 \rceil = 4$ állomás. A szűk keresztmetszet a KoboCella 74,1 s-mal, a sor kapacitása $3600/74{,}1 = 48{,}6$ motor/óra, az igény $3600/88{,}7 = 40{,}6$ motor/óra. Minden állomás a 75 s-os korlát alatt marad, de a KoboCella tartaléka mindössze 0,9 s: a cella ciklusidejének bármilyen romlása az egész sor kibocsátását veszélyezteti, ezért a cella ciklusidejét a gyártás során is folyamatosan követni kell. A kiegyenlítés hatásfoka $313{,}1/(5 \cdot 74{,}1) = 84{,}5\%$.
A diagram a kézi munka és a gépi idő szétválasztásával többet mutat a puszta állomásidőknél. A harmadik, negyedik és ötödik állomás kezelőjének kézi munkája együtt $18 + 12 + 35 = 65$ s, ami az ütemidőn belül marad, mert a kobot és a tesztberendezés a kezelő nélkül is dolgozik. A csapat ezért azt vizsgálta, átveheti-e a KoboCella kezelője a csomagolást is: ekkor kézi terhelése $18 + 35 = 53$ s lenne ütemenként, azaz az ütemidő 60%-a, a sor létszáma pedig ötről négyre csökkenne. A változtatást a csapat csak az elrendezés módosítása (a csomagolóasztal a cella kiadóoldalára kerül) és a terhelés ismételt mérése után vezette be (11.6. szakasz). A kezelő kis lekötöttsége ugyanis nem csak tartalék: a felügyeleti feladat és a zavarelhárítás figyelmet igényel, és a leromlott üzemben a kezelőnek kell átvennie a cella egyes feladatait (12. fejezet).

19.5. OEE és OLE: a berendezések és a munka hatékonysága

A **teljes eszközhatékonyság** (Overall Equipment Effectiveness, OEE) Nakajima teljes körű hatékony karbantartási (TPM) koncepciójának mérőszáma. Azt méri, hogy a tervezett üzemidőből mennyi fordítódik ténylegesen jó termék előállítására a gép elméleti sebességével. A veszteségeket hat csoportba sorolja, ezek a **hat nagy veszteség** (Nakajima 1988):

- **állásidő-veszteségek:** (1) meghibásodások és (2) beállítás, átállás;
- **sebességveszteségek:** (3) üresjárat és kisebb leállások, (4) csökkentett sebesség;
- **minőségi veszteségek:** (5) selejt és utómunka, (6) indítási (felfutási) veszteségek.

A három veszteségcsoportnak egy-egy tényező felel meg, és az OEE ezek szorzata:

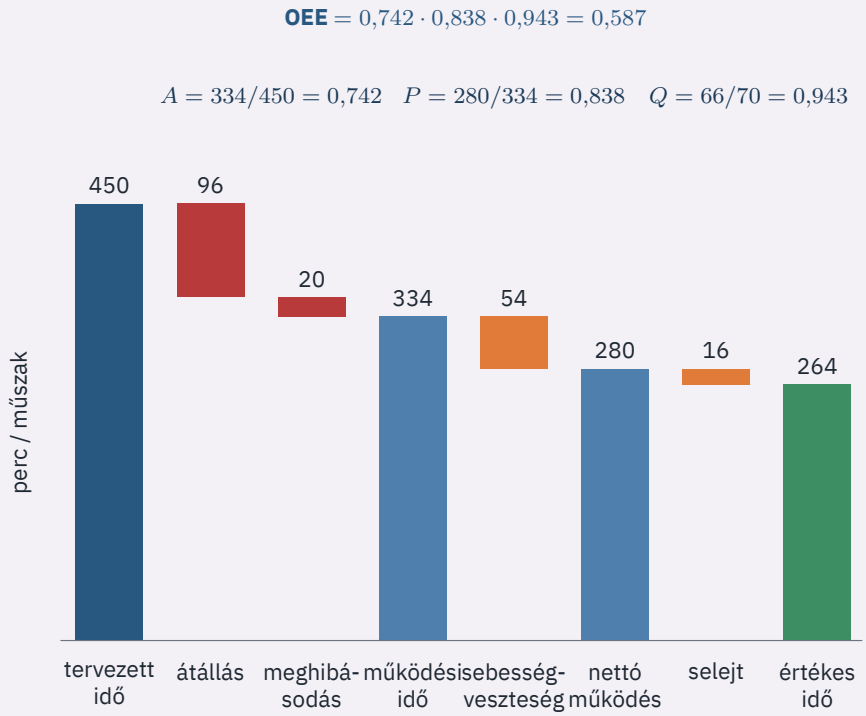
$$\mathrm{OEE} = A \cdot P \cdot Q, \qquad A = \frac{T_\mathrm{m}}{T_\mathrm{t}}, \qquad P = \frac{t_\mathrm{i} \cdot n}{T_\mathrm{m}}, \qquad Q = \frac{n_\mathrm{j}}{n}, \tag{19.6}$$

ahol T_t a tervezett üzemidő, T_m a tényleges működési idő, t_i az ideális ciklusidő, n a legyártott és n_j a jó darabok száma. A szorzat egyszerűsítve a jó darabok ideális gyártási idejének

és a tervezett üzemidőnek a hányadosa: OEE $= t_{\mathrm{i}} \cdot n_{\mathrm{j}} / T_{\mathrm{t}}$. A szakirodalom Nakajimára hivatkozva adja meg azokat az ideális értékeket, amelyek mellett a gép kiemelkedően hatékonynak tekinthető: rendelkezésre állás 90% felett, teljesítmény 95% felett, minőség 99% felett, ami mintegy 85%-os OEE-t ad (Nakajima 1988). A KoboCella követelményeiben rögzített 85%-os berendezéshatékonyság (6. fejezet) ezt a szintet célozza.

ForgácsTech: az OEE-vízesés

A ForgácsTech az M3 koncepció első ütemében gépi adatgyűjtést és OEE-monitorozást vezetett be (10. fejezet). Az egyik többtengelyes megmunkáló központ első négy hetének műszakonkénti átlagát a 19.4. ábra mutatja. A 450 perces tervezett üzemidőből 96 percet vitt el két átállás és 20 percet a meghibásodások, így a rendelkezésre állás $334/450 = 0{,}742$. A gép műszakonként 70 darabot gyártott; ideális ciklusideje 4 perc, így a teljesítménytényező $70 \cdot 4/334 = 0{,}838$. A 70 darabból 66 volt jó, a minőségi tényező $0{,}943$. Az OEE $0{,}742 \cdot 0{,}838 \cdot 0{,}943 = 0{,}587$: a tervezett üzemidőnek csak 59%-a jelent jó terméket. A legnagyobb veszteség az átállás; ezt a 19.7. szakasz vizsgálja.



19.4. ábra. A ForgácsTech megmunkáló központjának OEE-vízesése egy műszakra (perc).

Az OEE népszerűségét egyszerűsége adja, korlátait azonban ismerni kell. Muchiri és Pintelon irodalmi áttekintése szerint a mutató egyetlen gépre vonatkozik, és nem mutatja meg a gépek egymásra hatását; a három tényezőt egyenlő súllyal kezeli; értéke erősen függ az adatgyűjtés pontosságát (a kézzel rögzített kisebb leállások például gyakran kimaradnak); és mivel az üzemek céljai különböznek, nincs mindenhol érvényes viszonyítási érték. Kis sorozatú egyedi gyártásban a mutató kevésbé hasznos, pufferekkel tagolt folyamatokban pedig át kell értelmezni (Muchiri és Pintelon 2008). A ForgácsTech esetében ezért az OEE-t a gépek saját korábbi értékeihez hasonlítják, és nem iparági átlagokhoz; a szűk keresztmetszetként működő gépek OEE-je fontosabb, mint a többié. Ha a tervezett üzemidőt a naptári időhöz viszonyítjuk, a teljes effektív eszközteljesítmény (TEEP) azt is megmutatja, mennyi kapacitást hagyunk kihasználatlanul a műszakrend miatt (Muchiri és Pintelon 2008).

Az OEE gondolata a munkára is kiterjeszthető. A **teljes munkaerő-hatékonyság** (Overall Labour Effectiveness, OLE) a munkaerő rendelkezésre állásának, teljesítményének és minőségének szorzata: a rendelkezésre állást a távollétek, a várakozás és a szervezési veszteségek, a teljesítményt a normaidőhöz viszonyított lassúság, a minőséget az utómunka csökkenti. Braglia és szerzőtársai bírálták a mutató hagyományos formáját, mert a veszteségek egy része nehezen mérhető, és olyan veszteségstruktúrát javasoltak, amely közvetlenül mérhető mennyiségekre épül, és a szakaszos gyártásban is alkalmazható (Braglia és mtsai. 2021). A VárosiBringa műhelyének egyik szerelője egy 450 perces műszakban 405 percig dolgozhatott (a többi idő megbeszélésre és alkatrészre várásra ment el), a 405 perc alatt elvégzett munkák normaideje 340 perc volt, és a javítások 95%-a elsőre sikerült. Ekkor OLE $= 0{,}90 \cdot 0{,}84 \cdot 0{,}95 \approx 0{,}72$. A mutató értelme nem a dolgozók rangsorolása, hanem a munkaszervezési veszteségek feltárása: a 45 perces várakozás oka az alkatrész-ellátásban van, nem a szerelőben.

19.6. Értékáram és veszteségek

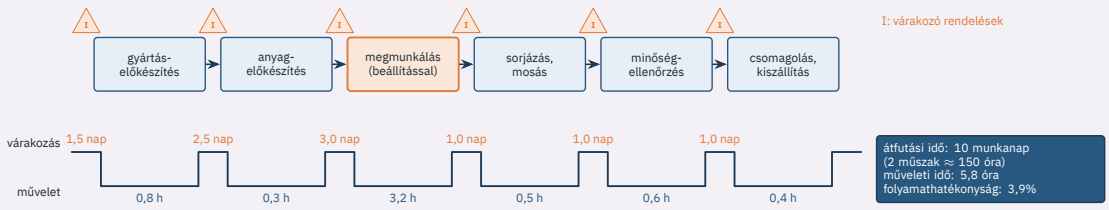
A Toyota termelési rendszerének egyik alapgondolata, hogy a termelésben minden tevékenység vagy értéket teremt a vevő számára, vagy veszteség. Ohno hét veszteségfajtát különböztetett meg: a túltermelést, a várakozást, a szállítást, a felesleges feldolgozást, a készletet, a felesleges mozgást és a hibás termék előállítását (Ohno 1988). Közülük a túltermelést tartotta a legsúlyosabbnak, mert a többi veszteséget eltakarja.

Az **értékáram-térkép** (value stream mapping, VSM) a termék útját ábrázolja a nyersanyagtól a vevőig: a folyamatlépéseket, az anyag- és információáramlást, a lépések közötti készleteket, és alul egy lépcsős idővonalon a várakozási és a műveleti időket (Rother és Shook 1999). Az idővonalból számítható a **folyamathatékonyság**, a műveleti idő és az átfutási idő hányadosa:

$$\text{PCE} = \frac{\sum T_{\text{művelet}}}{T_{\text{átfutás}}}. \tag{19.7}$$

Rother és Shook módszerében a térképezés két lépésből áll: először a jelenlegi állapotot rajzoljuk fel a gyártótérben mért adatokkal, majd a jövőbeli állapotot tervezzük meg, amelyben a termék a lehető legkevesebb várakozással, a vevői ütemhez igazodva áramlik (Rother és Shook 1999). A térkép elkészítésének lépéseit a Gyakorlati útmutató folyamatábra- és értékáramtérkép-eszközlapja mutatja be.

> **ForgácsTech: a rendelés értékárama**
>
> A ForgácsTech csapata egy jellemző rendelés útját térképezte fel (19.5. ábra). A hat lépés műveleti ideje együtt 5,8 óra, a lépések előtti várakozás együtt 10 munkanap, ami két műszakkal 150 óra. A folyamathatékonyság $5{,}8/150 = 3{,}9\%$: a rendelés az idejének több mint 96%-át várakozással tölti. A leghosszabb várakozás (3 nap) a megmunkáló központ előtt, a második leghosszabb (2,5 nap) az anyag-előkészítés előtt keletkezik; ez egyezik a problémaelemzés megállapításával, hogy a késések fele a beállításnál és az anyagbeszerzésnél keletkezik (4. fejezet). A 10 napos átfutás a Little-törvénnyel számított értékkel is egyezik (19.1. szakasz). A jövőbeli állapotban a csapat a rendeléskiadást a megmunkáló központ kapacitásához köti, az anyagot a rendelés kiadásakor foglalja le, és a beállítási időt csökkenti.



19.5. ábra. A ForgácsTech egy rendelésének egyszerűsített értékáram-térképe. Fent a folyamatlépések és a várakozó rendelések, lent az idővonal.

19.7. A beállítási idő csökkentése: SMED

A beállítási (átállási) idő az az idő, amely az egyik termék utolsó jó darabja és a következő termék első jó darabja között telik el. Ha hosszú, a gyártó nagy sorozatokat gyárt, hogy ritkán kelljen átállni, a nagy sorozatok pedig nagy készletet és hosszú átfutást okoznak. Shingo **SMED** (single-minute exchange of die, percre kész szerszámcsere) módszere a beállítási időt tíz perc alá kívánja csökkenteni. Kiindulópontja a beállítási műveletek két csoportja (Shingo 1985):

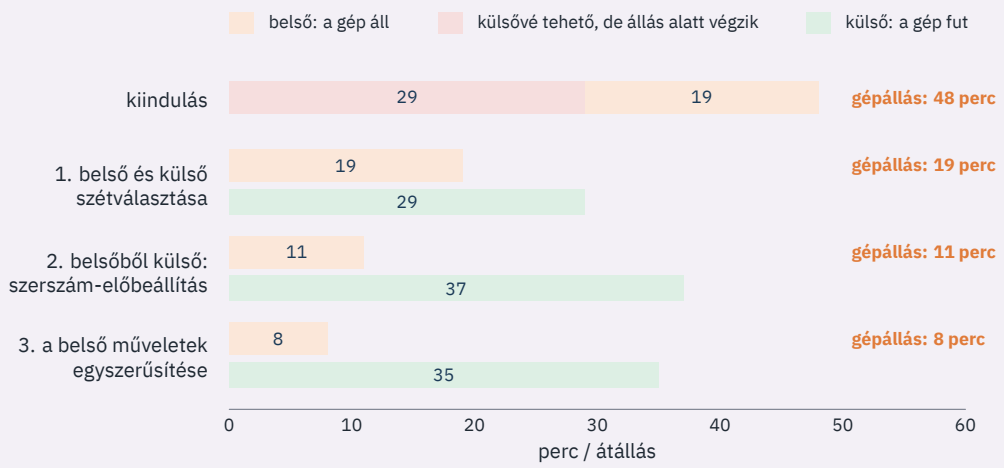
- **belső** műveletek: csak álló gépen végezhetők (például a szerszám behelyezése, a munkadarab befogása);
- **külső** műveletek: a gép futása közben is elvégezhetők (például a következő szerszám és készülék előkészítése, a program betöltése, a mérőeszközök odakészítése).

A módszer három szakaszból áll (Shingo 1985):

1. **a belső és a külső műveletek szétválasztása:** a külső műveleteket a gép futása alatt végezzük el, ami önmagában nagy megtakarítást ad, mert a gyakorlatban sok külső műveletet álló gépen végeznek;
2. **belső műveletek átalakítása külsővé:** például a szerszámok gépen kívüli előbeállításával vagy szabványos befogófelületekkel;
3. **minden művelet egyszerűsítése:** gyorsrögzítők, a csavarozás és az utánállítás kiküszöbölése, párhuzamos munkavégzés.

ForgácsTech: a megmunkáló központ átállása

A csapat videóra vette a megmunkáló központ átállásait, és minden lépést belsőnek vagy külsőnek minősített. A 48 perces átállásból 29 perc olyan tevékenység volt, amely a gép futása alatt is elvégezhető: a szerszámok és készülékek kikeresése, a program betöltése és ellenőrzése, a mérőeszközök odakészítése (19.6. ábra). Az első szakaszban ezeket a gép futása alatt végezték el, így a gépállás 19 percre csökkent. A második szakaszban a szerszámokat egy gépen kívüli előbeállító készüléken állították be, ami további 8 percet tett külsővé. A harmadik szakaszban nullponti gyorsbefogó rendszert vezettek be a készülékekhez, és a gépállás 8 percre csökkent.



19.6. ábra. A beállítási idő csökkentése a SMED három szakaszával a ForgácsTech megmunkáló központján.

A megtakarítás kétféleképpen használható fel. Ha az átállások száma marad (műszakonként kettő), a rendelkezésre állás $(450-16-20)/450 = 0{,}92$-re, az OEE változatlan teljesítmény- és minőségi tényező mellett $0{,}92 \cdot 0{,}838 \cdot 0{,}943 = 0{,}727$-re nő. Ha a csapat a sorozatokat felezi, és műszakonként négyszer áll át, a rendelkezésre állás $0{,}884$, az OEE $0{,}699$, ami még mindig jóval meghaladja a kiindulási 0,587-et, a gép előtt várakozó munka viszont a felére csökken, és a Little-törvény szerint vele együtt a gép előtti várakozási idő is. A ForgácsTech a második utat választotta, mert céljai között a határidő-tartás és az átfutás csökkentése szerepel, és nem a gépkihasználás maximalizálása.

Folyamatelemzés szolgáltatásokban

A fejezet fogalmai nem csak a gyártásban érvényesek. A Little-törvény egy kórházi osztály betegforgalmára, egy ügyfélszolgálat nyitott ügyeire vagy egy szoftverfejlesztő csapat folyamatban lévő feladataira ugyanúgy alkalmazható (Little 2011). Az OEE mintájára szolgáltatásokban is felbontható az idő: egy orvos fizetett munkaidejéből levonható a szabadság, a le nem foglalt idő, a lemondott vizsgálatok és az orvosi képesítést nem igénylő tevékenységek ideje, és a maradék mutatja a valóban értékteremtő időt. A szolgáltatási folyamatok sajátossága, hogy a vevő maga is a folyamat része, és a várakozást ő érzékeli. Ezért ott az átfutási idő és a várakozás gyakran fontosabb a kihasználtságnál.

A gépkihasználtság hajszolása

Gyakori hiba, hogy a vezetés minden gép kihasználtságát külön-külön maximalizálni akarja. Ha egy nem szűk keresztmetszet gép a szükségesnél többet gyárt, az nem növeli a kibocsátást, csak a készletet és az átfutási időt. Hasonló hiba az OEE-t gépek, műszakok vagy üzemek rangsorolására használni: a mutató értelme a saját veszteségek feltárása és a javulás követése. A kihasználtság és az átfutás közötti átváltást (a teljes kihasználtság közelében a várakozás gyorsan nő) tudatosan kell kezelni.

Összefoglalás

- A folyamatot az áram, a folyamatidő és a készlet írja le; a Little-törvény ($I = R \cdot T$) stabil rendszerben a hosszú távú átlagokra eloszlástól függetlenül érvényes.
- Az erőforrás kapacitása m/p; a folyamat kapacitását a szűk keresztmetszet határozza meg, az áram a kereslet és a kapacitás közül a kisebb, a kihasználtság az áram és a kapacitás hányadosa.
- Az átlagos munkaerő-kihasználtság a munkatartalom és a ciklusonként fizetett mun-

kaidő hányadosa; a kihasználatlan időt is ki kell fizetni.
- Az ütemidő a rendelkezésre álló idő és a kereslet hányadosa; a sorkiegyenlítés az állomásidőket az ütemidő alá és egymáshoz közel hozza. A Yamazumi-diagram a kézi és a gépi időt is megmutatja.
- Az OEE a rendelkezésre állás, a teljesítmény és a minőség szorzata, a hat nagy veszteségre épül; egy gépre vonatkozik, és értéke az adatgyűjtés minőségétől függ. Az OLE ugyanezt a munkára alkalmazza.
- Az értékáram-térkép a várakozásokat teszi láthatóvá; a folyamathatékonyság gyakran néhány százalék.
- A SMED a belső és a külső műveletek szétválasztásával, a belsők külsővé alakításával és egyszerűsítésével csökkenti a beállítási időt; a megtakarítás kapacitásra vagy kisebb sorozatokra és rövidebb átfutásra fordítható.

Önellenőrzés

1. Egy ügyfélszolgálaton átlagosan 45 nyitott ügy van, és naponta 15 ügyet zárnak le. Mennyi egy ügy átlagos átfutási ideje? Milyen feltételek mellett érvényes a számítás?
2. Egy háromlépéses folyamat lépésein 1, 2, illetve 1 dolgozó dolgozik, a munkaigények 4, 10, illetve 3 perc/egység. Melyik lépés a szűk keresztmetszet, és mennyi a folyamatkapacitás egységenként óránként?
3. Az előző folyamatban a kereslet 10 egység/óra. Mennyi az egyes lépések kihasználtsága és az átlagos munkaerő-kihasználtság?
4. Egy szerelősor két műszakban, műszakonként 7 óra nettó idővel dolgozik, a napi igény 420 termék. Mennyi az ütemidő? A munkatartalom 230 s; legalább hány állomás kell?
5. Mit mutat a Yamazumi-diagram, amit az állomásidők táblázata nem?
6. Egy gép tervezett üzemideje 480 perc, ebből 60 perc állás. Az ideális ciklusidő 1 perc, a gép 380 darabot gyártott, ebből 361 jó. Számítsd ki az A, P, Q tényezőt és az OEE-t!
7. Miért nem célszerű két különböző üzem OEE-értékeit közvetlenül összehasonlítani?
8. Sorold fel Ohno hét veszteségét, és adj mindegyikre példát egy javítóműhelyből!
9. Mi a különbség a belső és a külső beállítási művelet között? Miért ad már az első SMED-szakasz is nagy megtakarítást?

Megoldás.

1. $T = I/R = 45/15 = 3$ nap. A rendszernek stabilnak kell lennie (hosszú távon annyi ügy zárul le, amennyi beérkezik), és a három mennyiséget ugyanarra a rendszerhatárra és időszakra kell mérni.
2. Kapacitások: $60/4 = 15$, $2 \cdot 60/10 = 12$, $60/3 = 20$ egység/óra. A szűk keresztmetszet a második lépés, a folyamatkapacitás 12 egység/óra.
3. Az áram 10 egység/óra (keresletkorlátos). Kihasználtságok: $10/15 = 67\%$, $10/12 = 83\%$, $10/20 = 50\%$. A ciklusidő 6 perc, a munkatartalom 17 perc, a létszám 4, így az átlagos munkaerő-kihasználtság $17/(6 \cdot 4) = 70{,}8\%$.
4. $TT = 2 \cdot 7 \cdot 3600/420 = 120$ s; $N_{\min} = \lceil 230/120 \rceil = 2$ állomás.
5. Az állomásidők összetételét: a kézi munka, a gépi idő, a gyaloglás és a várakozás arányát, valamint az ütemidőhöz viszonyított tartalékot. Ebből látszik, hol lehet a kézi munkát átcsoportosítani úgy, hogy a gépi idő változatlan maradjon.
6. $A = 420/480 = 0{,}875$; $P = 380 \cdot 1/420 = 0{,}905$; $Q = 361/380 = 0{,}95$; OEE $= 0{,}875 \cdot 0{,}905 \cdot 0{,}95 = 0{,}752$ (ellenőrzés: $361/480 = 0{,}752$).
7. Mert eltérhet a veszteségek osztályozása és az adatgyűjtés pontossága (például a kisebb leállások rögzítése), az ideális ciklusidő meghatározása, a termékszerkezet és a sorozatnagyság. Az üzemek céljai is különböznek; a mutató a saját fejlődés követésére alkalmas.
8. Túltermelés: előre felújított, de hetekig álló kerékpárok; várakozás: alkatrészre váró szerelő; szállítás: a kerékpár többszöri mozgatása a műhelyrészek között; felesleges feldolgozás: a teljes átvizsgálás olyan hibánál is, amelyet a diagnosztika már lokalizált; készlet: nagy alkatrészkészlet; felesleges mozgás: szerszám keresése; hibás termék: ismételt javítás, mert a fék elsőre nem lett jó.
9. A belső művelet csak álló gépen, a külső a gép futása közben is elvégezhető. A gyakorlatban sok külső műveletet (keresés, előkészítés, szállítás) álló gépen végeznek, ezért már a szétválasztás és a gép futása alatti elvégzés is jelentősen csökkenti a gépállást, beruházás nélkül.

Projektlépés: folyamatelemzés

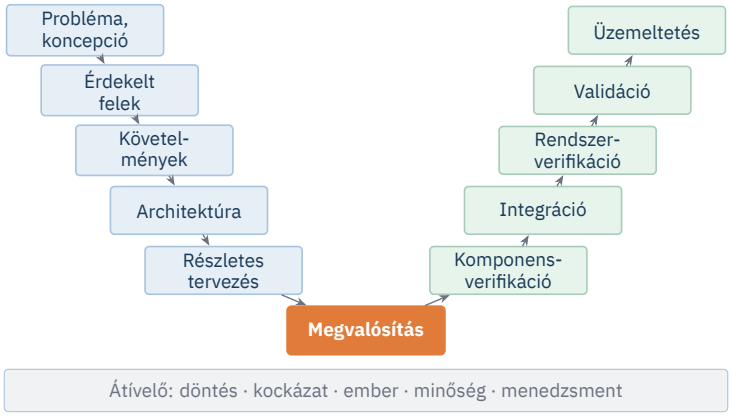
Ha saját fejlesztési feladatod egy gyártó- vagy szolgáltatófolyamatot is érint (vagy a rendszer gyártását, szerelését, javítását kell megtervezni):

1. rajzold fel a folyamatábrát, és jelöld az áramló egységet, a rendszerhatárt és az erőforrásokat;
2. becsüld vagy mérd az áramot, a készletet és a folyamatidőt, és ellenőrizd őket a Little-törvénnyel;
3. számítsd ki az erőforrások kapacitását és kihasználtságát, és azonosítsd a szűk keresztmetszetet;
4. ha a folyamat ütemezett, számítsd ki az ütemidőt, és készíts Yamazumi-diagramot;
5. számítsd ki egy kritikus berendezés OEE-jét, vagy készíts értékáram-térképet a teljes folyamatról;
6. használd a Gyakorlati útmutató folyamatábra-, kapacitás-, sorkiegyenlítési és OEE-eszközlapját.

Továbbolvasás

- Cachon és Terwiesch (2013), 2–4. fejezet: a folyamatelemzés alapjai, a kapacitás, a kihasználtság és a munkaerő-termelékenység; a fejezet számpéldái is erre a könyvre épülnek.
- Hopp és Spearman (2008): a gyártórendszerek „fizikája”: a Little-törvény, a változékonyság és a kihasználtság hatása az átfutásra.
- Little (2011): a Little-törvény elmélete és gyakorlata ötven év távlatából.
- Nakajima (1988) és Muchiri és Pintelon (2008): az OEE eredete, a hat nagy veszteség, valamint a mutató változatainak és korlátainak áttekintése.
- Rother és Shook (1999): az értékáram-térképezés gyakorlati kézikönyve.
- Shingo (1985): a SMED-módszer eredeti leírása.
- Scholl és Becker (2006): a szerelősor-kiegyenlítés pontos és heurisztikus megoldási módszereinek áttekintése.
- Lakatos (é. n.b): a termelési rendszerek magyar nyelvű egyetemi jegyzete.
- Goldberg és mtsai. (1994), 7.15–7.16. szakasz: a folyamatábra és a munkafolyamat-elemzés gyakorlati leírása.

20. Folyamatmodellezés BPMN-nel



Tanulási eredmények

A fejezet feldolgozása után képes leszel:

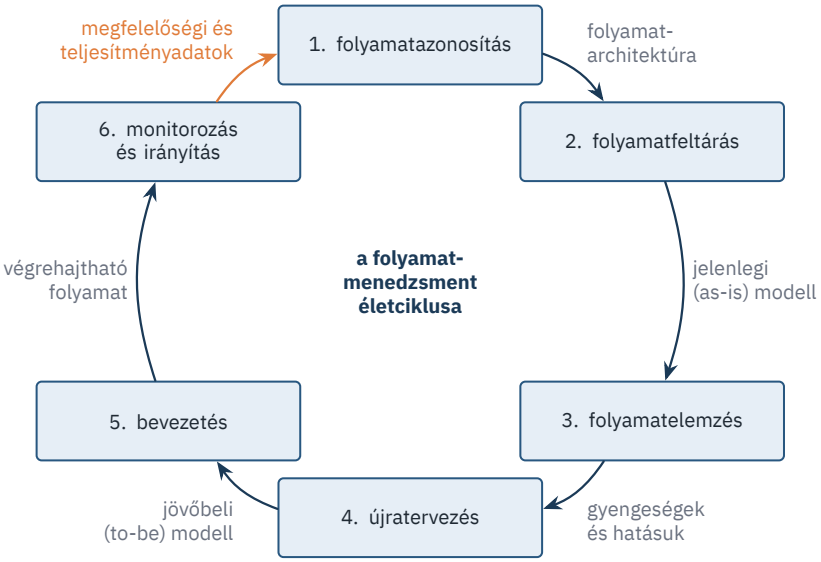
- elhelyezni a folyamatmodellezést a folyamatmenedzsment életciklusában;
- kiválasztani a folyamatfeltárás megfelelő módszereit, és összevetni erősségeiket és gyengeségeiket;
- BPMN 2.0 jelöléssel modellezni egy több résztvevős folyamatot eseményekkel, tevékenységekkel, átjárókkal, sávokkal és üzenetfolyamokkal;
- alkalmazni a modellezési irányelveket, hogy a modell érthető és hibamentes legyen;
- a modellből a folyamat átfutási idejét kiszámítani, és eldönteni, mikor van szükség szimulációra;
- megkülönböztetni a folyamatbányászat három feladatát, és a modellt eseménynaplóval összevetni;
- nagy nyelvi modellt felelősen használni a modell előállításában és ellenőrzésében.

Az előző fejezet a folyamatot számokkal írta le: árammal, kapacitással, hatékonysággal. Ehhez azonban tudni kell, milyen lépésekből áll a folyamat, ki végzi őket, hol ágazik el, és milyen információ áramlik a résztvevők között. Ezt a **folyamatmodell** rögzíti. A rendszermérnökségben a folyamatmodellnek kettős szerepe van: a fejlesztés tárgyát (egy gyártó- vagy szolgáltatófolyamatot) írja le, és egyben a fejlesztendő informatikai rendszer követelményeinek forrása, mert megmutatja, mely lépéseket kell a rendszernek támogatnia vagy automatizálnia. A fejezet a folyamatmenedzsment tankönyvi összefoglalására (Dumas és mtsai. 2018), a BPMN 2.0 szabványra (Object Management Group 2011) és a folyamatbányászat alapművére (Aalst 2016) épül.

20.1. A folyamatmenedzsment életciklusa

Dumas és szerzőtársai a folyamatmenedzsmentet hat szakaszból álló, ismétlődő ciklusként írják le (20.1. ábra) (Dumas és mtsai. 2018, 1. fejezet). A **folyamatazonosítás** a

szervezet folyamatait és kapcsolataikat (a folyamatarchitektúrát) tárja fel, és kiválasztja a fejlesztendő folyamatot. A **folyamatfeltárás** a jelenlegi (as-is) modellt készíti el. A **folyamatelemzés** a gyengeségeket és azok hatását tárja fel, minőségi és mennyiségi módszerekkel. Az **újratervezés** a jövőbeli (to-be) modellt állítja elő. A **bevezetés** a szervezeti változtatásokat és a folyamat informatikai támogatását valósítja meg. A **monitorozás és irányítás** a végrehajtás adataiból méri a folyamatot, és visszacsatol a következő ciklusba.



20.1. ábra. A folyamatmenedzsment életciklusa és a szakaszok kimenetei (Dumas és mtsai. 2018, 1. fejezet nyomán).

A ciklus a V-modellel jól összeilleszthető: a feltárás és az elemzés a probléma meghatározásához és a működési koncepcióhoz (4. fejezet), az újratervezés az architektúrához és a koncepcióválasztáshoz, a bevezetés a megvalósításhoz és az integrációhoz, a monitorozás az üzemeltetéshez tartozik. A ForgácsTech MES-projektje (10. fejezet) például a rendelésteljesítési folyamat feltárásával kezdődött, és a bevezetett rendszer adataiból a folyamat újabb ciklusban fejleszthető tovább.

A folyamatok három csoportját érdemes megkülönböztetni: az **üzleti** (alap)folyamatok értéket teremtenek a vevő számára (például a rendelésteljesítés), a **fejlesztési** folyamatok új termékeket és képességeket hoznak létre, az **üzemeltetési** (támogató) folyamatok a működés feltételeit biztosítják (karbantartás, beszerzés, informatika).

20.2. Folyamatfeltárás

A folyamatfeltárás nehézsége, hogy a folyamatot ismerő szakemberek általában nem modelleznek, a modellező pedig nem ismeri a folyamatot. A tudás szétszórt: mindenki a saját lépéseit ismeri, és ritkán látja a teljes folyamatot (Dumas és mtsai. 2018, 5. fejezet). A feltárás módszereit a 20.1. táblázat foglalja össze.

20.1. táblázat. A folyamatfeltárás módszerei (Dumas és mtsai. 2018, 5. fejezet nyomán).

Módszer	Erősség	Gyengeség
dokumentumelemzés	strukturált információ, nem igényli az érintettek idejét	elavult anyag, nem megfelelő részletezettség
megfigyelés	gazdag, a valóságot mutató kép a folyamatról	zavaró lehet, a megfigyeltek másképp viselkedhetnek, kevés esetet lát
automatikus feltárás (folyamatbányászat)	sok eset, objektív adat	az adatok minősége gyenge lehet
interjú	részletes, mélyre hatoló kérdezés	sok időt kér az érintettektől, több körös egyeztetés kell
workshop	az eltérő nézetek közvetlenül tisztázhatók	egyszerre több érintett jelenlétét igényli

A módszereket a gyakorlatban kombináljuk: a dokumentumokból és az informatikai rendszerek naplóiból vázlatot készítünk, interjúkkal és megfigyeléssel pontosítjuk, és workshopon hagyjuk jóvá. Az interjúk technikáját a 5. fejezet, a megfigyelés és a feladatelemzés módszereit a 11. fejezet tárgyalja.

20.3. A BPMN 2.0 jelölés

A **BPMN** (Business Process Model and Notation) az Object Management Group (OMG) szabványa, amelynek 2.0 változata 2011-ben jelent meg (Object Management Group 2011), és ISO/IEC szabványként is elfogadták (ISO/IEC 2013). Célja, hogy az üzleti és a műszaki szakemberek egyaránt megértsék, és hogy a modell végrehajtható folyamatleírássá is alakítható legyen. A szabvány az elemeket öt alapcsoportba sorolja: folyamatobjektumok (események, tevékenységek, átjárók), adatok, összekötők, résztvevők (medencék és sávok) és kiegészítők (Object Management Group 2011, 7.2. szakasz). A leggyakrabban használt elemeket a 20.2. ábra mutatja.

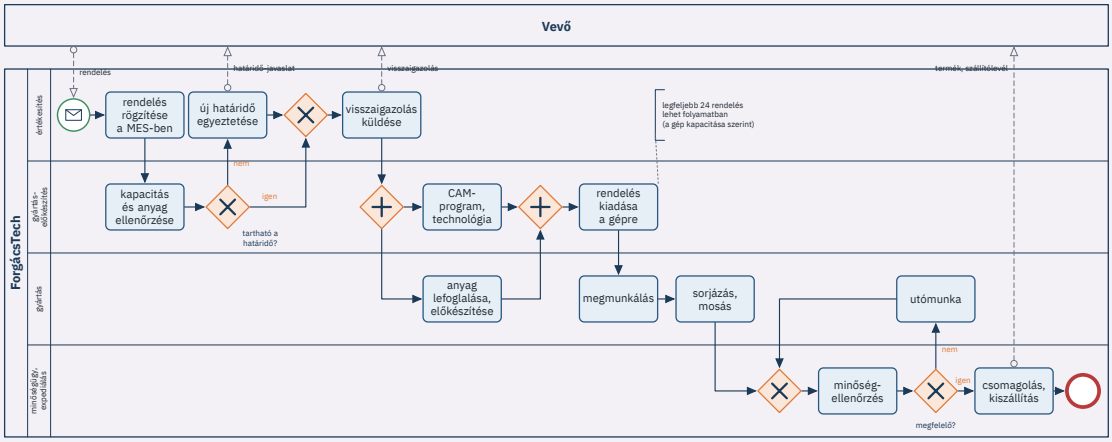


20.2. ábra. A BPMN 2.0 leggyakrabban használt elemei (Object Management Group 2011 nyomán).

- **Események:** valami, ami megtörténik. Az indító esemény elindítja, a záró esemény befejezi a folyamatot, a köztes esemény a folyamat közben következik be (például egy üzenet megérkezése vagy egy határidő lejárta). Az esemény típusát belső jel mutatja: üzenet, időzítő, hiba, jelzés.
- **Tevékenységek:** a munka egységei. A feladat tovább nem bontott tevékenység; az alfolyamat önálló modellben részletezett tevékenységcsoport. A jelölők ismétlődést, többszörös végrehajtást vagy kompenzációt jeleznek; a feladat típusa is jelölhető (felhasználói, automatikus, kézi, üzenetküldő).
- **Átjárók:** a folyamat elágazása és összefutása. A **kizáró** (XOR) átjáró a feltételek alapján pontosan egy ágat választ; a **párhuzamos** (ÉS) átjáró minden ágat elindít, illetve mindegyik befejeződését megvárja; a **befogadó** (VAGY) átjáró egy vagy több ágat indít; az **eseményalapú** átjáró annak alapján választ, melyik esemény következik be előbb.
- **Összekötők:** a **szekvenciafolyam** a végrehajtás sorrendjét mutatja egy résztvevőn belül, az **üzenetfolyam** két résztvevő közötti üzenetváltást, az **asszociáció** a kiegészítőket köti az elemekhez.
- **Résztvevők:** a **medence** egy önálló résztvevőt (szervezetet, vevőt, informatikai rendszert) jelöl, a **sáv** a medencén belüli szerepet vagy szervezeti egységet. Két medence között csak üzenetfolyam lehet.
- **Kiegészítők:** az adatobjektum a folyamatban keletkező vagy felhasznált dokumentumot, adatot jelöl; a csoport és a szöveges megjegyzés a modell olvasását segíti.

