Pannon Egyetem
University of Pannonia

Kézikönyv

Biztonságkritikus rendszerek

Kockázat, megbízhatóság és funkcionális biztonság
a veszélyazonosítástól a biztonsági érvelésig

Abonyi János

Pannon Egyetem · Veszprém, 2026

Pannon Egyetem
University of Pannonia

Kézikönyv

Biztonságkritikus rendszerek

Kockázat, megbízhatóság és funkcionális biztonság a veszélyazonosítástól a biztonsági érvelésig

Abonyi János

Pannon Egyetem
University of Pannonia

Veszprém, 2026



Abonyi János: Biztonságkritikus rendszerek – Kézikönyv

Kockázat, megbízhatóság és funkcionális biztonság a veszélyazonosítástól a biztonsági érvelésig

© 2026 Abonyi János

Kiadó: Pannon Egyetem, Veszprém

Változat: 1.0.0, 2026. november 21.

A mű a Creative Commons Nevezd meg! – Ne add el! – Így add tovább! 4.0 Nemzetközi licenc (CC BY-NC-SA 4.0) feltételei szerint használható fel; a mellékelt programkód licence MIT.

Webes változat: https://complexsystemsintelligence.github.io/biztonsagkritikus-rendszerek/kezikonyv/

Forrás, hibajelzés és javaslat: https://github.com/ComplexSystemsIntelligence/biztonsagkritikus-rendszerek

DOI: 10.5281/zenodo.24413011 (a kiadások közös azonosítója a Zenodón)

Betűkészletek: IBM Plex Serif, IBM Plex Sans, Raleway (SIL Open Font License 1.1). Szedés: Quarto és XeLaTeX.

Tartalom

Előszó 1

I. rész – Alapok 5

1. Miért hibáznak a rendszerek? Balesetek és balesetmodellek 7
2. Fogalmak: veszély, kockázat, biztonság, megbízhatóság 17
3. Kockázatértékelés és a kockázat elfogadhatósága 29
4. A biztonsági életciklus és a szabványok rendszere 41

II. rész – Veszélyazonosítás 55

5. Veszélyazonosítás: a veszélylistáktól a HAZOP-ig 57
6. FMEA, FMECA és FMEDA 67
7. Rendszerelméleti veszélyelemzés: STAMP és STPA 79

III. rész – Kvantitatív megbízhatóság és biztonság 89

8. Megbízhatóság-elmélet és élettartam-adatok 91
9. Rendszermegbízhatóság: blokkdiagramok és k-ból-n rendszerek 103
10. Hibafa-elemzés 113
11. Eseményfa, rétegzett védelem és csokornyakkendő-modell 129
12. Markov-modellek és rendelkezésre állás 141
13. A biztonsági integritás igazolása: PFD, PFH és SIL 151

IV. rész – Tervezés biztonságra 163

14. Biztonságos architektúrák és tervezési elvek 165
15. A szoftver biztonsága 175
16. Az ember a biztonságkritikus rendszerben 185

V. rész – Bizonyítás és új kihívások 193

17. Biztonsági érvelés és igazolás 195

18. Új kihívások: rendeltetésszerű funkció, mesterséges intelligencia és kiberbiztonság 203

Irodalom 213

Függelékek 221

A. Valószínűségszámítási összefoglaló 221

B. A három esettanulmány 225

C. Magyar–angol szószedet 229

D. Szabványok és jogszabályok 233

Előszó

Egy vegyipari reaktor hűtése leáll. Egy autó radarja egy kidobott fémdobozt gyalogosnak lát. Egy kollaboratív robot karja a megengedettnél gyorsabban közelít a mellette dolgozó szerelőhöz. Mindhárom helyzetben ugyanaz a kérdés: elvégzi-e a rendszer akkor is a dolgát, amikor valami elromlik, vagy amikor a környezete másképp viselkedik, mint ahogy a tervezők gondolták. Ez a könyv arról szól, hogyan lehet erre a kérdésre megalapozott, számokkal és érvekkel alátámasztott választ adni.

A biztonságkritikus rendszer olyan rendszer, amelynek hibás működése emberéletet, egészséget, a természeti környezetet vagy jelentős anyagi értéket veszélyeztethet. Ilyen a folyamatipari védelmi rendszer, a jármű fék- és kormányrendszere, a vasúti biztosítóberendezés, a repülőgép repülésvezérlése, az orvostechnikai eszköz és az ipari robotcella. A felsorolás egyre hosszabb, mert a szoftver, a hálózati kapcsolat és a gépi tanulás egyre több olyan döntést hoz meg, amelyet korábban ember hozott meg.

Kinek szól a könyv?

Minden mérnöknek és szakembernek, aki biztonságkritikus rendszert tervez, fejleszt, üzemeltet, értékel vagy engedélyez, iparágtól függetlenül. A gépészmérnök, a vegyészmérnök, a villamosmérnök, a mechatronikai mérnök, a mérnökinformatikus és a minőségügyi szakember ugyanazokra a módszerekre támaszkodik, csak más szabványok más szavaival. A könyv ezért a közös elméleti magot tanítja, és mindig megmutatja, hogyan jelenik meg ugyanaz a gondolat a folyamatiparban (IEC 61511), a közúti járműveknél (ISO 26262, ISO 21448), a gépiparban (ISO 13849, IEC 62061), a vasútnál (EN 50126-sorozat), a repülésben (ARP4754B, ARP4761A, DO-178C) és az orvostechnikában (ISO 14971, IEC 62304).

A könyv mesterképzésben és posztgraduális képzésben tankönyvként, a gyakorló mérnök számára kézikönyvként használható. Az előismeretek szerények: valószínűségszámítási alapok (amelyeket az A. függelék összefoglal), alapfokú lineáris algebra és differenciálegyenletek, valamint mérnöki rendszerszemlélet.

Mit ad a könyv?

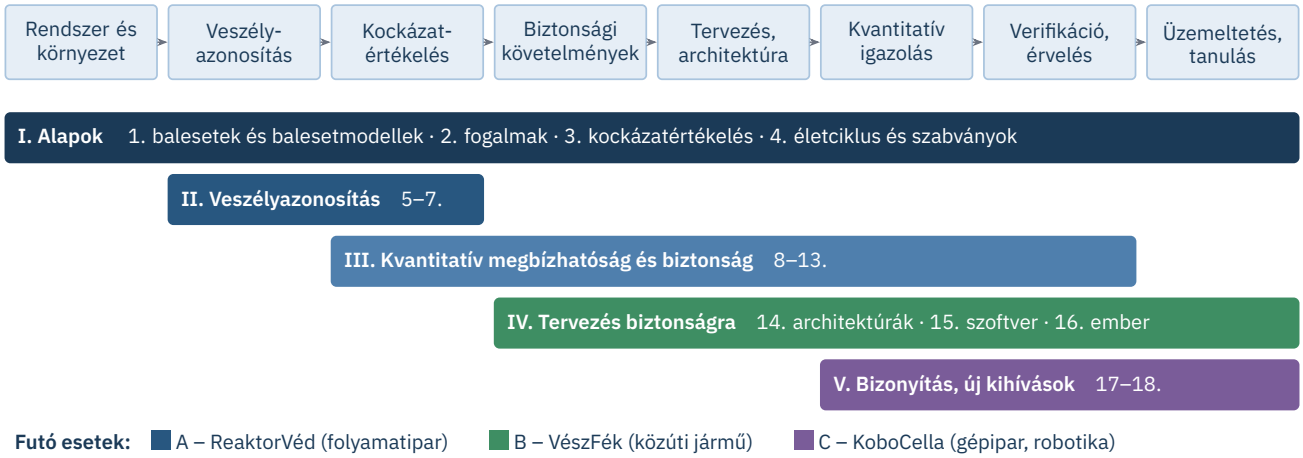
A könyv három dologra tanít meg.

1. **Látni a veszélyt.** A balesetek mintázatainak és a veszélyazonosítás módszereinek (HAZOP, FMEA, STPA) megismerése után az olvasó módszeresen tárja fel, mi romolhat el, és milyen következménnyel.
2. **Számolni a kockázattal.** A megbízhatóság-elmélet, a blokkdiagramok, a hibafák, az eseményfák, a Markov-modellek és a biztonsági funkciók PFD- és PFH-számítása megmutatja, mekkora a maradó kockázat, és hol érdemes csökkenteni.

3. **Érvelni a biztonságról.** A biztonsági architektúrák, a szoftverfejlesztési és emberi tényezős követelmények, valamint a biztonsági érvelés (safety case) eszközei abban segítenek, hogy a rendszer biztonságát a megrendelő, a hatóság és a független értékelő számára is átláthatóan igazolni lehessen.

A könyv felépítése

A könyv a biztonsági életciklust követi (1. ábra). Az I. rész leteszi az alapokat: a balesetek tanulságait, a fogalmakat, a kockázat elfogadhatóságának kérdését és a szabványok rendszerét. A II. rész a veszélyazonosítás három nagy módszercsaládját mutatja be. A III. rész a könyv módszertani gerince: a kvantitatív megbízhatósági és biztonsági modellezés a valószínűségi alapoktól a biztonsági funkciók integritásának igazolásáig. A IV. rész a tervezési elvekkel, a szoftverrel és az emberrel foglalkozik. Az V. rész a biztonsági érvelést és a legújabb kihívásokat tárgyalja: a rendeltetésszerű funkció biztonságát (SOTIF), a mesterséges intelligenciát és a kiberbiztonság és a biztonság kapcsolatát.



1. ábra. A könyv felépítése a biztonsági életciklus mentén. A színes sávok azt mutatják, hogy az egyes részek az életciklus mely szakaszait fedik le.

Minden fejezet egy kis életciklus-térképpel kezdődik, amely megmutatja, hol tartunk. A fejezetek elején tanulási eredmények állnak, a végén összefoglalás, önellenőrző kérdések (megoldással) és válogatott továbbolvasnivaló.

Három futó eset

A módszereket három, a könyvön végigvezetett esettanulmány köti össze. Mindhárom kitalált, de valós mérnöki problémákat modellez, a számértékek a szakirodalomban szokásos nagyságrendekben mozognak. Részletes leírásuk a B. függelékben található.