ForgácsTech: a rendelésteljesítés jövőbeli folyamata

A ForgácsTech csapata a MES első ütemének bevezetésekor a rendelésteljesítés jövőbeli folyamatát BPMN-nel modellezte (20.3. ábra). A vevő külön medence: belső működése nem érdekes, csak az üzenetváltás vele. A ForgácsTech medencéje négy sávra oszlik. A rendelés beérkezésekor az értékesítés rögzíti a rendelést a MES-ben, a gyártáselőkészítés a MES ütemezőjével ellenőrzi a kapacitást és az anyagot. Ha a kért határidő nem tartható, az értékesítés új határidőt egyeztet a vevővel; ezután küldi el a visszaigazolást. A CAM-program elkészítése és az anyag lefoglalása párhuzamosan fut. A rendelést csak akkor adják ki a gépre, ha a folyamatban lévő rendelések száma a korlát alatt van (19. fejezet). A minőségellenőrzés után a nem megfelelő darab utómunkára kerül, majd újra ellenőrzik.



20.3. ábra. A ForgácsTech rendelésteljesítési folyamatának jövőbeli BPMN-modellje.

A jelenlegi folyamat modellje két fontos különbséget mutatott. A kapacitást a termelésvezető becsülte heti egyszer, papíron, ezért a visszaigazolás átlagosan több mint egy napig tartott. A sürgős rendeléseket pedig szóban sorolták előre, ami a modellben sehol sem jelent meg, a valóságban viszont gyakori volt (20.6. szakasz).

20.4. Modellezési irányelvek

Ugyanaz a folyamat sokféleképpen modellezhető, és a modellek érthetősége, hibákra való hajlama nagyon különböző. Mendling, Reijers és van der Aalst empirikus vizsgálatok alapján hét irányelvet fogalmazott meg (7PMG) (Mendling, Reijers, és Aalst 2010):

1. a modell a lehető legkevesebb elemből álljon;
2. egy elembe a lehető legkevesebb ág fusson be és fusson ki;
3. egy indító és egy záró esemény legyen;
4. a modell a lehető legstrukturáltabb legyen: minden elágazó átjárónak legyen azonos típusú összefutó párja;
5. kerülni kell a befogadó (VAGY) átjárókat;
6. a tevékenységek neve „ige + tárgy” szerkezetű legyen (például „rendelés rögzítése”);
7. az 50 elemnél nagyobb modellt alfolyamatokra kell bontani.

Ezekhez a gyakorlatban néhány további szabály társul. A modellnek egyértelmű célja és nézőpontja legyen (kinek készül, mit kell megmutatnia). A részletezettség legyen egyenletes: egy modellben ne keveredjenek a stratégiai lépések a billentyűleütésekkel. A medencék közötti kapcsolat mindig üzenetfolyam, a sávok közötti átadás szekvenciafolyam. A ForgácsTech modellje (20.3. ábra) egy indító és egy záró eseményt használ, a kizáró átjárók összefutását is kirajzolja, és a párhuzamos ágakat azonos típusú átjáró zárja, így strukturált; 12 tevékenységével jóval az 50 elemes határ alatt marad.

20.5. Mértékek és szimuláció

A folyamat teljesítményét négy szempontból mérjük: idő, költség, minőség és rugalmasság (Dumas és mtsai. 2018, 2. fejezet). A négy szempont gyakran ütközik: a gyorsabb folyamat drágább lehet, a rugalmasabb lassabb. A mérőszámokat két szinten érdemes megkülönböztetni. A **kulcsmutatók** (KPI) a szervezet egészének céljaihoz kapcsolódnak, és hosszabb távon állandóak (például a határidőre teljesített rendelések aránya). Az **akciócentrikus mutatók** egy-egy folyamatlépést vagy részfolyamatot követnek, és közvetlen beavatkozásra alkalmasak (például a visszaigazolás átfutási ideje).

A modell elemeihez paraméterek rendelhetők: a tevékenységekhez ciklusidő, erőforrás, költség és rendelkezésre állás, az átjárók ágaihoz valószínűség. Ezekből a **folyamatáram-elemzés** kiszámítja a folyamat átlagos ciklusidejét: soros lépéseknél az idők összeadódnak, a kizáró átjáró ágainál az ágak idejének valószínűséggel súlyozott átlagát, a párhuzamos ágaknál a leghosszabb ág idejét vesszük; egy r valószínűséggel ismétlődő T idejű blokk várható ideje $T/(1-r)$ (Dumas és mtsai. 2018, 7. fejezet):

$$CT_{\text{soros}} = \sum_i T_i, \qquad CT_{\text{XOR}} = \sum_i p_i T_i, \qquad CT_{\text{ÉS}} = \max_i T_i, \qquad CT_{\text{ismétlés}} = \frac{T}{1-r}. \tag{20.1}$$

ForgácsTech: mennyi ideig tart a visszaigazolás?

A jelenlegi folyamatban a rögzítés fél óra, a kapacitás becslése (a termelésvezetőre várással együtt) átlagosan 16 óra, a rendelések 40%-ánál új határidőt kell egyeztetni, ami a vevő válaszával együtt 24 óra, a visszaigazolás elküldése fél óra. A visszaigazolás átlagos ideje $0{,}5 + 16 + 0{,}4 \cdot 24 + 0{,}5 = 26{,}6$ óra. A MES ütemezőjével a kapacitás-ellenőrzés 2

óra, és mivel a vállalt határidők a kapacitáson alapulnak, csak a rendelések 20%-ánál kell egyeztetni: $0{,}5 + 2 + 0{,}2 \cdot 24 + 0{,}5 = 7{,}8$ óra. A párhuzamos ágon a CAM-program 6, az anyag előkészítése 4 óra, a blokk ideje a hosszabb ág, 6 óra; ha sorban végeznék, 10 óra lenne. A minőségellenőrzés 0,6 óra, a darabok 5%-a 1 órás utómunkára és újraellenőrzésre szorul, így a blokk várható ideje $(0{,}6 + 0{,}05 \cdot 1)/(1 - 0{,}05) = 0{,}68$ óra.

A folyamatáram-elemzés átlagokkal számol, és nem veszi figyelembe, hogy a tevékenységek erőforrásokon osztoznak, és az ingadozó érkezések sorokat okoznak. Ezekre a kérdésekre a sorbanállási modellek (21. fejezet) és a **folyamatszimuláció** ad választ. A szimuláció a modellt eseményenként végrehajtja, véletlen érkezési és feldolgozási időkkel, és a „mi lenne, ha” kérdésekre ad választ: mi történik, ha nő a rendelések száma, ha egy erőforrás kiesik, vagy ha egy lépést automatizálunk (Dumas és mtsai. 2018, 7. fejezet). A szimuláció eredménye annyira megbízható, amennyire a bemenő adatok és a modell validálása (3. fejezet). A folyamatosan frissített adatokkal futó szimulációs modell a digitális iker egyik formája.

20.6. Folyamatbányászat

Az informatikai rendszerek (ERP, MES, ügyviteli rendszerek) minden lépést naplóznak. Az **eseménynapló** minden sora egy esemény: melyik esethez tartozik (esetazonosító), melyik tevékenység történt, mikor (időbélyeg), és gyakran az is, ki végezte. A **folyamatbányászat** ezekből a naplókból nyer ismeretet a valóságban futó folyamatról. Van der Aalst három feladattípust különböztet meg (Aalst 2016):

- **feltárás:** a naplóból modell készül, előzetes modell nélkül;
- **megfelelőség-ellenőrzés:** a meglévő modellt vetjük össze a naplóval, és feltárjuk, hol tér el a valóság a modelltől;
- **bővítés:** a meglévő modellt a napló adataival gazdagítjuk, például a várakozási időkkel vagy a szűk keresztmetszetekkel.

ForgácsTech: a napló, amely felfedte a rejtett lépést

A MES első három hónapjának naplójában a rendelések 17%-ánál szerepelt egy „rendelés átsorolása” esemény, amely a jövőbeli modellben nem szerepelt: a sürgős rendeléseket a termelésvezető továbbra is előre sorolta, csak most a rendszerben. Az átsorolt rendelések határidőre teljesültek, az általuk megelőzött rendelések viszont átlagosan csaknem két munkanappal később kerültek gépre, és több mint harmaduk késve teljesült. A csapat a lépést beépítette a modellbe, és szabályhoz kötötte: átsorolni csak akkor lehet, ha az ütemező szerint a megelőzött rendelések határideje így is tartható. A folyamatbányászat eszközeit és lépéseit a Gyakorlati útmutató folyamatbányászati eszközlapja mutatja be.

A folyamatbányászat eredménye a napló minőségén múlik: hiányzó vagy pontatlan időbélyegek, eltérő tevékenységnevek és rosszul meghatározott esetfogalom torz modellt adnak (Aalst 2016). Az eseménynapló a folyamat minden szereplőjéről személyes adatot is tartalmazhat, ezért az elemzés célját és az adatkezelés módját előre rögzíteni kell.

20.7. Nagy nyelvi modellek a folyamatmodellezésben

A nagy nyelvi modellek (LLM) szöveges folyamatleírásból folyamatmodellt tudnak előállítani. Grohs és szerzőtársai három feladaton vizsgálták a GPT-4-et: imperatív folyamatmo-

dell kinyerését szövegből, deklaratív modell kinyerését szövegből és annak megítélését, mely feladatok alkalmasak robotizált folyamatautomatizálásra. Mindhárom feladatban a célra fejlesztett megoldásokhoz hasonló vagy jobb eredményt kaptak, különösebb előkészítés nélkül (Grohs és mtsai. 2024). Kourani és szerzőtársai olyan keretrendszert javasoltak, amelyben a nyelvi modell nem közvetlenül rajzol, hanem egy köztes, ellenőrizhető folyamatleíró kódot állít elő; a hibákat a rendszer automatikusan visszajelzi, és a kész modell BPMN-be vagy Petri-hálóba exportálható (Kourani és mtsai. 2024). A szöveges modellleírás és a modellek iteratív, párbeszédes finomítása így a modellezés gyakorlati eszközévé vált.

A nyelvi modell azonban a valóságot nem ismeri: csak a kapott leírást és az általános tudását. Ebből három szabály következik:

1. **Kétlépcsős munkamenet.** A modell először csak értelmezze a leírást: sorolja fel a résztvevőket, az eseményeket, a tevékenységeket és a döntési pontokat, és tegyen fel tisztázó kérdéseket. A modellt csak a jóváhagyott értelmezés alapján generálja.
2. **Jelenlegi állapot a leírás szerint.** A feltárás során kifejezetten kérni kell, hogy a modell ne optimalizáljon és ne egészítsen ki: a „szokásos” lépések hozzáadása éppen a valóságos eltéréseket fedi el.
3. **Ellenőrzés.** A generált modellt modellezőeszközben meg kell nyitni (érvényes-e, olvasható-e), és a folyamat résztvevőivel jóvá kell hagyatni. A megfelelőségellenőrzést is támogathatja a nyelvi modell (az eseménynaplót és a modellt együtt megadva), de a következtetést a naplóból számított gyakoriságokkal kell igazolni.

A kidolgozott mintapromptokat a Gyakorlati útmutató BPMN-eszközlapja tartalmazza.

Végrehajtható modellek és munkafolyamat-motorok

A BPMN 2.0 nemcsak jelölés, hanem végrehajtási szemantikát is meghatároz: a modell munkafolyamat-motorban futtatható, amely a felhasználói feladatokat kiosztja, a szolgáltatásfeladatokat meghívja, és minden lépést naplóz (Object Management Group 2011). Ilyenkor a modell a működő rendszer része, és a naplózott adatok közvetlenül a folyamatbányászat bemenetei. A ForgácsTech MES-e a rendeléstelje sítés lépéseit így követi. A végrehajtható modellhez azonban jóval részletesebb leírás kell (adatok, feltételek, kivételek kezelése), mint a megértést szolgáló modellhez; a kettőt nem érdemes összekeverni.

A modell, amely a szabályzatot rajzolja le

Gyakori hiba, hogy a jelenlegi folyamat modellje a szabályzatból vagy a vezető elmondásából készül, és azt mutatja, hogyan kellene a folyamatnak működnie. Az ilyen modellből levont következtetések tévesek, mert a kerülőutak, a kivételek és a rejtett lépések hiányoznak belőle. A jelenlegi állapotot a folyamatot végzőkkel, megfigyeléssel és lehetőség szerint eseménynaplóval kell feltárni.

Összefoglalás

- A folyamatmenedzsment életciklusa: azonosítás, feltárás, elemzés, újratervezés, bevezetés, monitorozás; a ciklus a V-modell lépéseivel összeilleszthető.
- A feltárás módszerei (dokumentumelemzés, megfigyelés, automatikus feltárás, interjú, workshop) egymást kiegészítik.
- A BPMN 2.0 elemei: események, tevékenységek, átjárók, összekötők, résztvevők (medence, sáv) és kiegészítők; medencék között csak üzenetfolyam lehet.
- A 7PMG irányelvei (kevés elem, egy indító és egy záró esemény, strukturált modell, „ige + tárgy” nevek, alfolyamatok) a modell érthetőségét és helyességét javítják.
- A folyamatáram-elemzés a modellből átlagos ciklusidőt számít; az erőforrás-

versengéshez és az ingadozáshoz szimuláció kell.
- A folyamatbányászat feladatai: feltárás, megfelelőség-ellenőrzés, bővítés; az eredmény a napló minőségén múlik.
- A nagy nyelvi modell gyorsítja a modellezést, de a kimenetet kétlépcsős munkamenettel, modellezőeszközben és a résztvevőkkel ellenőrizni kell.

Önellenőrzés

1. Sorold fel a folyamatmenedzsment életciklusának szakaszait és kimeneteit!
2. Miért kevés egyetlen feltárási módszer? Melyik módszert választanád egy háromműszakos gyártóüzem anyagellátási folyamatának feltárására, és miért?
3. Mi a különbség a szekvenciafolyam és az üzenetfolyam között? Miért nem köthet össze két medencét szekvenciafolyam?
4. Mi a különbség a kizáró, a párhuzamos és a befogadó átjáró között? Adj példát mindháromra!
5. Egy folyamat három soros lépésből áll (1, 2 és 1 óra). A második és a harmadik lépés közé 30% valószínűséggel egy 4 órás ellenőrző lépés ékelődik. Mennyi az átlagos ciklusidő?
6. Egy 2 órás lépést 20% valószínűséggel meg kell ismételni. Mennyi a várható ideje?
7. Mi a folyamatbányászat három feladata? Melyikkel derítenéd ki, hogy a dolgozók megkerülik-e a jóváhagyási lépést?
8. Miért kell egy nyelvi modellel készített folyamatmodellt a folyamat résztvevőivel is jóváhagyatni?

Megoldás.

1. Azonosítás (folyamatarchitektúra), feltárás (jelenlegi modell), elemzés (gyengeségek és hatásuk), újratervezés (jövőbeli modell), bevezetés (végrehajtható folyamat), monitorozás és irányítás (megfelelőségi és teljesítményadatok).
2. Mert mindegyiknek más a gyengesége: a dokumentum elavult lehet, a megfigyelés kevés esetet lát, az interjú egy ember nézőpontját adja. Háromműszakos üzemben a megfigyelés több műszakban, a dolgozókkal készült interjúk és a raktári rendszer naplója együtt adja a teljes képet; a vázlatot workshopon érdemes jóváhagyatni.
3. A szekvenciafolyam a végrehajtás sorrendjét mutatja egy résztvevőn belül, az üzenetfolyam két önálló résztvevő közötti kommunikációt. Két medence egymástól függetlenül működik, a sorrendjüket nem egymás vezérlése, hanem üzenetek váltása kapcsolja össze.
4. Kizáró: pontosan egy ág (jóváhagyják vagy elutasítják a rendelést). Párhuzamos: minden ág (a CAM-program és az anyag előkészítése egyszerre). Befogadó: egy vagy több ág (a rendeléshez szükség lehet hőkezelésre, felületkezelésre vagy mindkettőre).
5. $1 + 2 + 0{,}3 \cdot 4 + 1 = 5{,}2$ óra: a kizáró ág idejét a valószínűségével súlyozzuk.
6. $2/(1 - 0{,}2) = 2{,}5$ óra.
7. Feltárás, megfelelőség-ellenőrzés, bővítés. A megkerülést a megfelelőség-ellenőrzés mutatja ki: azokat az eseteket keresi, amelyekben a jóváhagyás eseménye hiányzik, vagy a sorrend eltér a modelltől.
8. Mert a nyelvi modell csak a kapott leírásból és általános tudásából dolgozik: kitölthet hiányokat „szokásos” lépésekkel, és nem tudhatja, mi történik valójában. A résztvevők jóváhagyása az érvényesítés része.

Projektlépés: folyamatmodell

Saját fejlesztési feladatodhoz:

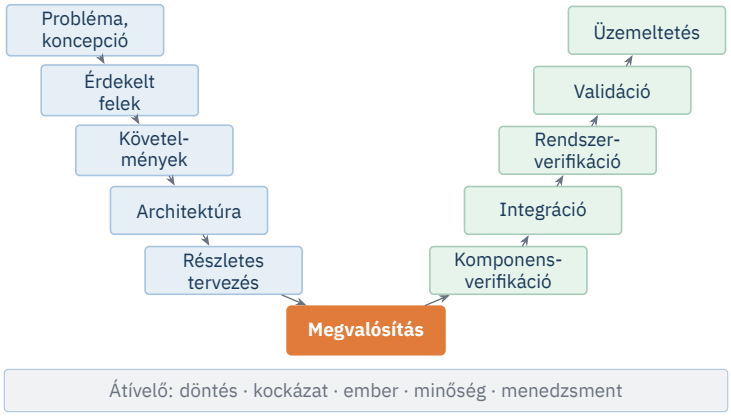
1. válaszd ki a megvalósítás szempontjából legfontosabb folyamatot, és határozd meg a modell célját és nézőpontját;
2. tárd fel a jelenlegi folyamatot legalább két módszerrel (például interjú és megfigyelés vagy dokumentum);
3. készíts BPMN 2.0 modellt legalább három résztvevővel (medencével vagy sávval) és legalább tíz tevékenységgel, és ellenőrizd a 7PMG irányelvei szerint;
4. rendelj a tevékenységekhez időt és az átjárókhoz valószínűséget, és számítsd ki az

átlagos ciklusidőt;
5. használd a Gyakorlati útmutató BPMN-eszközlapját, a modellt pedig egy BPMN-szerkesztőben (például a bpmn.io webes szerkesztőjében) készítsd el.

Továbbolvasás

- Dumas és mtsai. (2018): a folyamatmenedzsment alapvető tankönyve az életciklustól a feltáráson, a modellezésen és az elemzésen át a folyamatbányászatig.
- Object Management Group (2011): a BPMN 2.0 szabvány; ISO/IEC (2013): ugyanez nemzetközi szabványként.
- Mendling, Reijers, és Aalst (2010): a hét modellezési irányelv és empirikus háttere.
- Aalst (2016): a folyamatbányászat alapműve.
- Grohs és mtsai. (2024) és Kourani és mtsai. (2024): nagy nyelvi modellek a folyamatmodellezésben.

21. Ütemezés, készlet és sorbanállás



Tanulási eredmények

A fejezet feldolgozása után képes leszel:

- elhelyezni a gyártásirányítás feladatait a vállalat–irányítás integráció (ISA-95) szintjein;
- megfogalmazni egy ütemezési feladatot (feladatok, műveletek, erőforrások, korlátozások, célfüggvény), és Gantt-diagrammal ábrázolni;
- alkalmazni az SPT, az EDD és a Johnson-sorolást, és értelmezni, melyik célfüggvényre optimális;
- leírni egy diszkrét eseményű rendszert, és eldönteni, mikor kell szimulációval vizsgálni;
- a Kendall-jelöléssel osztályozni egy sorbanállási rendszert, kiszámítani az M/M/1 rendszer jellemzőit, és becsülni a változékonyság hatását;
- kiszámítani a készletforgást és a készlethatékonyságot, és ABC-elemzéssel differenciált készletgazdálkodást tervezni.

A 19. fejezet a folyamatot átlagokkal írta le: áram, kapacitás, kihasználtság. A 20. fejezet a lépések logikáját modellezte. Ez a fejezet három további kérdést vizsgál, amely a megvalósító folyamat tervezésében mindig felmerül:

- **Ki, mikor, min dolgozzon?** Ez az ütemezés kérdése.
- **Mennyit kell várni?** Ezt a sorbanállási modellek és a szimuláció válaszolják meg.
- **Mennyi anyagot és félkész terméket tartsunk?** Ez a készletgazdálkodás kérdése.

A három kérdést a Little-törvény köti össze (19. fejezet): a várakozó munka készlet, és a készlet átfutási idő. A fejezet Lakatos Béla egyetemi jegyzeteire (Lakatos, é. n.b, é. n.a) és a nemzetközi tankönyvekre épül (Pinedo 2022; Hopp és Spearman 2008; Silver, Pyke, és Thomas 2016).

21.1. A gyártásirányítás szintjei

Egy gyártó vállalat döntései különböző időtávokon születnek. Az IEC 62264 szabvány (az amerikai ISA-95 nemzetközi változata) a vállalatirányítási és a gyártásirányítási rendszerek integrációjához funkcionális hierarchiát határoz meg (21.1. ábra) (IEC 2013). A 4. szint az üzleti tervezés és a logisztika (jellemzően az ERP-rendszer), a 3. szint a gyártásirányítás (MOM, a gyakorlatban a MES): a részletes ütemezés, a gyártási rendelések kiadása, a teljesítmény követése. A 2. szint a felügyelet és a folyamatirányítás, az 1. szint az érzékelés és a beavatkozás, a 0. szint maga a fizikai folyamat. Az alsóbb szintek rövidebb időtávon döntenek: a 4. szint hónapokban és hetekben, a 3. szint napokban, műszakokban és órákban, az alsó szintek percekben és másodpercekben gondolkodnak.



21.1. ábra. A vállalat–irányítás integráció szintjei (IEC 2013 nyomán), a ForgácsTech példájával.

Lakatos ugyanezt a hierarchiát a tervezés felől írja le: hosszabb időhorizonton a koordinációs szint aggregált tervet készít, rövid időhorizonton a lokális szint műveletterveket; a kettő együtt már a rendszertervezés és a rendszerirányítás kombinációja (Lakatos, é. n.a, 8.1. szakasz). A ForgácsTech problémája (4. fejezet) éppen a 3. szinten volt: az ERP ismerte a rendeléseket, a gépvezérlők a műveleteket, de a kettő között a heti papíralapú ütemterv egyhetes késleltetéssel zárta az irányítási kört. A MES ezt a rést tölti ki.

21.2. Ütemezés

Az ütemezési feladat elemei: a **munkafeladatok** (mit kell gyártani), a **műveletek** (milyen lépésekből áll), a **műveleti egységek** (milyen erőforrásokon) és a **korlátozások**. A műveletek és a műveleti egységek összerendelése az allokáció. A korlátozások lehetnek időbeliek (kezdési és befejezési idő, megszakíthatóság), kizárásiak (a feladatok osztoznak egy erőforráson) és sorrendiek (egy feladat olyan anyagot vagy információt igényel, amelyet egy másik állít elő) (Lakatos, é. n.b, 6.1. szakasz). Két feladatosztály különböztethető meg: az **ütemezés** az összerendelést és a sorrendet úgy határozza meg, hogy a korlátozások teljesüljenek, és a célfüggvény optimális legyen; a **megvalósíthatósági vizsgálat** adott összerendelésről azt dönti el, teljesíthetők-e a korlátozások (Lakatos, é. n.b, 6.1. szakasz).

Az ütemezési feladatokat a gépkörnyezet szerint osztályozzuk: egy gép, párhuzamos gépek, **folyamatsor** (flow shop: minden feladat ugyanabban a sorrendben halad át a gépeken) és **műhelyrendszer** (job shop: minden feladatnak saját útvonala van) (Pinedo 2022, 2. fejezet). A ForgácsTech egyedi rendelései műhelyrendszert alkotnak, a KoboCella-sor folyamatsor.

A munkafeladatok időjellemzői: a $t_{p,i}$ műveleti idő, a d_i esedékesség (határidő), a $t_{f,i}$ átfutási idő (a feladat befejezéséig eltelt idő), az $L_i = t_{f,i} - d_i$ pontatlanság (lehet negatív), a $T_i = \max\{L_i, 0\}$ késedelem (Lakatos, é. n.b, 6.3. szakasz). A leggyakoribb célfüggvények az átlagos átfutási idő, a legnagyobb késedelem, a késő feladatok száma, a teljes átfutás (az utolsó feladat befejezése) és az átállási veszteség. Az ütemtervet a **Gantt-diagram** ábrázolja: minden sor egy erőforrás, minden téglalap egy művelet, amelynek hossza a műveleti idő. A diagram több mint száz éve a termelésirányítás alapeszköze (Wilson 2003).

Egy gépen a legegyszerűbb ütemezési szabályok a **sorolási elvek** (Lakatos, é. n.b, 6.3. szakasz):

- **FCFS** (first come, first served): érkezési sorrend;
- **SPT** (shortest processing time): a legrövidebb műveleti idő először; egy gépen ez minimalizálja az átlagos átfutási időt, az átlagos várakozási időt és az átlagos pontatlanságot;
- **EDD** (earliest due date): a legkorábbi határidő először; egy gépen ez minimalizálja a legnagyobb késedelmet;
- **CR** (critical ratio): a határidőig hátralévő idő és a műveleti idő hányadosa szerint, dinamikus helyzetekre.

ForgácsTech: hat rendelés egy gépen

A megmunkáló központra reggel hat rendelés vár (21.1. táblázat). Érkezési sorrendben négy rendelés késik, a legnagyobb késedelem 12 óra. SPT-sorolással az átlagos átfutási idő 16,8 óráról 13,0 órára csökken, és csak egy rendelés késik 2 órát. EDD-sorolással egyetlen rendelés sem késik, az átlagos átfutás 13,5 óra. A MES ütemezője ezért a határidő-tartást elsődleges célnak tekintve EDD-vel indul, és az egyforma határidejű rendeléseket SPT szerint rendezi.

21.1. táblázat. A megmunkáló központra váró rendelések.

Rendelés	R1	R2	R3	R4	R5	R6
műveleti idő [óra]	6	2	8	3	5	4
határidő [óra múlva]	20	6	30	10	12	24

21.2. táblázat. A sorolási elvek összevetése.

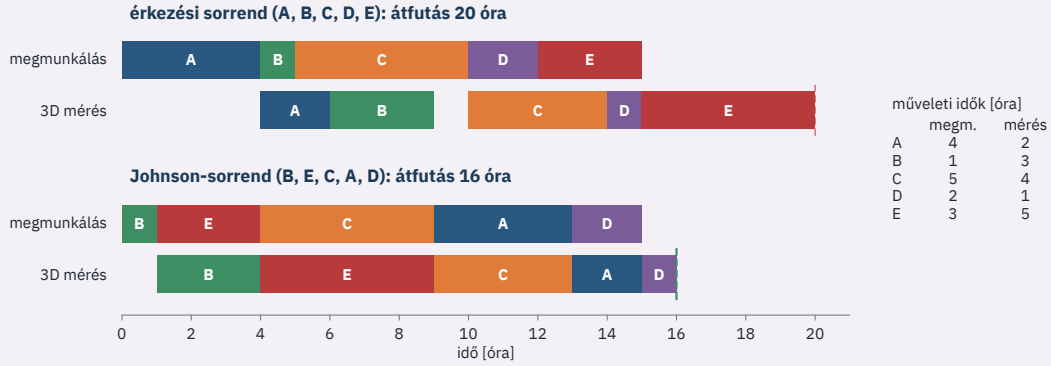
Sorolás	Sorrend	Átlagos átfutás [óra]	Késő rendelések	Legnagyobb késedelem [óra]
FCFS	R1, R2, R3, R4, R5, R6	16,8	4	12
SPT	R2, R4, R6, R5, R1, R3	13,0	1	2
EDD	R2, R4, R5, R1, R6, R3	13,5	0	0

Két gépből álló folyamatsorban, ahol minden feladat előbb az első, majd a második gépen halad át, a teljes átfutást **Johnson eljárása** minimalizálja (S. M. Johnson 1954): keresd meg a legkisebb műveleti időt a még nem sorolt feladatok között; ha ez az első gépen

van, a feladat a sorrend elejére kerül, ha a másodikon, a végére; ismételd a maradék feladatokra (Lakatos, é. n.b, 6.3. szakasz). Három vagy több gépre és műhelyrendszerre a feladat általában NP-nehéz, ezért heurisztikákat, egészértékű programozást vagy kényszerprogramozást használnak (Pinedo 2022).

ForgácsTech: megmunkálás és 3D mérés

Öt rendelés előbb a megmunkáló központon, majd a 3D mérőgépen halad át (21.2. ábra). Érkezési sorrendben a teljes átfutás 20 óra, mert a mérőgép az elején és a C rendelés előtt is áll. A Johnson-sorrend (B, E, C, A, D) a legrövidebb első gépi idejű B rendeléssel indít, hogy a mérőgép hamar munkát kapjon, és a legrövidebb második gépi idejű D rendeléssel zár. A teljes átfutás 16 óra; ez egyben az alsó korlát is (az első gép 15 órája plusz a legrövidebb mérés), tehát az ütemezés optimális.



21.2. ábra. Két gép ütemezése érkezési sorrendben és Johnson eljárásával.

21.3. Diszkrét eseményű rendszerek és szimuláció

A szakaszos gyártórendszerek, a műhelyek és a szolgáltatások **diszkrét eseményű rendszerek**: véges idő alatt véges számú esemény történik, és a rendszer állapota csak ezekben az időpontokban változik (Lakatos, é. n.a, 8.1. szakasz). Az esemény lehet egy munkafeladat beérkezése, egy művelet kezdete vagy vége, egy gép meghibásodása. Az állapot lehet diszkrét (hány feladat vár a gép előtt) vagy folytonos (folyadékszint egy tartályban). A rendszer leírásának elemei az állapot, a rendszeridő és az eseménylista (Lakatos, é. n.b, 8.1. szakasz).

A **diszkrét eseményű szimuláció** ezt a leírást hajtja végre: a szimulációs óra mindig a következő esemény időpontjára ugrik, végrehajtja az esemény hatását az állapotra, és az újonnan keletkező eseményeket felveszi az eseménylistára (Law 2015). A véletlen érkezési és műveleti időket valószínűségi eloszlásokból sorsolja, ezért egy futás egyetlen lehetséges lefolyás: a következtetéshez több független futás és konfidenciaintervallum kell (Law 2015). A szimuláció akkor szükséges, ha a rendszer túl összetett az analitikus modellekhez: több erőforráson osztozó termékek, meghibásodások, átállások, prioritások, munkaidő-beosztás. A szimulációs modell érvényességét a valós rendszerrel kell igazolni (3. fejezet).

21.4. Sorbanállás

Ha a munkafeladatok véletlenszerűen érkeznek, és a kiszolgálásuk ideje is ingadozik, akkor átlagosan elegendő kapacitás mellett is sorok alakulnak ki. A sorbanállási rendszer elemei az ügyfélpopuláció, az érkezési folyamat, a kiszolgáló egységek száma és kiszolgálási ideje, a rendszer kapacitása és a sor viselkedése (Lakatos, é. n.b, 8.3. szakasz). Kendall jelölése a rendszert az *érkezési folyamat / kiszolgálási idő eloszlása / kiszolgálók száma* hármassal írja le (Kendall 1953): M a Markov-tulajdonságú (Poisson-érkezés, exponenciális kiszolgálási idő), D a determinisztikus, G az általános eloszlás. Az M/M/1 rendszerben egy kiszolgáló dolgozik, az érkezési intenzitás λ, a kiszolgálási intenzitás μ, a kihasználtság $\rho = \lambda/\mu < 1$. A stacionárius állapot jellemzői:

$$L = \frac{\rho}{1-\rho}, \qquad W = \frac{1}{\mu-\lambda}, \qquad W_q = \frac{\rho}{\mu-\lambda}, \qquad L_q = \frac{\rho^2}{1-\rho}, \tag{21.1}$$

ahol L a rendszerben, L_q a sorban lévő feladatok várható száma, W a rendszerben, W_q a sorban töltött idő várható értéke. A jellemzőket a Little-törvény köti össze: $L = \lambda W$, $L_q = \lambda W_q$ (Lakatos, é. n.b, 8.3. szakasz; Little 1961).

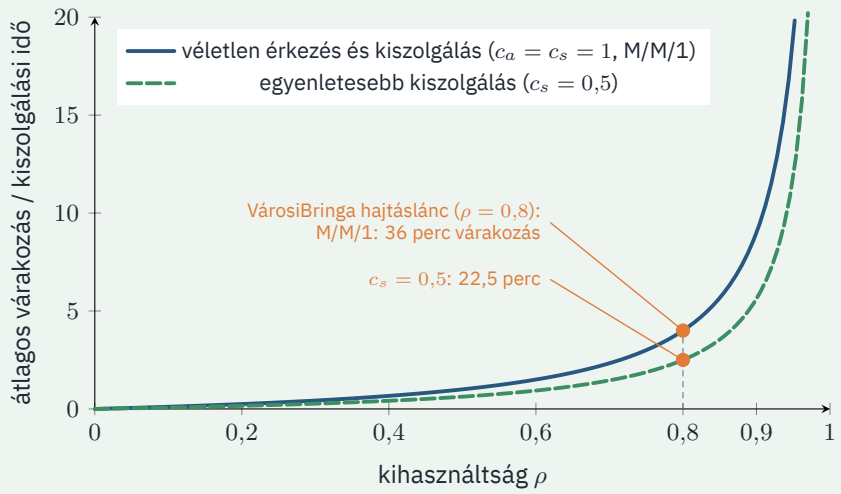
Az M/M/1 modell a véletlen érkezést és a nagyon ingadozó kiszolgálást feltételezi. Általános eloszlásokra Kingman közelítése ad becslést a sorban töltött időre egy kiszolgáló esetén, nagy kihasználtságnál (Kingman 1961). Hopp és Spearman ezt a változékonyság (V), a kihasználtság (U) és az idő (T) szorzataként fogalmazza meg (Hopp és Spearman 2008, 8. fejezet):

$$W_q \approx \underbrace{\frac{c_a^2 + c_s^2}{2}}_{V} \cdot \underbrace{\frac{\rho}{1-\rho}}_{U} \cdot \underbrace{t_s}_{T}, \tag{21.2}$$

ahol c_a és c_s az érkezések közötti idő és a kiszolgálási idő relatív szórása, t_s az átlagos kiszolgálási idő. Exponenciális eloszlásoknál ($c_a = c_s = 1$) a közelítés az M/M/1 pontos képletét adja vissza. A képlet két tanulsága a rendszertervezés alapszabálya: a várakozás a kihasználtság közelében korlátlanul nő, és a változékonyság csökkentése ugyanúgy rövidíti a várakozást, mint a kapacitás bővítése (21.3. ábra).

VárosiBringa: mennyit vár egy kerékpár a hajtásláncra?

A javítóműhely szűk keresztmetszete a hajtáslánc-javítás (19. fejezet): óránként átlagosan $\lambda = 5{,}33$ kerékpár érkezik, a szerelő óránként $\mu = 6{,}67$ kerékpárt javít, a kihasználtság $\rho = 0{,}8$. Ha az érkezések véletlenszerűek és a javítási idők exponenciálisak, egy kerékpár átlagosan $W_q = 0{,}8/(6{,}67 - 5{,}33) = 0{,}6$ órát, azaz 36 percet vár, a rendszerben átlagosan $L = 4$ kerékpár van. A javítási időket méréssel ellenőrizve kiderült, hogy relatív szórásuk csak 0,5, mert a szerelő szabványos munkautasítás szerint dolgozik. Kingman közelítésével a várakozás $\frac{1+0{,}25}{2} \cdot 4 \cdot 9 = 22{,}5$ perc. A nyári csúcsban, a feladatok átcsoportosítása után a mosás kihasználtsága 0,86 lesz; itt a várakozás már a kiszolgálási idő hatszorosa is lehet, ezért a csapat a csúcsidőszakra időpontfoglalást vezetett be, amely az érkezések változékonyságát is csökkenti.



21.3. ábra. Az átlagos várakozás a kihasználtság függvényében, véletlen és egyenletesebb kiszolgálásnál (Kingman közelítése).

Több párhuzamos kiszolgálóra (M/M/c) az Erlang-féle képletek adnak pontos eredményt; ezeket és az időbeli változásokat a Gyakorlati útmutató sorbanállási eszközlapja és notebookja számítja ki.

21.5. Készlet és készlethatékonyság

A készlet a termelési rendszer puffere: alapanyag, alkatrész, műveletközi félkész termék és végtermék formájában. Lakatos a készletezés motivációi között a gazdaságosságot, a bizonytalanságok csökkentését, a szállítási és logisztikai okokat, a termelés simítását és az irányítási költségek csökkentését sorolja fel; a készletezési rendszert az igény jellege, a szállítási idő, a készletellenőrzés periódusa és a készlet értékének változása jellemzi (Lakatos, é. n.b, 10.1. szakasz). A készlet azonban pénzt köt le, helyet foglal, avul, és a Little-törvény szerint meghosszabbítja az átfutást.

A készlet nagyságát a forgalomhoz viszonyítjuk:

$$\text{fordulatszám} = \frac{\text{forgalom (éves felhasználás)}}{\text{átlagos készlet}}, \qquad \text{készlet napokban} = \frac{\text{napok száma}}{\text{fordulatszám}}, \tag{21.3}$$

ahol a forgalmat és a készletet azonos értékalapon (például bekerülési áron) kell mérni (Silver, Pyke, és Thomas 2016). A **készlethatékonyság** az átlagos készlet és a forgalom hányadosa: azt mutatja, mekkora készlet kell ezer forint forgalomhoz. A készlet napokban kifejezve a Little-törvény alkalmazása: az átlagos idő, amelyet egy anyagegység a raktárban tölt.

A készletgazdálkodás erőforrásait a tételek fontossága szerint érdemes elosztani. Az **ABC-elemzés** a tételeket éves felhasználási értékük szerint csökkenő sorrendbe rendezi, és három csoportra osztja: a kevés, de nagy értékű A tételekre, a közepes B és a sok, kis értékű C tételekre (Silver, Pyke, és Thomas 2016). A csoporthatárok vállalatonként változnak; a könyv példái a tételek 20–40–40%-os megosztásával dolgoznak. A csoportokhoz

eltérő szabályozás tartozik: az A tételeknél pontos nyilvántartás, gyakori felülvizsgálat és előrejelzés, a C tételeknél egyszerű nyilvántartás, nagyobb rendelési tételek és biztonsági készlet.

ForgácsTech: az anyagkészlet ABC-elemzése

A ForgácsTech éves anyag- és szerszámfelhasználása 185 millió forint, az átlagos készlet értéke 31 millió forint. A fordulatszám 6, a készlet 230 munkanappal számolva 38 munkanapnyi felhasználásnak felel meg. A 20 leggyakrabban használt tétel ABC-elemzése szerint a tételek 20%-a (három rúdanyag és a keményfém marószár) adja a felhasználási érték 70%-át, a következő 40% 27%-át, az utolsó 40% csupán 3%-át. A csapat az A tételeknél heti készletfelülvizsgálatot és a rendelésekhez kötött anyaglefoglalást vezetett be a MES-ben; ez szüntette meg az anyagra várást a jövőbeli értékáramban (19. fejezet). A C tételeknél negyedéves keretrendelésre tértek át, ami csökkentette a beszerzés adminisztrációját.

A rendelési tételnagyság klasszikus modellje, a gazdaságos rendelési tételnagyság (EOQ), a rendelési (átállási) költség és a készlettartási költség egyensúlyát keresi: $Q^{*} = \sqrt{2DS/H}$, ahol D az éves igény, S a rendelésenkénti fix költség, H az egységnyi éves készlettartási költség (Silver, Pyke, és Thomas 2016). A képlet szerint a fix költség csökkentése kisebb optimális tételnagyságot ad: ugyanez a gondolat áll a SMED mögött (19. fejezet). A rövidebb átállás kisebb sorozatot, kisebb készletet és rövidebb átfutást tesz lehetővé.

Ütemezés és nagy nyelvi modellek

Az ütemezési feladatok nagy része kombinatorikus optimalizálási feladat, amelyet célalgoritmusok (heurisztikák, egészértékű és kényszerprogramozás) oldanak meg (Pinedo 2022). A nagy nyelvi modellek itt elsősorban a feladat megfogalmazásában segíthetnek: a szöveges igényekből (határidők, prioritások, kizárások) korlátozásokat és célfüggvényt írnak, vagy elmagyarázzák egy ütemterv döntéseit. Az ütemtervet azonban ellenőrizhető algoritmusnak kell előállítania, és a korlátozások teljesülését automatikusan ellenőrizni kell.

Átlaggal méretezett rendszer

Gyakori hiba, hogy a kapacitást az átlagos igényre méretezik, és 95–100%-os kihasználtsággal terveznek. A sorbanállási modellek szerint ilyenkor a várakozás a kiszolgálási idő sokszorosa, és a legkisebb zavar is hosszú sorokat okoz. A szűk keresztmetszeteken tartalékkapacitást kell tervezni, vagy csökkenteni kell az érkezések és a kiszolgálás változékonyságát (időpontfoglalás, szabványos munka, kisebb sorozatok).

Összefoglalás

- Az IEC 62264 (ISA-95) a vállalati (4.) és a gyártásirányítási (3.) szint integrációját írja le; a MES a 3. szinten végzi a részletes ütemezést és a teljesítménykövetést.
- Az ütemezési feladat elemei a munkafeladatok, a műveletek, az erőforrások, a korlátozások és a célfüggvény; az ütemtervet Gantt-diagram ábrázolja.
- Egy gépen az SPT az átlagos átfutást, az EDD a legnagyobb késedelmet minimalizálja; két gépes folyamatsorban a Johnson-eljárás a teljes átfutást.
- A diszkrét eseményű rendszer állapota csak eseményekkor változik; a szimuláció a következő esemény időpontjára léptet, és több futásból von le következtetést.
- Az M/M/1 rendszer várakozása $\rho/(\mu - \lambda)$; Kingman közelítése szerint a várakozás a változékonyság, a kihasználtság és a kiszolgálási idő szorzata.
- A készletforgás és a készletnapok a Little-törvény alkalmazásai; az ABC-elemzés a készletgazdálkodás figyelmét a nagy értékű tételekre irányítja.

Önellenőrzés

1. Melyik ISA-95 szinthez tartozik a heti termelési terv, a műszakon belüli rendeléssorrend és egy hajtás fordulatszám-szabályozása?
2. Négy feladat műveleti ideje 5, 2, 7 és 3 óra, határidejük 9, 4, 20 és 6 óra. Sorold SPT és EDD szerint, és számítsd ki az átlagos átfutást és a legnagyobb késedelmet!
3. Négy feladat ideje az első és a második gépen: A (3, 6), B (5, 2), C (1, 2), D (6, 6). Add meg a Johnson-sorrendet és a teljes átfutást!
4. Miért kell több független szimulációs futás egy sztochasztikus rendszer vizsgálatához?
5. Egy kiszolgálóhoz óránként 8 ügyfél érkezik, a kiszolgálás átlagosan 6 perc. Mennyi a kihasználtság, és M/M/1 feltevéssel mennyi a sorban töltött idő és a rendszerben lévő ügyfelek száma?
6. Az előző rendszerben a kiszolgálás relatív szórása 0,3-ra csökken. Mennyi a sorban töltött idő Kingman közelítésével?
7. Egy cég éves anyagfelhasználása 120 millió forint, átlagos készlete 20 millió forint. Mennyi a fordulatszám és a készlet napokban (365 nappal)?
8. Miért nem elég az ABC-elemzés önmagában a készletszabályozás megtervezéséhez?

Megoldás.

1. A heti terv a 4. szint (üzleti tervezés), a műszakon belüli sorrend a 3. szint (gyártásirányítás), a fordulatszám-szabályozás a 2. és az 1. szint.
2. SPT: 2, 3, 5, 7 (a második, a negyedik, az első, a harmadik feladat); befejezések 2, 5, 10, 17, átlagos átfutás 8,5 óra; késedelmek: 0, 0, 1 (az első feladat $10 > 9$), 0, legnagyobb 1 óra. EDD: határidő szerint 4, 6, 9, 20 (második, negyedik, első, harmadik): ugyanez a sorrend, átlagos átfutás 8,5 óra, legnagyobb késedelem 1 óra.
3. A legkisebb idő C első gépen (1) → elejére; aztán B második gépen (2) → végére; aztán A első gépen (3) → C után; végül D. Sorrend: C, A, D, B. Első gép: C 0–1, A 1–4, D 4–10, B 10–15; második gép: C 1–3, A 4–10, D 10–16, B 16–18. Teljes átfutás 18 óra.
4. Mert egy futás a véletlen sorsolások egyetlen lehetséges kimenete; a teljesítményjellemzők becsléséhez és a bizonytalanság számszerűsítéséhez (konfidenciaintervallum) több független futás eredményének eloszlása kell.
5. $\mu = 10$/óra, $\rho = 0{,}8$; $W_q = 0{,}8/(10-8) = 0{,}4$ óra = 24 perc; $L = 0{,}8/0{,}2 = 4$ ügyfél.
6. $W_q \approx \frac{1+0{,}09}{2} \cdot 4 \cdot 6 = 13{,}1$ perc.
7. Fordulatszám $120/20 = 6$, készlet napokban $365/6 \approx 61$ nap.
8. Mert csak a felhasználási értéket nézi; az igény változékonyságát, a beszerzési időt, a tétel kritikusságát (például egy olcsó, de pótolhatatlan alkatrész) és az avulást nem. Ezért gyakran kiegészítik az igény változékonysága szerinti (XYZ) vagy kritikussági besorolással.

Projektlépés: ütemezés, sorbanállás, készlet

Saját fejlesztési feladatodhoz, ha a megvalósító folyamatban releváns:

1. helyezd el a rendszer irányítási funkcióit az ISA-95 szintjein, és jelöld, melyik szinten milyen időtávú döntés születik;
2. készíts Gantt-diagramot egy jellemző napra vagy műszakra, és vesd össze legalább két sorolási elvet;
3. a szűk keresztmetszetnél becsüld meg a várakozást sorbanállási modellel, és vizsgáld meg a kihasználtság és a változékonyság hatását;
4. ha a folyamatban készlet van, számítsd ki a fordulatszámot, és készíts ABC-elemzést;
5. használd a Gyakorlati útmutató Gantt-, ABC- és sorbanállási eszközlapját.

Továbbolvasás

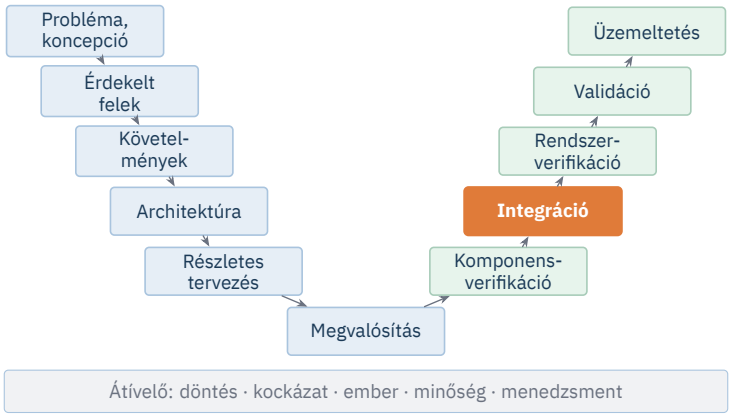
- Lakatos (é. n.b), 6., 8. és 10. fejezet, valamint Lakatos (é. n.a), 8. fejezet: a gyártásütemezés, a diszkrét eseményű rendszerek, a sorbanállás és a készletezés magyar nyelvű alapjai.
- Pinedo (2022): az ütemezéselmélet és -algoritmusok alapvető tankönyve.
- Hopp és Spearman (2008): a változékonyság, a kihasználtság és a várakozás kapcsolata

a gyártórendszerekben.
- Law (2015): a diszkrét eseményű szimuláció módszertana.
- Silver, Pyke, és Thomas (2016): a készletgazdálkodás és a termeléstervezés tankönyve.
- IEC (2013): a vállalat–irányítás integráció szabványa.

IV. rész – A V jobb ága: integrációtól az üzemeltetésig

Pannon Egyetem
University of Pannonia

22. Integráció és interfészmenedzsment



Tanulási eredmények

A fejezet feldolgozása után képes leszel:

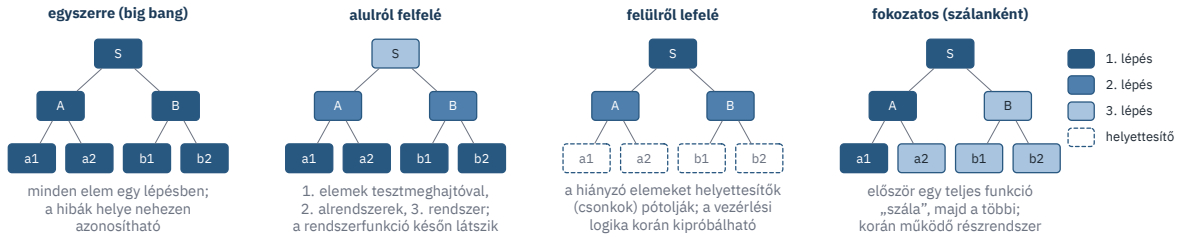
- elmagyarázni az integráció célját és helyét a V-modell jobb ágán;
- összevetni az integrációs stratégiákat, és indokoltan választani közülük;
- azonosítani és dokumentálni a rendszer interfészeit, és megkülönböztetni az interfészkövetelmény-dokumentumot (IRD) és az interfészvezérlő dokumentumot (ICD);
- integrációs tervet készíteni, amely lépésenként megadja az integrálandó elemeket, a segédeszközöket, a felelősöket és a lépést lezáró ellenőrzést;
- felismerni a jellegzetes integrációs kockázatokat, és tervezni a kezelésüket.

A V-modell bal ága a rendszert lebontja, a jobb ága összerakja. Az összerakás nem egyszerű szerelés: a külön-külön jól működő elemek együtt gyakran nem működnek, mert az interfészeken, a köztük lévő kölcsönhatásokban rejlő hibák csak ekkor jelennek meg. Az ISO/IEC/IEEE 15288 az integrációt önálló műszaki folyamatként írja le: célja, hogy a rendszerelemekből olyan megvalósult rendszert állítson össze, amely teljesíti a rendszerkövetelményeket, és előkészítse a verifikációt és az átadást (ISO/IEC/IEEE 2023, 6.4.8. szakasz; NASA 2016, 5.2. szakasz). Az integrációra a rendszerhierarchia minden szintjén szükség van: az alkatrészekből alrendszer, az alrendszerekből rendszer lesz.

Két dolgot érdemes már az elején rögzíteni. Az integráció nem egyszeri esemény a gyártás végén: elemzéssel és szimulációval kezdődik, és egyre valószerűbb, egyre több funkciót tartalmazó összeállításokon át jut el a végtermékig. Az integráció tervezését pedig már a koncepció kialakításakor el kell kezdeni, mert az architektúra döntései (például a modulhatárok) meghatározzák, milyen sorrendben lehet majd összeépíteni a rendszert (8. fejezet).

22.1. Integrációs stratégiák

Az **integrációs stratégia** megadja, milyen sorrendben, milyen csoportokban kerülnek össze az elemek, és mi ellenőrzi az egyes lépéseket. A szabvány szerint a stratégia célja, hogy az integráció idejét, költségét és kockázatát a lehető legkisebbre csökkentse (ISO/IEC/IEEE 2023, 6.4.8. szakasz). Minél nagyobb, összetettebb vagy érzékenyebb a rendszer, annál fontosabb a helyes sorrend. A sorrendet rendszeresen felül kell vizsgálni, mert a gyártási és szállítási ütemezés változásai érvényteleníthetik a korábbi döntések alapjait. A leggyakoribb stratégiákat a 22.1. ábra mutatja (INCOSE 2023b).



22.1. ábra. Integrációs stratégiák egy háromszintű rendszeren.

- **Egyszerre (big bang):** minden elemet egy lépésben kapcsolunk össze. Kis, jól ismert rendszernél gyors, de ha valami nem működik, nehéz kideríteni, melyik elem vagy interfész a hibás.
- **Alulról felfelé:** előbb az elemeket, majd az alrendszereket, végül a rendszert integráljuk. Minden szinten csak ellenőrzött építőelemek kerülnek össze, de a rendszer egészének viselkedése csak a végén látszik; a hiányzó felsőbb szintet tesztmeghajtók pótolják.
- **Felülről lefelé:** a rendszer vezérlő logikája készül el először, a hiányzó elemeket egyszerű helyettesítők (csonkok, szimulátorok) pótolják. A rendszerszintű működés korán kipróbálható, a helyettesítők azonban munkát igényelnek, és pontatlanságuk elfedhet hibákat.
- **Fokozatos, szálanként:** először egy teljes funkció „szálát” integráljuk a bemenettől a kimenetig, majd a többit. Korán van működő, bemutatható részrendszer, és a legfontosabb funkciók a legtöbbet tesztelődnek.

A választást a rendszer szerkezete, az elemek elkészülésének sorrendje, a kockázatok és a rendelkezésre álló segédeszközök határozzák meg. A gyakorlatban a stratégiák keverednek: a mechanikai alrendszereket alulról felfelé, a szoftvert felülről lefelé, a kritikus funkciót szálanként integrálják.

22.2. Interfészmenedzsment

Az **interfész** két rendszerelem vagy a rendszer és környezete közötti kapcsolat, amelyen anyag, energia, információ vagy erőhatás halad át. Az interfészek típusai: mechanikai (illesztés, rögzítés, tömeg), energetikai (villamos, pneumatikus), adat- és jelinterfész, ember–gép interfész és környezeti interfész (hő, rezgés, elektromágneses hatások). Az interfészeket a funkcióelemzés és az N²-diagram segítségével azonosítjuk (8. fejezet).

A nagy projektek gyakorlatában az interfészek dokumentálására három dokumentumtípus terjedt el (Defense Acquisition University 2001; NASA 2016, 5.2. szakasz és L függelék):

- az **interfészkövetelmény-dokumentum** (IRD) az interfész követelményeit rögzíti, és lehetőleg nem írja elő a megoldást;
- az **interfészvezérlő dokumentum** (ICD) végső formájában az IRD követelményeinek részletes megvalósítását írja le (méretek, csatlakozók, jelek, adatformátumok, időzítés);
- az **interfészvezérlési terv** az IRD-k és ICD-k kezelésének folyamatát határozza meg: hogyan tárják fel és oldják fel az összeférhetetlenségeket, és hogyan értékelik az interfészt érintő tervezési változtatások hatását.

Az interfészkezelés az életcikluson átívelő tevékenység: az interfészeket azonosítani, dokumentálni és a változásaikat szabályozottan kezelni kell a teljes életciklus alatt (INCOSE 2023b). A kompatibilitási elemzés igazolja, hogy az interfész-meghatározás teljes, és a változások nyomon követhetők; enélkül előfordulhat, hogy egy elem a rendszerbe illesztve nem a szándék szerint működik.

KoboCella: az interfészjegyzék

A csapat az N²-diagram (8. fejezet) alapján nyolc interfészt azonosított, és mindegyikhez felelőst és ICD-t rendelt (22.1. táblázat). A két legkockázatosabb interfész a kezelő és a kezelőfelület közötti (I-06), mert itt a tervezett működés az emberen múlik (11. fejezet), és a csavarozó vezérlője és a MES közötti (I-07), mert a két oldalt két különböző beszállító fejleszti.

22.1. táblázat. A KoboCella interfészjegyzéke.

Azonosító	Interfész	Típus	Mi halad át?	Felelős
I-01	kobot – csavarozó orsó	mechanikai, energetikai, jel	rögzítés, tömeg, 24 V, indítás/kész	robotintegrátor
I-02	csavarozó orsó – csavaradagoló	mechanikai, pneumatikus	csavar, sűrített levegő	robotintegrátor
I-03	kobot – cellavezérlés (PLC)	jel, adat	indítás, kész, hibakód	robotintegrátor
I-04	lézerszkenner – biztonsági vezérlés	biztonsági jel	védőmező megsértése, leállítás	biztonsági mérnök
I-05	befogó készülék – motorház	mechanikai	pozicionálás, rögzítés	gyártástechnológus
I-06	kezelő – kezelőfelület	ember–gép	utasítás, hibajelzés, nyugtázás	ergonómus, rendszermérnök
I-07	csavarozó vezérlő – MES	adat	nyomaték, szög, megfelelő/nem megfelelő, ciklusidő	informatika, beszállító
I-08	cella – üzem	energetikai, környezeti	400 V, sűrített levegő, padló, hőmérséklet	karbantartás

22.3. Az integrációs terv

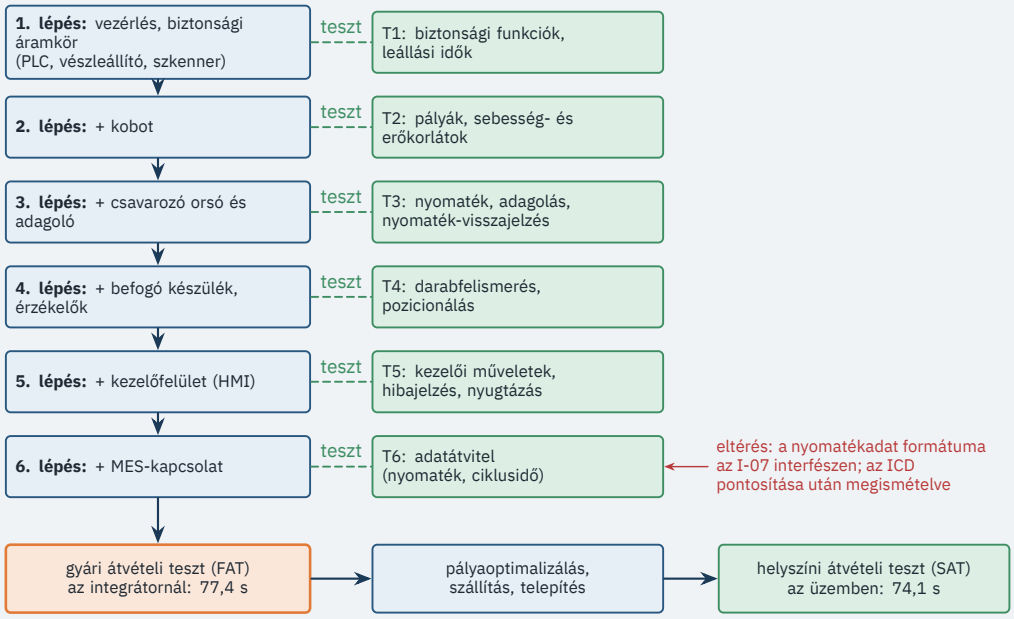
Az integrációs terv a stratégiát lépésekre bontja. Indokolja a választott stratégiát (ha szükséges, az alternatívák elemzésével), és lépésenként megadja a tevékenységek helyét, az integrálandó elemeket, a szükséges segédeszközöket (különleges szoftver, teszteszközök, csonkok és meghajtók a még hiányzó elemek szimulálására), a telepítés utáni integrációs

tevékenységeket, a lépést követő verifikációt, a felelősöket és az ütemezést. A stratégia egyben az építési, a verifikációs és a telepítési terv alapja is.

Minden integrációs lépést ellenőrzés zár: csak akkor lépünk tovább, ha az eddig összerakott rész működik. Így a hiba mindig a legutóbb hozzáadott elemben vagy interfészében keresendő. Az integráció két jellegzetes mérföldköve a **gyári átvételi teszt** (factory acceptance test, FAT) a beszállítónál vagy az integrátornál, és a **helyszíni átvételi teszt** (site acceptance test, SAT) a felhasználás helyén. A kettő között a rendszer környezete megváltozik: más az energiaellátás, a padló, a hálózat, és mások a kezelők.

KoboCella: integráció lépésenként

A csapat alulról felfelé és szálanként vegyes stratégiát választott (22.2. ábra). Elsőként a vezérlés és a biztonsági áramkör került össze, mert minden további lépés csak működő biztonsági funkciók mellett végezhető biztonságosan. Ezután a kobot, a csavarozó orsó és az adagoló, a befogó készülék, a kezelőfelület és végül a MES-kapcsolat következett; minden lépést teszt zárt. A hatodik lépésben derült ki, hogy a csavarozó vezérlője a nyomatékot más mértékegységben és tizedesjegy-pontossággal küldi, mint amit a MES vár (I-07). A hibát az ICD pontosítása és a fogadó oldal átalakítása oldotta meg, a T6 tesztet megismételték. A gyári átvételi teszten a ciklusidő 77,4 s volt; a kobot pályájának optimalizálása után a helyszíni átvételen 74,1 s (6. fejezet).



22.2. ábra. A KoboCella integrációs lépései, a lépéseket lezáró tesztek és az átvételi tesztek.

22.4. Integrációs kockázatok és tanulságok

Az integráció a projekt egyik legkockázatosabb szakasza, mert itt derül ki minden, amit a tervezés során nem gondoltak végig. A leggyakoribb kockázatok:

- **későn rögzített vagy kétértelmű interfészek:** ha az ICD csak a gyártás után készül el, a két oldal eltérően értelmezi;
- **beszállítói elemek ismeretlen viselkedése:** a kereskedelmi termékek dokumentációja ritkán írja le a határhelyzeteket;

- **a teszt- és az üzemi környezet eltérése:** ami az integrátor csarnokában működik, az üzemben másként viselkedhet;
- **szoftver- és firmware-változatok:** az integráció közben frissített szoftver más viselkedést mutat;
- **a segédeszközök hiánya:** ha nincs szimulátor vagy csonk, a lépés csak akkor végezhető el, amikor minden elem megérkezett, és a stratégia észrevétlenül „big bang”-gé válik.

Gyakran idézett mérnöki szabály, hogy úgy kell tesztelni, ahogyan a rendszert használni fogják, és úgy kell használni, ahogyan tesztelték. Ez az integrációra is igaz: a lépéseket záró teszteknek a valós működési feltételeket kell közelíteniük.

ForgácsTech: integráció a gépvezérlők garanciájának megőrzésével

A ForgácsTech problémamegfogalmazása korlátként rögzítette, hogy a gépek vezérlőinek gyártói garanciáját nem szabad veszélyeztetni (4. fejezet). Ezért a MES és a gépek közötti interfészt csak olvasó adatkapcsolatként határozták meg: a MES a vezérlők szabványos adatfelületéről olvassa a gépállapotot és a darabszámot, de nem ír a vezérlőbe. A régebbi esztergáknál, ahol ilyen felület nincs, külső érzékelők (áramfelvétel, ajtókapcsoló) adják a jelet. Az integrációt szálanként végezték: elsőként egyetlen megmunkáló központ teljes adatútját kötötték be a gépállapottól az OEE-kijelzőig, és csak a működő szál után következett a többi gép.

Folyamatos integráció

A szoftverfejlesztésben a folyamatos integráció azt jelenti, hogy a fejlesztők változtatásait naponta többször egy közös változatba olvasztják, és minden összeolvasztás után automatikus tesztek futnak. A gondolat a mechatronikai rendszerekre is átvihető: a digitális iker és a hardver a hurokban (hardware-in-the-loop) tesztelés lehetővé teszi, hogy a vezérlőszoftvert a valós hardver elkészülte előtt szimulált környezetben integrálják és teszteljék (3. fejezet). A virtuális üzembe helyezés így a FAT előtt feltárhatja az interfész- és logikai hibák nagy részét.

Integráció interfészjegyzék nélkül

Gyakori hiba, hogy az interfészeket nem azonosítják és nem dokumentálják külön, mert „a rajzokon úgyis rajta van”. A rajz azonban csak az egyik oldalt mutatja; a két oldal közötti megállapodást, a mértékegységeket, a tűréseket, az időzítést és a hibahelyzetek kezelését senki nem rögzíti. Az integráció során ezek a hiányok egyszerre jelentkeznek, amikor a legdrágább kijavítani őket.

Összefoglalás

- Az integráció az elemekből a magasabb szintű terméket állítja össze, és igazolja, hogy együtt működnek; a rendszerhierarchia minden szintjén szükséges, és tervezését a koncepciónál kell kezdeni.
- A stratégiák: egyszerre, alulról felfelé, felülről lefelé, szálanként; a gyakorlatban vegyesen alkalmazzák őket, a sorrendet műszaki és üzleti szempontok együtt határozzák meg.
- Az interfészeket azonosítani, dokumentálni és szabályozottan változtatni kell; az IRD a követelményeket, az ICD a megvalósítást, az interfészvezérlési terv a kezelés folyamatát rögzíti.
- Az integrációs terv lépésenként megadja az elemeket, a segédeszközöket, a felelősöket, az ütemet és a lépést lezáró ellenőrzést.
- A legnagyobb kockázatot a későn rögzített interfészek, az ismeretlen beszállítói elemek és a teszt- és az üzemi környezet eltérése jelenti.

Önellenőrzés

1. Miért nem egyszeri esemény az integráció?
2. Hasonlítsd össze az alulról felfelé és a felülről lefelé integrációt! Mikor melyiket választanád?
3. Mi a különbség az IRD és az ICD között? Miért nem szabad az IRD-ben megoldást előírni?
4. Sorolj fel öt interfésztípust, és adj mindegyikre példát egy háztartási robotporszívóból!
5. Mit tartalmaz az integrációs terv egy lépésének leírása?
6. Miért hasznos, ha minden integrációs lépést ellenőrzés zár?
7. Mi a FAT és a SAT, és miért nem helyettesíti az egyik a másikat?

Megoldás.

1. Mert elemzéssel és szimulációval kezdődik, és egyre valószerűbb, egyre több funkciót tartalmazó összeállításokon át jut el a végtermékig; ráadásul a rendszerhierarchia minden szintjén ismétlődik, és a tervezése már a koncepciónál elkezdődik.
2. Alulról felfelé: minden szinten ellenőrzött elemek kerülnek össze, de a rendszerszintű működés későn látszik; akkor jó, ha az elemek korán elkészülnek, és a fizikai illesztés a kritikus. Felülről lefelé: a vezérlési logika és a rendszerszintű működés korán kipróbálható, de helyettesítők kellenek; akkor jó, ha a vezérlés a legkockázatosabb, és a helyettesítők olcsón elkészíthetők.
3. Az IRD az interfész követelményeit rögzíti (mit kell teljesítenie), az ICD a részletes megvalósítást (hogyan). Ha az IRD megoldást ír elő, szükségtelenül szűkíti a tervezési teret, és a később jobb megoldás csak követelményváltozással vezethető be.
4. Mechanikai: a kefe rögzítése; energetikai: a töltőállomás érintkezői; adat és jel: a mobilalkalmazás és a robot közötti kapcsolat; ember–gép: a gombok és a hangjelzések; környezeti: a padló, a küszöbök, a por és a hő.
5. A tevékenységek helyét, az integrálandó elemeket, a segédeszközöket (szoftver, teszteszköz, csonk, meghajtó), a telepítés utáni integrációs tevékenységeket, a lépést követő verifikációt, a felelősöket és az ütemezést.
6. Mert így a hiba a legutóbb hozzáadott elemben vagy annak interfészében keresendő, és a hibakeresés gyors; enélkül a hibák összeadódnak, és a lokalizálásuk nehéz.
7. A FAT a gyártónál vagy az integrátornál, a SAT a felhasználás helyén zajló átvételi teszt. Nem helyettesítik egymást, mert a környezet (energiaellátás, padló, hálózat, kezelők, anyagok) eltér, és a helyszínen olyan hibák jelentkezhetnek, amelyek a gyárban nem.

Projektlépés: integrációs terv

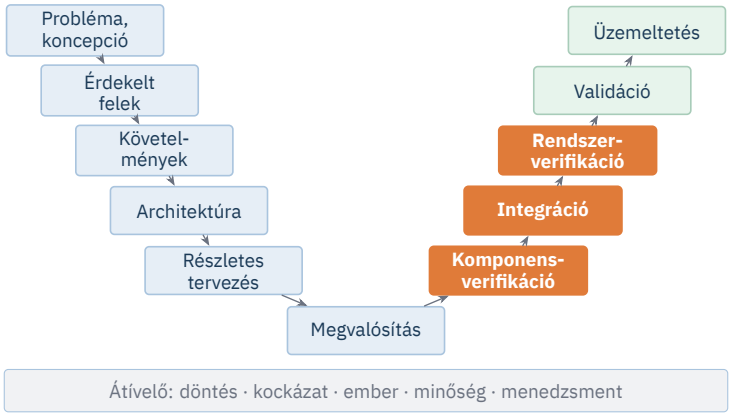
Saját fejlesztési feladatodhoz:

1. készíts interfészjegyzéket az N²-diagram alapján (azonosító, interfész, típus, mi halad át, felelős);
2. válassz integrációs stratégiát, és indokold a rendszer szerkezete és a kockázatok alapján;
3. bontsd a stratégiát legalább öt integrációs lépésre: mi kerül össze, milyen segédeszköz kell, mi zárja a lépést, ki a felelős;
4. azonosítsd a három legnagyobb integrációs kockázatot, és tervezd meg a kezelésüket;
5. használd a Gyakorlati útmutató integrációs terv eszközlapját.

Továbbolvasás

- ISO/IEC/IEEE (2023), 6.4.8. szakasz: az integrációs folyamat a szabványban.
- INCOSE (2023b): az integrációs stratégiák és az interfészmenedzsment gyakorlata.
- NASA (2016), 5.2. és 6.3. szakasz, H és L függelék: a termékintegráció folyamata, az interfészmenedzsment, az integrációs terv és az interfészkövetelmény-dokumentum vázlata.

23. Verifikáció



Tanulási eredmények

A fejezet feldolgozása után képes leszel:

- megkülönböztetni a verifikációt, a validációt, a minősítést, az átvételt és a tanúsítást;
- minden követelményhez megfelelő verifikációs módot rendelni (vizsgálat, elemzés, bemutatás, teszt);
- követelmény-verifikációs mátrixot készíteni sikerkritériumokkal;
- megtervezni egy verifikációs eljárást, és helyesen kezelni az eltéréseket;
- a mintavételből származó bizonytalanságot konfidenciahatárral és sikeres futás próbával figyelembe venni.

Az integráció végén a rendszer összeáll és működik. A verifikáció azt kérdezi: azt építettük-e meg, amit a követelmények előírnak? Az ISO/IEC/IEEE 15288 szerint a verifikáció célja objektív bizonyítékot szolgáltatni arról, hogy a rendszer vagy a rendszerelem teljesíti a rá megadott követelményeket és jellemzőket (ISO/IEC/IEEE 2023, 6.4.9. szakasz; NASA 2016, 5.3. szakasz). A V-modellben a verifikáció a bal ág követelményeit zárja le, szintről szintre: a komponenst a komponenskövetelményekkel, az alrendszert az alrendszer-követelményekkel, a rendszert a rendszerkövetelményekkel vetjük össze (2. fejezet). A rendszerek, szoftverek és hardverek verifikációs és validációs tevékenységeit az IEEE 1012 szabvány részletezi (IEEE 2024).

23.1. Verifikáció, validáció és rokon fogalmak

A **verifikáció** a megadott követelményekhez, a **validáció** az érdekelt felek elvárásaihoz és a működési koncepcióhoz méri a terméket. A különbség a tesztek jellegén is látszik. A verifikációs teszt a jóváhagyott követelményekre vezethető vissza: hivatalos, jegyzőkönyvezett teszt, amely mérőeszközökkel, ellenőrzött környezetben igazolja, hogy a rendszer teljesíti a rá allokált követelményeket, beleértve a fizikai és funkcionális interfészeket. A validációs teszt ezzel szemben a működési koncepcióra vezethető vissza: valószerű

vagy szimulált körülmények között, jellemző felhasználókkal vizsgálja, hogy a termék alkalmas-e a rendeltetésére (IEEE 2024; Engel 2010). A verifikáció tehát azt kérdezi, *jól építettük-e meg* a terméket; a validáció azt, hogy *a jó terméket építettük-e meg* (24. fejezet).

A verifikációhoz három rokon fogalom kapcsolódik (23.1. táblázat). A különbség abban áll, mit igazolnak, és hányszor kell elvégezni őket.

23.1. táblázat. A verifikáció, a minősítés, az átvétel és a tanúsítás. Az európai űripari verifikációs szabvány (ECSS 2018) és az INCOSE kézikönyve (INCOSE 2023b) alapján, saját összeállítás.

Fogalom	Mit igazol?	Hányszor?
verifikáció	a rendszer és elemei teljesítik a megadott követelményeket (teszttel, elemzéssel, vizsgálattal vagy bemutatással)	egyszer, amíg a terv nem változik
minősítés	a terv a várható környezeti szélsőértékek mellett is teljesíti a funkcionális és teljesítménykövetelményeket, a tervezett tartalékokkal	egyszer, amíg a terv nem változik
átvétel	az adott példány gyártása megfelel a már verifikált és minősített tervnek	minden legyártott példányon
tanúsítás	a verifikáció bizonyítékait az illetékes szervezet áttekinti, és a tervet használatra alkalmasnak nyilvánítja	egyszer, amíg a terv nem változik

A KoboCella gyári és helyszíni átvételi tesztje (FAT és SAT, 22. fejezet) ebben az értelemben egyszerre verifikáció és átvétel: a cella egyedi példány, így a terv verifikációja és a példány átvétele egybeesik. Egy sorozatban gyártott dokkolónál (VárosiBringa) viszont a verifikációt és a minősítést egyszer, néhány mintapéldányon végzik el, az átvételi tesztet pedig minden legyártott darabon, szűkebb tartalommal.

23.2. A verifikáció módjai

A szabványok és a szakirodalom jellemzően négy verifikációs módot különböztetnek meg (23.2. táblázat) (ISO/IEC/IEEE 2018; INCOSE 2023b). Az európai űripari szabvány a bemutatás helyett a tervdokumentáció átvizsgálását (review of design) tekinti negyedik módnak (ECSS 2018). Ezek már a követelmények megfogalmazásakor szóba kerültek (6. fejezet): ha egy követelményhez nem rendelhető verifikációs mód, a követelményt újra kell fogalmazni.

- **Elemzés:** matematikai modellek és elemzési módszerek használata annak előrejelzésére, hogy a terv megfelel-e az elvárásoknak, számított adatok vagy az alacsonyabb szintű elemek verifikációs eredményei alapján. Akkor alkalmazzák, ha prototípus vagy kész termék még nem áll rendelkezésre; ide tartozik a modellezés és a szimuláció, valamint a hasonlóság alapján végzett verifikáció egy korábban bevált termék esetén.
- **Bemutatás:** annak megmutatása, hogy a termék használata teljesíti a követelményt. Alapvető képességigazolás, amely a teszttől abban különbözik, hogy nem gyűjt részletes adatokat. Makettel vagy szimulátorral is végezhető.

- **Vizsgálat:** a kész termék szemrevételezése. Fizikai jellemzők, feliratok, azonosítók ellenőrzésére szolgál; kiterjedhet rajzok, dokumentumok és egyéb nyilvántartások ellenőrzésére is.
- **Teszt:** a termék használata a teljesítmény igazolásához szükséges részletes adatok megszerzésére, vagy annyi információ előállítására, amennyi a teljesítmény további elemzéssel való igazolásához kell. Ellenőrzött körülmények között, minden követelményre diszkrét pontokban szolgáltat adatot, és a legtöbb erőforrást igénylő verifikációs mód.

23.2. táblázat. A verifikáció négy módja (ISO/IEC/IEEE 2018; INCOSE 2023b).

Mód	Mikor választjuk?	KoboCella-példa
elemzés	nincs még kész termék, a teszt túl drága vagy nem kivitelezhető, valószínűségi követelmény	R-02: havi rendelkezésre állás megbízhatósági blokkdiagrammal (16. fejezet)
bemutatás	a képesség megléte a kérdés, részletes mérés nélkül	R-09: a nyomatékadatok megjelennek a MES-ben a motor azonosítójához rendelve
vizsgálat	fizikai jellemző, felirat, méret, dokumentum	R-15: a cella elfér a 6 × 4 m-es területen
teszt	mennyiségi teljesítménykövetelmény, ellenőrzött körülmények között	R-01: ciklusidő próbaüzemben; R-03: nyomaték kalibrált mérőeszközzel

A választás mérlegelés kérdése. A verifikáció a fejlesztés egyik legnagyobb költség- és ütemezési tényezője (Engel 2010), ezért a verifikációs módokról és a létesítményekről már az életciklus elején érdemes elemzéssel dönteni, például egy modellen végzett teszt és egy olcsóbb, de kevésbé feltáró elemzés között. A kockázatelemzés, például az FMEA, gyakran új aggályokat tár fel, amelyek további verifikációt indokolnak (15. fejezet).

23.3. A verifikáció tervezése: a követelmény-verifikációs mátrix

A verifikációs terv minden követelményhez megadja, hogyan, hol, mikor és ki igazolja a teljesülését. Ennek központi eszköze a **követelmény-verifikációs mátrix** (requirements verification matrix, RVM). Az európai űripari szabvány ezt verifikációs ellenőrző dokumentumnak (verification control document, VCD) nevezi (ECSS 2018; NASA 2016, D függelék). A mátrixba a kötelező („kell”) követelmények kerülnek, mindegyik egyedi azonosítóval és a forrásdokumentum megjelölésével. A javasolt minimális tartalom: azonosító, forrásdokumentum és bekezdés, a követelmény szövege, a sikerkritérium, a verifikációs mód, a létesítmény vagy labor, a fázis (szint), annak jelölése, hogy a követelményt minden példány átvételekor is ellenőrizni kell-e, a végrehajtó szervezet és az eredményt tartalmazó dokumentum.