> **A – ReaktorVéd: exoterm reaktor védelmi rendszere**
>
> Félszakaszos üzemű keverős reaktor, amelyben exoterm reakció zajlik. Ha a hűtés kiesik, a hőmérséklet megfuthat, a nyomás a tartály teherbírása fölé nőhet. Ezen az eseten követjük végig a HAZOP-ot, a rétegzett védelem elemzését (LOPA), a biztonsági integritási szint (SIL) kiosztását, a hibafát és a biztonsági funkció PFD-számítását.

B – VészFék: automatikus vészfékező asszisztens

Személygépkocsi radaros és kamerás vészfékező asszisztense (AEB) elektrohidraulikus fékaktuátorral. Ezen az eseten mutatjuk be a veszélyelemzést és kockázatbecslést (HARA), az ASIL-besorolást, az FMEA-t és FMEDA-t, a hibatűrő architektúrákat, valamint a rendeltetésszerű funkció biztonságát és a gépi tanulás biztonsági kérdéseit.

C – KoboCella: kollaboratív robotcella

Összeszerelő cella, amelyben ember és kollaboratív robot közös munkatérben dolgozik. Ezen az eseten mutatjuk be a gépbiztonsági kockázatértékelést (ISO 12100), a szükséges teljesítményszint (PL) meghatározását, a rendszerelméleti veszélyelemzést (STPA) és az emberi tényezők szerepét. A KoboCella a sorozat *Rendszermérnökség* kötetében is szerepel; itt a biztonsági szempontjait bontjuk ki.

Válogatott útvonalak

A fejezetek önállóan is olvashatók. Néhány javasolt útvonal:

1. táblázat. Javasolt olvasási útvonalak

Útvonal	Ajánlott fejezetek
Gyors bevezetés a biztonságtechnikába	1., 2., 3., 4., 5., 17.
Folyamatbiztonság (IEC 61511)	2., 3., 5., 10., 11., 13., 14.
Közúti járművek (ISO 26262, ISO 21448)	2., 4., 5., 6., 14., 15., 18.
Gépbiztonság és robotika	3., 4., 7., 9., 13., 16.
Megbízhatósági modellezés	8., 9., 10., 11., 12., A. függelék
Szoftver és MI-alapú rendszerek	7., 15., 17., 18.

A gyakorlati útmutató

A kézikönyvhöz külön kötet tartozik: a *Gyakorlati útmutató*. Ebben minden módszerhez lépésenként vezetett feladat, böngészőben futó interaktív alkalmazás és Google Colabban futtatható Python-notebook tartozik. A kézikönyv szövegében a **Gyakorlat** feliratú dobozok jelzik, melyik feladat kapcsolódik az adott részhez.

Jelölések és szóhasználat

A könyv a magyar szakszavakat használja, és első előfordulásukkor megadja az angol megfelelőjüket. A teljes magyar–angol szószedet a C. függelékben található. A kiemelődobozok jelentése:

- **Tanulási eredmények:** mit tudsz a fejezet végére.
- **Fogalom:** pontos definíció, lehetőleg szabványos megfogalmazásban.
- **Futó eset (A, B, C):** a három esettanulmány aktuális lépése.
- **Kidolgozott példa:** lépésenként végigszámolt feladat.
- **Gyakori buktató:** tipikus tévedés, amelyet a gyakorlatban sokszor látni.
- **Esettanulmány egy balesetből:** valós baleset tanulsága a vizsgálati jelentések alapján.
- **Kitekintés:** haladó vagy szomszédos téma.
- **Gyakorlat:** a kapcsolódó feladat a gyakorlati útmutatóban.
- **Önellenőrzés:** kérdések lenyitható megoldással.

A valószínűségeket és gyakoriságokat tudományos jelöléssel írjuk (10^{-4}), a meghibásodási rátákat óránkénti értékben (h^{-1}) vagy FIT-ben (1 FIT $= 10^{-9}\ \mathrm{h}^{-1}$) adjuk meg.

Előzmények és köszönetnyilvánítás

A könyv a Pannon Egyetem mechatronikai mesterképzéséhez 2014-ben készült *Biztonságkritikus rendszerek* című egyetemi jegyzet (Abonyi és Fülep 2014) teljes átdolgozásával jött létre. A jegyzet kockázatelemzési és megbízhatósági fejezeteiből a módszertani mag megmaradt, de minden fejezet újraíródott. A szabványi háttér a mai kiadásokat követi, és a hibás vagy elavult definíciók javítva lettek. Új fejezetek születtek a balesetmodellekről, a HAZOP-ról, az STPA-ról, a rétegzett védelemről, a Markov-modellekről, a biztonsági funkciók igazolásáról, az emberi megbízhatóságról, a biztonsági érvelésről, valamint a SOTIF, a mesterséges intelligencia és a kiberbiztonság témaköréről. Minden ábra új, saját vektorgrafika.

Köszönettel tartozom a jegyzet egykori társszerzőjének, Fülep Tímeának, akinek a járműipari biztonsági stratégiákról és a fékrendszer-architektúrákról írt fejezetei a 14. fejezet egyes példáinak kiindulópontjai voltak. Köszönöm a Pannon Egyetem hallgatóinak a jegyzet évekig tartó használata során adott visszajelzéseiket.

Javítások és visszajelzés

A könyv nyílt, folyamatosan javított kiadvány. Hibát, pontatlanságot vagy javaslatot a webes változat oldalain található „Hiba vagy javaslat bejelentése” hivatkozáson keresztül lehet jelezni. Minden kiadás saját DOI-azonosítót kap, így pontosan hivatkozható; a kiadások közös azonosítója 10.5281/zenodo.24413011, amely mindig a legfrissebb változatra mutat.

> **A szabványokról**
>
> A könyv sok szabványra hivatkozik, de egyiket sem helyettesíti. A szabványok pontos követelményeit mindig az érvényes kiadás eredeti szövegéből kell venni. A könyv a 2026 végén érvényes kiadásokat követi; ahol új kiadás készül (például az IEC 61508 és az ISO 26262 harmadik kiadása), ott ezt jelezzük.

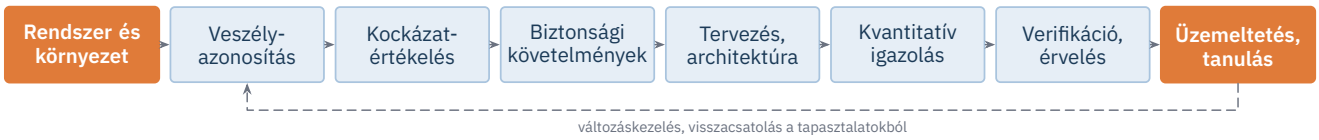
I. rész – Alapok



Pannon Egyetem
University of Pannonia

1.

Miért hibáznak a rendszerek? Balesetek és balesetmodellek



A biztonságtechnika a balesetekből tanult meg mindent, amit tud. A szabványok előírásai, a tervezési elvek és az elemzési módszerek mögött szinte mindig egy-egy konkrét katasztrófa áll, amelyet utólag sokan „elkerülhetőnek” neveztek. Ebben a fejezetben hat jól dokumentált balesetet vizsgálunk meg, és megkeressük, mi a közös bennük. Ezután áttekintjük, hogyan gondolkodott a mérnöki és a pszichológiai szakirodalom a balesetek keletkezéséről az elmúlt száz évben. A fejezet célja nem a félelemkeltés, hanem a szemléletformálás: a későbbi módszerek akkor használhatók jól, ha értjük, milyen jelenségek ellen dolgoznak.

Tanulási eredmények

A fejezet végére képes leszel

- hat jelentős baleset (Therac-25, Ariane 5, Texas City, Buncefield, Boeing 737 MAX, Uber ATG) okait és tanulságait a vizsgálati jelentések alapján összefoglalni;
- megkülönböztetni a lineáris oklánccmodelleket, az epidemiológiai modelleket és a rendszerszintű balesetmodelleket;
- a sajtszelet-modell, a Rasmussen-féle sodródás és a STAMP fogalmaival elemezni egy balesetet;
- megindokolni, miért nem elég a biztonsághoz az egyes alkatrészek megbízhatóságát növelni.

1.1. Mit tanulhatunk a balesetekből?

Edward Murphy amerikai mérnök nevéhez fűződik a mondás, hogy „ami elromolhat, az el is romlik”. A történet szerint 1949-ben egy gyorsulásmérési kísérletsorozatot kellett megismételni, mert a mérőátalakítókat fordítva szerelték be. A mondás szellemes, de a mérnök számára veszélyes is, mert azt sugallja, hogy a hiba sorsszerű. A balesetvizsgálatok tapasztalata ennél árnyaltabb: a súlyos balesetek ritkán egyetlen hibából fakadnak, és szinte mindig megelőzik őket figyelmeztető jelek, elmaradt intézkedések, rosszul értelmezett kockázatok.

A balesetvizsgálat célja ma már nem a bűnös megtalálása, hanem annak megértése, hogyan válhatott lehetővé a baleset, és hogyan lehet megakadályozni az ismétlődését. Ezt a szemléletet követik a független vizsgálószervek, például az amerikai Közlekedésbiztonsági Testület (NTSB), a Vegyipari

Biztonsági Testület (CSB) vagy a francia polgári repülési balesetvizsgáló hatóság (BEA). Jelentéseik nyilvánosak, és a biztonságtechnika legfontosabb tananyagai közé tartoznak.

1.2. Hat baleset, hat tanulság

Az alábbi esetek különböző iparágakból származnak, és mindegyikük más okból vált a szakma közös emlékezetének részévé. A leírások a hivatalos vizsgálati jelentéseken és az azokat feldolgozó szakirodalmon alapulnak.

Therac-25: amikor a szoftver átveszi a hardveres védelem szerepét (1985–1987)

A Therac-25 sugárterápiás gyorsító két üzemmódban működött: kis intenzitású elektronnyalábbal és nagy intenzitású, céltárgyon átvezetett röntgensugárzással. Az elődmodellekben hardveres reteszelés akadályozta meg, hogy a nagy intenzitású nyaláb céltárgy nélkül érje a beteget. A Therac-25-ben ezt a reteszelést szoftveres ellenőrzés váltotta fel. Hat ismert balesetben a betegek a tervezett dózis sokszorosát kapták, több eset halálos kimenetelű volt (Leveson és Turner 1993).
A vizsgálat több egymást erősítő okot tárt fel. A szoftverben versenyhelyzet (*race condition*) volt: ha a kezelő gyorsan módosította a bevitt adatokat, a gép beállítása és a képernyőn látható érték eltérhetett egymástól. A hibaüzenetek rejtélyesek voltak („MALFUNCTION 54”), a kezelők megszokták, hogy egyszerűen továbblépnek. A gyártó az első bejelentések után nem tudta reprodukálni a hibát, és hosszú ideig kizártnak tartotta, hogy a szoftver okozza. A szoftvert nem vetették alá független felülvizsgálatnak, a régebbi modellekből átvett kódrészleteket újrafelhasználták, pedig azok korábban a hardveres reteszelés „árnyékában” működtek.
Tanulság: a szoftver nem hibázik véletlenszerűen, hanem szisztematikusan; a hardveres védelem kiváltása szoftverrel új veszélyeket teremt; a kezelőfelület és a hibajelzések tervezése biztonsági kérdés.

Ariane 5, 501-es repülés: hibátlan kód rossz környezetben (1996)

1996. június 4-én az Európai Űrügynökség új hordozórakétája az első repülésén mintegy negyven másodperccel a start után letért a pályájáról, és megsemmisült. A vizsgálóbizottság szerint az inerciális referenciarendszer (SRI) szoftvere egy 64 bites lebegőpontos értéket (a vízszintes sebességhez kapcsolódó változót) 16 bites előjeles egész számmá alakított, és a túlcsordulás kezeletlen kivételt okozott (Lions 1996).
A kódot az Ariane 4-ből vették át, ahol a rakéta pályája miatt ez az érték soha nem nőhetett ekkorára. Az Ariane 5 nagyobb vízszintes gyorsulása mellett viszont igen. Ráadásul az érintett függvénynek a start után már nem volt szerepe.
A redundancia nem segített: a két SRI-egység ugyanazt a szoftvert futtatta, így ugyanabban a pillanatban, ugyanazért hibásodott meg. A fedélzeti számítógép a hibás egység diagnosztikai adatait repülési adatként értelmezte, és teljes kitérítést adott a fúvókáknak.
Tanulság: az újrafelhasznált komponens csak az eredeti feltételei között bizonyított; az azonos redundáns egységek közös okú hibája ellen a megtöbbszörözés nem véd; a követelmények és feltevések nyomon követése (mit tételezett fel a kód írója?) biztonsági kérdés.

BP Texas City: a figyelmeztető jelek erodálódása (2005)

2005. március 23-án a texasi finomító izomerizációs üzemének indításakor egy desztillálókolonnát túltöltöttek. A túlnyomás miatt nyíló biztonsági szelepeken keresztül éghető folyadék került egy légköri lefúvató tartályba és kéménybe, onnan a szabadba. A kialakuló gőzfelhő berobbant. Tizenöt ember meghalt, több mint százhetvenen megsérültek; a halálos áldozatok a közelben elhelyezett ideiglenes konténerirodákban tartózkodtak (U.S. Chemical Safety and Hazard Investigation Board 2007).
A CSB jelentése a technikai okok (hibás szintmérés, a magasszint-riasztás elmaradása, elavult lefúvató rendszer) mellett hangsúlyosan szervezeti okokat nevezett meg: költségcsökkentést a karbantartásban, a kezelők fáradtságát, a korábbi hasonló események figyelmen kívül hagyását, és azt, hogy a vállalat a munkabiztonsági mutatók (például a munkaidő-kieséses balesetek száma) javulását a folyamatbiztonság javulásának tekintette.

Tanulság: a személyi munkabiztonság és a folyamatbiztonság két különböző dolog, más mutatókkal mérhető; a kisebb események és a „majdnem balesetek” a nagy balesetek előjelei; a telepítési döntések (hová kerülnek az emberek) a kockázat részei.

Buncefield: a nem független védelmi réteg (2005)

2005. december 11-én hajnalban az angliai Buncefield üzemanyag-tárolójában egy tartályt túltöltöttek. A szintmérő megakadt, és állandó értéket mutatott, a független magasszint-kapcsoló pedig nem működött, mert egy szerelési és karbantartási hiba miatt hatástalan volt. A kiömlő benzin gőzfelhőt alkotott, amely berobbant és a környék több tartályára átterjedő tűzvészt okozott. Halálos áldozat nem volt, de negyvenhárom ember megsérült, és a kár Európa egyik legnagyobb ipari kára volt (Buncefield Major Incident Investigation Board 2008).
Tanulság: a papíron független védelmi rétegek a gyakorlatban közös gyengeséggel rendelkezhetnek (közös karbantartási gyakorlat, közös téves feltevés); a védelmi funkciót rendszeresen, valósághűen kell vizsgálni (*proof test*); a tervezett kockázatcsökkentés csak akkor valós, ha az üzemeltetés is fenntartja.

Boeing 737 MAX: egyetlen érzékelő és egy rejtett automatizmus (2018–2019)

A Lion Air 610-es (2018. október 29.) és az Ethiopian Airlines 302-es (2019. március 10.) járatának lezuhanásában összesen 346 ember halt meg. Mindkét esetben a repülésvezérlés egy új funkciója, az MCAS (*Maneuvering Characteristics Augmentation System*) hibás állásszögjel alapján ismételten lefelé trimmelte a gépet. Az MCAS egyetlen állásszög-érzékelő jelére támaszkodott, ismételten aktiválódhatott, és a hatásköre a fejlesztés során megnőtt. A pilóták képzési anyagai nem írták le részletesen (Joint Authorities Technical Review 2019).
A nemzetközi szakértői vizsgálat (JATR) többek között azt állapította meg, hogy a funkció hibás működésének veszélyességi besorolása és a tanúsítás során tett feltevések (például hogy a pilóták néhány másodpercen belül felismerik és kezelik a helyzetet) nem követték a funkció módosításait.
Tanulság: egy biztonságkritikus funkció nem támaszkodhat egyetlen érzékelőre; a tervezés során módosuló funkciók veszélyelemzését újra kell értékelni; az emberi beavatkozásra vonatkozó feltevéseket igazolni kell, nem csak feltételezni.

Uber ATG, Tempe: az automatizálás és a felügyelő ember (2018)

2018. március 18-án éjjel az arizonai Tempe városában egy fejlesztés alatt álló önvezető tesztjármű halálosan elütött egy kerékpárját toló gyalogost. Az NTSB vizsgálata szerint a rendszer körülbelül 5,6 másodperccel az ütközés előtt észlelte a gyalogost, de többször átsorolta (jármű, egyéb objektum, kerékpár), és minden átsorolásnál elvetette a mozgáspályájának előzményeit. A jármű saját vészfékező funkcióját a tesztek idejére kikapcsolták, és a rendszer a beavatkozást a biztonsági vezetőre bízta, aki a vizsgálat szerint a baleset előtt a telefonján nézett videót (National Transportation Safety Board 2019).
Az NTSB a valószínű okok között a biztonsági vezető figyelmetlenségét, de a hozzájáruló tényezők között a vállalat elégtelen biztonsági kockázatértékelését, a kezelők figyelmének hiányzó felügyeletét és a „automatizálási elkényelmesedés” (*automation complacency*) kezelésének hiányát is megnevezte.
Tanulság: a passzív felügyeletre szoruló ember megbízhatatlan biztonsági tartalék; az érzékelés és osztályozás bizonytalanságát a rendszer architektúrájának kell kezelnie; a fejlesztési és tesztelési tevékenység maga is biztonsági kockázatot hordoz.

1.3. Közös mintázatok

A hat eset között évtizedek és iparágak húzódnak, mégis feltűnően hasonlóak. A 1.1. táblázat összefoglalja azokat a mintázatokat, amelyekre a könyv további fejezetei módszereket kínálnak.

1.1. táblázat. A balesetek közös mintázatai és a kapcsolódó fejezetek

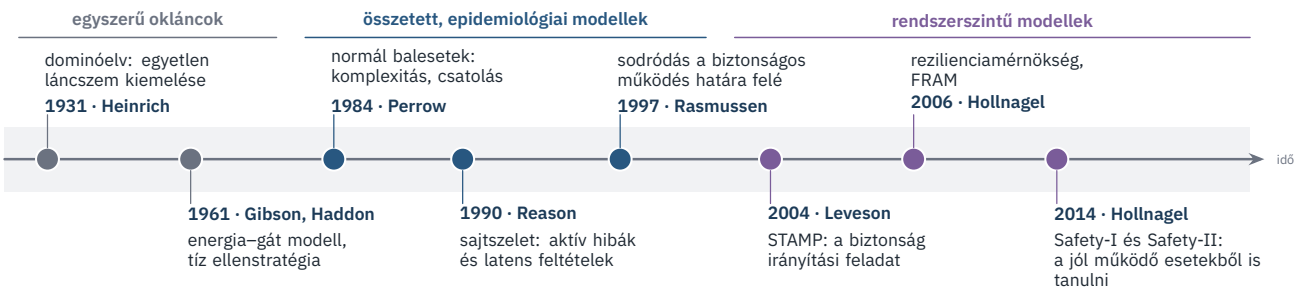
Mintázat	Példa	A könyv válasza
A hiba a specifikációban vagy a feltevésekben van, nem az alkatrészben	Ariane 5, Therac-25, 737 MAX	veszélyazonosítás (5–7. fejezet), követelmények nyomon követése (15. fejezet)
A redundáns egységek közös okból hibásodnak meg	Ariane 5, Buncefield	közös okú hibák modellezése (10. és 13. fejezet)
A védelmi rétegek idővel erodálódnak	Texas City, Buncefield	proof teszt, üzemeltetési fázis (13. fejezet)
A szoftver új, szisztematikus hibamódokat hoz	Therac-25, Ariane 5	szoftverbiztonság (15. fejezet)
Az ember felügyelőként és tartalékként nem megbízható	Uber, 737 MAX	ember a rendszerben (16. fejezet)
Szervezeti nyomás és gyenge biztonsági kultúra	Texas City, Uber	biztonsági érvelés és menedzsment (17. fejezet)
Az új technológia a meglévő módszerek határain túl működik	Uber	SOTIF, gépi tanulás (18. fejezet)

Két általános következtetés adódik. Az első, hogy **a megbízhatóság és a biztonság nem ugyanaz.** Az Ariane 5 szoftvere azt tette, amit specifikáltak; a Therac-25 szoftvere a specifikációja szerinti esetek túlnyomó többségében helyesen működött. Egy rendszer lehet megbízható és mégis veszélyes, és lehet kevéssé megbízható, mégis biztonságos (például ha minden hibájakor biztonságos állapotba áll). A 2. fejezet ezt a különbséget pontosan definiálja.