A mátrix legfontosabb oszlopa a **sikerkritérium**. Az R-01 követelmény („legfeljebb 75 s”) önmagában nem mondja meg, hány ciklust kell mérni, milyen körülmények között, és mi számít teljesülésnek, ha az egyes ciklusok szórnak. A sikerkritérium ezt rögzíti, még a teszt előtt. A kritérium utólagos megfogalmazása, amikor az eredmény már ismert, a verifikáció hitelességét rontja.

KoboCella: a követelmény-verifikációs mátrix részlete

A 23.3. táblázat a KoboCella követelményeinek (6. fejezet) verifikációs mátrixát mutatja a helyszíni átvétel után. Két követelmény a gyári átvételen még nem teljesült: az R-01 (77,4 s) és az R-09 (az I-07 interfész adatformátuma, 22. fejezet). Mindkettő javítás után, ismételt verifikációval teljesült.

23.3. táblázat. A KoboCella követelmény-verifikációs mátrixának részlete.

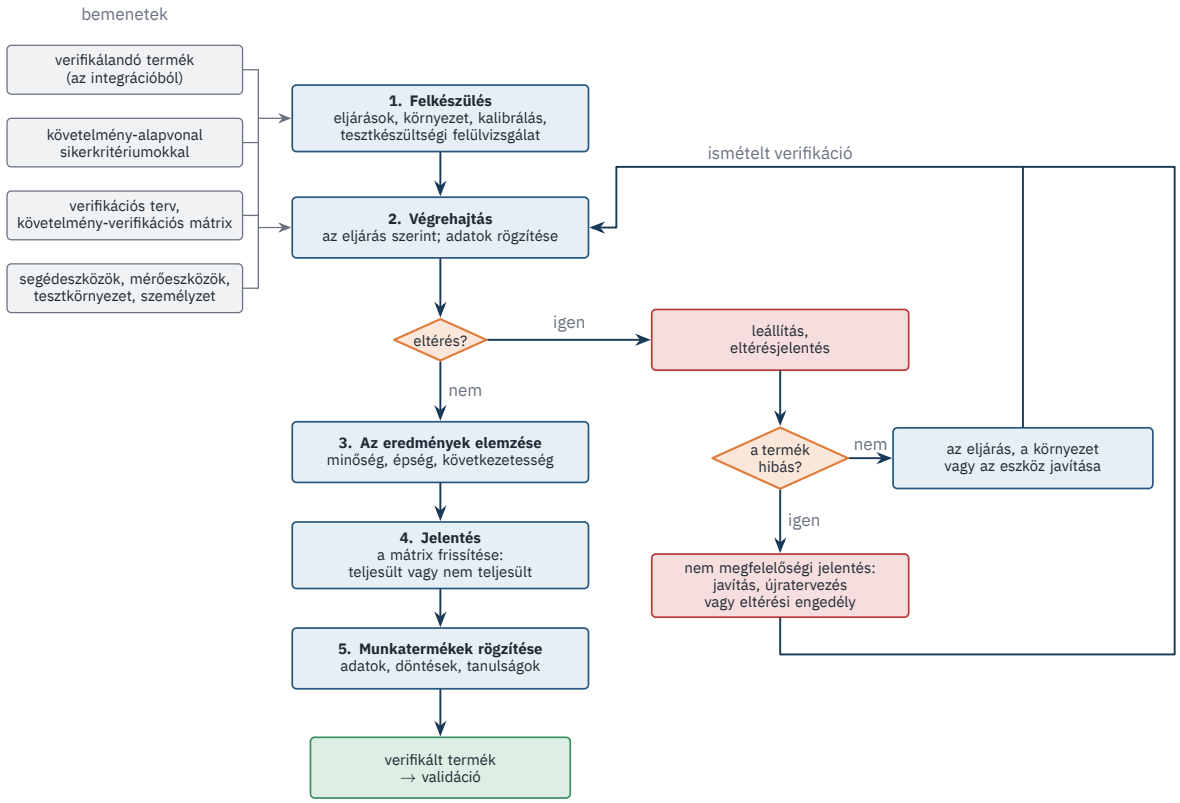
Az.	Követelmény	Sikerkritérium	Mód	Szint, helyszín	Átvétel?	Eredmény
R-01	ciklusidő ≤ 75 s	a háromműszakos próbaüzem műszakátlagaiból számolt átlag 95%-os egyoldali felső konfidenciahatára ≤ 75 s	teszt (előtte elemzés)	rendszer, SAT	igen	FAT: 77,4 s, nem teljesült; SAT: 74,1 s, felső határ 74,7 s, teljesült
R-02	havi rendelkezésre állás ≥ 95%	a megbízhatósági blokkdiagram és a tervezett állásidők alapján ≥ 95%	elemzés	rendszer	nem	teljesült; az üzemi adatok az első három hónapban megerősítendők
R-03	nyomaték 4,0 ± 0,2 Nm	a próbaüzem minden kötése a tűrésmezőn belül; legalább 2302 kötés hiba nélkül	teszt	csavarozó egység, FAT és SAT	igen	8096 kötés hiba nélkül, teljesült
R-05	törzshajlítás ≤ 20°	digitális emberi modell az 5. és a 95. percentilisre, majd próba három kezelővel	elemzés és bemutatás	munkahely, FAT	nem	teljesült
R-07	tűrésen kívüli kötés jelölése	20 szándékosan hibás kötésből (hiányzó, laza csavar) mind hibásként jelölve	teszt	rendszer, FAT	igen	teljesült
R-09	nyomaték-adatok a MES-ben	10 motor adatai a MES-ben a motor azonosítójához rendelve, egyezésben a vezérlő naplójával	bemutatás	rendszer és MES, FAT	igen	FAT: nem teljesült (I-07); javítás után teljesült
R-11	sebesség a védőzónában ≤ 250 mm/s	sebességmérés három pozícióban, próbatesttel a védőzónában	teszt	rendszer, FAT	igen	teljesült
R-15	elfér 6 × 4 m-en	helyszíni felmérés és elrendezési rajz	vizsgálat	üzem	nem	teljesült

A mátrix egyben a verifikáció állapotjelentése. Ha a verifikációs módot és a sikerkritériumot már a követelmények alapvonalba helyezésekor kitöltik, kiderül, mely követelmények nem verifikálhatók, és melyekhez kell drága segédeszköz vagy létesítmény. A verifikációt akkor tekintjük lezártnak, ha minden követelményre van dokumentált objektív bizonyí-

ték vagy jóváhagyott eltérési engedély, és minden eltérés- és nem megfelelőségi jelentés lezárult.

23.4. A verifikáció végrehajtása

Az ISO/IEC/IEEE 15288 a verifikációt három tevékenységre bontja: az előkészítésre, a végrehajtásra és az eredmények kezelésére (ISO/IEC/IEEE 2023, 6.4.9. szakasz). A gyakorlatban ezt öt lépésben követjük: felkészülés, végrehajtás, az eredmények elemzése, jelentés és a munkatermékek rögzítése (23.1. ábra).



23.1. ábra. A verifikáció folyamata és az eltérések kezelése. Az ISO/IEC/IEEE 15288 verifikációs folyamata (ISO/IEC/IEEE 2023, 6.4.9. szakasz) nyomán, saját szerkesztés.

Felkészülés. Összegyűjtik és jóváhagyják a verifikációs tervet és a követelményeket, előkészítik a terméket és a segédeszközöket, véglegesítik a lépésenkénti eljárásokat, és felkészítik a környezetet (létesítmény, eszközök, mérőeszközök, klimatikus feltételek). Ha a rendszerrel ember dolgozik, a verifikációban az embert megfelelően kell képviselni: testméret, képességek, tudás, képzettség, ruházat és eszközök szempontjából; az ilyen teszteket „ember a hurokban” (human-in-the-loop) tesztnek nevezik. A felkészülést jellemzően tesztkészültségi felülvizsgálat zárja, amely ellenőrzi, hogy a létesítmény, a személyzet, a mérőeszközök és a segédeszközök készen állnak-e.

A **verifikációs eljárás** és a **verifikációs jelentés** tartalmát előre rögzíteni kell; a jelentésre az európai űripari szabvány részletes mintát is ad (ECSS 2018). Az eljárásban szerepel többek között a vizsgált termék azonosítása és a tesztkonfiguráció eltérése az üzemitől, a célok és a sikerkritériumok, a mérendő jellemzők tűrésekkel, a lépések sorrendje, a mérőeszközök (méréshatár, pontosság), a kalibrálás rögzítése, a veszélyes helyzetek és a

biztonsági előírások, valamint az eltérések esetén követendő eljárás. A jelentés a célok teljesülését, az eltéréseket, az egyes lépések eredményeit, a nem megfelelőségek összefoglalását a javító intézkedésekkel, a következtetéseket és a ténylegesen végrehajtott eljárás (a feljegyzésekkel ellátott példány) másolatát tartalmazza.

Végrehajtás. A verifikáció a jóváhagyott eljárás szerint folyik; a vezető ellenőrzi, hogy az eljárást betartották, a mérőeszközöket helyesen kalibrálták, és az adatokat rögzítették. A verifikáció a termékhierarchia több szintjén zajlik; ha lehet, a végén a teljes működési láncot átfogó (end-to-end) teszt következik, amely működési forgatókönyvekkel igazolja az interfészek összeférhetőségét és a teljes funkcionalitást. A KoboCellánál ilyen teszt a motor útja a behelyezéstől a nyomatékadat MES-be kerüléséig.

Eltérések. Ha a teszt során eltérés jelentkezik, vagyis bármilyen különbség a megkövetelt vagy várt eredménytől, a verifikációt le kell állítani, és eltérésjelentést kell készíteni. Ezután azt kell kideríteni, hogy a termék nem felel meg, vagy az eljárás, a környezet, illetve a végrehajtás hibás. Termékhiba esetén nem megfelelőségi jelentés készül; a javítás a terv módosítását és ismételt verifikációt igényelhet, vagy, ha a követelmény enyhítése elfogadható, eltérési engedélyt (waiver) kérnek. A nem megfelelőségek a kockázatkezelés bemenetei is (14. fejezet).

Elemzés, jelentés, rögzítés. Az adatokat minőség, épség, helyesség, következetesség és érvényesség szempontjából elemzik, lehetőleg még akkor, amikor a tesztkonfiguráció érintetlen, hogy egy esetleges ismétlés gyors legyen. Az eredmények a verifikációs mátrixba kerülnek. Az eredmények és a kapcsolódó adatok (döntések és indoklásuk, feltételezések, tanulságok) rögzítése nagyon fontos, de gyakran elhanyagolt lépés.

> **ForgácsTech: a MES verifikációja a teljes adatúton**
>
> A ForgácsTech MES-ének egyik követelménye, hogy a gépállapot (üzemel, áll, átállás, hiba) változását legfeljebb 60 s késéssel jelezze, és a MES által számolt darabszám műszakonként legfeljebb 1%-kal térjen el a kézi számlálástól. A csapat a verifikációt a teljes adatúton végezte: a gép vezérlőjétől vagy a régebbi esztergák külső érzékelőitől az OEE-kijelzőig (22. fejezet). A késést 40 kiváltott állapotváltáson mérték, a darabszámot tíz műszakon át vetették össze a műszakvezetők számlálásával. A tesztek két régi esztergánál 2–3%-os darabszám-eltérést mutattak: az ajtókapcsoló a mérőhelyi ellenőrzéskor is jelzett, és a MES ezt új ciklusként számolta. Az eltérés oka nem a MES volt, hanem a jelfeldolgozás szabálya, amelyet a ciklusidő alsó korlátjával egészítettek ki. Az ismételt verifikáció után a követelmény teljesült.

23.5. Statisztikai alapok

A teszt mintát vesz a termék viselkedéséből. Ha a mért jellemző szór, a minta átlaga csak becslése a valódi átlagnak, és a döntésnek ezt a bizonytalanságot is figyelembe kell vennie. Kis mintákra a Student-féle t-eloszlásra épülő elemzés való, amely annak eldöntésére szolgál, hogy egy mintaátlag adott hibavalószínűség mellett egyezik-e egy előírt értékkel; feltétele, hogy a mért jellemző eloszlása ne térjen el lényegesen a normálistól. Ugyanerre épül a konfidenciaintervallum, amely megadja, milyen tartományban van a valódi átlag a választott valószínűséggel (Montgomery 2020; Goldberg és mtsai. 1994, 6.1. és 6.5. szakasz).

23.5.1. Konfidenciahatár egy felső korlátos követelményhez

Ha a követelmény felső korlát ($\mu \le U$), az átlag egyoldali felső konfidenciahatárát vetjük össze a korláttal:

$$\bar{x} + t_{1-\alpha,\,n-1}\,\frac{s}{\sqrt{n}} \le U, \tag{23.1}$$

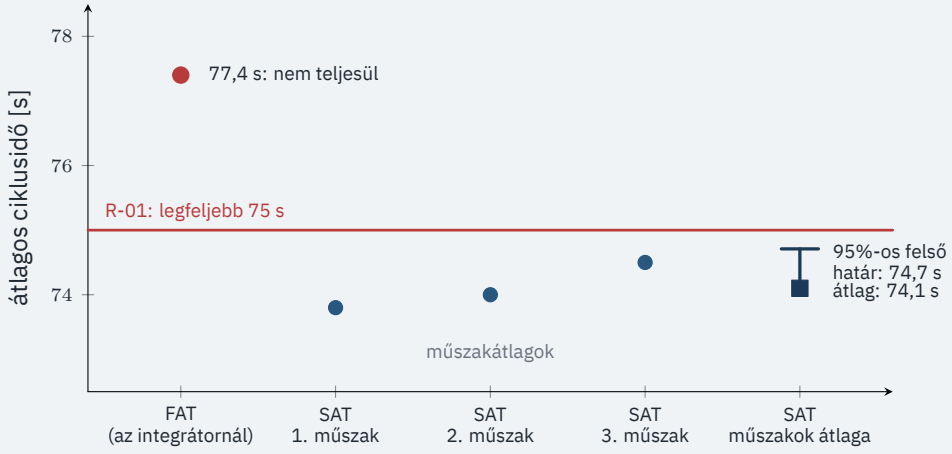
ahol $\bar{x}$ a mintaátlag, s a minta szórása, n a mintaelemszám, $t_{1-\alpha,\,n-1}$ a t-eloszlás kvantilise $n-1$ szabadsági fokkal. Ha az egyenlőtlenség teljesül, $1-\alpha$ megbízhatósággal állíthatjuk, hogy a valódi átlag a korlát alatt van. Ha csak a mintaátlag van a korlát alatt, a határ azonban nem, a teszt nem elég bizonyító erejű: vagy több mérés kell, vagy kisebb szórás.

KoboCella: teljesül-e a 75 s?

A helyszíni próbaüzem három műszakjában a MES minden ciklus idejét rögzítette. Az egymást követő ciklusok azonban nem függetlenek: egy lassabb kezelő vagy egy beállítási hiba több egymást követő ciklust érint. A csapat ezért óvatos megközelítést választott: a műszakokat tekintette független megfigyelésnek ($n = 3$). A műszakátlagok 73,8, 74,0 és 74,5 s voltak (23.2. ábra); az átlag $\bar{x} = 74{,}1$ s, a szórás $s = 0{,}36$ s. A $t_{0{,}95;\,2} = 2{,}92$ kvantilissel a 23.1. egyenlet szerinti felső határ

$$74{,}1 + 2{,}92 \cdot \frac{0{,}36}{\sqrt{3}} = 74{,}7\ \text{s} \le 75\ \text{s},$$

így az R-01 követelmény 95%-os megbízhatósággal teljesül. A gyári átvételen mért 77,4 s ezzel szemben egyértelműen a korlát fölött volt.



23.2. ábra. Az R-01 ciklusidő-követelmény verifikációja: a gyári átvétel mérése, a helyszíni próbaüzem műszakátlagai és az átlag 95%-os egyoldali felső konfidenciahatára.

23.5.2. Sikeres futás próba

Nagy megbízhatóságú követelményeknél (egy kötés legalább 0,999 valószínűséggel megfelelő) a hibák ritkák, így a mintában jellemzően nincs hibás darab. A **sikeres futás próba**

(success run) azt adja meg, hány egymást követő hibátlan próba igazolja az R megbízhatóságot C konfidenciaszinten. Ha n független próba mind sikeres, és a hibavalószínűség állandó, akkor

$$R^{n} \le 1 - C \quad \Longrightarrow \quad n \ge \frac{\ln(1-C)}{\ln R} \tag{23.2}$$

(O'Connor és Kleyner 2012). A KoboCella R-03 követelményéhez ($R = 0{,}999$, $C = 0{,}9$) $n \ge \ln 0{,}1 / \ln 0{,}999 \approx 2302$ hibátlan kötés szükséges, ami 288 motornak, körülbelül hat üzemórának felel meg. A háromműszakos próbaüzem 1012 motorján 8096 kötés készült hiba nélkül; ez 90%-os konfidenciaszinten $R \ge 0{,}1^{1/8096} \approx 0{,}9997$ megbízhatóságot igazol. Ha egyetlen hiba is előfordul, a szükséges próbaszám jelentősen nő, ezért a nagy megbízhatóságú követelményeket elemzéssel és alacsonyabb szintű tesztekkel együtt szokás verifikálni (16. fejezet).

Az első és a másodfajú hiba

A verifikációs döntés két hibát ejthet. Elsőfajú hiba, ha a megfelelő terméket elutasítjuk; másodfajú, ha a nem megfelelőt elfogadjuk. A konfidenciaszint az utóbbit korlátozza: a 95%-os felső határral legfeljebb 5% az esélye annak, hogy a ténylegesen 75 s feletti átlagú cellát megfelelőnek nyilvánítjuk. A szükséges mintaelemszám a két hiba megengedett valószínűségétől, a mért jellemző szórásától és attól függ, mennyire van a várt érték a korlát alatt (Montgomery 2020). A mintaelemszámot ezért a teszt megtervezésekor kell meghatározni, nem a mérés után.

23.6. Verifikáció modellekkel és a hierarchia szintjein

A verifikáció nem a rendszer elkészültével kezdődik. A koncepció és a tervezés szakaszában az elemzés és a szimuláció a fő eszköz, és a modellek maguk is verifikációt és validációt igényelnek (3.4. szakasz). A hardver a hurokban (hardware-in-the-loop) teszt a valós vezérlőt szimulált környezethez köti, így a vezérlőszoftver a mechanika elkészülte előtt verifikálható. A követelmények egy része csak alacsonyabb szinten verifikálható (a csavarozó orsó nyomatékpontossága a komponensen), más része csak a rendszer szintjén (a ciklusidő). A verifikációs terv ezért a mátrixban minden követelményhez a szintet is megadja.

VárosiBringa: a dokkoló reteszcsapjának verifikációja

A dokkoló reteszcsapjára vonatkozó követelmény a nagy terheléses események elviselése. A csapat a verifikációt három szinten végezte. Elemzéssel igazolta, hogy a terhelés és a szilárdság eloszlása alapján a várható tönkremenetel elfogadható (18. fejezet). Teszttel, tíz próbacsap tönkremenetelig terhelésével igazolta a nyírószilárdság várható értékét és szórását. Vizsgálattal pedig minden beérkező csapszállítmánynál ellenőrzi az anyagbizonylatot. Az első kettő a terv verifikációja és minősítése, a harmadik az átvétel része.

A teszt mint formalitás

Gyakori hiba, hogy a verifikációt a projekt végére hagyják, és úgy tekintenek rá, mint a kész termék „lepecsételésére". Ilyenkor a sikerkritériumokat utólag igazítják az eredményekhez, az eltéréseket „mérési hibának" minősítik, és a nem megfelelőségi jelentés helyett szóbeli megállapodás születik. A verifikáció célja a hibák megtalálása, amíg olcsón javíthatók; ezt csak előre rögzített sikerkritériumokkal, dokumentált eltéréskezeléssel és a teszt

végrehajtójának függetlenségével lehet elérni.

Összefoglalás

- A verifikáció azt igazolja, hogy a termék teljesíti a megadott követelményeket; a validáció azt, hogy megfelel az érdekelt felek elvárásainak és a működési koncepciónak.
- A minősítés a környezeti szélsőértékek melletti működést, az átvétel az egyes példányok gyártását, a tanúsítás a bizonyítékok hatósági vagy megrendelői elfogadását jelenti.
- A négy verifikációs mód az elemzés, a bemutatás, a vizsgálat és a teszt; a teszt adja a legrészletesebb adatot, de a legdrágább.
- A követelmény-verifikációs mátrix minden kötelező követelményhez megadja a sikerkritériumot, a módot, a szintet, a helyszínt, a felelőst és az eredményt; a sikerkritériumot a teszt előtt kell rögzíteni.
- Az eltérésnél a verifikációt le kell állítani, és el kell dönteni, hogy a termék vagy az eljárás hibás; a verifikáció akkor zárul, ha minden követelményre van bizonyíték vagy eltérési engedély, és minden eltérésjelentés lezárult.
- A mintavétel bizonytalanságát konfidenciahatárral, a nagy megbízhatóságú követelményeket sikeres futás próbával vagy elemzéssel kell kezelni.

Önellenőrzés

1. Miben különbözik a verifikációs és a validációs teszt?
2. Mi a különbség a minősítés és az átvétel között? Melyiket kell minden példányon elvégezni?
3. Milyen verifikációs módot választanál a következő követelményekhez: (a) a dokkoló kijelzőjén a feliratok magyar és angol nyelvűek; (b) a kerékpár akkumulátora 50 km hatótávot biztosít; (c) a kezelő a vészleállítót a munkahelyéről eléri; (d) a rendszer éves rendelkezésre állása legalább 99%?
4. Miért kell a sikerkritériumot a teszt előtt rögzíteni?
5. Egy teszt során a mért érték kívül esik a tűrésen. Milyen lépések következnek?
6. Egy követelmény szerint a válaszidő átlaga legfeljebb 2,0 s. Öt független mérés átlaga 1,8 s, szórása 0,2 s. Teljesül-e a követelmény 95%-os megbízhatósággal? ($t_{0{,}95;\,4} = 2{,}13$)
7. Hány hibátlan próba kell ahhoz, hogy 95%-os konfidenciaszinten 0,99-es megbízhatóságot igazoljunk?

Megoldás.

1. A verifikációs teszt a jóváhagyott követelményekre vezethető vissza, ellenőrzött környezetben, mérőeszközökkel, mérnökök és technikusok végzik. A validációs teszt a működési koncepcióra vezethető vissza, valószerű vagy szimulált körülmények között, jellemző felhasználókkal vizsgálja a termék alkalmasságát.
2. A minősítés azt igazolja, hogy a terv a környezeti szélsőértékek mellett is működik, egyszer, amíg a terv nem változik. Az átvétel azt igazolja, hogy az adott példány gyártása megfelel a tervnek; ezt minden példányon el kell végezni.
3. (a) vizsgálat; (b) teszt (vagy, a korai szakaszban, elemzés az akkumulátor és a fogyasztás adataiból); (c) bemutatás, a kezelők testméret-tartományát képviselő személyekkel; (d) elemzés, mert egy év üzem nem várható ki a verifikáció során, majd az üzemi adatokkal megerősítés.
4. Mert az eredmény ismeretében megfogalmazott kritérium az eredményhez igazodhat, és így a verifikáció nem bizonyít semmit. A kritérium határozza meg a mintaelemszámot és a körülményeket is, ezeket a teszt előtt kell megtervezni.
5. A verifikációt le kell állítani, és eltérésjelentést kell készíteni. Meg kell vizsgálni, hogy a termék hibás, vagy az eljárás, a környezet, a mérőeszköz. Termékhiba esetén nem megfelelőségi jelentés készül, javítás vagy eltérési engedély következik, majd ismételt verifikáció; nem termékhiba esetén az eljárást vagy a környezetet javítják, és megismétlik a tesztet.
6. A felső határ $1{,}8 + 2{,}13 \cdot 0{,}2/\sqrt{5} = 1{,}8 + 0{,}19 = 1{,}99$ s, ami kisebb 2,0 s-nál, így a követelmény (szűken) teljesül. A tartalék kicsi; egy újabb mérés vagy kisebb szórás erősítené a bizonyítékot.

7. $n \geq \ln 0{,}05 / \ln 0{,}99 \approx 298{,}1$, vagyis 299 hibátlan próba.

Projektlépés: verifikációs terv

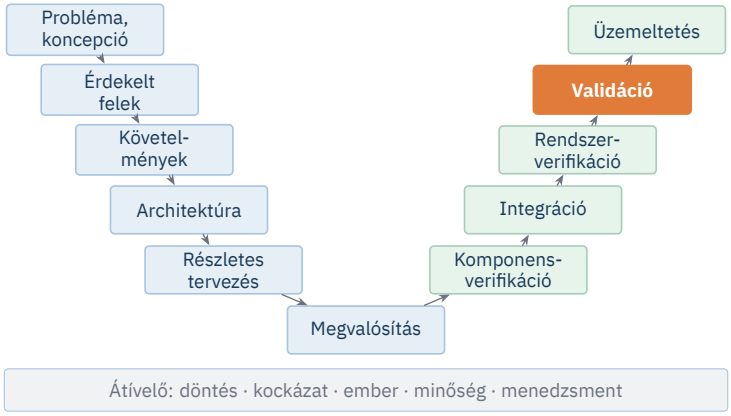
Saját fejlesztési feladatodhoz:

1. válassz ki legalább nyolc kötelező követelményt, köztük legalább egy teljesítmény-, egy interfész-, egy biztonsági és egy emberi tényezőkre vonatkozó követelményt;
2. készíts követelmény-verifikációs mátrixot: sikerkritérium, mód, szint, helyszín, átvételi követelmény-e, felelős;
3. írd meg egy teszt eljárását a 23.4. szakaszban leírt fő tartalommal (cél, konfiguráció, mérőeszközök, lépések, biztonság, eltérések kezelése);
4. egy mennyiségi követelménynél határozd meg a döntési szabályt konfidenciahatárral, és becsüld meg a szükséges mintaelemszámot;
5. használd a Gyakorlati útmutató követelmény-verifikációs mátrix eszközlapját.

Továbbolvasás

- Engel (2010): a verifikáció, a validáció és a tesztelés alapvető tankönyve, költség- és kockázati modellekkel.
- ECSS (2018): az európai űripar verifikációs szabványa a minősítés, az átvétel és a verifikációs ellenőrző dokumentum leírásával.
- ISO/IEC/IEEE (2023), 6.4.9. szakasz; IEEE (2024): a verifikáció folyamata a szabványokban.
- Montgomery (2020): a t-próba és a konfidenciaintervallum.
- O'Connor és Kleyner (2012): a megbízhatóság igazolása teszttel.
- NASA (2016), 5.3. szakasz és D függelék: a termékverifikáció folyamata, a verifikáció módjai, az eljárás és a jelentés tartalma, a követelmény-verifikációs mátrix.
- Goldberg és mtsai. (1994), 6.1. és 6.5. szakasz: a t-próba és a konfidenciaintervallum gyakorlati alkalmazása.

24. Validáció



Tanulási eredmények

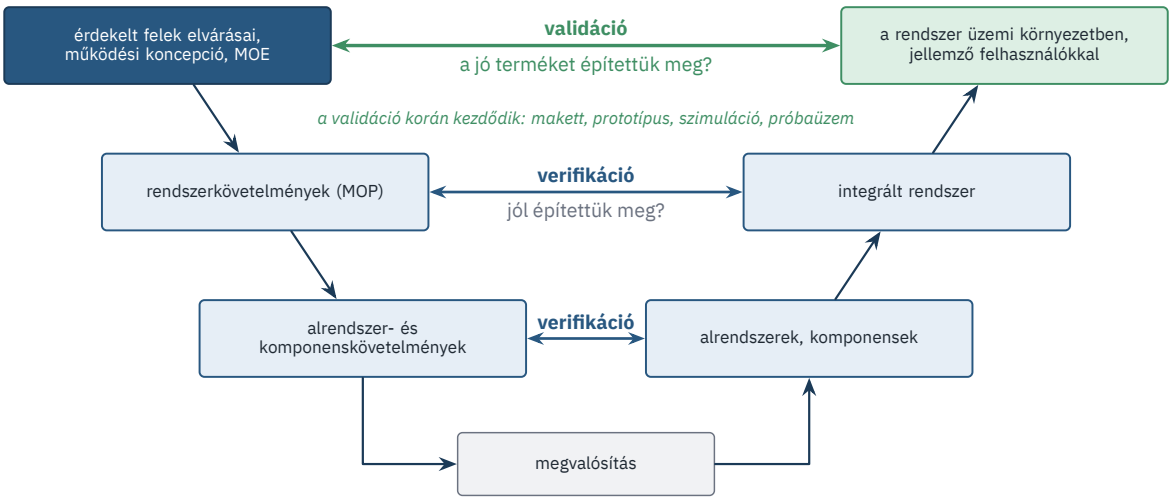
A fejezet feldolgozása után képes leszel:

- elmagyarázni, miben különbözik a validáció a verifikációtól, és miért kezdődik már a tervezés korai szakaszában;
- validációs tervet és validációs mátrixot készíteni az érdekelt felek elvárásaiból és a MOE-kból;
- megtervezni egy használhatósági tesztet, és értelmezni a hatékonyság, a hatásosság és az elégedettség mérőszámait;
- megbecsülni a résztvevők szükséges számát, és a mért arányokat konfidenciaintervallummal értékelni;
- felismerni, miért bukhat meg a validáción egy verifikáción átjutott rendszer.

A verifikáció azt igazolja, hogy a termék teljesíti a követelményeket. A követelmények azonban az érdekelt felek elvárásainak egy fordítását jelentik, és a fordítás lehet hiányos vagy téves. Boehm rövid kérdéseivel: a verifikáció azt bizonyítja, hogy *jól építettük-e meg* a terméket, a validáció azt, hogy *a jó terméket építettük-e meg* (B. W. Boehm 1984). A verifikáció objektív bizonyítékot ad arra, hogy a követelményekben szereplő minden „kell” teljesült. Az ISO/IEC/IEEE 15288 szerint a validáció pedig arra ad objektív bizonyítékot, hogy a rendszer használat közben teljesíti az üzleti vagy küldetési célokat és az érdekelt felek követelményeit, vagyis a tervezett környezetben betölti a rendeltetését (ISO/IEC/IEEE 2023, 6.4.11. szakasz; NASA 2016, 5.4. szakasz). A két folyamat viszonyát a 24.1. ábra mutatja.

24.1. A validáció célja és tárgya

A validáció tehát azt igazolja, hogy a végtermék kielégíti az érdekelt felek elvárásait (a MOE-kat) a tervezett üzemi környezetben, és lehetőség szerint a várható kezelők és felhasználók



24.1. ábra. A verifikáció a követelményekhez, a validáció az érdekelt felek elvárásaihoz és a működési koncepcióhoz méri a rendszert.

végzik. Az emberi rendszerintegráció szemléletében (11.2. szakasz) ennek két oldala van (Booher 2003; ISO 2019):

- **a rendeltetésszerű használat igazolása:** a termék a tervezett környezetben üzemeltetve betölti a rendeltetését; a rendszerstruktúra minden szintjén, a legalsó végterméktől a rendszerig;
- **az ember megfelelő integrálásának igazolása:** a felhasználói felület megfelel az emberi tényezők szempontjainak, a kezelők és a karbantartók elsajátították a szükséges készségeket, az utasítások elkészültek, a képzés megszervezése megtörtént, és a munkakörnyezet támogatja az egészséget és a biztonságot (11. fejezet).

A validáció módszerei ugyanazok, mint a verifikációé: elemzés, bemutatás, vizsgálat és teszt; a mérce azonban az érdekelt felek céljai, a működési koncepció és a MOE-k, nem a követelmények szövege (IEEE 2024). A validációs teszt a várható környezetre és működésre összpontosít, a minősítő teszt viszont a legrosszabb esetet jelentő terhelésekre és környezeti feltételekre (23. fejezet).

A validáció nem a fejlesztés végén kezdődik. A validációt a lehető legkorábban és iteratívan kell elvégezni, mert minél korábban derül ki az újratervezés szükségessége, annál olcsóbb (2.1. szakasz) (INCOSE 2023b). Az emberközpontú tervezés szabványa ugyanezt írja elő: a terveket a felhasználók bevonásával, a tervezés során többször kell értékelni (ISO 2019). A koncepció szakaszában a működési forgatókönyvek közös végiggondolása, később a makettek, a prototípusok és a szimulációk, a megvalósítás után a próbaüzem és az üzemi adatok szolgálnak validációs bizonyítékul.

24.2. A validációs terv és a validációs mátrix

A validációs terv forrásai a működési koncepció, az érdekelt felek szükségletei, céljai és célkitűzései, a követelmények indoklása, a korábbi projektek tanulságai, a rendszerarchitektúra modelljei, valamint a rendszermérnöki, az emberi rendszerintegrációs és a verifikációs-validációs tervek. A validáció eszközei sokfélék lehetnek: az érdekelt felek értékelése és visszajelzése, szakmai átvizsgálások, különböző hűségű fizikai modellek,

szimulációk és virtuális modellek, tesztek, illesztési próbák, az eljárások próbái és a felülvizsgálatokon kapott visszajelzések (Engel 2010).

A **validációs mátrix** minden validációs tevékenységhez megadja az azonosítót, a tevékenység leírását, a célt (mit kell az érdekelt félnek értékelnie), a módszert, a helyszínt, az életciklus-fázist, a felelős szervezetet és az eredményt igazoló dokumentumot. A verifikációs mátrixszal szemben itt nem a követelmények, hanem az érdekelt felek céljai és a MOE-k a sorok kiindulópontjai.

VárosiBringa: a validációs mátrix

A VárosiBringa három célkitűzése (5. fejezet) a validáció mércéje: C1 (csúcsidőben az állomások legalább 90%-án legalább két kerékpár), C2 (a második évben a belvárosi, 3 km-nél rövidebb autós utak aránya legalább 10%-kal csökkenjen) és C3 (a 65 év felettiek körében a sikeres első kölcsönzések aránya segítség nélkül legalább 85%). A 24.1. táblázat a validációs mátrix részlete. A tevékenységek a tervezés korai szakaszától az üzemeltetés második évéig tartanak.

24.1. táblázat. A VárosiBringa validációs mátrixának részlete.

Az.	Tevékenység	Cél	Módszer	Fázis	Felelős	Eredmény
V-1	a kölcsönző alkalmazás két prototípusának próbája 12 résztvevővel	a kölcsönzés minden felhasználói csoport számára érthető (C3)	teszt (használhatósági)	részletes tervezés	UX-csapat	SUS 64 → 79; feladatsiker 75% → 96%
V-2	az áthelyezési stratégia szimulációja a kölcsönzési adatokkal	a C1 elérhető a tervezett furgonszámmal	elemzés	részletes tervezés	rendszermérnök	93% a szimulációban
V-3	első kölcsönzés próbaállomáson 30, 65 év feletti résztvevővel	a C3 elérhetőségének előzetes értékelése	teszt	próbaüzem	UX-csapat, idősügyi tanács	26/30 sikeres (87%)
V-4	az állomások telítettségének mérése az első három hónapban	C1	teszt (üzemi adatok)	üzemeltetés	üzemeltető	86% → az áthelyezés módosítása után 92%
V-5	az első kölcsönzések naplója a 65 év feletti regisztrált felhasználóknál	C3	teszt (üzemi adatok)	üzemeltetés, első félév	üzemeltető	571/640 sikeres (89%)
V-6	forgalomszámlálás és háztartási felmérés	C2	teszt (terepi felmérés)	üzemeltetés, második év	önkormányzat	folyamatban

A táblázat két fontos tanulságot hordoz. A validáció egy része csak üzem közben végezhető el: a C2 hatásmutató a második év végéig nem igazolható, ezért a validációs terv az üzemeltetés első két évére is kiterjed (25. fejezet). A korai bizonyítékok (V-1–V-3) pedig nem igazolják a MOE teljesülését, de jelzik, jó irányba halad-e a terv, és olcsón lehetővé teszik a javítást.

24.3. Felhasználók bevonása és használhatósági teszt

A validáció leggyakoribb, a felhasználókat közvetlenül bevonó formája a **használhatósági teszt**. Az ISO 9241-11 szabvány szerint a használhatóság annak mértéke, hogy egy rendszert, terméket vagy szolgáltatást meghatározott felhasználók meghatározott célok elérésére mennyire hatásosan, hatékonyan és elégedetten használhatnak egy meghatározott használati környezetben (ISO 2018c). A három összetevőhöz jellemző mérőszámok tartoznak:

- **hatásosság:** a feladatok sikeres befejezésének aránya, a hibák száma és súlyossága;
- **hatékonyság:** a feladatra fordított idő és erőforrás;
- **elégedettség:** a felhasználók értékelése, például a System Usability Scale (SUS) kérdőívvel (Brooke 1996); a SUS-pontszámok átlaga a nagy adatbázisok alapján 68 körül van (Sauro és Lewis 2016).

A használhatósági tesztben a felhasználói népességet képviselő résztvevők a működési forgatókönyvekből levezetett, valószerű feladatokat hajtanak végre, miközben a moderátor megfigyeli őket, és rögzíti a sikert, az időt, a hibákat és az elakadásokat (Rubin és Chisnell 2008). A validáció felkészítésénél ügyelni kell arra, hogy a résztvevők megfelelően képviseljék a felhasználókat (testméret, készségek, tudás, képzettség, ruházat, eszközök), és ahol lehet, valódi végfelhasználókat kell bevonni (ISO 2019).

A tesztek két célt szolgálhatnak. A **formatív** teszt a tervezés közben a problémák feltárását célozza: mi akadályozza a felhasználót, és miért. Az **összegző** (szummatív) teszt a kész vagy közel kész termék mért teljesítményét veti össze a célértékekkel, például a C3 85%-os sikerarányával. A kettő eltérő mintaelemszámot kíván.

24.3.1. Hány résztvevő kell a problémák feltárásához?

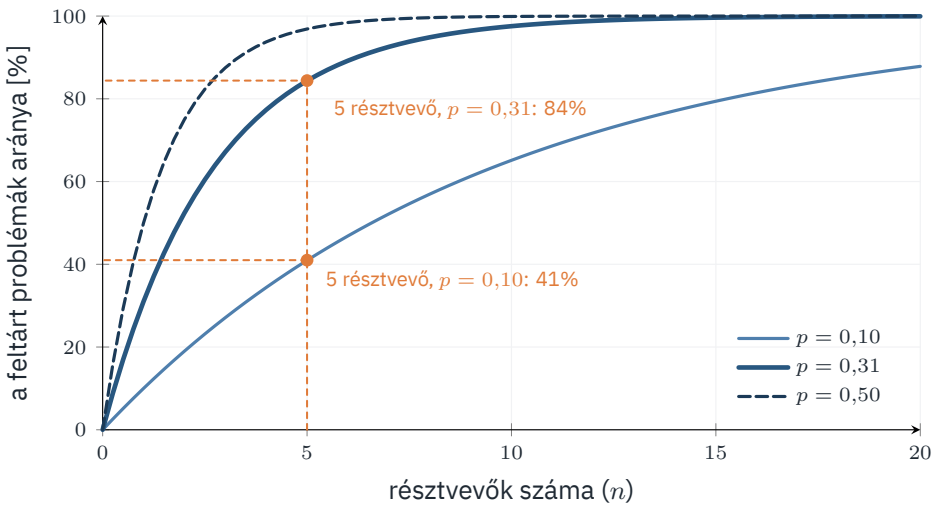
Nielsen és Landauer modellje szerint, ha egy probléma egy résztvevőnél p valószínűséggel jelentkezik, akkor n résztvevő a problémák

$$1 - (1 - p)^n \tag{24.1}$$

hányadát tárja fel (Nielsen és Landauer 1993). A szerzők által elemzett projektekben a p átlagosan 0,31 volt; ekkor öt résztvevő a problémák mintegy 84%-át találja meg (24.2. ábra). Ebből ered a gyakran idézett „öt résztvevő elég” szabály, amelyet Virzi korábbi vizsgálatai is alátámasztottak (Virzi 1992). A szabály azonban feltételes: ha a p kicsi, például mert a felhasználói csoportok eltérően viselkednek, vagy a problémák csak ritka helyzetekben jelentkeznek, öt résztvevő a problémák kisebbik felét sem tárja fel. Sauro és Lewis ezért azt javasolja, hogy a p-t a saját tesztadatokból becsüljük, és vegyük figyelembe, hogy kis mintán a becslés felfelé torzít (Sauro és Lewis 2016). Több felhasználói csoport esetén csoportonként kell résztvevőket bevonni.

24.3.2. Arányok és konfidenciaintervallum

Az összegző tesztben a sikerarány is mintából származó becslés. Kis mintán a normális közelítésen alapuló („Wald-féle”) intervallum megbízhatatlan; Agresti és Coull a módosított Wald-intervallumot javasolja, amely x sikerre n próbából



24.2. ábra. A feltárt problémák aránya a résztvevők számának függvényében (Nielsen és Landauer 1993 modellje alapján).

$$\tilde{p} = \frac{x + z^2/2}{n + z^2}, \qquad \tilde{p} \pm z\sqrt{\frac{\tilde{p}(1-\tilde{p})}{n + z^2}}, \tag{24.2}$$

ahol 95%-os szinten $z = 1{,}96$ (Agresti és Coull 1998). Sauro és Lewis a használhatósági tesztek sikerarányaira ezt az intervallumot ajánlja (Sauro és Lewis 2016).

VárosiBringa: elég-e 26 sikeres első kölcsönzés 30-ból?

A próbaállomáson (V-3) a 30, 65 év feletti résztvevő közül 26 segítség nélkül kölcsönzött elsőre, ami 87%, vagyis a mintaarány meghaladja a 85%-os célt. A 24.2. egyenlet szerinti 95%-os intervallum azonban 70–95%: a 30 fős minta nem elég annak igazolására, hogy a népességben a sikerarány legalább 85%. A csapat ezért a próbát formatív eredménynek tekintette: a négy sikertelen résztvevő közül három a dokkolón a zár kioldására szolgáló gombot kereste, ezért a kioldás menetét a dokkoló kijelzőjén nagyobb betűs, hangos útmutatóval egészítették ki. A C3 teljesülését az üzemi napló (V-5) igazolta: az első félévben a 65 év feletti regisztrált felhasználók 640 első kölcsönzéséből 571 sikerült ügyfélszolgálati segítség nélkül (89%; 95%-os intervallum 87–91%).

KoboCella: validáció a kezelőkkel

A KoboCella verifikációja igazolta a ciklusidőt, a nyomatékot és a biztonsági funkciókat (23. fejezet). A validáció azt kérdezte, hogy a cella a valós műszakrendben, a valós kezelőkkel betölti-e a rendeltetését: tartja-e az évi 140 ezer motor ütemét, és a kezelők munkája valóban könnyebb-e. A helyszíni átvétel utáni két hétben mind a hat betanított kezelő legalább három műszakot dolgozott a cellánál. Az első héten a kezelők a hibaelhárítási forgatókönyvekben (elakadt csavar, hiányzó fedél) a nyugtázó gomb és a hibaüzenet kijelzőjének távolsága miatt gyakran a cella mögé mentek; a csapat a nyugtázást a kezelői oldalra helyezte. A második héten a kezelők átlagos SUS-pontszáma 74, a NASA-TLX fizikai terhelési összetevője a korábbi kézi szerelésénél lényegesen kisebb volt (11. fejezet). A heti termelés a tervezett ütemet elérte.

24.4. Validáció üzem közben

Sok MOE csak a rendszer üzemelése során mérhető: a VárosiBringa elérhetősége és hatása, a ForgácsTech határidő-tartása. Az ilyen MOE-k validációja az átadás után folytatódik, és a validációs tervnek meg kell határoznia, milyen adatokból, mennyi ideig és ki értékeli az eredményt (25. fejezet). Az üzemi adatokkal végzett validáció előnye a nagy mintaelemszám és a valós környezet, hátránya, hogy a feltárt hiányosság javítása már drága. A validáció addig nem tekinthető befejezettnek, amíg minden feltárt probléma és intézkedés le nem zárult.

24.5. Ha a verifikáció sikeres, de a validáció nem

Különösen kellemetlen az a helyzet, amikor a rendszer sikeresen átjut a verifikáción, de a validáció egy kritikus szakaszában megbukik. Ez késést, jelentős átdolgozást és az érdekelt felekkel kötött kompromisszumokat okozhat. A megelőzés kulcsa a korán kidolgozott és a követelmények és a tervezés során folyamatosan finomított működési koncepció, a felhasználók bevonásának tisztázása az emberi rendszerintegrációs tervben, valamint a gyakori, iteratív kommunikáció az érdekelt felekkel. Arra is ügyelni kell, hogy a validációs hiányosságok javítása nem ütközhet az érdekelt felek elvárásaival azok előzetes egyeztetése nélkül, és a javítás nem okozhat új hibát a korábban megfelelően működő részekben; ennek eszköze a korábbi átvételi tesztek megismétlése (regressziós teszt).

A validáció sikertelenségének nem mindig a terv az oka. Három okcsoportot érdemes megkülönböztetni: a validáció rossz végrehajtását (hiányzó vagy hibás segédeszközök, betanítatlan kezelők, be nem tartott eljárások, nem kalibrált eszközök, nem megfelelő környezet), a verifikáció hiányosságát, valamint modellek és szimulációk esetén a hibás kezdeti és peremfeltételeket, a pontatlan egyenleteket és a nem megfelelő felbontást (3.4. szakasz). Csak ezek kizárása után állítható, hogy a terv nem felel meg az elvárásoknak.

ForgácsTech: a verifikált MES és a 89%-os határidő-tartás

A ForgácsTech MES-e minden követelményt teljesített a verifikáción (23. fejezet). A siker kritériuma azonban a problémamegfogalmazás szerint az volt, hogy a rendelések legalább 95%-a teljesüljön határidőre három egymást követő hónapban (4. fejezet). Az első három hónapban az arány 92, 87 és 89% lett: jelentős javulás a kiinduló 62%-hoz képest, de a cél alatt. Az elemzés kimutatta, hogy a műszakvezetők az ütemező átsorolási javaslatainak 41%-át módosították. A naplózott indoklásokból kiderült, hogy az ütemező nem tudta, melyik kezelő jogosult az öttengelyes megmunkáló központ kezelésére, ezért javaslatai egy része a gyakorlatban végrehajthatatlan volt (12. fejezet). Ez az információ egyik követelményben sem szerepelt, mert a feltáró interjúkban senki nem említette: a műszakvezetők számára magától értetődő volt. A csapat a kezelők jogosultsági mátrixát az ütemező korlátai közé vette, az érintett modulokon megismételte a korábbi átvételi teszteket, és a következő három hónapban az arány 95,6, 96,1 és 95,2% lett. A MOE teljesült.

Validáció és tanúsítás szabályozott területeken

Az orvostechnikai eszközök, a járművek és a vasúti rendszerek területén a validációt jogszabályok és harmonizált szabványok írják elő. Ilyenkor a validáció bizonyítékai a megfelelőségértékelés részei, és a tervezési dokumentáció mellett a hatóság vagy a kijelölt tanúsító szervezet vizsgálja őket. Az orvostechnikai eszközöknél a használhatósági mérnöki folyamatot, benne a használhatóság összegző értékelését, külön szabvány írja elő, mert a használati hibák a betegre veszélyesek lehetnek (IEC 2015). A rendszermérnöki elv ugyanaz, mint a

fejezet többi részében: a validáció tervét korán kell elkészíteni, és a működési koncepcióra és a felhasználókra kell építeni.

Validáció a fejlesztőkkel

Gyakori hiba, hogy a „felhasználói próbát” a fejlesztőcsapat tagjai vagy a rendszert jól ismerő kollégák végzik. Ők tudják, hol kell keresni a gombot, mit jelent a hibaüzenet, és milyen sorrendben kell végrehajtani a lépéseket, ezért nem akadnak el ott, ahol a valódi felhasználó elakadna. A validáció résztvevőinek a felhasználói népességet kell képviselniük, beleértve azokat a csoportokat is (idősek, gyengénlátók, alkalmi felhasználók, új kezelők), amelyek a fejlesztők között nem fordulnak elő.

Összefoglalás

- A verifikáció azt igazolja, hogy a terméket jól készítettük el, a validáció azt, hogy a jó terméket készítettük el: a mérce az érdekelt felek céljai, a működési koncepció és a MOE-k.
- A validációt a tervezett környezetben, lehetőleg valódi felhasználókkal végzik, és az ember megfelelő integrálását is igazolja.
- A validációt korán és iteratívan kell végezni: makettel, prototípussal, szimulációval, próbaüzemmel; egyes MOE-k csak üzem közben igazolhatók.
- A validációs mátrix minden tevékenységhez megadja a célt, a módszert, a fázist, a felelőst és az eredményt.
- A használhatóság hatásosságból, hatékonyságból és elégedettségből áll; a problémák feltárásához kevés, a sikerarány igazolásához sok résztvevő kell, az arányokat módosított Wald-intervallummal értékeljük.
- A verifikáción átjutott rendszer is megbukhat a validáción, ha a követelmények hiányosan fordították le az elvárásokat; a megelőzés kulcsa a jó működési koncepció és a felhasználók folyamatos bevonása.

Önellenőrzés

1. Fogalmazd meg egy-egy mondatban a verifikáció és a validáció kérdését!
2. Milyen két oldala van a validációnak az emberi rendszerintegráció szemléletében?
3. Miben különbözik a validációs teszt a minősítő teszttől?
4. Miért kezdődik a validáció már a tervezés korai szakaszában? Milyen eszközei vannak ekkor?
5. Egy alkalmazás tesztjén a problémák egy résztvevőnél átlagosan 0,2 valószínűséggel jelentkeznek. Hány százalékukat tárja fel öt, illetve tíz résztvevő?
6. Egy próbán 20 résztvevőből 18 sikeres. Számítsd ki a módosított Wald-intervallumot! Igazolja-e ez, hogy a sikerarány legalább 80%?
7. Sorolj fel három okot, amelyek miatt egy validáció sikertelen lehet, pedig a terv megfelelő!
8. Mit tanít a ForgácsTech esete a követelmények feltárásáról?

Megoldás.

1. Verifikáció: jól építettük-e meg a terméket, vagyis teljesíti-e a követelményeket? Validáció: a jó terméket építettük-e meg, vagyis kielégíti-e az érdekelt felek elvárásait a tervezett környezetben?
2. Annak igazolása, hogy a termék a tervezett környezetben betölti a rendeltetését, és hogy az ember megfelelően integrálódott a rendszerbe (felület, készségek, utasítások, képzés, munkakörnyezet).
3. A validációs teszt a várható környezetre és működésre összpontosít, a minősítő teszt a legrosszabb esetet jelentő terhelésekre és környezeti feltételekre.
4. Mert minél korábban derül ki, hogy a terv nem felel meg az elvárásoknak, annál olcsóbb a javítás. Eszközei: a működési forgatókönyvek közös végiggondolása az érdekelt felekkel, makettek, prototípusok, szimulációk, virtuális modellek, felülvizsgálati visszajelzések.
5. $1 - 0{,}8^5 = 0{,}67$, vagyis 67%; $1 - 0{,}8^{10} = 0{,}89$, vagyis 89%.

6. $\tilde{p} = (18 + 1{,}92)/(20 + 3{,}84) = 0{,}836$; a félszélesség $1{,}96\sqrt{0{,}836 \cdot 0{,}164/23{,}84} = 0{,}149$; az intervallum 69–98%. Nem igazolja, mert az alsó határ 80% alatt van, bár a mintaarány (90%) fölötte.
7. Hiányzó vagy hibásan működő segédeszközök, betanítatlan kezelők, be nem tartott eljárás, nem kalibrált eszközök, nem megfelelő környezet; a verifikáció hiányossága; modelleknél hibás kezdeti vagy peremfeltételek.
8. Azt, hogy a magától értetődőnek tartott, ki nem mondott tudás (a kezelők jogosultsága) könnyen kimarad a követelményekből, és csak a valós használatban derül ki. A feltárásnál a munka megfigyelése és a forgatókönyvek közös végigjátszása segít, az üzem közbeni validáció pedig felszínre hozza az ilyen hiányokat.

Projektlépés: validációs terv

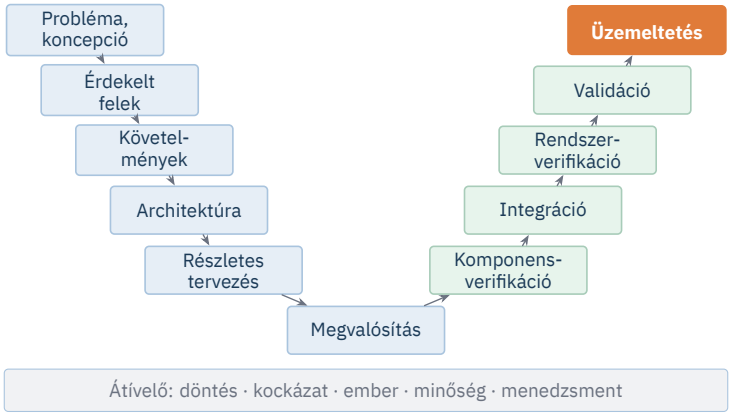
Saját fejlesztési feladatodhoz:

1. gyűjtsd össze a MOE-kat és a működési forgatókönyveket, és jelöld, melyik érdekelt csoportot érintik;
2. készíts validációs mátrixot legalább négy tevékenységgel, köztük legalább egy korai (makett, prototípus, szimuláció) és egy üzem közbeni tevékenységgel;
3. tervezz egy használhatósági tesztet: résztvevők csoportonként, feladatok sikerkritériummal és célidővel, mérőszámok (siker, idő, SUS);
4. becsüld meg a résztvevők számát a problémák feltárásához és a sikerarány igazolásához, és indokold a különbséget;
5. használd a Gyakorlati útmutató használhatósági teszt eszközlapját.

Továbbolvasás

- Engel (2010): a validáció tervezése és eszközei a verifikációval és a teszteléssel együtt.
- ISO/IEC/IEEE (2023), 6.4.11. szakasz; ISO (2018c); ISO (2019): a validáció, a használhatóság és az emberközpontú értékelés a szabványokban.
- Rubin és Chisnell (2008); Sauro és Lewis (2016): a használhatósági tesztek tervezése és statisztikai értékelése.
- Nielsen és Landauer (1993); Virzi (1992); Agresti és Coull (1998): a résztvevők száma és az arányok konfidenciaintervalluma.
- NASA (2016), 5.4. szakasz és E függelék: a termékvalidáció folyamata, céljai, a sikertelen validáció okai és a validációs mátrix.

25. Átadás, üzemeltetés és folyamatos fejlesztés



Tanulási eredmények

A fejezet feldolgozása után képes leszel:

- megtervezni egy rendszer átadását és üzembe helyezését, és megnevezni az átadás lezárásának feltételeit;
- megkülönböztetni a karbantartás fő fajtáit, és a rendszer jellemzőihez illő karbantartási stratégiát választani;
- alkalmazni a minőségügy alapeszközeit (Pareto-diagram, ok-okozati diagram, hisztogram, kontrollkártya) üzemi problémák elemzésére;
- kontrollkártyát készíteni és értelmezni, és kiszámolni a folyamatképességi indexeket;
- PDCA-, 8D- vagy DMAIC-ciklusban vezetni egy fejlesztést, és üzemi adatokból trendeket felismerni;
- már a tervezéskor figyelembe venni a rendszer életciklusának végét.

A V-modell jobb ága nem ér véget a validációval. A rendszert át kell adni, üzembe kell helyezni, üzemeltetni és karban kell tartani, a teljesítményét követni és javítani kell, végül ki kell vonni a használatból. Az üzemeltetés célja, hogy a rendszer kielégítse azt a szükségletet, amelyre létrehozták, és ezt tartósan fenntartsa. A továbbfejlesztés is ide tartozik, de csak addig, amíg nem igényli az architektúra jelentős megváltoztatását: az ilyen mértékű változás már új szükségletet jelent, és új fejlesztési ciklust indít. Az ISO/IEC/IEEE 15288 ezeket a tevékenységeket négy műszaki folyamatban írja le: átadás, üzemeltetés, karbantartás és kivonás (ISO/IEC/IEEE 2023, 6.4.10., 6.4.12–6.4.14. szakasz).

25.1. Átadás és üzembe helyezés

Az ISO/IEC/IEEE 15288 szerint az átadás célja, hogy a rendszer a tervezett üzemi környezetben képes legyen nyújtani az érdekelt felek követelményeiben rögzített szolgáltatásokat

(ISO/IEC/IEEE 2023, 6.4.10. szakasz; NASA 2016, 5.5. szakasz). Az átadásnak két formája van: az alacsonyabb szintű termék átadása integrálásra a rendszerstruktúra következő szintjén, illetve a végtermék átadása a felhasználónak, aki a tervezett üzemi környezetben használja. A rendszermérnök felelőssége, hogy csak megfelelően tesztelt, verifikált és validált termék kerüljön át a következő szintre.

Az átadás fő tevékenységei (ISO/IEC/IEEE 2023, 6.4.10. szakasz):

- **felkészülés:** a termék, a személyzet és a segédeszközök készenlétének értékelése; a termékkel együtt átadandó dokumentáció (kezelési, telepítési és karbantartási utasítások, a konfigurációs alapvonal, a verifikáció és a validáció bizonyítékai); a személyzet szükséges képzése;
- **a helyszín előkészítése:** helyszíni felmérés, a létesítmények szükséges átalakítása, a logisztikai támogatás, a felhasználók, kezelők és karbantartók képzése; a helyszíni felmérést már az életciklus korai szakaszában érdemes elvégezni, mert a helyszín adottságai a tervet is befolyásolják;
- **a termék előkészítése és szállítása:** csomagolás, kezelés, tárolás, a sérülések elleni védelem;
- **az átadás és a telepítés:** jóváhagyott telepítési eljárással, majd funkcionális és átvételi teszttel, amely igazolja, hogy a szállítás nem okozott sérülést, és a termék támogatásra kész;
- **a munkatermékek rögzítése:** helyszíntervek, kezelési eljárások, képzési anyagok, tanulságok.

A felhasználónak történő átadás akkor zárul le, ha a termék a helyszínen telepítve van, az átvételi és tanúsítási tevékenységek befejeződtek, a felhasználók, kezelők és karbantartók képzése megtörtént, és az átadást átvételi dokumentum zárja.

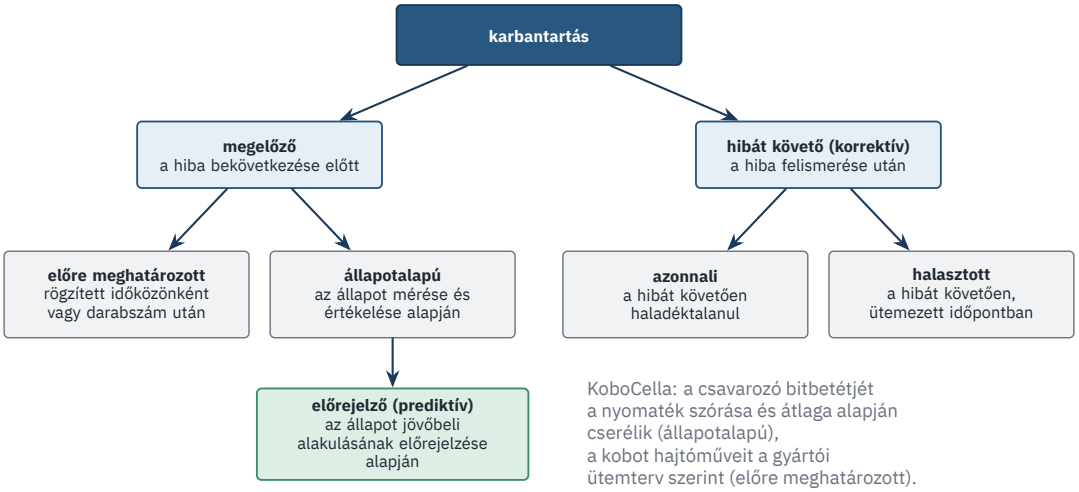
> **ForgácsTech: bevezetés a termelés leállítása nélkül**
>
> A ForgácsTech problémamegfogalmazása korlátként rögzítette, hogy a bevezetés alatt a termelés egy hétnél tovább nem állhat le (4. fejezet). A csapat ezért szakaszos bevezetést tervezett. Az első két hétben a MES párhuzamosan futott a papíralapú heti ütemtervvel: a műszakvezetők a megszokott módon dolgoztak, de minden átsorolást a MES-ben is rögzítettek, és naponta összevetették a két ütemtervet. A gépeket csoportonként, hétvégén kötötték be. A harmadik héten a papíralapú ütemterv megszűnt. Az átadás része volt a műszakvezetők és a kezelők képzése, a kezelési útmutató, a hibaelhárítási eljárás (mi a teendő, ha a MES nem elérhető) és a felelősök kijelölése az üzemeltetés első három hónapjára.

25.2. Üzemeltetés és karbantartás

Az üzemeltetés és a karbantartás jellemző tevékenységei (ISO/IEC/IEEE 2023, 6.4.12–13. szakasz): a pótalkatrész-terv végrehajtása, a műszaki adatok gyűjtése, a cserekezelők és -karbantartók képzése, az üzemeltetési és karbantartási naplók vezetése, a rendszer karbantartása és korszerűsítése, a kockázatok azonosítása és frissítése, a hiba- és problémajelentések kezelése, a tanulságok rögzítése, valamint a kivonás előkészítése.

A **karbantartás** az EN 13306 szabvány szerint két fő ágra bomlik (25.1. ábra) (CEN 2017). A **megelőző karbantartást** a hiba bekövetkezése előtt végzik: előre meghatározott időközönként vagy darabszám után, illetve az állapot mérése és értékelése alapján (állapotalapú karbantartás). Az állapotalapú karbantartás egyik formája az **előrejelző** (prediktív) karbantartás, amely az állapot jövőbeli alakulását is előre jelzi. A **hibát követő** (korrektív)

karbantartás a hiba felismerése után állítja helyre a működőképességet, azonnal vagy halasztva.



25.1. ábra. A karbantartás fajtái az EN 13306 szerint (CEN 2017 alapján), KoboCella-példával.

A választás a meghibásodás jellegétől és következményeitől függ. Ha a meghibásodási ráta nem nő az idővel (16. fejezet), az időalapú csere nem csökkenti a hibák számát, csak a jó alkatrészeket cseréli le. Ha a romlás mérhető (kopás, rezgés, hőmérséklet, a nyomaték szórása), az állapotalapú karbantartás a hibák előtt beavatkozik, de csak a szükséges helyen. A karbantartás a rendelkezésre állást és az OEE-t közvetlenül befolyásolja (19. fejezet), ezért a karbantartási stratégia a rendszer teljesítménykövetelményeinek része.

A rendszer üzemi **teljesítménykövetése** a fejlesztés során használt mérőszámok folytatása: a MOE-k és a TPM-ek (6. fejezet) az üzemben mutatószámokká válnak. A KoboCellánál a ciklusidő, a rendelkezésre állás és a nyomaték, a VárosiBringánál az állomások telítettsége és a kölcsönzések száma, a ForgácsTechnél a határidő-tartás és az OEE.

25.3. A minőségügy alapeszközei

Az üzemi problémák többsége egyszerű eszközökkel elemezhető. Ishikawa hét alapeszközt javasolt, amelyekkel szerinte az üzemi minőségi problémák túlnyomó része megoldható (Ishikawa 1982); Montgomery a statisztikai folyamatszabályozás „nagy hetesének” nevezi őket (25.1. táblázat) (Montgomery 2020).

25.1. táblázat. A minőségügy hét alapeszköze (Ishikawa 1982; Montgomery 2020).

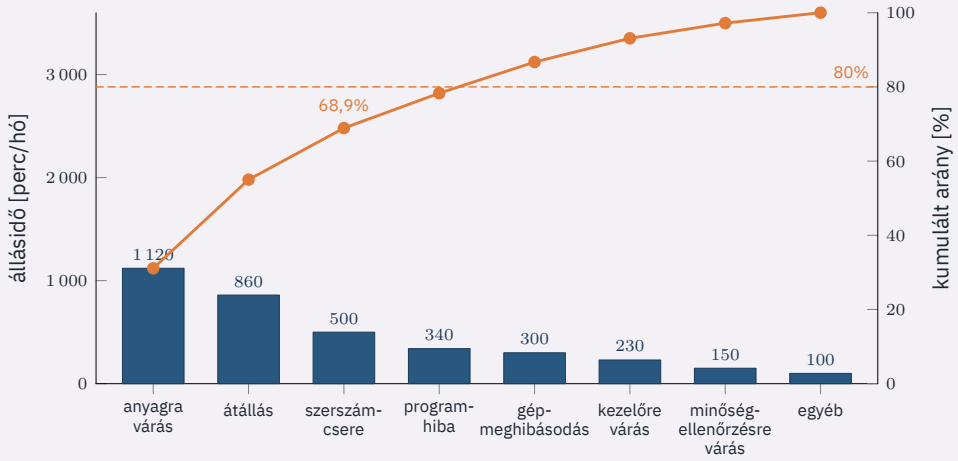
Eszköz	Mire jó?
ellenőrzőlap	az adatok egyszerű, egységes gyűjtése (hol, mikor, milyen hiba)
hisztogram	egy jellemző eloszlásának megjelenítése; a tűrésmezőhöz viszonyítva a képesség szemléltetése
Pareto-diagram	a problémák vagy okok fontossági sorrendje; a „kevés lényeges” kiválasztása
ok-okozati (halszálka-) diagram	egy probléma lehetséges okainak rendszerezése

Eszköz	Mire jó?
rétegzés	az adatok csoportosítása (gép, műszak, anyag) a különbségek feltárására
szórásdiagram	két változó kapcsolatának vizsgálata
kontrollkártya	a folyamat időbeli stabilitásának figyelése; a különleges okok jelzése

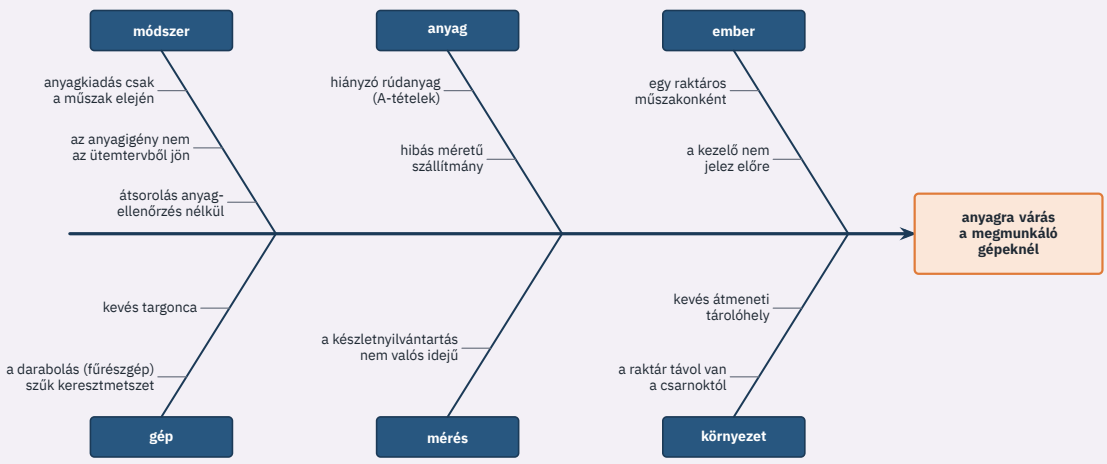
A **Pareto-diagram** az okokat gyakoriságuk vagy hatásuk szerint csökkenő sorrendben ábrázolja, a kumulált aránnyal együtt. A tapasztalat szerint a hatás nagy részét az okok kis része okozza; ez a „kevés lényeges és sok kevésbé lényeges” elve (Montgomery 2020). A Pareto-diagram így a problémák rangsorolására és a fejlesztési erőfeszítések összpontosítására szolgál. Az **ok-okozati diagram** egy probléma lehetséges okait fő csoportok (gyakran: ember, gép, módszer, anyag, mérés, környezet) szerint rendezi; a csapatmunkát segíti, de a lehetséges okok közül a valódit adatokkal kell igazolni (Ishikawa 1982).

ForgácsTech: mi okozza az állásidőt?

A MES bevezetése után a gépek állásidejének túlnyomó része okhoz rendelve került rögzítésre, ami a siker egyik kritériuma volt (4. fejezet). Egy hónap adataiból a csapat Pareto-diagramot készített (25.2. ábra): a 3600 perc állásidőből 31% anyagra várás, 24% átállás, 14% szerszámcsere volt; a három ok együtt az állásidő 69%-át adta. Az átállás csökkentésén már dolgoztak (19. fejezet), ezért az anyagra várást vizsgálták tovább ok-okozati diagrammal (25.3. ábra). A hat okcsoportból a csapat két lehetséges gyökérokot tartott valószínűnek, és a MES-naplóval ellenőrizte őket: a várakozások 64%-a egy rendelés átsorolása után keletkezett, mert az anyagot csak a műszak elején adták ki, az átsorolt rendeléshez tehát nem volt kikészítve anyag.



25.2. ábra. A ForgácsTech gépeinek havi állásideje okok szerint (Pareto-diagram).



25.3. ábra. Az anyagra várás lehetséges okai (ok-okozati diagram); pirossal az adatokkal igazolt gyökérok.

25.4. Statisztikai folyamatszabályozás

Minden folyamat szór. Shewhart megkülönböztette a **véletlen** (közös) okokat, amelyek a folyamat természetes ingadozását adják, és a **különleges** (azonosítható) okokat, amelyek a folyamat megváltozását jelzik (Shewhart 1931). A folyamat akkor „szabályozott”, ha csak véletlen ingadozás van jelen; a statisztikai folyamatszabályozás célja, hogy a különleges okokat még azelőtt felismerje, mielőtt hibás termék keletkezne (Montgomery 2020).

A **kontrollkártya** a mérési eredményeket (vagy részcsoportok átlagát) időrendben ábrázolja, a középvonallal és a ±3σ-nál húzott beavatkozási határokkal. Részcsoportok esetén a szórást az átlagos terjedelemből becsüljük:

$$\hat{\sigma} = \frac{\bar{R}}{d_2}, \qquad \text{FEH, AEH} = \bar{\bar{x}} \pm 3\,\frac{\hat{\sigma}}{\sqrt{n}} = \bar{\bar{x}} \pm A_2 \bar{R}, \tag{25.1}$$

ahol $\bar{R}$ a részcsoport-terjedelmek átlaga, n a részcsoport mérete, d_2 és $A_2 = 3/(d_2\sqrt{n})$ táblázatos állandók ($n = 5$ esetén $d_2 = 2{,}326$, $A_2 = 0{,}577$) (Montgomery 2020). A határokat egy stabil időszak adataiból számoljuk (I. szakasz), és ezekkel figyeljük a folyamatot (II. szakasz).

A különleges ok jelei az úgynevezett Western Electric-szabályok szerint (Montgomery 2020; Goldberg és mtsai. 1994, 7.14. szakasz): (1) egy pont a ±3σ határon kívül; (2) három egymást követő pontból kettő ugyanazon az oldalon a 2σ-n túl; (3) öt egymást követő pontból négy ugyanazon az oldalon az 1σ-n túl; (4) nyolc egymást követő pont a középvonal ugyanazon oldalán. A szabályok a határon kívüli pont előtt is jelezhetik a folyamat elmozdulását.

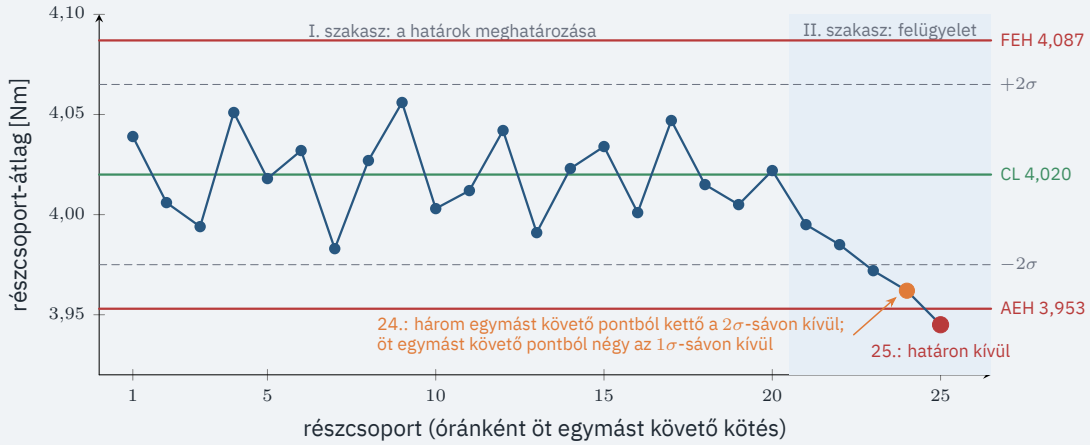
A kontrollkártya a folyamat stabilitásáról szól, nem arról, hogy a termék megfelel-e a tűrésnek. Ezt a **folyamatképességi indexek** mutatják:

$$C_p = \frac{\mathrm{FTH} - \mathrm{ATH}}{6\sigma}, \qquad C_{pk} = \frac{\min(\mathrm{FTH} - \mu,\ \mu - \mathrm{ATH})}{3\sigma}, \tag{25.2}$$

ahol FTH és ATH a felső és az alsó tűréshatár. A C_p a tűrésmező és a folyamat szórásának viszonya, a C_{pk} a folyamat középértékének elhelyezkedését is figyelembe veszi (Montgomery 2020). A képességindex csak stabil folyamatra értelmezhető.

KoboCella: a nyomaték kontrollkártyája

A csavarozó egység óránként öt egymást követő kötés nyomatékát rögzíti. Az első húsz részcsoportból a középvonal 4,020 Nm, az átlagos terjedelem 0,116 Nm, így $\hat{\sigma} = 0{,}116/2{,}326 = 0{,}050$ Nm, a beavatkozási határok 3,953 és 4,087 Nm (25.4. ábra). A folyamat stabil, a képességindexek a 4,0 ± 0,2 Nm tűrésre $C_p = 0{,}4/(6{\cdot}0{,}050) = 1{,}33$ és $C_{pk} = 0{,}18/(3{\cdot}0{,}050) = 1{,}20$. A C_{pk} azért kisebb a C_p-nél, mert a középérték 0,02 Nm-rel a célérték fölött van; a középérték beállításával a C_{pk} a C_p értékéig növelhető (18. fejezet). A 21. részcsoporttól az átlag csökkenni kezdett. A 24. részcsoportnál két szabály is jelzett (három pontból kettő a 2σ-n túl, öt pontból négy az 1σ-n túl), a 25. pont már a beavatkozási határ alá esett. Az első gyanú a bitbetét kopása volt, de a kísérlettervezés szerint a választott kétlépcsős stratégiánál a kopás csak néhány század Nm-rel csökkenti a nyomatékot (18. fejezet), a megfigyelt elmozdulás ennél jóval nagyobb volt. A rétegzés megmutatta, hogy az elmozdulás egy új csavartétel felhasználásával kezdődött: a beszállító gyártási helyet váltott, és a csavarok felületi bevonata eltért a korábbiakétól. A csapat a tételt visszatartotta, a beszállítói specifikációt a bevonatra is kiterjesztette, és a beérkező tételekre mintavételes csavarozási próbát vezetett be; a kontrollkártya jelzése egyben a bitbetét állapotalapú cseréjének is egyik feltétele lett.

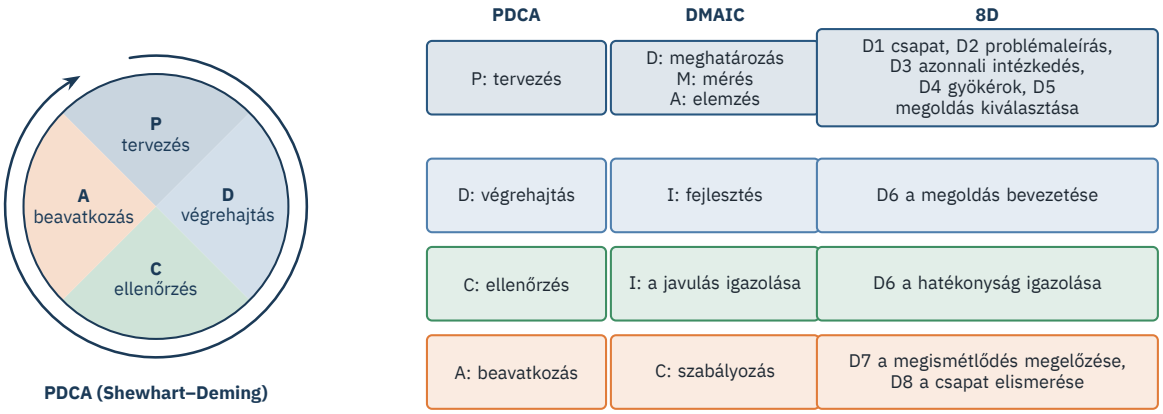


25.4. ábra. A KoboCella csavarozási nyomatékának átlag-kontrollkártyája (FEH és AEH: felső és alsó beavatkozási határ, CL: középvonal).