A második, hogy **a balesetek mögött rendszerszintű okok állnak.** Az alkatrészhiba, az emberi tévedés vagy a szoftverhiba közvetlen ok, de az, hogy ebből baleset lett, a rendszer felépítésén, a szervezet működésén és a szabályozási környezeten múlt. Ezt a felismerést követték a balesetmodellek.

1.4. A balesetmodellek fejlődése

A balesetmodell az a gondolati keret, amellyel a balesetek keletkezését magyarázzuk. A választott modell meghatározza, mit keresünk a vizsgálat során, és milyen intézkedéseket tartunk hatékonynak. A 1.1. ábra a legfontosabb modelleket mutatja időrendben.



1.1. ábra. A balesetmodellek fejlődése: az egyszerű okláncoktól a rendszerszintű modellekig.

1.4.1. Lineáris oklánccmodellek

Herbert W. Heinrich biztosítási szakember 1931-ben megjelent könyvében a balesetet egymás után dőlő dominók sorozataként írta le: a társadalmi környezet és a személyes hiányosságok vezetnek a nem biztonságos cselekedethez vagy állapothoz, ez a balesethez, az pedig a sérüléshez (Heinrich 1931). Ha egy dominót kiveszünk, a lánc megszakad. Heinrich nevéhez fűződik az a sokat idézett, de módszertanilag gyenge állítás is, hogy a balesetek túlnyomó többségét emberi „nem biztonságos cselekedet" okozza.

A dominóelv egyszerű és jól kommunikálható, ezért ma is él például a gyökérok-elemzés (*root cause analysis*) naiv változataiban. Két korlátja van: a balesetet egyetlen láncra szűkíti, holott a valóságban több, egymással kölcsönható ok vezet hozzá, és könnyen az utolsó láncszemnél, a kezelő hibájánál áll meg.

1.4.2. Energia–gát modell

James Gibson és William Haddon az 1960-as és 1970-es években a balesetet a nem kívánt **energiaátadás** jelenségeként írta le (Haddon 1973). A kár akkor keletkezik, ha mechanikai, hő-, vegyi, villamos vagy sugárzási energia olyan mértékben jut el az emberhez vagy a védendő értékhez, amely meghaladja annak tűrőképességét. A megelőzés eszközei a **gátak** (*barriers*): az energia mennyiségének csökkentése, a felhalmozódás megakadályozása, a felszabadulás lassítása, a térbeli vagy időbeli elválasztás, a védendő érték ellenállóbbá tétele. Haddon tíz ilyen ellenstratégiát sorolt fel.

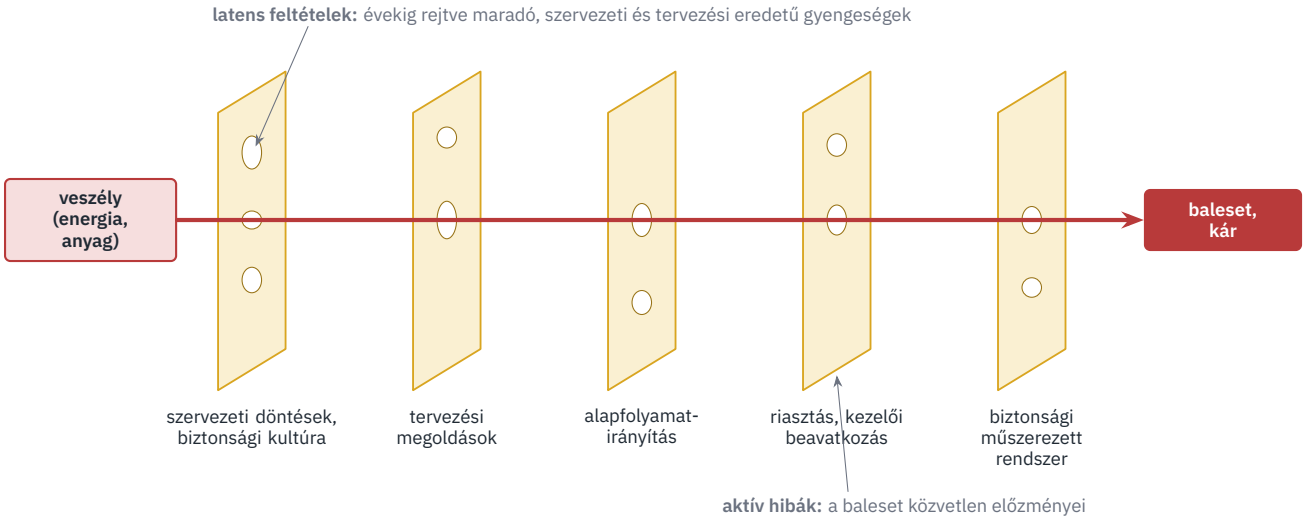
Az energia–gát gondolat a mai folyamatbiztonság alapja. A 11. fejezetben tárgyalt rétegzett védelem (LOPA) és a bow-tie diagram közvetlenül erre épül, és a 14. fejezetben tárgyalt inherens biztonság elve (kevesebb veszélyes anyag, alacsonyabb nyomás) is az energiaforrás csökkentéséről szól.

1.4.3. A sajtszelet-modell: aktív hibák és latens feltételek

James Reason pszichológus a *Human Error* (1990) és a *Managing the Risks of Organizational Accidents* (1997) című könyveiben a szervezeti balesetek magyarázatára vezette be a sokszor „svájci sajt modellnek" nevezett ábrázolást (Reason 1990, 1997). A rendszer védelmi rétegeit (szervezeti döntések, tervezés, irányítás, riasztás, védelmi rendszer) szeletekként képzeljük el, amelyeken lyukak vannak (1.2. ábra). Baleset akkor következik be, ha egy pillanatra minden szelet lyukai egy vonalba kerülnek, és a veszély átjut rajtuk.

Reason két hibatípust különböztet meg. Az **aktív hibák** (*active failures*) a frontvonalban dolgozók (kezelők, pilóták, karbantartók) tévedései és szabályszegései, amelyeknek azonnali hatásuk van. A **latens feltételek** (*latent conditions*) a tervezők, a vezetők és a szabályozók döntéseiből fakadnak: elégtelen képzés, félrevezető kezelőfelület, karbantartási hiányosság, időnyomás. Ezek évekig rejtve maradhatnak, amíg egy aktív hibával találkozva balesethez nem vezetnek. A modell legfontosabb üzenete, hogy a kezelő hibája gyakran csak a lánc utolsó, látható eleme; a tartós javulást a latens feltételek felszámolása hozza.

A sajtszelet-modell a gyakorlatban rendkívül népszerű, és a 2014-es jegyzet is ezzel szemléltette a független védelmi rétegek jelentőségét. Fontos azonban látni a korlátait: a lyukak „egy vonalba kerülését" nem magyarázza meg, a rétegeket egymástól függetlennek ábrázolja, és könnyen arra csábít, hogy egyszerűen több szeletet tegyünk a rendszerbe.



1.2. ábra. A sajtszelet-modell. A védelmi rétegek gyengeségei (lyukai) folyamatosan változnak; a baleset akkor következik be, amikor egy pillanatra mind egy vonalba kerülnek.

> **„Még egy védelmi réteg" mindig segít?**
>
> A sajtszelet-modell félreérthető üzenete, hogy minél több a szelet, annál biztonságosabb a rendszer. A valóságban az új réteg új hibamódokat, új bonyolultságot és hamis biztonságérzetet is hozhat. A kezelők például jobban támaszkodhatnak az automatikus védelemre, és ritkábban figyelnek a folyamatra. A rétegek csak akkor szorozzák a kockázatcsökkentést, ha valóban függetlenek egymástól. Ezt a 11. fejezetben a független védelmi réteg (IPL) szigorú kritériumaival fogjuk megfogalmazni.

1.4.4. Normál balesetek: komplexitás és csatolás

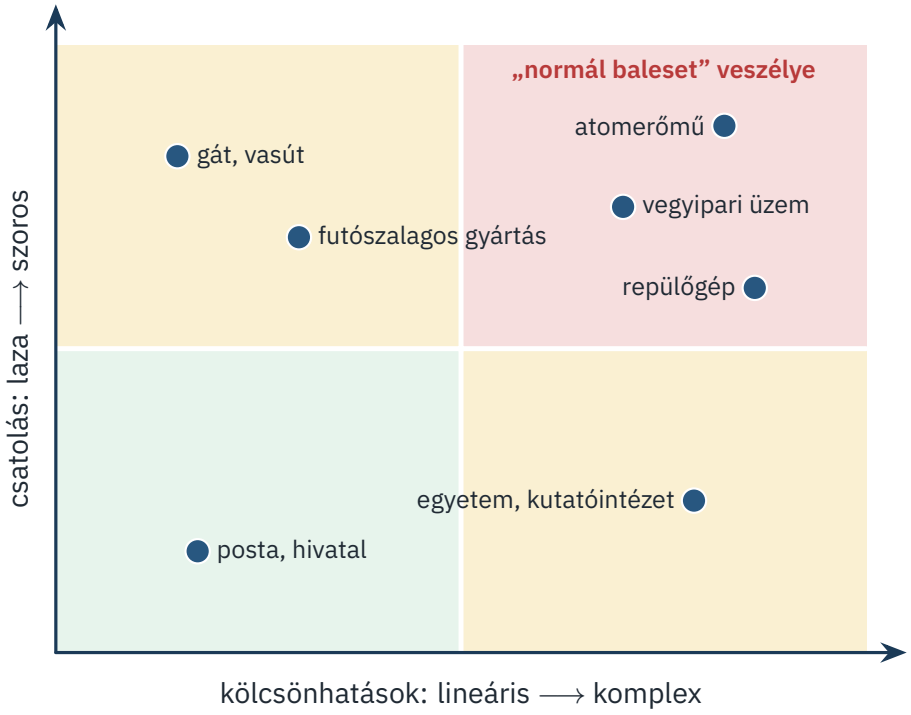
Charles Perrow szociológus az 1979-es Three Mile Island-i atomerőmű-baleset elemzése nyomán jutott arra a következtetésre, hogy bizonyos rendszerekben a balesetek elkerülhetetlenek, „normálisak" (Perrow 1984). Két jellemzőt emelt ki (1.3. ábra):

- **interaktív komplexitás:** a rendszer elemei váratlan, nem tervezett és nehezen átlátható módon hathatnak egymásra;
- **szoros csatolás:** a folyamatok gyorsan, kötött sorrendben és kevés tartalékkal futnak, így nincs idő beavatkozni, ha valami eltér a tervezettől.