25.5. Fejlesztési ciklusok: PDCA, 8D, DMAIC

A folyamatos fejlesztés szervezett ciklusokban halad (25.5. ábra). A **PDCA**-ciklus (tervezés, végrehajtás, ellenőrzés, beavatkozás) Shewhart munkájára épül, és Deming tette ismertté (Deming 1986): a fejlesztés tervvel és mérhető céllal indul, kis léptékben valósul meg, a hatását mérik, és a bevált változtatást szabványosítják, a be nem váltat elvetik.

A **8D** (nyolc fegyelem) a reklamációk és a súlyos hibák rendszerezett, csapatban végzett kezelésére szolgál, különösen a járműiparban (VDA QMC 2018). Lépései: D1 a csapat



25.5. ábra. A PDCA-ciklus, valamint a DMAIC és a 8D lépéseinek megfeleltetése a PDCA szakaszainak.

összeállítása, D2 a probléma leírása, D3 azonnali (elszigetelő) intézkedések, D4 a gyökérok feltárása és igazolása, D5 a javító intézkedések kiválasztása, D6 a bevezetésük és hatékonyságuk igazolása, D7 a megismétlődés megelőzése, D8 a csapat elismerése és a lezárás. A 8D sajátossága, hogy az ügyfél védelmét szolgáló azonnali intézkedést elválasztja a gyökérokot megszüntető tartós intézkedéstől.

A **DMAIC** (meghatározás, mérés, elemzés, fejlesztés, szabályozás) a hat szigma módszertan projektciklusa. Schroeder és szerzőtársai szerint a hat szigma lényege a párhuzamos szervezeti struktúra (képzett projektvezetők), a strukturált fejlesztési eljárás és a mérőszámokra épülő célok (Schroeder és mtsai. 2008). A DMAIC adatvezérelt: a probléma meghatározása után a jelenlegi teljesítményt mérik, a gyökérokokat statisztikai eszközökkel igazolják, a megoldást kísérletileg kipróbálják, végül a javulást szabályozással (például kontrollkártyával) tartják fenn (Montgomery 2020).

25.2. táblázat. A három fejlesztési ciklus összevetése (Deming 1986; VDA QMC 2018; Schroeder és mtsai. 2008; Montgomery 2020 alapján).

	PDCA	8D	DMAIC
tipikus kiváltó ok	bármilyen fejlesztési lehetőség	reklamáció, súlyos hiba	krónikus, adatokkal mérhető teljesítményhiány
léptéke	kicsi, gyorsan ismételhető	egy probléma, csapat, hetek	projekt, hónapok
fő eszközök	egyszerű mérés, kísérlet	ok-okozati diagram, 5 miért, azonnali intézkedés	statisztika, kísérlettervezés, kontrollkártya
a lezárás feltétele	a változtatás szabványosítása	a megismétlődés megelőzése igazolt	a javulás szabályozott és fenntartott

VárosiBringa: 8D a téli zárhibákra

Az első télen három hét alatt 12 bejelentés érkezett arról, hogy a dokkoló a kerékpár visszahelyezése után nem zárt; mind a négy érintett állomás árnyékos helyen volt (D2). Az azonnali intézkedés: az alkalmazás a visszaadás után a zár állapotát is ellenőrizte, és sikertelen zárás esetén a felhasználót a szomszédos dokkolóhoz irányította; a diszpécser reggelente ellenőrizte az érintett állomásokat (D3). A csapat a lehetséges okokat ok-okozati diagrammal gyűjtötte össze, és hűtőkamrás teszttel igazolta a gyökérokot: −5 °C alatt a zárnyelv

kenőanyaga besűrűsödött, és a reteszcsap nem ugrott be teljesen (D4). A javító intézkedés alacsony hőmérsékleten is működő kenőanyag és a zárnyelv rugójának erősítése lett (D5), amelyet hűtőkamrában és két állomáson igazoltak (D6). A megelőzés: a dokkoló FMEA-jának frissítése (15. fejezet), a kenőanyag-specifikáció módosítása és egy téli ellenőrzőlista (D7).

ForgácsTech: DMAIC az anyagra várásra

A Pareto-elemzés után a csapat DMAIC-projektet indított. *Meghatározás:* az anyagra várás havi 1120 perc, a cél legfeljebb 400 perc. *Mérés:* a MES-napló gépenként és okonként rögzíti a várakozásokat. *Elemzés:* az ok-okozati diagram és a napló szerint a várakozások 64%-a átsorolás után keletkezett. *Fejlesztés:* a MES-ütemező átsorolás előtt ellenőrzi az anyag rendelkezésre állását, és az anyagot a műszak eleje helyett kétóránként, az ütemterv alapján adják ki; két gépen egy hónapig próbálták ki. *Szabályozás:* a napi várakozási percek egyedi értékes kontrollkártyáját a műszakvezetői megbeszélésen követik. Három hónap után a havi anyagra várás 380 perc lett.

25.6. Trendelemzés és üzemi visszacsatolás

A kontrollkártya a hirtelen változást jelzi, a **trendelemzés** a lassú, fokozatos romlást. A trendelemzés több területre is kiterjedhet: a teljesítmény (egy működési jellemző alakulása), a problémák (a hibajelentések gyakorisága és jellege), a programmenedzsment (költség, ütemezés, erőforrások), a támogathatóság (pótalkatrészek, karbantartás) és a megbízhatóság trendjére. A trendelemzés célja, hogy a romlást még a követelmény megsértése vagy a meghibásodás előtt felismerje, és a beavatkozásra időt adjon.

Az üzemi adatok a fejlesztés számára is értékesek. A folyamatbányászat a valós folyamatot tárja fel az eseménynaplókból (20. fejezet), a hibajelentések és a karbantartási naplók a megbízhatósági modellek paramétereit pontosítják (16. fejezet), az üzem közbeni validáció (24. fejezet) pedig a ki nem mondott elvárásokat hozza felszínre. Ezért a tanulságokat az üzemeltetés és a lezárás minden szakaszában rögzíteni kell; a következő fejlesztés ezekből indul.

25.7. Az életciklus vége és a körforgásos szemlélet

Minden rendszer egyszer kikerül a használatból. Az ISO/IEC/IEEE 15288 a kivonást önálló folyamatként kezeli: célja, hogy a rendszer vagy elem működését befejezze, a kivont elemeket megfelelően kezelje, és a kritikus kivonási igényeket kielégítse, például a környezetvédelmi előírásokat (ISO/IEC/IEEE 2023, 6.4.14. szakasz). A rendszermérnök feladata, hogy a műszaki információkat azonosítsák és archiválják. A kivonás tevékenységei hosszúak és összetettek lehetnek, és visszahatnak a tervre, ezért a lehetőségeket és költségeiket már a korábbi szakaszokban mérlegelni kell.

A **körforgásos gazdaság** ezt a szemléletet a teljes termékláncra kiterjeszti. Kirchherr és szerzőtársai 114 meghatározás elemzése alapján a körforgásos gazdaságot olyan gazdasági rendszerként írják le, amely a termékek életciklusának végét a csökkentéssel, az újrahasználattal, az újrahasznosítással és az anyagok visszanyerésével váltja fel a termelés, az elosztás és a fogyasztás folyamataiban; a gyakran használt hierarchikus keretrendszerek a lehetőségeket az elutasítástól és újragondolástól az újrahasználaton, javításon, felújításon és újragyártáson át az újrahasznosításig és az energia visszanyeréséig rangsorolják (Kirchherr, Reike, és Hekkert 2017). A rendszermérnöki következmény:

a javíthatóságot, a szétszerelhetőséget, a modularitást és az anyagok azonosíthatóságát követelményként kell kezelni, nem utólagos szempontként. Egyes termékcsoportoknál ezt jogszabály is előírja: az Európai Unió elemrendelete például a könnyű közlekedési eszközök (köztük az elektromos kerékpárok) akkumulátoraira is gyűjtési, újrafeldolgozási és tájékoztatási kötelezettségeket állapít meg (Európai Parlament és a Tanács 2023).

VárosiBringa: az akkumulátorok életciklusa

A VárosiBringa csapata az akkumulátorok életciklus végét már a koncepció szakaszában a követelmények közé vette: a cserélhető, szabványos csatlakozójú akkumulátorok kapacitását a flottakezelő rendszer folyamatosan követi; ha egy akkumulátor kapacitása a névleges 80%-a alá csökken, kikerül a kerékpárból. A kivont akkumulátorokat a szerződés szerint a gyártó veszi vissza, és helyhez kötött energiatárolóként vagy újrafeldolgozásra hasznosítja. A követelmény hatással volt a tervre is: a kerékpár akkumulátortartója szerszám nélkül, de csak szervizkulccsal nyitható.

Prediktív karbantartás és digitális iker

Az érzékelők, az ipari adatgyűjtés és a gépi tanulás lehetővé teszi, hogy a karbantartás az állapot jövőbeli alakulásának előrejelzésére épüljön. A rendszer digitális ikre (3. fejezet) az üzemi adatokkal folyamatosan frissül, és szimulációval mutatja meg, hogyan hat egy beavatkozás. A rendszermérnöki kérdések ugyanazok, mint a fejezet többi részében: mely jellemzők romlása mérhető, mekkora a téves riasztás és az elmulasztott jelzés költsége, és ki dönt a beavatkozásról (12. fejezet).

Átadás után magára hagyott rendszer

Gyakori hiba, hogy a fejlesztőcsapat az átvétellel lezártnak tekinti a munkát, és senki nem felel a teljesítmény követéséért. A mérőszámokat nem gyűjtik, a kezelők a hibákat megkerülő megoldásokkal kezelik, a tudás a fejlesztőkkel együtt távozik. Az átadás része kell legyen az üzemeltetés felelőseinek kijelölése, a mérőszámok gyűjtésének és értékelésének rendje, és a fejlesztők elérhetősége egy meghatározott időszakra.

Összefoglalás

- Az átadás a verifikált és validált terméket a következő szintre vagy a felhasználóhoz juttatja; része a dokumentáció, a képzés, a helyszín előkészítése, a telepítés és az átvételi teszt.
- Az üzemeltetés során a rendszer továbbfejleszthető, de az architektúra jelentős változása új szükségletet és új életciklust jelent.
- A karbantartás megelőző (előre meghatározott vagy állapotalapú, ezen belül előrejelző) vagy hibát követő; a választást a meghibásodás jellege és következménye határozza meg.
- A hét alapeszköz (ellenőrzőlap, hisztogram, Pareto-diagram, ok-okozati diagram, rétegzés, szórásdiagram, kontrollkártya) az üzemi problémák többségének elemzésére elég.
- A kontrollkártya a különleges okokat jelzi; a képességindexek a tűréshez viszonyított teljesítményt mérik, és csak stabil folyamatra értelmezhetők.
- A PDCA, a 8D és a DMAIC a fejlesztés szervezett ciklusai; a trendelemzés a lassú romlást, az üzemi adatok a következő fejlesztés bemeneteit adják.
- Az életciklus végét a tervezéskor kell figyelembe venni: javíthatóság, szétszerelhetőség, újrahasználat és újrahasznosítás.

Önellenőrzés

1. Mikor zárul le az átadás?
2. Miért kell a helyszíni felmérést az életciklus korai szakaszában elvégezni?
3. Mi a különbség az előre meghatározott és az állapotalapú karbantartás között? Mikor nem csökkenti a hibák számát az időalapú csere?
4. Egy hónapban a reklamációk okai: csomagolási sérülés 46, hibás címke 12, méreteltérés 31, felületi hiba 8, egyéb 3. Készíts Pareto-sorrendet, és számold ki, mekkora arányt ad a két leggyakoribb ok!
5. Mi a különbség a véletlen és a különleges ok között?
6. Egy folyamat középértéke 10,02 mm, szórása 0,01 mm, a tűrés 10,00 ± 0,05 mm. Számítsd ki a C_p és a C_{pk} értékét!
7. Miben különbözik a 8D azonnali (D3) és tartós (D5–D6) intézkedése, és miért fontos a kettő elválasztása?
8. Milyen tervezési követelmények következnek a körforgásos szemléletből?

Megoldás.

1. Ha a termék a helyszínen telepítve van, az átvételi és tanúsítási tevékenységek befejeződtek, a felhasználók, kezelők és karbantartók képzése megtörtént, és az átadást átvételi dokumentum zárja.
2. Mert a létesítmények átalakítását, a logisztikai támogatást és az esetleges engedélyeket jóval az üzembe helyezés előtt meg kell tervezni; egy későn felismert hiány (energiaellátás, alapozás, hálózat) késést és többletköltséget okoz.
3. Az előre meghatározott karbantartás rögzített időközönként vagy darabszám után végzik, az állapotalapú az állapot mérése alapján. Ha a meghibásodási ráta nem nő az idővel (véletlen hibák), az időalapú csere nem csökkenti a hibák számát.
4. Sorrend: csomagolási sérülés 46, méreteltérés 31, hibás címke 12, felületi hiba 8, egyéb 3; összesen 100. A két leggyakoribb ok 77%.
5. A véletlen ok a folyamat természetes, sok kis tényezőből adódó ingadozása; a különleges ok azonosítható, a folyamat megváltozását jelzi (kopás, új anyag, beállítási hiba), és beavatkozást kíván.
6. $C_p = 0{,}10/(6\cdot 0{,}01) = 1{,}67$; $C_{pk} = \min(10{,}05-10{,}02;\ 10{,}02-9{,}95)/(3\cdot 0{,}01) = 0{,}03/0{,}03 = 1{,}00$.
7. Az azonnali intézkedés az ügyfelet védi a hiba következményeitől, amíg a gyökérok nem ismert; a tartós intézkedés a gyökérokot szünteti meg. Ha nem választják el őket, az azonnali intézkedés (például szigorított ellenőrzés) tartóssá válik, és a gyökérok megmarad.
8. Javíthatóság, szétszerelhetőség, modularitás, szabványos csatlakozók és alkatrészek, az anyagok azonosíthatósága, visszavételi és újrahasznosítási lehetőség, valamint az ezekhez szükséges információk átadása.

Projektlépés: üzemeltetés és fejlesztés

Saját fejlesztési feladatodhoz:

1. sorold fel az átadás feltételeit: dokumentáció, képzés, a helyszín előkészítése, átvételi teszt, felelősök;
2. válassz karbantartási stratégiát a rendszer két legfontosabb eleméhez, és indokold;
3. nevezz meg három üzemi mutatót (a MOE-kból és a TPM-ekből), és add meg, hogyan, milyen gyakran és ki méri őket;
4. tervezz egy kontrollkártyát egy kritikus jellemzőhöz: részcsoport, mintavételi gyakoriság, a jelzések kezelése;
5. írd le, mi történik a rendszerrel az életciklus végén, és milyen követelmény következik ebből.

Továbbolvasás

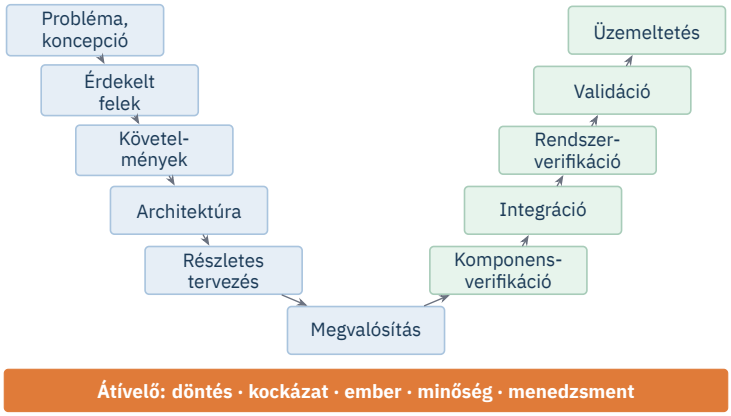
- ISO/IEC/IEEE (2023), 6.4.10. és 6.4.12–6.4.14. szakasz: az átadás, az üzemeltetés, a karbantartás és a kivonás folyamata.
- ISO/IEC/IEEE (2023), 6.4.10., 6.4.12–6.4.14. szakasz; CEN (2017): az átadás, üzemelte-

tés, karbantartás és kivonás folyamatai, a karbantartás fogalmai.
- Ishikawa (1982); Montgomery (2020): a minőségügy alapeszközei és a statisztikai folyamatszabályozás.
- Shewhart (1931); Deming (1986); VDA QMC (2018); Schroeder és mtsai. (2008): a folyamatszabályozás és a fejlesztési ciklusok alapjai.
- Kirchherr, Reike, és Hekkert (2017); Európai Parlament és a Tanács (2023): a körforgásos gazdaság fogalma és szabályozási példa.
- NASA (2016), 3.8., 3.9. és 5.5. szakasz: az üzemeltetési és a lezárási szakasz, valamint a termékátadás folyamata.
- Goldberg és mtsai. (1994), 5., 7. és 8. fejezet: a minőségügy alapeszközei, a statisztikai folyamatszabályozás és a trendelemzés gyakorlati leírása.

V. rész – Átívelő technikai menedzsment

Pannon Egyetem
University of Pannonia

26. Technikai tervezés és irányítás



Tanulási eredmények

A fejezet feldolgozása után képes leszel:

- áttekinteni a technikai irányítás folyamatait, és elhelyezni őket a V-modell mentén;
- munkalebontási struktúrát készíteni, és arra ütemezést és költségvetést építeni;
- a projekt előrehaladását technikai mérőszámokkal és a megszerzett érték módszerével követni;
- megtervezni a konfigurációkezelést, és szabályozottan kezelni a változtatási kérelmeket;
- technikai felülvizsgálatot előkészíteni és lebonyolítani;
- mérlegelni, mikor és hogyan illeszthetők az agilis módszerek egy rendszerfejlesztésbe.

A V-modell mentén haladó műszaki munkát irányítani kell: meg kell tervezni, követni kell az előrehaladását, kezelni kell a változásokat és a kockázatokat, és rendszeresen értékelni kell az eredményeket. Ezek a **technikai irányítási folyamatok** a teljes életcikluson átívelnek, és a termékszerkezet minden szintjén alkalmazandók; az ISO/IEC/IEEE 15288 nyolc ilyen folyamatot ír le (ISO/IEC/IEEE 2023, 6.3. szakasz). A 26.1. táblázat mutatja, hol tárgyalja a könyv az egyes folyamatokat.

26.1. táblázat. A technikai irányítás tevékenységei és az ISO/IEC/IEEE 15288 folyamatai (ISO/IEC/IEEE 2023).

Irányítási tevékenység	ISO/IEC/IEEE 15288 folyamat (szakasz)	Ahol a könyv tárgyalja
technikai tervezés	projekttervezés (6.3.1)	ez a fejezet
az előrehaladás értékelése	projektértékelés és -irányítás (6.3.2); mérés (6.3.7)	ez a fejezet; 6. fejezet
döntések előkészítése	döntéskezelés (6.3.3)	10. fejezet
kockázatkezelés	kockázatkezelés (6.3.4)	14. fejezet
konfigurációkezelés	konfigurációkezelés (6.3.5)	ez a fejezet

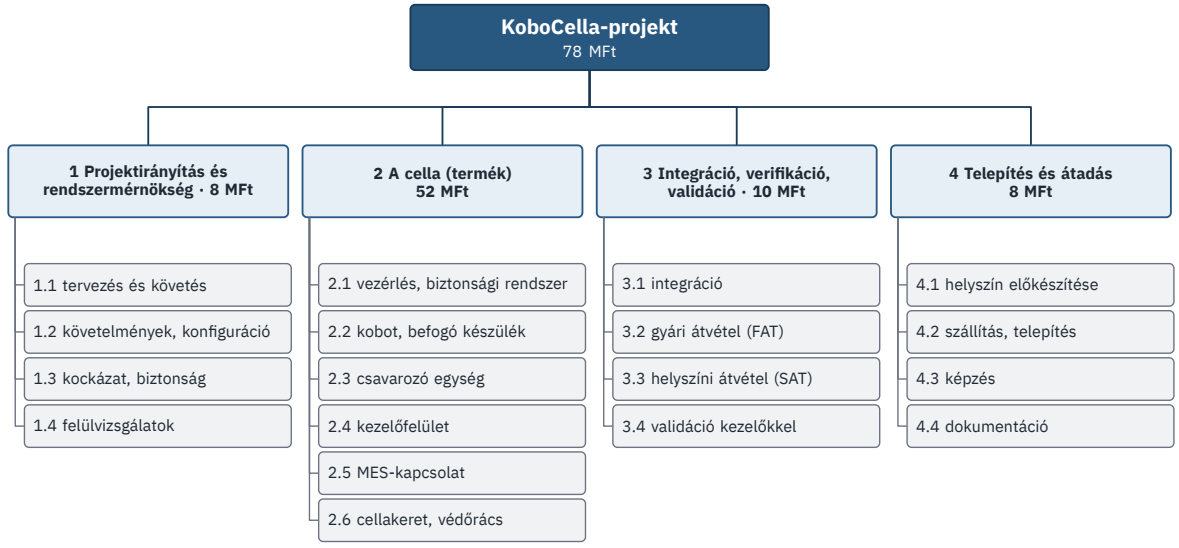
Irányítási tevékenység	ISO/IEC/IEEE 15288 folyamat (szakasz)	Ahol a könyv tárgyalja
technikai adatkezelés	információkezelés (6.3.6)	ez a fejezet
minőségbiztosítás	minőségbiztosítás (6.3.8)	25. fejezet
követelménykezelés	a rendszerkövetelmények meghatározásának része (6.4.3)	6. fejezet
interfészkezelés	az architektúra- és a tervmeghatározás része (6.4.4–6.4.5)	22. fejezet

26.1. Technikai tervezés: munkalebontás, ütemezés, költségvetés

26.1.1. Munkalebontási struktúra

A **munkalebontási struktúra** (work breakdown structure, WBS) a projekt végtermékeinek előállításához szükséges hardver, szoftver, szolgáltatások, adatok és létesítmények termékorientált, hierarchikus felosztása (U.S. Department of Defense 2022; NASA 2016, 6.1. szakasz). Azt is tükrözi, hogyan végzik a munkát, és hogyan gyűjtik, összesítik és jelentik a költség-, ütemezési, műszaki és kockázati adatokat (Project Management Institute 2019). A WBS tehát a termékfára (a termékstruktúrára) épül, de a terméket létrehozó munkát is tartalmazza: a projektirányítást, a rendszermérnöki munkát, az integrációt, a verifikációt, a telepítést és a képzést (26.1. ábra).

A WBS legfontosabb szabálya a **100%-szabály**: minden szint pontosan a felette álló elem teljes munkáját tartalmazza, se többet, se kevesebbet (Project Management Institute 2019). A legalsó szint elemei a **munkacsomagok**: önállóan tervezhető, felelőshöz rendelhető, becsülhető és követhető munkaegységek.



100%-szabály: minden szint pontosan a felette álló elem teljes munkáját tartalmazza (8 + 52 + 10 + 8 = 78 MFt); a legalsó szint elemei a munkacsomagok.

26.1. ábra. A KoboCella-projekt munkalebontási struktúrájának két felső szintje a költségvetéssel.

26.1.2. Ütemezés és költségvetés

A WBS munkacsomagjai időt igényelnek, és sorrendi kapcsolatban állnak egymással. A hálós ütemterv kifejezetten megadja az egyes tevékenységek függését más tevékenységektől, az erőforrások rendelkezésre állásától és a külső beszállításoktól. A **kritikus út** a függő tevékenységek leghosszabb sorozata, amely a projekt időtartamát meghatározza (Kelley és Walker 1959). A kritikus út a projekt előrehaladásával változik, ezért folyamatosan figyelni kell; a módszert a Gyakorlati útmutató integrációs terv és WBS eszközlapja mutatja be. A tapasztalat szerint a költség- és műszaki problémák gyakran először ütemezési problémaként jelentkeznek, ezért a csúszás korai jelzés.

A költségvetés a WBS-ből és az ütemtervből épül fel: a munkaerő- és egyéb erőforrásigényből, a közvetlen és közvetett költségekből, a beszállítói és anyagköltségekből és a tartalékból. A WBS, az ütemterv és a költségvetés egymástól kölcsönösen függ, és együtt kell kezelni őket (Fleming és Koppelman 2010).

26.1.3. A rendszermérnöki irányítási terv

A **rendszermérnöki irányítási terv** (systems engineering management plan, SEMP) a projekt legfelső szintű technikai irányítási dokumentuma: leírja, milyen technikai folyamatokat alkalmaz a projekt, milyen tevékenységekkel, milyen szervezetben, milyen költséggel és ütemezéssel (INCOSE 2023b; SEBoK Editorial Board 2025). Már a koncepcionális szakaszban készül, és az életciklus során folyamatosan frissül; a többi technikai tervnek összhangban kell lennie vele. A SEMP fő értéke nem maga a dokumentum, hanem a tervezési munka, amely létrehozza: a terjedelme lényegtelen, az számít, hogy a projekt műszaki szükségleteit lefedje.

26.2. A teljesítmény követése

A technikai értékelés folyamatos visszacsatolási hurok: tervezés, végrehajtás, állapotjelentés, értékelés, döntés, és ha szükséges, újratervezés (ISO/IEC/IEEE 2023, 6.3.2. és 6.3.7. szakasz; NASA 2016, 6.7. szakasz). A jó gyakorlat elvei: előre egyeztetett, jól meghatározott mérőszámok; egységes formátum minden szinten; a mérőszámok WBS szerinti összesítése; a történeti adatok megőrzése a trendek felismeréséhez; és színkódos (piros, sárga, zöld) jelzőzónák a mérőszámokra. A mérőszámokat rendszeresen, például havonta kell követni, a fontosabb felülvizsgálatokon pedig a trendjeiket is értékelni kell (Roedler és Jones 2005).

A műszaki előrehaladást a technikai teljesítménymutatók (TPM) követik (6. fejezet), a költséget és az ütemezést együtt a **megszerzett érték módszere** (earned value management, EVM). A megszerzett érték az adott időpontig ténylegesen elvégzett munka tervezett költsége; a módszer a műszaki tartalmat, az ütemezést és a költséget együtt kezelve méri a teljesítményt. Három mennyiséggel dolgozik (Fleming és Koppelman 2010):

- **tervezett érték** (planned value, PV): az adott időpontig tervezett munka tervezett költsége;
- **megszerzett érték** (earned value, EV): az adott időpontig ténylegesen elvégzett munka tervezett költsége;
- **tényleges költség** (actual cost, AC): az elvégzett munka tényleges költsége.

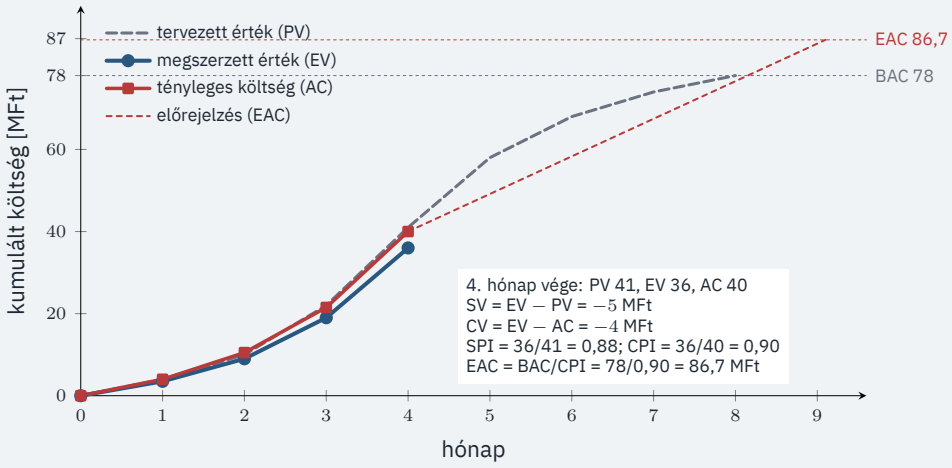
Ezekből

$$\mathrm{SV} = \mathrm{EV} - \mathrm{PV}, \quad \mathrm{CV} = \mathrm{EV} - \mathrm{AC}, \quad \mathrm{SPI} = \frac{\mathrm{EV}}{\mathrm{PV}}, \quad \mathrm{CPI} = \frac{\mathrm{EV}}{\mathrm{AC}}, \quad \mathrm{EAC} = \frac{\mathrm{BAC}}{\mathrm{CPI}}, \qquad (26.1)$$

ahol SV az ütemezési, CV a költségeltérés, SPI és CPI az ütemezési és a költséghatékonysági index, BAC a teljes jóváhagyott költségvetés, EAC a befejezéskori költség előrejelzése (azzal a feltevéssel, hogy a további munka is a mostani költséghatékonysággal halad) (Fleming és Koppelman 2010).

KoboCella: a projekt a negyedik hónap végén

A KoboCella-projekt költségvetése 78 MFt, időtartama nyolc hónap (26.1. ábra). A negyedik hónap végén a tervezett érték 41 MFt, a megszerzett érték 36 MFt, a tényleges költség 40 MFt (26.2. ábra). Az ütemezési index 0,88, a költséghatékonysági index 0,90: a projekt késésben van, és a vártnál drágábban halad. A befejezéskori költség előrejelzése $78/0{,}90 = 86{,}7$ MFt. Az elemzés két okot talált: a kobot beszállítója két hetet késett, és a kockázatértékelés (14. fejezet) után előírt kiegészítő biztonsági mérések a tervezettnél több munkát igényeltek. A csapat a kritikus úton lévő integrációs lépésekhez tartalék segédeszközt (szimulátort) rendelt, a többletköltségre pedig a tartalékból kért jóváhagyást.



26.2. ábra. A KoboCella-projekt megszerzett érték szerinti követése a negyedik hónap végén.

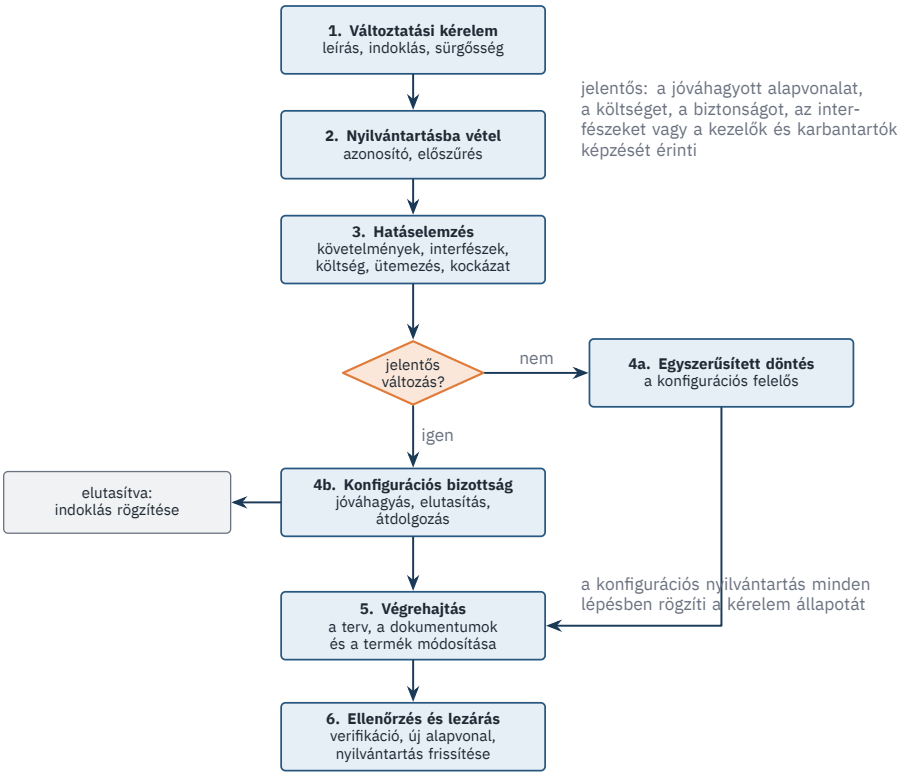
26.3. Konfigurációkezelés és technikai adatkezelés

A **konfigurációkezelés** célja, hogy a termék konfigurációja mindig ismert legyen és tükröződjön a termékinformációkban, hogy minden változtatás indokolt legyen és káros következmények nélkül valósuljon meg, és hogy a változások kezelése szabályozott legyen. Csökkenti a technikai kockázatot, megkülönbözteti a termékváltozatokat, és összhangban tartja a terméket és a róla szóló információkat (ISO 2017; U.S. Department of Defense 2020; NASA 2016, 6.5. szakasz). Az ISO 10007 szabvány öt tevékenységet különít el (ISO 2017):

- **tervezés és irányítás:** a konfigurációkezelés stratégiája és terve;
- **azonosítás:** a konfigurációs elemek (dokumentumok, hardver, szoftver, modellek) kijelölése, egyedi azonosítása és alapvonalba helyezése (2.5. szakasz);

- **változáskezelés:** a változtatási javaslatok rendszeres benyújtása, indoklása, értékelése, a jóváhagyott változások végrehajtása és ellenőrzése;
- **nyilvántartás** (configuration status accounting): a jelenlegi és korábbi konfigurációk, a változtatási kérelmek, eltérések és engedélyek állapotának rögzítése és jelentése;
- **ellenőrzés és auditálás:** annak igazolása, hogy a termék teljesíti a követelményeket (funkcionális audit), és hogy a terve dokumentált és összhangban van a termékkel (fizikai audit).

A változásokat a hatásuk szerint két osztályba sorolják (U.S. Department of Defense 2020). **Jelentős** (I. osztályú) a változás, ha az alapvonal konfigurációs dokumentációját jelentősen érinti: utólagos átalakítást kíván a már leszállított termékeken, vagy érinti a specifikációt, a költséget, a biztonságot, az interfésszel kapcsolódó termékekkel való összeférhetőséget, illetve a kezelők vagy karbantartók képzését. **Kisebb** (II. osztályú) a változás, ha a dokumentációt vagy a folyamatokat úgy javítja, hogy a termékek és elemek cserélhetőségét nem érinti. Az **eltérési engedély** (waiver) dokumentált megállapodás egy követelmény teljesítésének szándékos elengedéséről; nem változtatja meg az alapvonalat (U.S. Department of Defense 2020). A változáskezelés folyamatát a 26.3. ábra mutatja; a jelentős változásokról a **konfigurációs bizottság** (configuration control board, CCB) dönt.



26.3. ábra. A változáskezelés folyamata az ISO 10007 (ISO 2017) és a MIL-HDBK-61B (U.S. Department of Defense 2020) nyomán, egyszerűsítve.

KoboCella: egy második nyugtázó gomb

A validáció során a csapat egy második nyugtázó gombot javasolt a kezelői oldalra (24. fejezet). A változtatás első ránézésre apróság: egy nyomógomb és két vezeték. A hatáselemzés azonban kimutatta, hogy érinti a kezelőfelület interfészvezérlő dokumentumát (I-06), a kezelési útmutatót, a feladatelemzést és a kezelők képzését, és a biztonsági mérnöknek igazolnia kellett, hogy a biztonsági funkciókat nem érinti. A képzésre gyakorolt hatás miatt a változás jelentős, ezért a konfigurációs bizottság hagyta jóvá. A végrehajtás után a változáskérelem

> csak akkor zárult le, amikor minden érintett dokumentum új változata alapvonalba került, és a kezelők képzése megtörtént.

A **technikai adatkezelés** a projekt során keletkező műszaki adatok (tervek, modellek, elemzések, tesztadatok, jelentések) azonosítását, tárolását, védelmét, elérhetőségét és archiválását jelenti; a szabványban ez az információkezelési folyamat (ISO/IEC/IEEE 2023, 6.3.6. szakasz). A konfigurációkezeléssel együtt ezen múlik, hogy mindenki ugyanazt a jóváhagyott változatot használja, és hogy a döntések indoklása később is visszakereshető legyen.

26.4. Technikai felülvizsgálatok

A technikai felülvizsgálatok helyét az életciklusban a 2.5. szakasz mutatta be. A felülvizsgálatokhoz kapcsolódó tevékenységek (IEEE 2014): a szakaszok közötti felülvizsgálatok azonosítása, tervezése és lebonyolítása; minden felülvizsgálat céljának, tárgyának, valamint **belépési** és **sikerességi feltételeinek** rögzítése; a felülvizsgáló csoport összeállítása; végül a felülvizsgálat nyomán született intézkedések azonosítása és lezárása. A felülvizsgálatok azok a pontok, ahol a technikai mérőszámokat és trendjeiket gondosan meg kell vizsgálni a lehetséges problémák korai jelei után kutatva. A felülvizsgálatot jegyzőkönyv zárja, amely rögzíti az eredményeket, az ajánlásokat és az intézkedéseket a sikerességi feltételek teljesüléséről.

A formális felülvizsgálatok mellett a **szakmai átvizsgálások** (peer review) kisebb körben, egy-egy munkaterméket (tervet, kódot, elemzést) vizsgálnak meg részletesen. Legszigorúbb formájuk Fagan inspekciós módszere, a fajtáikat az IEEE 1028 szabvány rendszerezi (Fagan 1976; IEEE 2008; NASA 2016, N függelék).

VárosiBringa: az előzetes tervfelülvizsgálat

A VárosiBringa előzetes tervfelülvizsgálatának belépési feltétele volt a rendszerarchitektúra, az allokált követelmények, az interfészjegyzék, a dokkoló tervezési FMEA-ja, a kölcsönző alkalmazás első prototípusának használhatósági eredményei és az áthelyezési stratégia szimulációja. A dokumentumokat a bírálók tíz nappal a felülvizsgálat előtt kapták meg; a felülvizsgáló csoportban az önkormányzat, az üzemeltető, az akadálymentességi érdekképviselet és egy független rendszermérnök is részt vett. A felülvizsgálaton 23 észrevétel érkezett, köztük kettő súlyos: a dokkoló és a járda szélességének összeférhetősége két állomáson nem volt igazolva, és a téli üzemre (hó, jég) nem készült forgatókönyv. A felülvizsgálat feltételesen sikeres lett: a következő szakasz elkezdődhetett, de a két súlyos észrevételt felelőssel és határidővel le kellett zárni a részletes tervfelülvizsgálat előtt. A téli forgatókönyv hiánya utólag különösen tanulságos volt (25. fejezet).

26.5. Agilis és hibrid irányítás

Az agilis fejlesztés rövid ciklusokban, működő részeredményeket szállítva halad (2. fejezet). Az Agilis kiáltvány az egyéneket és az együttműködést, a működő terméket, a megrendelővel való együttműködést és a változásra való reagálást helyezi előtérbe (Beck és mtsai. 2001). A legelterjedtebb keretrendszer, a Scrum, rögzített hosszúságú (legfeljebb egyhónapos) sprintekben dolgozik: a termékcélból levezetett, rangsorolt teendőlistából a csapat sprintenként egy használható növekményt állít elő, amelyet a sprint végén az

érdekelt felekkel közösen áttekint, majd a munkamódszerét is értékeli (Schwaber és Sutherland 2020).