Perrow szerint a jobb felső negyedbe eső rendszerekben a több független hiba váratlan kölcsönhatása előbb-utóbb balesethez vezet, és a redundancia tovább növelheti a komplexitást. Az elmélet vitatott, mert a magas megbízhatóságú szervezetek (például a repülőgép-hordozók vagy a légiforgalmi irányítás) kutatói kimutatták, hogy megfelelő szervezeti kultúrával ezek a rendszerek is tartósan biztonságosan működtethetők. A mérnök számára mégis fontos üzenetet hordoz: a komplexitás és a csatolás tervezési paraméter, és csökkentésük maga is biztonsági intézkedés.

1.4.5. Sodródás a határ felé

Jens Rasmussen dán mérnök és pszichológus 1997-es tanulmányában a biztonságot dinamikus egyensúlyként írta le (Rasmussen 1997). A rendszer működése egy olyan térben mozog, amelyet



1.3. ábra. Perrow kölcsönhatás–csatolás tere. A jobb felső negyedbe eső rendszerekben a „normál baleset” kockázata a legnagyobb.

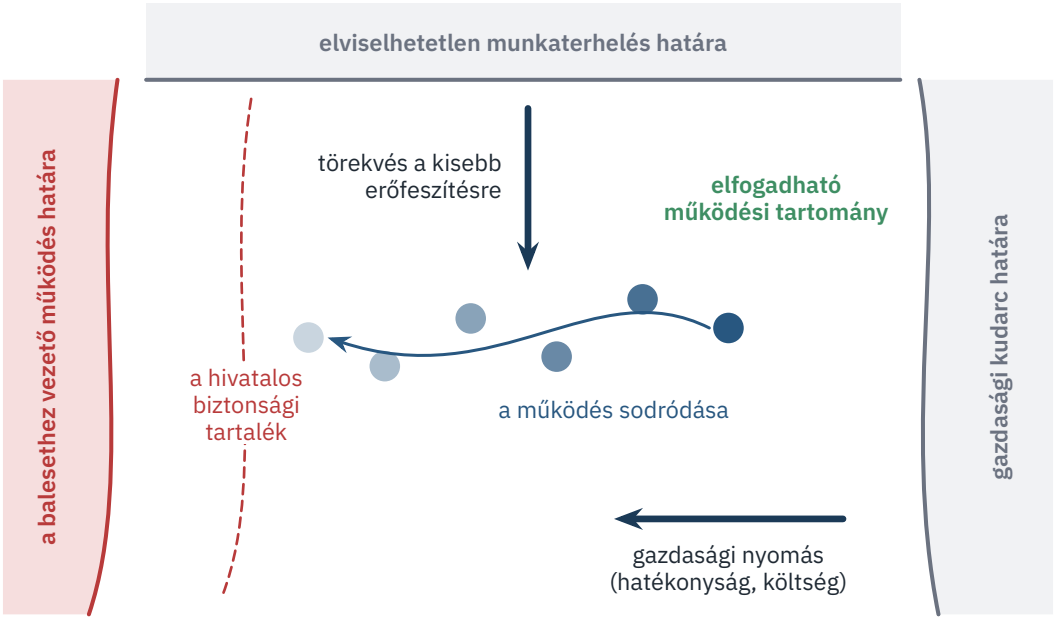
három határ fog közre (1.4. ábra): a gazdasági kudarc határa, az elviselhetetlen munkaterhelés határa és a balesethez vezető működés határa. A vezetés a hatékonyság növelésére, a dolgozók az erőfeszítés csökkentésére törekednek. Ez a két nyomás a működést lassan, észrevétlenül a biztonsági határ felé sodorja. Minden egyes lépés ésszerűnek tűnik, és mivel a baleset elmarad, a kisebb tartalék megszokottá válik.

A Texas City-i baleset jó példája ennek a sodródásnak: az elhalasztott karbantartások, a mellőzött eljárások és az elfogadott rendellenességek külön-külön nem okoztak balesetet, együtt viszont eltüntették a biztonsági tartalékot. A Rasmussen-modell üzenete, hogy a biztonságot nem lehet egyszer „beépíteni” a rendszerbe, hanem folyamatosan figyelni kell, és a határokat láthatóvá kell tenni a döntéshozók számára.

1.4.6. STAMP: a biztonság irányítási feladat

Nancy Leveson az MIT professzoraként a 2000-es évek elején fogalmazta meg a STAMP-modellt (*Systems-Theoretic Accident Model and Processes*), amelyet *Engineering a Safer World* című könyvében foglalt össze (Leveson 2011). A STAMP a balesetet nem hibák láncolataként, hanem **irányítási problémaként** értelmezi: a biztonság a rendszer emergens tulajdonsága, amely akkor áll fenn, ha a rendszer viselkedésére vonatkozó biztonsági korlátokat (*safety constraints*) a hierarchikus irányítási struktúra minden szintje érvényesíti. A baleset oka az, hogy egy korlát nem érvényesült: az irányító hibás vagy hiányos folyamatmodell alapján döntött, a visszacsatolás hiányzott vagy késett, vagy az irányító beavatkozása nem hatott.

Ebben a szemléletben az alkatrészhiba csak egy a lehetséges okok közül. A baleset akkor is bekövetkezhet, ha egyetlen alkatrész sem hibásodik meg, de az alkatrészek kölcsönhatása nem



1.4. ábra. Rasmussen modellje: a gazdasági nyomás és a kisebb erőfeszítésre való törekvés a működést a biztonsági határ felé sodorja.

biztonságos. Erre a Mars Polar Lander 1999-es elvesztése a klasszikus példa: a leszállóláb kinyitásakor keletkező rezgést a szoftver a talajérés jelének értelmezte, és idő előtt leállította a fékezőhajtóművet. Az érzékelő és a szoftver is a specifikációja szerint működött.

A STAMP-ra épülő veszélyelemzési módszert, az STPA-t a 7. fejezet részletesen tárgyalja.

1.4.7. Safety-I és Safety-II

Erik Hollnagel és a rezilienciamérnökség (*resilience engineering*) képviselői arra hívták fel a figyelmet, hogy a hagyományos biztonságtechnika (Safety-I) a ritka hibákból és balesetekből tanul, és a biztonságot a nem kívánt események hiányaként határozza meg (Hollnagel 2014). A Safety-II szemlélet szerint a biztonság az a képesség, hogy a rendszer változó körülmények között is sikeresen működjön. Ezt pedig leginkább a jól működő esetekből, az emberek folyamatos alkalmazkodásából lehet megérteni. A két szemlélet nem zárja ki, hanem kiegészíti egymást: a tervezésnél a Safety-I módszerei (veszélyazonosítás, kockázatcsökkentés), az üzemeltetésnél a Safety-II szemlélete (a mindennapi alkalmazkodás megértése és támogatása) egyaránt szükséges.

1.5. Mit jelent mindez a mérnök számára?

A balesetmodellek nem egymást felváltó divatok, hanem egymásra épülő rétegek. A könyv módszertana mindegyikből merít:

- a **redukcionista módszerek** (FMEA, hibafa, megbízhatósági blokkdiagram, Markov-modell) az alkatrészhibák és kombinációik hatását számszerűsítik, és nélkülözhetetlenek a biztonsági funkciók integritásának igazolásához (III. rész);
- a **gát- és rétegszemlélet** (LOPA, bow-tie) a kockázatcsökkentés felépítését teszi átláthatóvá (11. fejezet);

- a **rendszerelméleti szemlélet** (STAMP, STPA) a kölcsönhatásokból, a specifikációs hibákból és az ember–automatizálás együttműködéséből fakadó veszélyeket tárja fel (7. fejezet);
- a **szervezeti és emberi szemlélet** (Reason, Rasmussen, Hollnagel) az üzemeltetés és a biztonsági kultúra kérdéseire figyelmeztet (16. és 17. fejezet).

ReaktorVéd: melyik baleset tanulsága érinti a reaktort?

A ReaktorVéd reaktorát exoterm reakció hőfejlődése teszi veszélyessé. A Texas City- és a Buncefield-baleset közvetlenül releváns: mindkettőben túltöltés és a védelmi rétegek eróziója vezetett a katasztrófához. A tervezőcsapat ezért már a koncepciós fázisban három kérdést tesz fel: Mennyi energiát tárolunk, és csökkenthető-e (kisebb adag, lassabb adagolás)? Mely védelmi rétegek valóban függetlenek egymástól? Hogyan biztosítjuk, hogy a védelmi funkciók az üzemeltetés évei alatt is működőképesek maradjanak?

VészFék: az Uber-baleset tanulsága

A VészFék fejlesztőcsapata az Uber-baleset jelentéséből két tervezési elvet szűr le. Az érzékelés bizonytalanságát (az objektum osztályozásának ingadozását) a funkció logikájának kell kezelnie, nem a vezetőnek. Továbbá a rendszer nem építhet olyan emberi beavatkozásra, amelyre a vezetőnek nincs ideje vagy figyelme. Ezek a szempontok a 18. fejezetben, a rendeltetésszerű funkció biztonságánál (SOTIF) térnek vissza.

Gyakorlat

A gyakorlati útmutató első feladata egy választott, nyilvános balesetvizsgálati jelentés elemzése a sajtszelet-modell és a STAMP szemléletével. A feladatlap mintapromptokat is ad ahhoz, hogyan lehet egy nagy nyelvi modellt kritikus olvasótársként használni a jelentés feldolgozásához (lásd a gyakorlati útmutató 1. laborját).

Összefoglalás

- A súlyos balesetek ritkán egyetlen hibából fakadnak; mögöttük specifikációs hibák, közös okú hibák, erodálódó védelmi rétegek, emberi tényezők és szervezeti nyomás áll.
- A megbízhatóság (a specifikáció szerinti működés) és a biztonság (a nem elfogadható kár elkerülése) két különböző tulajdonság.
- A balesetmodellek a lineáris okláncoktól (Heinrich) az energia–gát modellen és az epidemiológiai modelleken (Reason) át a rendszerszintű modellekig (Rasmussen, STAMP, Safety-II) fejlődtek.
- A modern biztonságtechnika a redukcionista, a gát-, a rendszerelméleti és a szervezeti szemléletet együtt alkalmazza.

Önellenőrzés

1. Miért nem védett a két inerciális referenciaegység az Ariane 5 balesete ellen?
2. Mi a különbség Reason fogalmai szerint az aktív hiba és a latens feltétel között? Adj egy-egy példát a Texas City-i balesetből!
3. Hogyan magyarázza a Rasmussen-modell, hogy egy üzem évekig baleset nélkül működik, majd egyszer csak súlyos baleset történik?
4. Lehet-e baleset akkor, ha a rendszer minden eleme a specifikációja szerint működik? Indokold egy példával!

Megoldás.

1. Mert ugyanazt a szoftvert futtatták, így ugyanaz a bemenet ugyanabban a pillanatban mindkettőben ugyanazt a kezeletlen kivételt váltotta ki. Ez közös okú hiba; ellene az azonos egységek megtöbbszörözése nem véd, csak a diverzitás (eltérő kialakítás) vagy a hiba okának megszüntetése.
2. Az aktív hiba azonnali hatású, a frontvonalban elkövetett tévedés vagy szabályszegés (például a kolonna túltöltése az indításkor, a szintjelzés figyelmen kívül hagyása). A latens feltétel tervezési vagy szervezeti

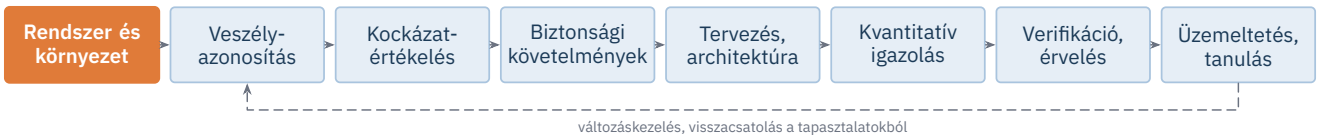
eredetű, rejtett gyengeség (például az elavult lefúvató rendszer, a kezelők túlterhelése, a karbantartási költségek csökkentése, az ideiglenes irodák rossz elhelyezése).
3. A gazdasági nyomás és a kisebb erőfeszítésre való törekvés lassan csökkenti a biztonsági tartalékot. Mivel a baleset sokáig elmarad, minden kisebb lépés elfogadottá válik, és a működés észrevétlenül a biztonsági határig sodródik. Ekkor már egy kis zavar is elegendő a balesethez.
4. Igen. A STAMP szerint a baleset az elemek nem biztonságos kölcsönhatásából is fakadhat. Példa a Mars Polar Lander: a lábak kinyitásakor keletkező jelet a szoftver a talajérés jelének vette, és leállította a hajtóművet, holott minden komponens a specifikációja szerint működött. A hiba a specifikációban (a kölcsönhatás figyelmen kívül hagyásában) volt.

Továbbolvasás

- Leveson, N. G. (2011): *Engineering a Safer World.* MIT Press – nyílt hozzáférésű; a 2. és 3. fejezet a balesetmodellek kritikai áttekintése (Leveson 2011).
- Reason, J. (1997): *Managing the Risks of Organizational Accidents.* Ashgate – a latens feltételek és a szervezeti balesetek klasszikus műve (Reason 1997).
- Rasmussen, J. (1997): Risk management in a dynamic society. *Safety Science* 27, 183–213 (Rasmussen 1997).
- A vizsgálati jelentések: Leveson és Turner (1993) a Therac-25-ről, Lions (1996) az Ariane 5-ről, a CSB (2007) a Texas City-i és az NTSB (2019) a tempei balesetről.

2.

Fogalmak: veszély, kockázat, biztonság, megbízhatóság



A biztonságtechnikában a pontos szóhasználat nem formalitás. Ha a tervező „veszélyről” beszél, a menedzser „kockázatról”, a szabvány pedig „veszélyes eseményről”, és mindhárman mást értenek rajta, akkor a kockázatértékelés eredménye értelmezhetetlen lesz. Ebben a fejezetben a könyv fogalmi alapjait rakjuk le a nemzetközi szabványok (ISO/IEC Guide 51, IEC 61508-4, IEC 60050-192) és az üzembiztonság-kutatás alapművének számító taxonómia (Avižienis és mtsai. 2004) szerint. Először a veszélytől a kárig vezető utat írjuk le, majd a kockázat és a biztonság fogalmát, végül a meghibásodások osztályozását, amelyre a III. rész kvantitatív módszerei épülnek.

Tanulási eredmények

A fejezet végére képes leszel

- pontosan definiálni és megkülönböztetni a veszély, a veszélyes helyzet, a veszélyes esemény, a kár és a kockázat fogalmát;
- elmagyarázni, mit jelent a biztonság, a funkcionális biztonság és a biztonsági funkció;
- leírni az üzembiztonság (dependability) tulajdonságait, fenyegetéseit és eszközeit;
- a hibaok–hiba–meghibásodás láncot több rendszerszinten alkalmazni;
- osztályozni a meghibásodásokat (véletlenszerű és szisztematikus, biztonságos és veszélyes, észlelt és rejtett, közös okú).

2.1. A veszélytől a kárig

A biztonsági szabványok fogalomrendszerének közös alapja az ISO/IEC Guide 51:2014 (ISO/IEC 2014), amely a szabványírók számára rögzíti a biztonsággal kapcsolatos alapfogalmakat. A definíciók láncot alkotnak (2.1. ábra).

Kár, veszély, veszélyes helyzet, veszélyes esemény

- **Kár** (*harm*): sérülés vagy az emberi egészség károsodása, illetve vagyontárgyak vagy a környezet károsodása.
- **Veszély** (*hazard*): a kár lehetséges forrása.

- **Veszélyes helyzet** (*hazardous situation*): olyan körülmény, amelyben ember, vagyontárgy vagy a környezet egy vagy több veszélynek ki van téve.
- **Veszélyes esemény** (*hazardous event*): olyan esemény, amely kárt okozhat.



2.1. ábra. A veszélytől a kárig vezető út és a három beavatkozási pont: a veszély elkerülése, a veszélyes esemény megelőzése és a következmények csökkentése.

A veszély a rendszer tulajdonsága: a reaktorban tárolt reakcióhő, a jármű mozgási energiája, a robotkar tömege és sebessége. A veszély önmagában nem okoz kárt; ehhez az kell, hogy valaki vagy valami ki legyen téve neki (veszélyes helyzet), és bekövetkezzen egy esemény, amely a veszélyt kárrá alakítja. A műszaki rendszerekben tipikus veszélyhelyzet az, amikor egy anyag vagy berendezés a környezeténél magasabb energiaszinten van, és a működés eltér a tervezett állapottól.

A lánc három pontján lehet beavatkozni. A leghatékonyabb a **veszély elkerülése vagy csökkentése**: kevesebb veszélyes anyag, kisebb nyomás, kisebb sebesség. Ha a veszély nem szüntethető meg, a **veszélyes esemény megelőzése** következik, például egy védelmi funkcióval, amely leállítja a folyamatot, mielőtt a veszély kárrá alakulna. Végül a **következmények csökkentése**: kármentő, védőburkolat, légzsák, vészhelyzeti terv. Ez a sorrend a kockázatcsökkentés hierarchiájának alapja, amelyre a 3. fejezetben visszatérünk.

A veszélyeket a hatásuk és eredetük szerint is csoportosíthatjuk (2.1. táblázat). A könyv elsősorban a műszaki rendszerek belső veszélyeivel foglalkozik, de a természeti és a szándékos (biztonsági, *security*) veszélyek a kockázatértékelésnél külső hatásként jelennek meg.

2.1. táblázat. A veszélyek csoportosítása eredetük szerint

Eredet	Példa veszélyre	Tipikus veszélyes esemény	Lehetséges kár
Mechanikai	mozgó géprész, nyomás alatti tartály	ütközés, becsípődés, tartályhasadás	sérülés, halál, anyagi kár
Termikus, vegyi	exoterm reakció, gyúlékony gőz	megfutó reakció, robbanás, kibocsátás	égés, mérgezés, környezetszennyezés
Villamos	feszültség alatt álló rész	áramütés, ív, tűz	sérülés, halál
Sugárzás	ionizáló sugárzás, lézer	túladagolás, besugárzás	egészségkárosodás
Természeti	árvíz, földrengés, villám	infrastruktúra-kiesés, tartályfelúszás	közvetett műszaki baleset

Eredet	Példa veszélyre	Tipikus veszélyes esemény	Lehetséges kár
Szándékos	kibertámadás, szabotázs	jogosulatlan beavatkozás a vezérlésbe	bármelyik fenti

2.2. Kockázat

Kockázat

Kockázat (*risk*): a kár bekövetkezése valószínűségének és a kár súlyosságának együttese (ISO/IEC 2014).