Hardvert is tartalmazó rendszereknél a tisztán agilis megközelítés ritkán járható: a hosszú beszerzési idők, a biztonsági igazolás és a fizikai integráció nem bontható kéthetes ciklusokra. A gyakorlatban **hibrid** megoldás terjedt el: a rendszerszintű követelményeket, az architektúrát, az interfészeket és a biztonsági igazolást V-modell szerint, alapvonalakkal és felülvizsgálatokkal kezelik, a szoftveren és egyes alrendszereken belül pedig agilis ciklusokban dolgoznak (INCOSE 2023b). Az inkrementális elköteleződésen alapuló spirálmodell ezt kockázatvezérelt döntési pontokkal köti össze: minden ciklus végén az érdekelt felek a bizonyítékok alapján döntenek a folytatásról (B. Boehm és mtsai. 2014).

ForgácsTech: a MES bevezetése hibrid módon

A ForgácsTech MES-ének bevezetését a csapat hibrid módon szervezte. A rendszerszintű követelményeket (határidő-tartás, nyomonkövethetőség, a gépvezérlők garanciájának megőrzése), az architektúrát és a gépekkel való interfészeket (22. fejezet) alapvonalba helyezte, és csak konfigurációs bizottsági döntéssel változtatta. A MES beállítását és a kiegészítő szoftverfejlesztést kéthetes ciklusokban végezték: minden ciklus végén a műszakvezetők kipróbálták az új funkciókat a tesztkörnyezetben, és a visszajelzéseik alapján rangsorolták a következő ciklus teendőit. Ez a gyakorlat azonban nem tárta fel időben a kezelői jogosultságok hiányát (24. fejezet), mert a tesztkörnyezet nem tartalmazta a valós munkabeosztást: az agilis visszajelzés csak olyan jó, amilyen valószerű a környezet, amelyben a felhasználók kipróbálják a terméket.

Modellalapú projektirányítás

A modellalapú rendszermérnökség (3. fejezet) a technikai irányítást is átalakítja. Ha a követelmények, az architektúra, az interfészek és a verifikációs kapcsolatok egyetlen modellben élnek, a konfigurációkezelés a modell változatainak kezelésévé válik, a változtatás hatáselemzése a modellben lévő kapcsolatok bejárásával automatizálható, a felülvizsgálatok pedig a dokumentumok helyett a modell nézeteire épülhetnek. A modell azonban csak akkor hiteles, ha a szervezet a modell karbantartását is megtervezi és finanszírozza.

Terv a fiókban

Gyakori hiba, hogy a projekt elején gondosan elkészítik a WBS-t, az ütemtervet és a SEMP-et, majd soha többé nem frissítik őket. A frissítés nélküli terv elavul, és a projekt észrevétlenül ellenőrizetlen állapotba csúszhat. A tervek élő dokumentumok: a változáskezelés, a felülvizsgálatok és a havi állapotértékelés teszi őket a projekt irányításának eszközeivé.

Összefoglalás

- A technikai irányítás folyamatai (tervezés, követelmény-, interfész-, kockázat-, konfiguráció- és adatkezelés, értékelés, döntéselemzés) a teljes életciklusban, minden szinten működnek.
- A WBS termékorientált, hierarchikus munkafelosztás; a 100%-szabály szerint minden szint pontosan a felette álló elem teljes munkáját tartalmazza. Erre épül az ütemterv, a kritikus út és a költségvetés.
- A SEMP a legfelső szintű technikai irányítási terv; fő értéke a tervezési munka, és folyamatosan frissíteni kell.
- A teljesítményt a TPM-ek és a megszerzett érték módszere követi: SPI = EV/PV, CPI = EV/AC, EAC = BAC/CPI.
- A konfigurációkezelés öt eleme a tervezés, az azonosítás, a változáskezelés, a nyilvántartás és az auditálás; a jelentős változásokról a konfigurációs bizottság dönt.
- A felülvizsgálatoknak előre rögzített belépési és sikerességi feltételei vannak, és az

intézkedéseket felelőssel és határidővel le kell zárni.
- Hardvert is tartalmazó rendszereknél a hibrid irányítás terjedt el: alapvonalak és felülvizsgálatok a rendszerszinten, agilis ciklusok a szoftverben és egyes alrendszerekben.

Önellenőrzés

1. Mi a különbség a termékfa és a WBS között?
2. Mit mond ki a 100%-szabály, és miért fontos?
3. Miért kell a kritikus utat folyamatosan figyelni?
4. Egy projekt költségvetése 120 MFt. Egy adott időpontban PV = 50, EV = 45, AC = 55 MFt. Számítsd ki az SV, CV, SPI, CPI és EAC értékét, és értelmezd!
5. Sorold fel a konfigurációkezelés öt elemét!
6. Mikor jelentős egy változás? Miért volt jelentős a KoboCella második nyugtázó gombja?
7. Mi a különbség a változtatás és az eltérési engedély között?
8. Mikor ajánlott a hibrid irányítás, és mi a korlátja az agilis visszajelzésnek?

Megoldás.

1. A termékfa a rendszer elemeinek hierarchiája; a WBS ezekre épül, de a termék létrehozásához szükséges teljes munkát tartalmazza: a projektirányítást, a rendszermérnöki munkát, az integrációt, a verifikációt, a telepítést, a képzést és a dokumentációt is.
2. Minden szint pontosan a felette álló elem teljes munkáját tartalmazza, se többet, se kevesebbet. Enélkül munka maradhat ki a tervből (és a költségvetésből), vagy ugyanaz a munka kétszer is szerepelhet.
3. Mert a kritikus út a projekt időtartamát meghatározó tevékenységsorozat, és a projekt során változik: ha egy nem kritikus tevékenység annyit késik, hogy elfogy a tartaléka, kritikussá válik.
4. SV = 45 − 50 = −5 MFt; CV = 45 − 55 = −10 MFt; SPI = 0,90; CPI = 0,82; EAC = 120/0,82 ≈ 146,7 MFt. A projekt késésben van, és a költséghatékonysága rossz: ha így folytatódik, mintegy 27 MFt-tal lépi túl a költségvetést.
5. Tervezés és irányítás, azonosítás, változáskezelés, nyilvántartás, ellenőrzés és auditálás.
6. Ha az alapvonal konfigurációs dokumentációját jelentősen érinti: utólagos átalakítást kíván a leszállított termékeken, vagy érinti a specifikációt, a költséget, a biztonságot, az interfészekkel való összeférhetőséget, illetve a kezelők vagy karbantartók képzését. A nyugtázó gomb a kezelők képzését és a kezelőfelület interfész-dokumentumát érintette.
7. A változtatás az alapvonalat módosítja; az eltérési engedély dokumentált megállapodás egy követelmény teljesítésének szándékos elengedéséről, az alapvonal változatlan marad.
8. Ha a rendszerszintű követelmények, az architektúra és a biztonsági igazolás stabil alapvonalat kíván, de a szoftver vagy egyes alrendszerek gyorsan változó igényekkel, rövid ciklusokban fejleszthetők. Az agilis visszajelzés csak olyan jó, amilyen valószerű a kipróbálás környezete; a valós használat feltételeit (munkabeosztás, terhelés, környezet) a próbában is képviselni kell.

Projektlépés: technikai terv

Saját fejlesztési feladatodhoz:

1. készíts legalább háromszintű WBS-t a 100%-szabály betartásával, és rendelj a munkacsomagokhoz felelőst és becsült ráfordítást;
2. készíts hálós ütemtervet, és határozd meg a kritikus utat;
3. jelöld ki a konfigurációs elemeket, és írd le, hogyan kezeled a változtatási kérelmeket;
4. tervezd meg a felülvizsgálatokat belépési és sikerességi feltételekkel;
5. használd a Gyakorlati útmutató WBS, felülvizsgálati és konfigurációkezelési eszközlapját.

Továbbolvasás

- U.S. Department of Defense (2020) és U.S. Department of Defense (2022): a konfigurációkezelés és a munkalebontás részletes, gyakorlati útmutatói.
- IEEE (2014) és IEEE (2008): a technikai felülvizsgálatok és a szakmai átvizsgálások szabványai.

- ISO/IEC/IEEE (2023), 6.3. szakasz; ISO (2017): a technikai irányítás és a konfigurációkezelés a szabványokban.
- Project Management Institute (2019); Kelley és Walker (1959); Fleming és Koppelman (2010): a munkalebontás, a kritikus út és a megszerzett érték módszere.
- Beck és mtsai. (2001); Schwaber és Sutherland (2020); B. Boehm és mtsai. (2014); INCOSE (2023b): agilis és hibrid irányítás.
- NASA (2016), 6. fejezet, B, J és N függelék: a technikai irányítás folyamatai, a WBS, a SEMP, a konfigurációkezelés, a technikai értékelés és a szakmai átvizsgálás.

Irodalom

Aalst, Wil M. P. van der. 2016. *Process Mining: Data Science in Action*. 2. kiad. Berlin: Springer. https://doi.org/10.1007/978-3-662-49851-4.

Abonyi, János, és Tímea Czvetkó. 2022. „Hypergraph and Network Flow-Based Quality Function Deployment". *Heliyon* 8 (12): e12263. https://doi.org/10.1016/j.heliyon.2022.e12263.

Ackermann, Fran, és Colin Eden. 2011. „Strategic Management of Stakeholders: Theory and Practice". *Long Range Planning* 44 (3): 179–96. https://doi.org/10.1016/j.lrp.2010.08.001.

Agresti, Alan, és Brent A. Coull. 1998. „Approximate Is Better than »Exact« for Interval Estimation of Binomial Proportions". *The American Statistician* 52 (2): 119–26. https://doi.org/10.1080/00031305.1998.10480550.

AIAG, és VDA. 2019. *Failure Mode and Effects Analysis – FMEA Handbook*. 1. kiad. Southfield, MI: Automotive Industry Action Group.

Akao, Yoji. 1990. *Quality Function Deployment: Integrating Customer Requirements into Product Design*. Cambridge, MA: Productivity Press.

Alexander, Ian F. 2005. „A Taxonomy of Stakeholders: Human Roles in System Development". *International Journal of Technology and Human Interaction* 1 (1): 23–59. https://doi.org/10.4018/jthi.2005010102.

Altshuller, Genrich S. 1984. *Creativity as an Exact Science: The Theory of the Solution of Inventive Problems*. New York: Gordon; Breach.

Annett, John, és Keith D. Duncan. 1967. „Task Analysis and Training Design". *Occupational Psychology* 41: 211–21.

Arrow, Kenneth J. 1951. *Social Choice and Individual Values*. New York: Wiley.

ASME. 2018. „ASME Y14.5-2018 Dimensioning and Tolerancing". American Society of Mechanical Engineers, New York.

Avižienis, Algirdas, Jean-Claude Laprie, Brian Randell, és Carl Landwehr. 2004. „Basic Concepts and Taxonomy of Dependable and Secure Computing". *IEEE Transactions on Dependable and Secure Computing* 1 (1): 11–33. https://doi.org/10.1109/TDSC.2004.2.

Az Európai Unió Tanácsa. 2017. „A Tanács ajánlása (2017. május 22.) az egész életen át tartó tanulás Európai Képesítési Keretrendszeréről és a 2008. április 23-i európai parlamenti és tanácsi ajánlás hatályon kívül helyezéséről (2017/C 189/03)". Az Európai Unió Hivatalos Lapja, C 189, 2017. 6. 15., 15–28. o.

Bainbridge, Lisanne. 1983. „Ironies of Automation". *Automatica* 19 (6): 775–79. https://doi.org/10.1016/0005-1098(83)90046-8.

Baldwin, Carliss Y., és Kim B. Clark. 2000. *Design Rules, Volume 1: The Power of Modularity*. Cambridge, MA: MIT Press.

Bangor, Aaron, Philip T. Kortum, és James T. Miller. 2008. „An Empirical Evaluation of the System Usability Scale". *International Journal of Human-Computer Interaction* 24 (6): 574–94. https://doi.org/10.1080/10447310802205776.

Bangor, Aaron, Philip Kortum, és James Miller. 2009. „Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale". *Journal of Usability Studies* 4 (3): 114–23.

Beck, Kent és mtsai. 2001. „Manifesto for Agile Software Development”. https://agilemanifesto.org.
Belton, Valerie, és Tony Gear. 1983. „On a Short-Coming of Saaty’s Method of Analytic Hierarchies”. *Omega* 11 (3): 228–30. https://doi.org/10.1016/0305-0483(83)90047-6.
Ben-Akiva, Moshe, és Steven R. Lerman. 1985. *Discrete Choice Analysis: Theory and Application to Travel Demand*. Cambridge, MA: MIT Press.
Berger, Charles, Robert Blauth, David Boger, Christopher Bolster, Gary Burchill, William DuMouchel, Fred Pouliot, és mtsai. 1993. „Kano’s Methods for Understanding Customer-Defined Quality”. *Center for Quality of Management Journal* 2 (4): 3–36.
Bertalanffy, Ludwig von. 1968. *General System Theory: Foundations, Development, Applications*. New York: George Braziller.
Biomimicry Institute. 2026. „AskNature: Innovation Inspired by Nature”. https://asknature.org.
Blanchard, Benjamin S., és Wolter J. Fabrycky. 2011. *Systems Engineering and Analysis*. 5. kiad. Upper Saddle River, NJ: Prentice Hall.
Boardman, Anthony E., David H. Greenberg, Aidan R. Vining, és David L. Weimer. 2018. *Cost-Benefit Analysis: Concepts and Practice*. 5. kiad. Cambridge: Cambridge University Press.
Boehm, Barry W. 1984. „Verifying and Validating Software Requirements and Design Specifications”. *IEEE Software* 1 (1): 75–88. https://doi.org/10.1109/MS.1984.233702.
———. 1988. „A Spiral Model of Software Development and Enhancement”. *Computer* 21 (5): 61–72. https://doi.org/10.1109/2.59.
Boehm, Barry, Jo Ann Lane, Supannika Koolmanojwong, és Richard Turner. 2014. *The Incremental Commitment Spiral Model: Principles and Practices for Successful Systems and Software*. Upper Saddle River, NJ: Addison-Wesley.
Bono, Edward de. 1985. *Six Thinking Hats*. Boston: Little, Brown.
Booher, Harold R., szerk. 2003. *Handbook of Human Systems Integration*. Hoboken, NJ: Wiley.
Bowles, John B. 2003. „An Assessment of RPN Prioritization in a Failure Modes Effects and Criticality Analysis”. In *Annual Reliability and Maintainability Symposium, 2003 Proceedings*, 380–86. https://doi.org/10.1109/RAMS.2003.1182019.
Box, George E. P. 1979. „Robustness in the Strategy of Scientific Model Building”. In *Robustness in Statistics*, szerkesztette Robert L. Launer és Graham N. Wilkinson, 201–36. New York: Academic Press.
Braglia, Marcello, Davide Castellano, Marco Frosolini, Mosè Gallo, és Leonardo Marrazzini. 2021. „Revised Overall Labour Effectiveness”. *International Journal of Productivity and Performance Management* 70 (6): 1317–35. https://doi.org/10.1108/IJPPM-08-2019-0368.
Breque, Maija, Lars De Nul, és Athanasios Petridis. 2021. „Industry 5.0: Towards a Sustainable, Human-Centric and Resilient European Industry”. Luxembourg: European Commission, Directorate-General for Research; Innovation.
Brooke, John. 1996. „SUS: A »Quick and Dirty« Usability Scale”. In *Usability Evaluation in Industry*, szerkesztette Patrick W. Jordan, B. Thomas, Bernard A. Weerdmeester, és Ian L. McClelland, 189–94. London: Taylor & Francis.
Browning, Tyson R. 2001. „Applying the Design Structure Matrix to System Decomposition and Integration Problems: A Review and New Directions”. *IEEE Transactions on Engineering Management* 48 (3): 292–306.
Butler, John, Jianmin Jia, és James Dyer. 1997. „Simulation Techniques for the Sensitivity Analysis of Multi-Criteria Decision Models”. *European Journal of Operational Research* 103 (3): 531–46. https://doi.org/10.1016/S0377-2217(96)00307-4.
Cachon, Gérard, és Christian Terwiesch. 2013. *Matching Supply with Demand: An Introducti-*

on to Operations Management. 3. kiad. New York: McGraw-Hill/Irwin.
CEN. 2017. „EN 13306:2017 Maintenance – Maintenance Terminology". European Committee for Standardization, Brüsszel.
Center for Chemical Process Safety. 2001. *Layer of Protection Analysis: Simplified Process Risk Assessment*. New York: American Institute of Chemical Engineers.
Chan, Lai-Kow, és Ming-Lu Wu. 2002. „Quality Function Deployment: A Literature Review". *European Journal of Operational Research* 143 (3): 463–97. https://doi.org/10.1016/S0377-2217(02)00178-9.
Chase, Kenneth W., és Alan R. Parkinson. 1991. „A Survey of Research in the Application of Tolerance Analysis to the Design of Mechanical Assemblies". *Research in Engineering Design* 3 (1): 23–37. https://doi.org/10.1007/BF01580066.
Cockburn, Alistair. 2001. *Writing Effective Use Cases*. Boston: Addison-Wesley.
Conway, Melvin E. 1968. „How Do Committees Invent?" *Datamation* 14 (4): 28–31.
Cowan, Nelson. 2001. „The Magical Number 4 in Short-Term Memory: A Reconsideration of Mental Storage Capacity". *Behavioral and Brain Sciences* 24 (1): 87–114. https://doi.org/10.1017/S0140525X01003922.
Cox, Louis Anthony. 2008. „What's Wrong with Risk Matrices?" *Risk Analysis* 28 (2): 497–512. https://doi.org/10.1111/j.1539-6924.2008.01030.x.
Crawley, Edward, Bruce Cameron, és Daniel Selva. 2016. *System Architecture: Strategy and Product Development for Complex Systems*. Hoboken, NJ: Pearson.
Czvetkó, Tímea, Viktor Sebestyén, és János Abonyi. 2025. „Key Factors of Industry 5.0-Based Organizational Sustainability". *Technology in Society* 83: 102966. https://doi.org/10.1016/j.techsoc.2025.102966.
Dalkey, Norman, és Olaf Helmer. 1963. „An Experimental Application of the Delphi Method to the Use of Experts". *Management Science* 9 (3): 458–67. https://doi.org/10.1287/mnsc.9.3.458.
Deb, Kalyanmoy, Amrit Pratap, Sameer Agarwal, és T. Meyarivan. 2002. „A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II". *IEEE Transactions on Evolutionary Computation* 6 (2): 182–97.
Defense Acquisition University. 2001. *Systems Engineering Fundamentals*. Fort Belvoir, VA: Defense Acquisition University Press.
Dekker, Sidney. 2014. *The Field Guide to Understanding „Human Error"*. 3. kiad. Farnham: Ashgate.
Delbecq, Andre L., és Andrew H. Van de Ven. 1971. „A Group Process Model for Problem Identification and Program Planning". *The Journal of Applied Behavioral Science* 7 (4): 466–92. https://doi.org/10.1177/002188637100700404.
Deming, W. Edwards. 1986. *Out of the Crisis*. Cambridge, MA: MIT Center for Advanced Engineering Study.
Design Council. 2019. „Framework for Innovation: Design Council's Evolved Double Diamond". Design Council, London (a Double Diamond modell első változata: 2005). https://www.designcouncil.org.uk/our-resources/the-double-diamond/.
Diehl, Michael, és Wolfgang Stroebe. 1987. „Productivity Loss in Brainstorming Groups: Toward the Solution of a Riddle". *Journal of Personality and Social Psychology* 53 (3): 497–509.
Doran, George T. 1981. „There's a S.M.A.R.T. Way to Write Management's Goals and Objectives". *Management Review* 70 (11): 35–36.
Dorofee, Audrey J., Julie A. Walker, Christopher J. Alberts, Ronald P. Higuera, Richard L. Murphy, és Ray C. Williams. 1996. *Continuous Risk Management Guidebook*. Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University.
Doshi, Anil R., és Oliver P. Hauser. 2024. „Generative AI Enhances Individual Creativity but Reduces the Collective Diversity of Novel Content". *Science Advances* 10 (28): eadn5290.

https://doi.org/10.1126/sciadv.adn5290.
Dumas, Marlon, Marcello La Rosa, Jan Mendling, és Hajo A. Reijers. 2018. *Fundamentals of Business Process Management*. 2. kiad. Berlin: Springer.
ECSS. 2018. „ECSS-E-ST-10-02C Rev.1 Space Engineering – Verification". European Cooperation for Space Standardization, ESA-ESTEC, Noordwijk.
Eden, Colin, és Fran Ackermann. 1998. *Making Strategy: The Journey of Strategic Management*. London: Sage.
Edwards, Ward, és F. Hutton Barron. 1994. „SMARTS and SMARTER: Improved Simple Methods for Multiattribute Utility Measurement". *Organizational Behavior and Human Decision Processes* 60 (3): 306–25.
Endsley, Mica R. 1995. „Toward a Theory of Situation Awareness in Dynamic Systems". *Human Factors* 37 (1): 32–64. https://doi.org/10.1518/001872095779049543.
Engel, Avner. 2010. *Verification, Validation, and Testing of Engineered Systems*. Hoboken, NJ: John Wiley & Sons.
Eppinger, Steven D., és Tyson R. Browning. 2012. *Design Structure Matrix Methods and Applications*. Cambridge, MA: MIT Press.
Epstein, Joshua M. 2008. „Why Model?" *Journal of Artificial Societies and Social Simulation* 11 (4): 12. https://www.jasss.org/11/4/12.html.
Ericson, Clifton A., II. 2015. *Hazard Analysis Techniques for System Safety*. 2. kiad. Hoboken, NJ: John Wiley & Sons.
Európai Parlament és a Tanács. 2023. „Az (EU) 2023/1542 rendelet (2023. július 12.) az elemekről és a hulladékelemekről". Az Európai Unió Hivatalos Lapja, L 191.
European Commission. 2004. „Aid Delivery Methods, Volume 1: Project Cycle Management Guidelines". Brussels: European Commission, EuropeAid Cooperation Office.
Fagan, Michael E. 1976. „Design and Code Inspections to Reduce Errors in Program Development". *IBM Systems Journal* 15 (3): 182–211.
Fitts, Paul M., szerk. 1951. „Human Engineering for an Effective Air-Navigation and Traffic-Control System". Washington, DC: National Research Council.
Fleming, Quentin W., és Joel M. Koppelman. 2010. *Earned Value Project Management*. 4. kiad. Newtown Square, PA: Project Management Institute.
Forsberg, Kevin, és Harold Mooz. 1991. „The Relationship of System Engineering to the Project Cycle". In *Proceedings of the First Annual Symposium of the National Council on Systems Engineering (NCOSE) and the American Society for Engineering Management*, 57–65. Chattanooga, TN.
Freeman, R. Edward. 1984. *Strategic Management: A Stakeholder Approach*. Boston: Pitman.
Friedenthal, Sanford, Alan Moore, és Rick Steiner. 2014. *A Practical Guide to SysML: The Systems Modeling Language*. 3. kiad. Waltham, MA: Morgan Kaufmann.
Friedrich, Jan, Ulrike Hammerschall, Marco Kuhrmann, és Marc Sihling. 2009. *Das V-Modell XT: Für Projektleiter und QS-Verantwortliche kompakt und übersichtlich*. 2. kiad. Berlin: Springer.
Gilb, Tom. 2005. *Competitive Engineering: A Handbook for Systems Engineering, Requirements Engineering, and Software Engineering Using Planguage*. Oxford: Elsevier Butterworth-Heinemann.
Goldberg, B. E., K. Everhart, R. Stevens, N. Babbitt, P. Clemens, és L. Stout. 1994. „System Engineering »Toolbox« for Design-Oriented Engineers". NASA Reference Publication 1358. NASA Marshall Space Flight Center.
Goldratt, Eliyahu M., és Jeff Cox. 1984. *The Goal: A Process of Ongoing Improvement*. Great Barrington, MA: North River Press.
Grieves, Michael, és John Vickers. 2017. „Digital Twin: Mitigating Unpredictable, Undesirable Emergent Behavior in Complex Systems". In *Transdisciplinary Perspectives on Complex Systems*, szerkesztette Franz-Josef Kahlen, Shannon Flumerfelt, és Anabela

Alves, 85–113. Cham: Springer.
Griffin, Michael D. 2007. „System Engineering and the »Two Cultures« of Engineering". Boeing Lecture, Purdue University, West Lafayette, IN.
Grohs, Michael, Luka Abb, Nourhan Elsayed, és Jana-Rebecca Rehse. 2024. „Large Language Models Can Accomplish Business Process Management Tasks". In *Business Process Management Workshops (BPM 2023)*, 492:453–65. Lecture Notes in Business Information Processing. Cham: Springer. https://doi.org/10.1007/978-3-031-50974-2_34.
Guilford, J. P. 1950. „Creativity". *American Psychologist* 5 (9): 444–54.
Haddon, William. 1973. „Energy Damage and the Ten Countermeasure Strategies". *Journal of Trauma* 13 (4): 321–31. https://doi.org/10.1097/00005373-197304000-00011.
Hart, Sandra G. 2006. „NASA-Task Load Index (NASA-TLX); 20 Years Later". In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 50:904–8. 9. https://doi.org/10.1177/154193120605000909.
Hart, Sandra G., és Lowell E. Staveland. 1988. „Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research". In *Human Mental Workload*, szerkesztette Peter A. Hancock és Najmedin Meshkati, 52:139–83. Advances in Psychology. Amsterdam: North-Holland. https://doi.org/10.1016/S0166-4115(08)62386-9.
Hauser, John R., és Don Clausing. 1988. „The House of Quality". *Harvard Business Review* 66 (3): 63–73.
Health and Safety Executive. 2001. *Reducing Risks, Protecting People: HSE's Decision-Making Process*. Sudbury: HSE Books.
Hignett, Sue, és Lynn McAtamney. 2000. „Rapid Entire Body Assessment (REBA)". *Applied Ergonomics* 31 (2): 201–5. https://doi.org/10.1016/S0003-6870(99)00039-3.
Holland, John H. 1975. *Adaptation in Natural and Artificial Systems*. Ann Arbor: University of Michigan Press.
Hopp, Wallace J., és Mark L. Spearman. 2008. *Factory Physics*. 3. kiad. New York: McGraw-Hill/Irwin.
IEC. 2006. „IEC 61025:2006 Fault tree analysis (FTA)". International Electrotechnical Commission, Genf.
———. 2010a. „IEC 61508-1:2010 Functional safety of electrical/electronic/programmable electronic safety-related systems – Part 1: General requirements". International Electrotechnical Commission, Genf.
———. 2010b. „IEC 62502:2010 Analysis techniques for dependability – Event tree analysis (ETA)". International Electrotechnical Commission, Genf.
———. 2013. „IEC 62264-1:2013 Enterprise-control system integration – Part 1: Models and terminology". International Electrotechnical Commission, Genf.
———. 2015. „IEC 62366-1:2015 Medical Devices – Part 1: Application of Usability Engineering to Medical Devices". Nemzetközi szabvány; módosítás: AMD1:2020.
———. 2016a. „IEC 61078:2016 Reliability block diagrams". International Electrotechnical Commission, Genf.
———. 2016b. „IEC 61511-1:2016 Functional safety – Safety instrumented systems for the process industry sector – Part 1: Framework, definitions, system, hardware and application programming requirements". International Electrotechnical Commission, Genf.
———. 2018. „IEC 60812:2018 Failure modes and effects analysis (FMEA and FMECA)". International Electrotechnical Commission, Genf.
IEEE. 1998. „IEEE Std 1362-1998 IEEE Guide for Information Technology – System Definition – Concept of Operations (ConOps) Document". Institute of Electrical and Electronics Engineers, New York.
———. 2008. „IEEE Std 1028-2008: IEEE Standard for Software Reviews and Audits". New York: IEEE.

———. 2014. „IEEE Std 15288.2-2014: IEEE Standard for Technical Reviews and Audits on Defense Programs”. New York: Szabvány; nemzetközi változata ISO/IEC/IEEE 24748-8:2019; IEEE.
———. 2024. „IEEE Std 1012-2024 IEEE Standard for System, Software, and Hardware Verification and Validation”. Szabvány, IEEE Standards Association.
Ilevbare, Imoh M., David Probert, és Robert Phaal. 2013. „A Review of TRIZ, and Its Benefits and Challenges in Practice”. *Technovation* 33 (2–3): 30–37. https://doi.org/10.1016/j.technovation.2012.11.003.
INCOSE. 2007. „Systems Engineering Vision 2020”. INCOSE-TP-2004-004-02. International Council on Systems Engineering.
———. 2015. *INCOSE Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities*. Szerkesztette David D. Walden, Garry J. Roedler, Kevin J. Forsberg, R. Douglas Hamelin, és Thomas M. Shortell. 4. kiad. Hoboken, NJ: John Wiley & Sons.
———. 2018. „INCOSE Systems Engineering Competency Framework”. INCOSE-TP-2018-002-01.0. International Council on Systems Engineering.
———. 2019. „Systems Engineering and System Definitions”. International Council on Systems Engineering, technikai termék. https://www.incose.org/about-systems-engineering/system-and-se-definitions.
———. 2022. „Systems Engineering Vision 2035: Engineering Solutions for a Better World”. International Council on Systems Engineering. https://www.incose.org/sevision.
———. 2023a. „Guide to Writing Requirements”. INCOSE-TP-2010-006-04, Rev. 4. International Council on Systems Engineering, Requirements Working Group.
———. 2023b. *INCOSE Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities*. Szerkesztette David D. Walden, Thomas M. Shortell, Garry J. Roedler, Bernardo A. Delicado, Odile Mornas, Yew-Seng Yip, és David Endler. 5. kiad. Hoboken, NJ: John Wiley & Sons.
Ishikawa, Kaoru. 1982. *Guide to Quality Control*. 2. kiad. Tokyo: Asian Productivity Organization.
ISO. 2000. „ISO 11226:2000 Ergonomics – Evaluation of Static Working Postures”. International Organization for Standardization, Genf.
———. 2010. „ISO 12100:2010 Safety of machinery – General principles for design – Risk assessment and risk reduction”. International Organization for Standardization, Genf.
———. 2014. „ISO/IEC Guide 51:2014 Safety aspects – Guidelines for their inclusion in standards”. International Organization for Standardization, Genf.
———. 2016. „ISO/TS 15066:2016 Robots and robotic devices – Collaborative robots”. International Organization for Standardization, Genf.
———. 2017. „ISO 10007:2017 Quality Management – Guidelines for Configuration Management”. Nemzetközi szabvány.
———. 2018a. „ISO 26262-3:2018 Road vehicles – Functional safety – Part 3: Concept phase”. International Organization for Standardization, Genf.
———. 2018b. „ISO 31000:2018 Risk management – Guidelines”. International Organization for Standardization, Genf.
———. 2018c. „ISO 9241-11:2018 Ergonomics of Human-System Interaction – Part 11: Usability: Definitions and Concepts”. International Organization for Standardization, Genf.
———. 2019. „ISO 9241-210:2019 Ergonomics of Human-System Interaction – Part 210: Human-Centred Design for Interactive Systems”. International Organization for Standardization, Genf.
———. 2021. „ISO 23247-1:2021 Automation Systems and Integration – Digital Twin Framework for Manufacturing – Part 1: Overview and General Principles”. Nemzetközi szabvány.

———. 2023. „ISO 13849-1:2023 Safety of machinery – Safety-related parts of control systems – Part 1: General principles for design”. International Organization for Standardization, Genf.
———. 2025a. „ISO 10218-1:2025 Robotics – Safety Requirements – Part 1: Industrial Robots”. International Organization for Standardization, Genf.
———. 2025b. „ISO 10218-2:2025 Robotics – Safety Requirements – Part 2: Industrial Robot Applications and Robot Cells”. International Organization for Standardization, Genf.
ISO/IEC. 2013. „ISO/IEC 19510:2013 Information technology – Object Management Group Business Process Model and Notation”. International Organization for Standardization, Genf.
———. 2023. „ISO/IEC 25010:2023 Systems and Software Engineering – Systems and Software Quality Requirements and Evaluation (SQuaRE) – Product Quality Model”. Nemzetközi szabvány.
ISO/IEC/IEEE. 2018. „ISO/IEC/IEEE 29148:2018 Systems and Software Engineering – Life Cycle Processes – Requirements Engineering”. Nemzetközi szabvány, 2. kiadás.
———. 2022. „ISO/IEC/IEEE 42010:2022 Software, Systems and Enterprise – Architecture Description”. Nemzetközi szabvány, 2. kiadás.
———. 2023. „ISO/IEC/IEEE 15288:2023 Systems and Software Engineering – System Life Cycle Processes”. Nemzetközi szabvány, 2. kiadás; ISO, IEC és IEEE.
———. 2024. „ISO/IEC/IEEE 24748-1:2024 Systems and Software Engineering – Life Cycle Management – Part 1: Guidelines for Life Cycle Management”. Nemzetközi szabvány, 2. kiadás.
Jansson, David G., és Steven M. Smith. 1991. „Design Fixation”. *Design Studies* 12 (1): 3–11. https://doi.org/10.1016/0142-694X(91)90003-F.
Johnson, Eric J., és Daniel Goldstein. 2003. „Do Defaults Save Lives?” *Science* 302 (5649): 1338–39.
Johnson, S. M. 1954. „Optimal Two- and Three-Stage Production Schedules with Setup Times Included”. *Naval Research Logistics Quarterly* 1 (1): 61–68. https://doi.org/10.1002/nav.3800010110.
Kano, Noriaki, Nobuhiku Seraku, Fumio Takahashi, és Shinichi Tsuji. 1984. „Attractive Quality and Must-Be Quality”. *Hinshitsu (Journal of the Japanese Society for Quality Control)* 14 (2): 39–48.
Kaplan, Stanley, és B. John Garrick. 1981. „On the Quantitative Definition of Risk”. *Risk Analysis* 1 (1): 11–27. https://doi.org/10.1111/j.1539-6924.1981.tb01350.x.
Karplus, Walter J. 1977. „The Spectrum of Mathematical Modeling and Systems Simulation”. *Mathematics and Computers in Simulation* 19 (1): 3–10.
Keeney, Ralph L., és Robin S. Gregory. 2005. „Selecting Attributes to Measure the Achievement of Objectives”. *Operations Research* 53 (1): 1–11. https://doi.org/10.1287/opre.1040.0158.
Keeney, Ralph L., és Howard Raiffa. 1976. *Decisions with Multiple Objectives: Preferences and Value Tradeoffs*. New York: Wiley.
Kelley, James E., és Morgan R. Walker. 1959. „Critical-Path Planning and Scheduling”. In *Papers Presented at the December 1–3, 1959, Eastern Joint IRE-AIEE-ACM Computer Conference*, 160–73. New York: ACM. https://doi.org/10.1145/1460299.1460318.
Kendall, David G. 1953. „Stochastic Processes Occurring in the Theory of Queues and Their Analysis by the Method of the Imbedded Markov Chain”. *The Annals of Mathematical Statistics* 24 (3): 338–54. https://doi.org/10.1214/aoms/1177728975.
Kennedy, James, és Russell Eberhart. 1995. „Particle Swarm Optimization”. In *Proceedings of ICNN’95 – International Conference on Neural Networks*, 4:1942–48. IEEE.
Kepner, Charles H., és Benjamin B. Tregoe. 1965. *The Rational Manager: A Systematic*

Approach to Problem Solving and Decision Making. New York: McGraw-Hill.
Kingman, J. F. C. 1961. „The Single Server Queue in Heavy Traffic". *Mathematical Proceedings of the Cambridge Philosophical Society* 57 (4): 902–4. https://doi.org/10.1017/S0305004100036094.
Kirchherr, Julian, Denise Reike, és Marko Hekkert. 2017. „Conceptualizing the Circular Economy: An Analysis of 114 Definitions". *Resources, Conservation and Recycling* 127: 221–32. https://doi.org/10.1016/j.resconrec.2017.09.005.
Kirwan, Barry. 1994. *A Guide to Practical Human Reliability Assessment*. London: Taylor & Francis.
Knight, John C. 2002. „Safety Critical Systems: Challenges and Directions". In *Proceedings of the 24th International Conference on Software Engineering (ICSE 2002)*, 547–50. New York: ACM.
Kossiakoff, Alexander, William N. Sweet, Samuel J. Seymour, és Steven M. Biemer. 2011. *Systems Engineering Principles and Practice*. 2. kiad. Hoboken, NJ: John Wiley & Sons.
Kourani, Humam, Alessandro Berti, Daniel Schuster, és Wil M. P. van der Aalst. 2024. „Process Modeling with Large Language Models". In *Enterprise, Business-Process and Information Systems Modeling (BPMDS 2024, EMMSAD 2024)*, 229–44. Lecture Notes in Business Information Processing. Cham: Springer. https://doi.org/10.1007/978-3-031-61007-3_18.
Kritzinger, Werner, Matthias Karner, Georg Traar, Jan Henjes, és Wilfried Sihn. 2018. „Digital Twin in Manufacturing: A Categorical Literature Review and Classification". *IFAC-PapersOnLine* 51 (11): 1016–22.
Lakatos, Béla. é. n.a. „Rendszerek tervezése és irányítása". Egyetemi jegyzet, Veszprémi Egyetem, Folyamatmérnöki Tanszék.
———. é. n.b. „Termelési rendszerek". Egyetemi jegyzet, Veszprémi Egyetem, Folyamatmérnöki Tanszék.
Lano, Robert J. 1977. „The N2 Chart". Redondo Beach, CA: TRW Software Series.
Law, Averill M. 2015. *Simulation Modeling and Analysis*. 5. kiad. New York: McGraw-Hill.
Lee, John D., és Katrina A. See. 2004. „Trust in Automation: Designing for Appropriate Reliance". *Human Factors* 46 (1): 50–80. https://doi.org/10.1518/hfes.46.1.50_30392.
Leveson, Nancy G. 2011. *Engineering a Safer World: Systems Thinking Applied to Safety*. Cambridge, MA: MIT Press.
Leveson, Nancy G., és John P. Thomas. 2018. „STPA Handbook". Cambridge, MA: Massachusetts Institute of Technology.
Little, John D. C. 1961. „A Proof for the Queuing Formula: $L = \lambda W$". *Operations Research* 9 (3): 383–87. https://doi.org/10.1287/opre.9.3.383.
———. 2011. „Little's Law as Viewed on Its 50th Anniversary". *Operations Research* 59 (3): 536–49. https://doi.org/10.1287/opre.1110.0940.
Liu, Hu-Chen, Long Liu, és Nan Liu. 2013. „Risk Evaluation Approaches in Failure Mode and Effects Analysis: A Literature Review". *Expert Systems with Applications* 40 (2): 828–38. https://doi.org/10.1016/j.eswa.2012.08.010.
Löfgren, Martin, Lars Witell, és Anders Gustafsson. 2011. „Theory of Attractive Quality and Life Cycles of Quality Attributes". *The TQM Journal* 23 (2): 235–46.
Magyarország Kormánya. 2012. „1229/2012. (VII. 6.) Korm. határozat a Magyar Képesítési Keretrendszer bevezetéséhez kapcsolódó feladatokról". Melléklet: A Magyar Képesítési Keretrendszer szintjei.
Maier, Mark W. 1998. „Architecting Principles for Systems-of-Systems". *Systems Engineering* 1 (4): 267–84.
Maier, Mark W., és Eberhardt Rechtin. 2009. *The Art of Systems Architecting*. 3. kiad. Boca Raton, FL: CRC Press.
Matzler, Kurt, és Hans H. Hinterhuber. 1998. „How to Make Product Development Projects

More Successful by Integrating Kano's Model of Customer Satisfaction into Quality Function Deployment". *Technovation* 18 (1): 25–38.
Mavin, Alistair, Philip Wilkinson, Adrian Harwood, és Mark Novak. 2009. „Easy Approach to Requirements Syntax (EARS)". In *Proceedings of the 17th IEEE International Requirements Engineering Conference (RE'09)*, 317–22. IEEE. https://doi.org/10.1109/RE.2009.9.
McAtamney, Lynn, és E. Nigel Corlett. 1993. „RULA: A Survey Method for the Investigation of Work-Related Upper Limb Disorders". *Applied Ergonomics* 24 (2): 91–99. https://doi.org/10.1016/0003-6870(93)90080-S.
Meadows, Donella H. 2008. *Thinking in Systems: A Primer*. White River Junction, VT: Chelsea Green Publishing.
Mendelow, Aubrey L. 1981. „Environmental Scanning: The Impact of the Stakeholder Concept". In *Proceedings of the Second International Conference on Information Systems (ICIS 1981)*. Cambridge, MA.
Mendling, Jan, Hajo A. Reijers, és Wil M. P. van der Aalst. 2010. „Seven Process Modeling Guidelines (7PMG)". *Information and Software Technology* 52 (2): 127–36. https://doi.org/10.1016/j.infsof.2009.08.004.
Miller, George A. 1956. „The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information". *Psychological Review* 63 (2): 81–97.
Mitchell, Ronald K., Bradley R. Agle, és Donna J. Wood. 1997. „Toward a Theory of Stakeholder Identification and Salience: Defining the Principle of Who and What Really Counts". *Academy of Management Review* 22 (4): 853–86. https://doi.org/10.5465/amr.1997.9711022105.
Montgomery, Douglas C. 2017. *Design and Analysis of Experiments*. 9. kiad. Hoboken, NJ: Wiley.
———. 2020. *Introduction to Statistical Quality Control*. 8. kiad. Hoboken, NJ: Wiley.
Muchiri, Peter, és Liliane Pintelon. 2008. „Performance Measurement Using Overall Equipment Effectiveness (OEE): Literature Review and Practical Application Discussion". *International Journal of Production Research* 46 (13): 3517–35. https://doi.org/10.1080/00207540601142645.
Mullen, Brian, Craig Johnson, és Eduardo Salas. 1991. „Productivity Loss in Brainstorming Groups: A Meta-Analytic Integration". *Basic and Applied Social Psychology* 12 (1): 3–23. https://doi.org/10.1207/s15324834basp1201_1.
Nakajima, Seiichi. 1988. *Introduction to TPM: Total Productive Maintenance*. Cambridge, MA: Productivity Press.
NASA. 2015. „Human Systems Integration (HSI) Practitioner's Guide". NASA/SP-2015-3709. National Aeronautics; Space Administration.
———. 2016. „NASA Systems Engineering Handbook". NASA/SP-2016-6105 Rev2. Washington, DC: National Aeronautics; Space Administration.
Neumann, John von, és Oskar Morgenstern. 1944. *Theory of Games and Economic Behavior*. Princeton: Princeton University Press.
Newman, M. E. J. 2006. „Modularity and Community Structure in Networks". *Proceedings of the National Academy of Sciences* 103 (23): 8577–82. https://doi.org/10.1073/pnas.0601602103.
Nielsen, Jakob, és Thomas K. Landauer. 1993. „A Mathematical Model of the Finding of Usability Problems". In *Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems*, 206–13. New York: ACM. https://doi.org/10.1145/169059.169166.
NIST. 1993. „Integration Definition for Function Modeling (IDEF0)". Federal Information Processing Standards Publication 183.
Norman, Don. 2013. *The Design of Everyday Things*. Revised and expanded. New York: Basic Books.