A definíció szándékosan nem mondja meg, hogyan kell a két összetevőt „összekombinálni”. A legegyszerűbb, a mérnöki gyakorlatban gyakran használt forma a szorzat, a várható kár:

$$R = F \cdot C, \tag{2.1}$$

ahol F a veszélyes esemény gyakorisága (például 1/év), C a következmény súlyossága (például halálesetek száma vagy pénzben kifejezett kár). Ha a rendszerben több, egymást kizáró veszélyes forgatókönyv lehetséges, a teljes kockázat ezek összege:

$$R = \sum_{i=1}^{n} F_i \, C_i. \tag{2.2}$$

Várható kár két forgatókönyvből

Egy tárolótelepen két veszélyes forgatókönyvet azonosítottak. A kisebb tömítetlenség becsült gyakorisága $F_1 = 0{,}1$/év, kára $C_1 = 2$ millió Ft; a tartály katasztrofális meghibásodásáé $F_2 = 10^{-4}$/év, kára $C_2 = 5$ milliárd Ft. A várható éves kár

$$R = 0{,}1 \cdot 2 \cdot 10^{6} + 10^{-4} \cdot 5 \cdot 10^{9} = 2 \cdot 10^{5} + 5 \cdot 10^{5} = 7 \cdot 10^{5} \text{ Ft/év}.$$

A két forgatókönyv várható kára azonos nagyságrendű, mégsem egyenértékűek: a második ritka, de katasztrofális esemény, amely a vállalat létét is veszélyeztetheti, és emberéleteket követelhet. Az ilyen esetekben a várható érték önmagában nem elég a döntéshez. A 3. fejezet ezért a kockázat elfogadhatóságát a gyakoriság és a súlyosság kétdimenziós terében vizsgálja.

A kockázat fogalmát árnyalja, hogy kit érint. Az **egyéni kockázat** (*individual risk*) annak a valószínűsége, hogy egy meghatározott személy egy év alatt meghal (vagy megsérül) a vizsgált tevékenység miatt. A **társadalmi kockázat** (*societal risk*) azt fejezi ki, milyen gyakorisággal következnek be adott számú áldozatot követelő balesetek. A kettő különbözik: egy ritkán lakott környezetben álló üzem egyéni kockázata nagy lehet a kerítés mellett, a társadalmi kockázata mégis kicsi, és fordítva. Mindkettő mérőszámaival a 3. fejezet foglalkozik.

Veszély vagy kockázat?

A két szó a köznyelvben összemosódik, de a biztonságtechnikában nem cserélhető fel. A **veszély** a kár forrása, a **kockázat** a kár valószínűségének és súlyosságának együttese. „Ennek a gépnek nagy a veszélye” helyett pontosabb így: „a gép mozgó részei veszélyt jelentenek; a kezelő becsípődésének kockázata nem elfogadható”. Egy nagy veszély (például egy atomreaktor maghője) kis kockázattal is társulhat, ha a védelmi rendszerek miatt a kár bekövetkezése rendkívül valószínűtlen.

2.3. Biztonság és funkcionális biztonság

Biztonság, tolerálható kockázat

- **Biztonság** (*safety*): nem tolerálható kockázattól való mentesség (ISO/IEC 2014).
- **Tolerálható kockázat** (*tolerable risk*): az a kockázatszint, amelyet adott körülmények között, a társadalom aktuális értékrendje alapján elfogadnak.

A definíció két fontos dolgot mond ki. Először, a biztonság nem a kockázat teljes hiánya: kockázatmentes rendszer nem létezik, ezért a biztonság mindig egy elfogadott maradó kockázathoz viszonyítva értelmezhető. Másodszor, a „tolerálható" szint nem a mérnök egyéni döntése, hanem társadalmi, jogi és szervezeti megállapodás eredménye. Az, hogy mi számít elfogadhatónak, iparáganként, országonként és koronként változik.

A tervezés során a kockázat három szintjét érdemes megkülönböztetni:

- az **eredeti kockázat** az, amelyet a rendszer védelmi intézkedések nélkül jelentene;
- a **tolerálható kockázat** a cél;
- a **maradó kockázat** (*residual risk*) az, amely a kockázatcsökkentő intézkedések után megmarad; sikeres kockázatkezelés esetén ez nem nagyobb a tolerálható kockázatnál.

A nem azonosított kockázat külön figyelmet érdemel: az a kockázat, amelyet egyik elemzés sem tárt fel. Ezt nem lehet közvetlenül csökkenteni, csak a veszélyazonosítás alaposságával és a módszerek sokféleségével szűkíteni.

2.3.1. Funkcionális biztonság

A modern műszaki rendszerekben a kockázatcsökkentés jelentős részét villamos, elektronikus és programozható elektronikus (E/E/PE) rendszerek végzik: a reaktor hőmérséklet-védelme, a jármű blokkolásgátlója, a robot sebességfelügyelete. Az IEC 61508 (IEC 2010a) ezért vezette be a funkcionális biztonság fogalmát.

Funkcionális biztonság, biztonsági funkció, biztonsági rendszer

- **Funkcionális biztonság** (*functional safety*): a teljes biztonság azon része, amely az irányított berendezéssel és annak irányítórendszerével kapcsolatos, és amely az E/E/PE biztonsági rendszerek és más kockázatcsökkentő intézkedések helyes működésétől függ.
- **Irányított berendezés** (*equipment under control, EUC*): az a gép, berendezés vagy üzem, amelynek veszélyeit a biztonsági funkciók kezelik.
- **Biztonsági funkció** (*safety function*): az a funkció, amelyet egy biztonsági rendszer valósít meg, hogy egy meghatározott veszélyes eseményre vonatkozóan az irányított berendezés biztonságos állapotát elérje vagy fenntartsa.
- **Biztonsági rendszer** (*safety-related system*): az a rendszer, amely a szükséges biztonsági funkciókat megvalósítja, és ehhez elegendő biztonsági integritással rendelkezik.

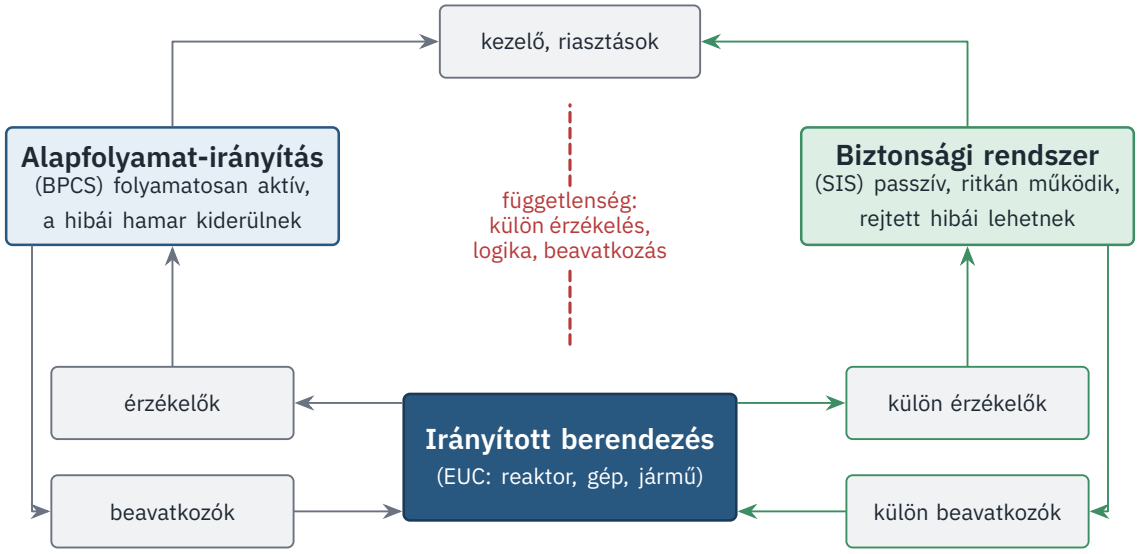
A funkcionális biztonság tehát nem a teljes biztonságot jelenti. A tartály falvastagsága, a robbanásbiztos kialakítás vagy a védőkorlát a biztonság része, de nem a funkcionális biztonságé, mert nem egy irányítórendszer helyes működésén múlik. Ugyanakkor a funkcionális biztonság szemlélete (meghatározott funkció, meghatározott integritás, igazolt teljesítés) a mechanikai és az emberi védelmi rétegekre is jól alkalmazható.

A **biztonsági integritás** (*safety integrity*) annak valószínűsége, hogy a biztonsági rendszer a megkövetelt biztonsági funkciókat meghatározott körülmények között, meghatározott időtartam alatt kielégítően

végrehajtja. Ennek diszkrét lépcsői a biztonsági integritási szintek (SIL), amelyekkel a 4. és a 13. fejezet foglalkozik.

2.3.2. Az alapfolyamat-irányítás és a védelmi rendszer

A folyamatiparban az irányított berendezést két, egymástól független rendszer felügyeli (2.2. ábra). Az **alapfolyamat-irányító rendszer** (*basic process control system*, BPCS) a normál üzemet valósítja meg: szabályozza a hőmérsékletet, a nyomást és a szintet. A **biztonsági műszerezett rendszer** (*safety instrumented system*, SIS) csak akkor lép működésbe, ha a folyamat a biztonságos tartományból kilép, és ilyenkor biztonságos állapotba viszi azt (például leállítja az adagolást).



2.2. ábra. Az alapfolyamat-irányítás és a független biztonsági rendszer. A két rendszer viselkedése és jellemző hibái alapvetően különböznek.

A két rendszer hibáinak jellege eltérő (2.2. táblázat). Az alapirányítás folyamatosan aktív, ezért hibái hamar kiderülnek: a kezelő látja, ha a szabályozott jellemző eltér az alapjeltől, és beavatkozik. A védelmi rendszer passzív: hónapokig vagy évekig nem kap igényt, és ha közben egy eleme meghibásodik, az rejtve maradhat egészen a következő igényig. Ezért kell a védelmi rendszert rendszeresen, a valós működést utánzó **bizonyító erejű vizsgálattal** (*proof test*) ellenőrizni, amely feltárja a rejtett hibákat, és a rendszert a lehető legjobban az „újszerű” állapotába hozza vissza.