O'Connor, Patrick D. T., és Andre Kleyner. 2012. *Practical Reliability Engineering*. 5. kiad. Chichester: Wiley.
Object Management Group. 2011. „Business Process Model and Notation (BPMN), Version 2.0". OMG Document formal/2011-01-03.
Ohno, Taiichi. 1988. *Toyota Production System: Beyond Large-Scale Production*. Cambridge, MA: Productivity Press.
OMG. 2025. „OMG Systems Modeling Language (SysML), Version 2.0". Object Management Group, formális specifikáció (a KerML 1.0 és a Systems Modeling API and Services 1.0 specifikációval együtt). https://www.omg.org/spec/SysML/2.0/.
Osborn, Alex F. 1953. *Applied Imagination: Principles and Procedures of Creative Thinking*. New York: Scribner.
Pahl, Gerhard, Wolfgang Beitz, Jörg Feldhusen, és Karl-Heinrich Grote. 2007. *Engineering Design: A Systematic Approach*. 3. kiad. London: Springer. https://doi.org/10.1007/978-1-84628-319-2.
Parasuraman, Raja, és Victor Riley. 1997. „Humans and Automation: Use, Misuse, Disuse, Abuse". *Human Factors* 39 (2): 230–53. https://doi.org/10.1518/001872097778543886.
Parasuraman, Raja, Thomas B. Sheridan, és Christopher D. Wickens. 2000. „A Model for Types and Levels of Human Interaction with Automation". *IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans* 30 (3): 286–97. https://doi.org/10.1109/3468.844354.
Parnell, Gregory S., szerk. 2017. *Trade-off Analytics: Creating and Exploring the System Tradespace*. Hoboken, NJ: John Wiley & Sons.
Parnell, Gregory S., Terry A. Bresnick, Steven N. Tani, és Eric R. Johnson. 2013. *Handbook of Decision Analysis*. Hoboken, NJ: John Wiley & Sons.
Phadke, Madhav S. 1989. *Quality Engineering Using Robust Design*. Englewood Cliffs, NJ: Prentice Hall.
Pinedo, Michael L. 2022. *Scheduling: Theory, Algorithms, and Systems*. 6. kiad. Cham: Springer. https://doi.org/10.1007/978-3-031-05921-6.
Price, Harold E. 1985. „The Allocation of Functions in Systems". *Human Factors* 27 (1): 33–45. https://doi.org/10.1177/001872088502700104.
Project Management Institute. 2019. *Practice Standard for Work Breakdown Structures*. 3. kiad. Newtown Square, PA: Project Management Institute.
Pugh, Stuart. 1991. *Total Design: Integrated Methods for Successful Product Engineering*. Wokingham: Addison-Wesley.
Rasmussen, Jens. 1983. „Skills, Rules, and Knowledge; Signals, Signs, and Symbols, and Other Distinctions in Human Performance Models". *IEEE Transactions on Systems, Man, and Cybernetics* SMC-13 (3): 257–66. https://doi.org/10.1109/TSMC.1983.6313160.
Reason, James. 1990. *Human Error*. Cambridge: Cambridge University Press.
———. 2000. „Human Error: Models and Management". *BMJ* 320 (7237): 768–70. https://doi.org/10.1136/bmj.320.7237.768.
Rittel, Horst W. J., és Melvin M. Webber. 1973. „Dilemmas in a General Theory of Planning". *Policy Sciences* 4 (2): 155–69.
Roedler, Garry J., és Cheryl Jones. 2005. „Technical Measurement: A Collaborative Project of PSM, INCOSE, and Industry". INCOSE-TP-2003-020-01. San Diego, CA: International Council on Systems Engineering.
Rohrbach, Bernd. 1969. „Kreativ nach Regeln – Methode 635, eine neue Technik zum Lösen von Problemen". *Absatzwirtschaft* 12: 73–76.
Rook, Paul. 1986. „Controlling Software Projects". *Software Engineering Journal* 1 (1): 7–16. https://doi.org/10.1049/sej.1986.0003.
Rother, Mike, és John Shook. 1999. *Learning to See: Value Stream Mapping to Add Value and Eliminate MUDA*. Brookline, MA: Lean Enterprise Institute.

Rowe, Gene, és George Wright. 1999. „The Delphi Technique as a Forecasting Tool: Issues and Analysis”. *International Journal of Forecasting* 15 (4): 353–75. https://doi.org/10.1016/S0169-2070(99)00018-7.
Royce, Winston W. 1970. „Managing the Development of Large Software Systems”. In *Proceedings of IEEE WESCON*, 1–9.
Rubin, Jeffrey, és Dana Chisnell. 2008. *Handbook of Usability Testing: How to Plan, Design, and Conduct Effective Tests*. 2. kiad. Indianapolis: Wiley.
Saaty, Thomas L. 1980. *The Analytic Hierarchy Process*. New York: McGraw-Hill.
———. 1990. „How to Make a Decision: The Analytic Hierarchy Process”. *European Journal of Operational Research* 48 (1): 9–26.
Saltelli, Andrea, Marco Ratto, Terry Andres, Francesca Campolongo, Jessica Cariboni, Debora Gatelli, Michaela Saisana, és Stefano Tarantola. 2008. *Global Sensitivity Analysis: The Primer*. Chichester: John Wiley & Sons.
Sargent, Robert G. 2013. „Verification and Validation of Simulation Models”. *Journal of Simulation* 7 (1): 12–24.
Sauro, Jeff, és James R. Lewis. 2016. *Quantifying the User Experience: Practical Statistics for User Research*. 2. kiad. Cambridge, MA: Morgan Kaufmann.
Scholl, Armin, és Christian Becker. 2006. „State-of-the-Art Exact and Heuristic Solution Procedures for Simple Assembly Line Balancing”. *European Journal of Operational Research* 168 (3): 666–93. https://doi.org/10.1016/j.ejor.2004.07.022.
Schroeder, Roger G., Kevin Linderman, Charles Liedtke, és Adrian S. Choo. 2008. „Six Sigma: Definition and Underlying Theory”. *Journal of Operations Management* 26 (4): 536–54. https://doi.org/10.1016/j.jom.2007.06.007.
Schwaber, Ken, és Jeff Sutherland. 2020. „The Scrum Guide: The Definitive Guide to Scrum: The Rules of the Game”. scrumguides.org.
SEBoK Editorial Board. 2025. „The Guide to the Systems Engineering Body of Knowledge (SEBoK), v. 2.13”. Hoboken, NJ: The Trustees of the Stevens Institute of Technology, https://sebokwiki.org.
Shah, Jami J., Steve M. Smith, és Noe Vargas-Hernandez. 2003. „Metrics for Measuring Ideation Effectiveness”. *Design Studies* 24 (2): 111–34. https://doi.org/10.1016/S0142-694X(02)00034-0.
Sheard, Sarah A. 1996. „Twelve Systems Engineering Roles”. In *Proceedings of the 6th Annual International Symposium of INCOSE*. Boston, MA.
Sheridan, Thomas B., és William L. Verplank. 1978. „Human and Computer Control of Undersea Teleoperators”. Cambridge, MA: MIT Man-Machine Systems Laboratory.
Shewhart, Walter A. 1931. *Economic Control of Quality of Manufactured Product*. New York: Van Nostrand.
Shingo, Shigeo. 1985. *A Revolution in Manufacturing: The SMED System*. Stamford, CT: Productivity Press.
———. 1986. *Zero Quality Control: Source Inspection and the Poka-Yoke System*. Cambridge, MA: Productivity Press.
Silver, Edward A., David F. Pyke, és Douglas J. Thomas. 2016. *Inventory and Production Management in Supply Chains*. 4. kiad. Boca Raton: CRC Press. https://doi.org/10.1201/9781315374406.
Simon, Herbert A. 1956. „Rational Choice and the Structure of the Environment”. *Psychological Review* 63 (2): 129–38. https://doi.org/10.1037/h0042769.
———. 1962. „The Architecture of Complexity”. *Proceedings of the American Philosophical Society* 106 (6): 467–82.
Snowden, David J., és Mary E. Boone. 2007. „A Leader’s Framework for Decision Making”. *Harvard Business Review* 85 (11): 68–76.
Sosa, Manuel E., Steven D. Eppinger, és Craig M. Rowles. 2004. „The Misalignment of

Product Architecture and Organizational Structure in Complex Product Development". *Management Science* 50 (12): 1674–89. https://doi.org/10.1287/mnsc.1040.0289.
Stanton, Neville A. 2006. „Hierarchical Task Analysis: Developments, Applications, and Extensions". *Applied Ergonomics* 37 (1): 55–79. https://doi.org/10.1016/j.apergo.2005.06.003.
Sterman, John D. 2000. *Business Dynamics: Systems Thinking and Modeling for a Complex World*. Boston: Irwin/McGraw-Hill.
Steward, Donald V. 1981. „The Design Structure System: A Method for Managing the Design of Complex Systems". *IEEE Transactions on Engineering Management* EM-28 (3): 71–74.
Sullivan, William G., Elin M. Wicks, és C. Patrick Koelling. 2019. *Engineering Economy*. 17. kiad. Hoboken, NJ: Pearson.
Swain, Alan D., és H. E. Guttmann. 1983. „Handbook of Human Reliability Analysis with Emphasis on Nuclear Power Plant Applications". NUREG/CR-1278. Washington, DC: U.S. Nuclear Regulatory Commission.
Taguchi, Genichi, Subir Chowdhury, és Yuin Wu. 2005. *Taguchi's Quality Engineering Handbook*. Hoboken, NJ: Wiley.
Tversky, Amos, és Daniel Kahneman. 1974. „Judgment under Uncertainty: Heuristics and Biases". *Science* 185 (4157): 1124–31.
Ulrich, Karl. 1995. „The Role of Product Architecture in the Manufacturing Firm". *Research Policy* 24 (3): 419–40.
U.S. Department of Defense. 1980. „MIL-STD-1629A Procedures for Performing a Failure Mode, Effects and Criticality Analysis". U.S. Department of Defense, Washington, DC.
———. 2011. „MIL-STD-46855A Department of Defense Standard Practice: Human Engineering Requirements for Military Systems, Equipment, and Facilities". U.S. Department of Defense, Washington, DC.
———. 2012. „MIL-STD-882E Department of Defense Standard Practice: System Safety". U.S. Department of Defense, Washington, DC.
———. 2020. „MIL-HDBK-61B: Configuration Management Guidance". Kézikönyv.
———. 2022. „MIL-STD-881F: Work Breakdown Structures for Defense Materiel Items". Szabvány.
VDA QMC. 2018. „8D – Problem Solving in 8 Disciplines: Method, Process, Report". Verband der Automobilindustrie, Qualitäts Management Center, Berlin, 1. kiadás.
Vesely, William E., Francine F. Goldberg, Norman H. Roberts, és David F. Haasl. 1981. „Fault Tree Handbook". NUREG-0492. Washington, DC: U.S. Nuclear Regulatory Commission.
Vincent, Julian F. V., Olga A. Bogatyreva, Nikolaj R. Bogatyrev, Adrian Bowyer, és Anja-Karina Pahl. 2006. „Biomimetics: Its Practice and Theory". *Journal of the Royal Society Interface* 3 (9): 471–82.
Virzi, Robert A. 1992. „Refining the Test Phase of Usability Evaluation: How Many Subjects Is Enough?" *Human Factors* 34 (4): 457–68. https://doi.org/10.1177/001872089203400407.
Vogel, Steven. 1998. *Cats' Paws and Catapults: Mechanical Worlds of Nature and People*. New York: W. W. Norton.
Voirin, Jean-Luc. 2017. *Model-Based System and Architecture Engineering with the Arcadia Method*. London: ISTE Press – Elsevier.
Wallas, Graham. 1926. *The Art of Thought*. New York: Harcourt, Brace; Company.
Weck, Olivier L. de, Daniel Roos, és Christopher L. Magee. 2011. *Engineering Systems: Meeting Human Needs in a Complex Technological World*. Cambridge, MA: MIT Press.
Weibull, Waloddi. 1951. „A Statistical Distribution Function of Wide Applicability". *Journal of Applied Mechanics* 18 (3): 293–97.

Wheatcraft, Louis S., Michael J. Ryan, és Tami Edner Katz. 2025. *INCOSE Needs and Requirements Manual: Needs, Requirements, Verification, Validation Across the Lifecycle*. Hoboken, NJ: John Wiley & Sons.

Williams, Christine, és Mary-Ellen Derro. 2008. „NASA Systems Engineering Behavior Study". NASA Office of the Chief Engineer. https://appel.nasa.gov/critical-knowledge/case-studies/nasa-systems-engineering-behavior-study/.

Williams, J. C. 1988. „A Data-Based Method for Assessing and Reducing Human Error to Improve Operational Performance". In *Proceedings of the IEEE Fourth Conference on Human Factors and Power Plants*, 436–50. Monterey, CA.

Wilson, James M. 2003. „Gantt Charts: A Centenary Appreciation". *European Journal of Operational Research* 149 (2): 430–37. https://doi.org/10.1016/S0377-2217(02)00769-5.

Winter, J. C. F. de, és D. Dodou. 2014. „Why the Fitts List Has Persisted throughout the History of Function Allocation". *Cognition, Technology & Work* 16 (1): 1–11. https://doi.org/10.1007/s10111-011-0188-1.

Zwicky, Fritz. 1969. *Discovery, Invention, Research through the Morphological Approach*. New York: Macmillan.

A. A futó esettanulmányok

A könyv minden módszert ugyanazon a három rendszeren mutat be. Az esetek szándékosan különböző világokból valók: egy mechatronikai gyártóberendezés, egy városi szolgáltatás alkalmazással és infrastruktúrával, valamint egy gyártóüzem működési folyamatai. Így ugyanaz a technika háromféle környezetben is látható, és kiderül, mi benne az általános, és mi múlik a rendszer természetén.

Mindhárom cég, város és szereplő kitalált. A számadatok szemléltető jellegűek: nagyságrendjük életszerű, de nem valós vállalatoktól származnak. Az esetek a fejezetekben színes dobozokban térnek vissza. A kék doboz az A, a zöld a B, a lila a C esetet jelöli.

A.1. A eset – KoboCella: kollaboratív robotos összeszerelő cella

Esetkártya: KoboCella

Szervezet: Kobold Hajtástechnika Kft., 180 fős beszállító, elektromos kerékpárok agymotorjait szereli össze.
Kiinduló helyzet: a motorfedél felhelyezése és a nyolc csavar meghúzása jelenleg kézi munkaállomáson zajlik. Az éves igény két év alatt várhatóan 90 ezerről 140 ezer darabra nő. Az operátorok csukló- és vállpanaszokra panaszkodnak. A nem megfelelő nyomatékkal meghúzott csavarok miatt a darabok mintegy 3,5%-a utómunkára kerül.
Fejlesztési cél: olyan kollaboratív robotos cella kialakítása, amelyben az operátor és egy kobot (együttműködő robot) megosztva végzi a munkát. A cellának a növekvő igényt ki kell szolgálnia, csökkentenie kell az ergonómiai terhelést és a selejtet, és illeszkednie kell a gyár gyártásirányítási (MES) rendszeréhez.
Miért tanulságos? Mechatronikai rendszer, amelyben a mechanika, az irányítás, a szoftver és az ember együtt határozza meg a teljesítményt. Kiemelten jól mutatja az emberi rendszerintegrációt, a funkciómegosztást, a biztonságkritikus tervezést, az FMEA-t és a berendezéshatékonyságot (OEE).

Fő érdekelt felek: - az operátorok és a műszakvezetők; - a karbantartás és a minőségügy; - a munkavédelem; - az üzemi tanács; - a gyár vezetése és pénzügye; - a kerékpárgyártó vevő; - a robotintegrátor; - az informatika.

A könyvben feldolgozott kérdések (példák): ki mit csináljon a cellában, az ember vagy a robot? Hogyan mérjük, hogy a cella „jó”? Milyen hibamódok fenyegetnek? Mekkora a cella valódi kapacitása, ha a robot időnként leáll?

A.2. B eset – VárosiBringa: közösségi elektromos kerékpárrendszer

Esetkártya: VárosiBringa

Szervezet: Bakonyvár megyei jogú város önkormányzata és az üzemeltetésre kiválasztott konzorcium. Bakonyvár kitalált, mintegy 60 ezer lakosú, dombos fekvésű egyetemi város.
Kiinduló helyzet: a belváros zsúfolt, a parkolóhelyek szűkösek. A város klímastratégiája csökkenteni kívánja a rövid autós utak számát. A domborzat miatt a hagyományos kerékpár nem vonzó a lakosság jelentős része számára.
Fejlesztési cél: 300 elektromos rásegítésű kerékpárból, 30 dokkolóállomásból és mobilalkalmazásból álló közösségi rendszer. A rendszert be kell illeszteni a helyi közösségi közlekedésbe, gazdaságosan kell üzemeltetni, és vonzónak kell lennie diákok, ingázók, turisták és idősebb lakók számára egyaránt.
Miért tanulságos? Termék, infrastruktúra, szoftver és szolgáltatás egyben. Sok, egymásnak sokszor ellentmondó érdekelt fél van benne, a siker pedig a felhasználói viselkedésen múlik. Kiválóan szemlélteti az érdekeltfél-elemzést, a Kano-modellt, a QFD-t, a többszempontú döntést és a szolgáltatásfolyamatok modellezését.

Fő érdekelt felek: - a lakosok, köztük a felhasználók és a nem felhasználók (gyalogosok, üzlettulajdonosok); - az önkormányzat; - az üzemeltető konzorcium; - a helyi közlekedési társaság; - az egyetem; - a karbantartók és a „töltő-áthelyező” szerviz; - az alkalmazásfejlesztő; - az energiaszolgáltató; - a rendőrség; - a fogyatékossággal élők érdekképviselete; - a környezetvédő civil szervezetek.

A könyvben feldolgozott kérdések (példák): kinek a hangja számít, és mennyire? Mely szolgáltatásjellemzők elvártak, és melyek okoznak meglepetésszerű elégedettséget? Hány állomás és hol? Hogyan alakul az áthelyezési folyamat költsége?

A.3. C eset – ForgácsTech: egy alkatrészgyártó üzem digitális átállása

Esetkártya: ForgácsTech

Szervezet: ForgácsTech Kft., 85 fős, egyedi és kis sorozatú CNC-megmunkálással foglalkozó alkatrészgyártó. Vevői főként járműipari és gépipari beszállítók.
Kiinduló helyzet: a rendelések útja a nyersanyag darabolásától a kész munkadarabig papíralapú műhelyrajzokon és szóbeli egyeztetéseken alapul. A határidő-csúszások gyakoriak. Senki sem tudja pontosan, mennyi ideig állnak a gépek. A vevők egyre gyakrabban kérnek nyomonkövethetőséget és rövidebb átfutást.
Fejlesztési cél: a folyamatok feltárása és modellezése, gyártásirányítási rendszer (MES) bevezetése, a gépek hatékonyságának mérése, átláthatóbb ütemezés és digitális munkautasítások.
Miért tanulságos? Itt a rendszer maga a szervezet és a folyamat. Ebben az esetben a folyamatelemzés, a Little-törvény, a szűk keresztmetszet, az OEE, a BPMN-modellezés, az ütemezés és a folyamatos fejlesztés eszközei kerülnek előtérbe.

Fő érdekelt felek: - a tulajdonos-ügyvezető; - a termelésvezető; - a CNC-gépkezelők és -programozók; - a gyártáselőkészítés; - a minőségügy; - a logisztika; - a vevők; - a MES-szállító; - a könyvelés.

A könyvben feldolgozott kérdések (példák): hogyan halad valójában egy rendelés az üzemen át? Hol a szűk keresztmetszet? Mekkora a gépek valós hatékonysága? Hogyan lehet a változtatást úgy bevezetni, hogy a gépkezelők támogassák?

B. Magyar–angol szószedet

A szószedet a könyvben használt fontosabb szakkifejezéseket és angol megfelelőiket gyűjti össze. A rövidítések az angol szakirodalomban használt alakban szerepelnek. A magyarázatok tömörek; a pontos meghatározást és a forrásokat az adott fejezet tartalmazza.

B.1. táblázat. A könyv szakkifejezései és angol megfelelőik.

Magyar	Angol	Rövid magyarázat
100%-szabály	100% rule	A munkalebontás minden szintje pontosan a felette álló elem teljes munkáját tartalmazza
alapvető (elvárt) jellemző	must-be quality	Kano-kategória: hiánya elégedetlenséget okoz, megléte nem növeli az elégedettséget
alapvonal	baseline	A rendszerleírás jóváhagyott, csak szabályozott változáskezeléssel módosítható állapota
allokált alapvonal	allocated baseline	Az alrendszerek és komponensek jóváhagyott teljesítményleírása
analitikus hierarchia eljárás	analytic hierarchy process (AHP)	Páros összehasonlításokon alapuló súlyozási és rangsorolási módszer
architektúra	architecture	A rendszer alapvető fogalmai és tulajdonságai, elemei és kapcsolatai, valamint a megvalósítását és fejlődését irányító elvek
architektúra-meghatározás	architecture definition	Az ISO/IEC/IEEE 15288 folyamata: architektúraváltozatok létrehozása, értékelése és a választott architektúra rögzítése nézetekben és modellekben
bemutatás	demonstration	Verifikációs mód: a működés megfigyelése, mérés nélkül vagy csak egyszerű méréssel
biztonsági integritási szint	safety integrity level (SIL)	A biztonsági funkció megkövetelt megbízhatóságának szintje (IEC 61508)
célkitűzés	objective	Konkrét, mérhető célszint, amely egy célhoz kapcsolódik
csatolás	coupling	Az elemek közötti függés erőssége; a jó architektúrában kicsi
cselekvési prioritás	action priority (AP)	Az FMEA rangsoroló eljárása a súlyosság, az előfordulás és az észlelés együttese alapján

Magyar	Angol	Rövid magyarázat
digitális fonal	digital thread	Az életciklus adatait összekötő, nyomon követhető adatkapcsolat
egydimenziós jellemző	one-dimensional quality	Kano-kategória: az elégedettség a teljesüléssel arányosan nő
elemzés	analysis	Verifikációs mód: számítással, modellel vagy szimulációval igazolt teljesülés
életciklus-modell	life cycle model	A fejlesztési tevékenységek sorrendjének és visszacsatolásainak modellje
előzetes tervfelülvizsgálat	preliminary design review (PDR)	Annak értékelése, hogy az előzetes terv elfogadható kockázattal teljesítheti-e a követelményeket
előzetes veszélyelemzés	preliminary hazard analysis (PHA)	A veszélyek korai, rendszerszintű azonosítása és értékelése
eltérési engedély	waiver	Dokumentált megállapodás egy követelmény teljesítésének szándékos elengedéséről; az alapvonal nem változik
emberi megbízhatóság-elemzés	human reliability analysis (HRA)	Az emberi hibák valószínűségének és következményeinek becslése
emberi rendszerintegráció	human systems integration (HSI)	Az emberi szempontok beépítése a rendszerfejlesztés teljes életciklusába
érdekelt fél	stakeholder	Akit a rendszer érint, vagy aki hatással van rá
értékáram-térkép	value stream mapping (VSM)	Az anyag- és információáramlás, valamint az átfutási idők ábrázolása
eseményfa-elemzés	event tree analysis (ETA)	Egy kezdeti eseményből kiinduló lehetséges kimenetek és valószínűségeik elemzése
fizikai audit	physical configuration audit	Annak igazolása, hogy a termék és a dokumentációja összhangban van
folyamatbányászat	process mining	Folyamatmodellek feltárása és ellenőrzése eseménynaplókból
folyamatos kockázatkezelés	continuous risk management (CRM)	Ismétlődő ciklus: azonosítás, elemzés, tervezés, követés, irányítás
fontossági (szaliencia-) modell	stakeholder salience model	Az érdekelt felek osztályozása hatalom, legitimitás és sürgősség alapján
funkciófa	functional hierarchy	A funkciók hierarchikus lebontása a fő funkciótól a részfunkciókig
funkcionális alapvonal	functional baseline	A rendszerszintű, teljesítményben megfogalmazott jóváhagyott leírás
funkcionális audit	functional configuration audit	Annak igazolása, hogy a konfigurációs elem teljesíti a követelményeit
funkcionális folyamatábra	functional flow block diagram (FFBD)	A funkciók sorrendjét és logikai kapcsolatait mutató ábra
gyári átvételi teszt	factory acceptance test (FAT)	Átvételi vizsgálat a gyártónál, a szállítás előtt

Magyar	Angol	Rövid magyarázat
hagymamodell	onion model	Az érdekelt felek szerepeinek koncentrikus rendszerezése
használati környezet	context of use	A felhasználók, a feladatok, az eszközök és a környezet, amelyben a terméket használják
használhatóság	usability	Milyen hatékonyan, eredményesen és elégedetten érhetik el céljaikat a felhasználók adott használati környezetben
hatalom–érdekeltség rács	power–interest grid	Az érdekelt felek négy negyedbe sorolása
hatáselemzés	impact analysis	Annak feltárása, mely elemeket, dokumentumokat és tevékenységeket érint egy változtatás
hatékonyság mérőszáma	measure of effectiveness (MOE)	A rendszer egészének sikerét az érdekelt felek szemszögéből mérő mutató
helyszíni átvételi teszt	site acceptance test (SAT)	Átvételi vizsgálat a telepítés helyén
helyzettudatosság	situation awareness	A környezet elemeinek észlelése, megértése és jövőbeli állapotuk előrejelzése
hibafa-elemzés	fault tree analysis (FTA)	Egy nemkívánatos csúcseseményhez vezető okok logikai (ÉS, VAGY) elemzése
hibajelenség	failure	Az az esemény, amikor egy elem nem látja el a megkövetelt funkcióját
hibamód- és hatáselemzés	failure mode and effects analysis (FMEA)	A lehetséges hibamódok, hatásaik és okaik rendszeres feltárása és rangsorolása
hibaok	fault	A hibajelenség oka, például alkatrész-, tervezési vagy szoftverhiba
hierarchikus feladatelemzés	hierarchical task analysis (HTA)	A feladatok célokra és részfeladatokra bontása, a végrehajtás terveivel
integráció	integration	Az elemek lépésenkénti összeépítése és a kapcsolatok ellenőrzése
interfészkövetelmény-dokumentum	interface requirements document (IRD)	Egy interfész követelményeinek leírása
interfészvezérlő dokumentum	interface control document (ICD)	Egy interfész jóváhagyott, részletes műszaki leírása
iteráció	iteration	Egy szinten belüli ismételt végighaladás a folyamatokon
jelentős változás	major change	A specifikációt, a költséget, a biztonságot, az interfészeket, a képzést vagy a leszállított terméket érintő változás
kár	harm	Személyi sérülés, egészségkárosodás, vagyoni vagy környezeti kár
képességindex	process capability index (Cp, Cpk)	A folyamat szórásának és helyzetének viszonya a tűréshatárokhoz

Magyar	Angol	Rövid magyarázat
keretadó	context setter	Nagy hatalmú, de kis érdekeltségű érdekelt fél
kihagyás	lapse	Emlékezeti hiba: egy szándékolt lépés elmarad
kisebb változás	minor change	A dokumentációt vagy a folyamatot javító változás, amely a cserélhetőséget nem érinti
kísérlettervezés	design of experiments (DoE)	Kísérletek tervezése a tényezők hatásainak és kölcsönhatásainak hatékony becslésére
kockázat	risk	Egy nemkívánatos esemény valószínűségének és következményének együttese
kockázatinformált döntéshozatal	risk-informed decision making (RIDM)	Döntés a változatok kockázatainak és bizonytalanságainak ismeretében
konfigurációkezelés	configuration management	A konfiguráció azonosítása, a változások szabályozása, nyilvántartása és auditálása
konfigurációs bizottság	configuration control board (CCB)	A jelentős változásokról döntő testület
konfigurációs elem	configuration item (CI)	Konfigurációkezelés alá vont dokumentum, hardver, szoftver, adat vagy modell
kontrollkártya	control chart	Mintastatisztika időbeli ábrázolása középvonallal és beavatkozási határokkal
követelmény-nyomonkövetési mátrix	requirements traceability matrix (RTM)	A követelmények forrásainak, leszármazottainak és verifikációjának kapcsolatai
követelmény-verifikációs mátrix	requirements verification matrix (RVM)	Követelményenként a verifikáció módja, sikerkritériuma, felelőse és eredménye
kritikus út	critical path	A projekt időtartamát meghatározó, tartalék nélküli tevékenységsorozat
kulcsszereplő	player	Nagy hatalmú és nagy érdekeltségű érdekelt fél
megbízhatósági blokkdiagram	reliability block diagram (RBD)	A rendszer megbízhatósági szerkezetét soros és párhuzamos blokkokkal leíró modell
megcsúszás	slip	Figyelmi hiba: a helyes szándék hibás végrehajtása
megszerzett érték	earned value (EV)	Az adott időpontig elvégzett munka tervezett költsége
megszerzett érték módszere	earned value management (EVM)	A műszaki tartalom, az ütemezés és a költség együttes teljesítménymérése
minőségfunkció-telepítés	quality function deployment (QFD)	A vevői elvárások műszaki jellemzőkre fordítása mátrixok láncolatával
minőségház	house of quality	A QFD első mátrixa: az elvárások és a műszaki jellemzők kapcsolata

Magyar	Angol	Rövid magyarázat
modellalapú rendszermérnökség	model-based systems engineering (MBSE)	Rendszermérnöki munka dokumentumok helyett összekapcsolt modellekre építve
morfológiai doboz	morphological box	A részfunkciók megoldásváltozatainak táblázata a koncepciók kombinálásához
munkalebontási struktúra	work breakdown structure (WBS)	A projekt teljes munkájának termékorientált, hierarchikus felosztása
működési koncepció	concept of operations (ConOps)	A rendszer használatának leírása a felhasználó szemszögéből
műszaki teljesítménymutató	technical performance measure (TPM)	A kritikus műszaki jellemzők követése a fejlesztés során a célértékhez képest
nyilvántartás (konfigurációs)	configuration status accounting	A konfigurációk, a változáskérelmek és az eltérések állapotának rögzítése és jelentése
nyomonkövethetőség	traceability	A követelmények kapcsolata forrásaikkal, a tervvel és a verifikációval
N²-diagram	N-squared diagram	Az elemek közötti interfészeket mátrixban ábrázoló diagram
ötletroham	brainstorming	Ötletgenerálás csoportban, az értékelés elhalasztásával
Pareto-front	Pareto front	A nem dominált megoldások halmaza több célkitűzés esetén
rekurzió	recursion	Ugyanazon folyamatok alkalmazása a termékszerkezet egymás alatti szintjein
rendszerek rendszere	system of systems (SoS)	Önállóan is működő rendszerek együttese, amely új képességet hoz létre
rendszerelméleti folyamatelemzés	system-theoretic process analysis (STPA)	Veszélyelemzés a nem biztonságos irányítási műveletek feltárásával
rendszermérnöki irányítási terv	systems engineering management plan (SEMP)	A projekt legfelső szintű technikai irányítási dokumentuma
részletes tervfelülvizsgálat	critical design review (CDR)	A gyártásra, kódolásra kész terv értékelése
reteszelés	interlock	Műszaki megoldás, amely veszélyes állapotban megakadályozza egy művelet végrehajtását
robusztus tervezés	robust design	A teljesítmény érzéketlenné tétele a zajtényezőkre és a gyártási szórásra
sikeres futás próba	success run testing	Hibátlan próbák számából számolt alsó megbízhatósági korlát
sorkiegyenlítés	line balancing	A műveletek munkaállomásokra osztása az ütemidő figyelembevételével
szabályszegés	violation	Szándékos eltérés az előírt eljárástól

Magyar	Angol	Rövid magyarázat
szakmai átvizsgálás	peer review	Egy munkatermék részletes vizsgálata kollégák által, kisebb körben
szükséglet	need	A megoldandó probléma megoldásfüggetlen megfogalmazása
tartalék (ütemezési)	float, slack	Ennyivel késhet egy tevékenység a projekt befejezésének csúszása nélkül
technikai felülvizsgálat	technical review	Bizonyítékokra épülő, tervezett értékelés a szakaszok határán
teljes eszközhatékonyság	overall equipment effectiveness (OEE)	A rendelkezésre állás, a teljesítmény és a minőség szorzata
teljesítmény mérőszáma	measure of performance (MOP)	A rendszer egy műszaki jellemzőjét mérő mutató, amely a hatékonyság mérőszámait támogatja
tényleges költség	actual cost (AC)	Az adott időpontig elvégzett munka tényleges költsége
termékalapvonal	product baseline	A gyártáshoz szükséges részletes, jóváhagyott leírás
tervezési struktúramátrix	design structure matrix (DSM)	Az elemek vagy tevékenységek közötti függéseket négyzetes mátrixban ábrázoló módszer
tervezett érték	planned value (PV)	Az adott időpontig tervezett munka költsége
teszt	test	Verifikációs mód: mérés ellenőrzött körülmények között
tévedés	mistake	Hiba a szándékban vagy a tervben
ütemidő	takt time	A rendelkezésre álló munkaidő és a kereslet hányadosa
validáció	validation	Annak igazolása, hogy a rendszer a tervezett környezetben betölti rendeltetését
védelmi rétegek elemzése	layer of protection analysis (LOPA)	A független védelmi rétegek elégségességének félkvantitatív vizsgálata
verifikáció	verification	Annak igazolása, hogy a rendszer megfelel a követelményeinek
veszély	hazard	A kár lehetséges forrása
vizsgálat	inspection	Verifikációs mód: szemrevételezés, méret- vagy dokumentumellenőrzés
vonzó jellemző	attractive quality	Kano-kategória: hiánya nem okoz elégedetlenséget, megléte kiemelkedő elégedettséget kelt



Pannon Egyetem

University of Pannonia

Nyílt tankönyv · https://complexsystemsintelligence.github.io/rendszermernokseg/

CC BY-NC-SA 4.0