2.2. táblázat. Az alapirányítás és a védelmi rendszer jellemző meghibásodási módjai

Alapfolyamat-irányítás (BPCS)	Védelmi rendszer (SIS)
A beavatkozó szerv véghelyzetbe szorul vagy beragad	Igénykor nem működik (veszélyes meghibásodás)
A szabályozó kimenete túl nagy vagy túl kicsi	Késve működik (veszélyes meghibásodás)
A távadó jele vagy a beavatkozó mozgása akadozik	Igény nélkül működik, téves leállítás (biztonságos meghibásodás)
A hiba üzem közben kiderül	A veszélyes hiba rejtve maradhat a következő igényig vagy proof tesztig

A két rendszer függetlenségét a szabványok kezdettől fogva megkövetelik. Az 1990-es évek közepéig a fizikai szétválasztás volt az egyetlen elfogadott megoldás. Ma, amikor mindkét funkciót programozható eszközök valósítják meg, a szabványok (IEC 61511) a funkcionális függetlenséget követelik: külön érzékelőket, külön logikai egységet és külön beavatkozó szerveket (vagy ezek igazoltan független részeit). Az adatkommunikáció közös hálózaton is történhet, ha nem befolyásolhatja a védelmi funkció működését. A függetlenség hiánya, ahogyan Buncefield példáján láttuk, a rétegzett védelem egyik leggyakoribb gyengesége.

2.4. Az üzembiztonság fogalomrendszere

A biztonság csak az egyik tulajdonsága annak, amit az angol szakirodalom *dependability* néven foglal össze, és amelyet ebben a könyvben **üzembiztonságnak** nevezünk: a rendszer azon képessége, hogy olyan szolgáltatást nyújtson, amelyben jogosan meg lehet bízni. Avižienis, Laprie, Randell és Landwehr 2004-es tanulmánya ennek egységes taxonómiáját adta meg (Avižienis és mtsai. 2004), amelyet a 2.3. ábra foglal össze.



2.3. ábra. Az üzembiztonság fogalomrendszere: tulajdonságok, fenyegetések és eszközök Avižienis és munkatársai nyomán.

Az üzembiztonság tulajdonságai

- **Rendelkezésre állás** (*availability*): a rendszer adott időpontban kész a helyes működésre.
- **Megbízhatóság** (*reliability*): a rendszer adott időtartamon át folyamatosan helyesen működik.
- **Biztonság** (*safety*): a rendszer működése nem okoz katasztrofális következményt a felhasználókra és a környezetre.
- **Integritás** (*integrity*): a rendszer nem változik meg nem megengedett módon.
- **Karbantarthatóság** (*maintainability*): a rendszer javítható és módosítható.

A magyar szakirodalomban a *reliability* fordítására a megbízhatóság és a hibamentesség szó egyaránt használatos. Ebben a könyvben a **megbízhatóság** szót a szűkebb, matematikailag pontos értelemben használjuk: annak valószínűsége, hogy a rendszer egy adott időtartamon át hibamentesen működik. Ezt az $R(t)$ megbízhatósági függvény írja le (8. fejezet).

A megbízhatóság és a rendelkezésre állás különbségét érdemes kiemelni. A megbízhatóság a $[0, t]$ intervallum *egészére* vonatkozik: az első meghibásodásig eltelt időről szól. A rendelkezésre állás egy *adott pillanatra* vonatkozik: lehet, hogy a rendszer közben többször meghibásodott, de a vizsgált pillanatban (a javítás után) éppen működik. Javítható rendszereknél a rendelkezésre állás tartósan magas lehet akkor is, ha a megbízhatóság hosszú időre nézve közel nullára csökken. Ha a

meghibásodások közötti átlagos idő (MTTF) T_1, az átlagos helyreállítási idő (MTTR) T_2, akkor hosszú távon a rendelkezésre állás

$$A = \frac{T_1}{T_1 + T_2} = \frac{\mathrm{MTTF}}{\mathrm{MTTF} + \mathrm{MTTR}}. \tag{2.3}$$

Ezt az összefüggést a 12. fejezetben Markov-modellből vezetjük le.

A biztonság és a megbízhatóság között sajátos feszültség is lehet. Egy olyan védelmi rendszer, amely minden kétes helyzetben leállítja a folyamatot, biztonságos, de a gyakori téves leállítások rontják a rendelkezésre állást, és arra csábítják a kezelőket, hogy áthidalják a védelmet. Egy jól tervezett rendszer mindkét szempontot egyensúlyban tartja; ezt a 13. fejezetben a téves leállítások gyakoriságának (*spurious trip rate*) elemzésével is vizsgáljuk.

2.5. Hibaok, hiba, meghibásodás

A magyar „hiba” szó három különböző angol fogalmat takar, és a pontos elemzéshez ezeket szét kell választani.

A hibaok–hiba–meghibásodás lánc

- **Meghibásodás** (*failure*): az az esemény, amikor a rendszer által nyújtott szolgáltatás eltér a helyes szolgáltatástól. Más megfogalmazásban: a rendszer elveszíti azt a képességét, hogy a megkövetelt funkciót ellássa.
- **Hiba** (*error*): a rendszer állapotának az a része, amely a meghibásodáshoz vezethet; hibás belső állapot.
- **Hibaok** (*fault*): a hiba megállapított vagy feltételezett oka.

A lánc működése: egy hibaok (például egy forrasztási repedés vagy egy szoftverben elírt határérték) sokáig **szunnyadhat** (*dormant fault*), amíg egy bemenet vagy egy környezeti feltétel **aktiválja**, és hibás belső állapotot hoz létre. Ha a hibás állapot eléri a rendszer kimenetét, és a szolgáltatás eltér a helyestől, meghibásodás következik be.

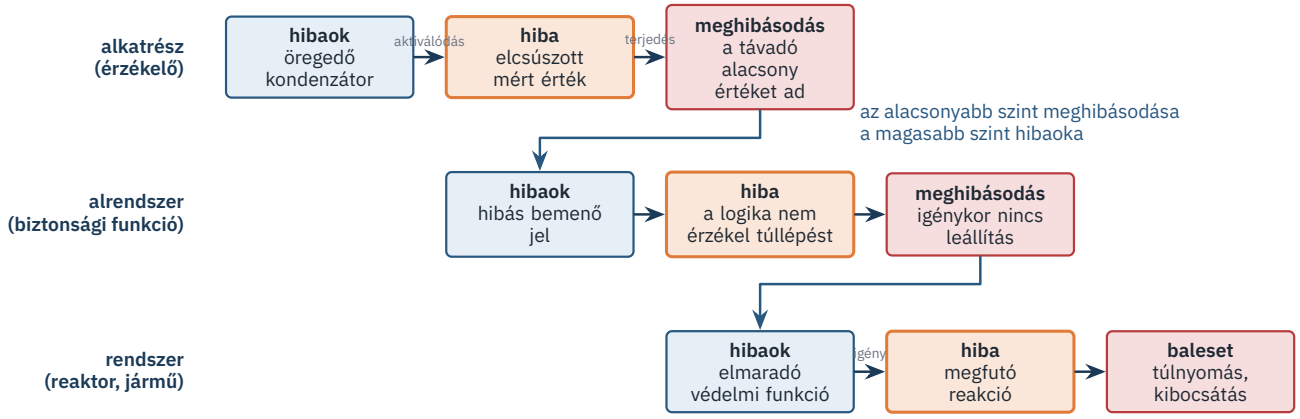
A lánc **rekurzív** (2.4. ábra): az alacsonyabb rendszerszint meghibásodása a magasabb szint szempontjából hibaok. Az érzékelő meghibásodása a biztonsági funkció hibaoka, a biztonsági funkció meghibásodása a reaktor szintjén hibaok, amely egy igény (a hűtés kiesése) esetén balesethez vezet. Az ISO 26262 fogalomhasználatában a jármű (az *item*) szintjén megjelenő meghibásodás a veszély, míg egy komponens meghibásodása a jármű szintjén hibaokként jelenik meg.

Ez a rekurzió a hibafa-elemzés lényege: a csúcseseménytől lefelé haladva minden szinten megkeressük azokat a hibaokokat, amelyek az adott szint meghibásodásához vezetnek (10. fejezet). Az FMEA a lánc másik irányát követi: egy alkatrész meghibásodásából kiindulva felfelé keresi a hatásokat (6. fejezet).

2.6. A meghibásodások osztályozása

2.6.1. Véletlenszerű és szisztematikus meghibásodások

Az IEC 61508 két alapvetően különböző meghibásodásfajtát különböztet meg, és ez a megkülönböztetés a funkcionális biztonság egész módszertanát meghatározza.



2.4. ábra. A hibaok–hiba–meghibásodás lánc rekurzív terjedése a rendszerszintek között a ReaktorVéd példáján.

Véletlenszerű hardvermeghibásodás és szisztematikus meghibásodás

- **Véletlenszerű hardvermeghibásodás** (*random hardware failure*): véletlen időpontban bekövetkező meghibásodás, amely a hardver egy vagy több lehetséges degradációs mechanizmusából ered (öregedés, kopás, korrózió, félvezető-meghibásodás).
- **Szisztematikus meghibásodás** (*systematic failure*): olyan meghibásodás, amely determinisztikusan egy meghatározott okra vezethető vissza, és csak a terv, a gyártási folyamat, az üzemeltetési eljárás, a dokumentáció vagy más tényező módosításával szüntethető meg.

A véletlenszerű meghibásodások egy alkatrész-populációban statisztikailag jól leírhatók: a meghibásodási ráta mérhető, adatbázisokban gyűjthető, és valószínűségi modellekkel számolható. A szisztematikus meghibásodások ezzel szemben nem „ritkák” vagy „gyakoriak”, hanem vagy jelen vannak a rendszerben, vagy nincsenek. Egy specifikációs hiba minden példányban ott van, és mindig ugyanazon feltételek mellett aktiválódik. Az Ariane 5 mindkét inerciális egysége azért hibásodott meg egyszerre, mert ugyanaz a szisztematikus hiba volt bennük.

A **szoftverhibák kivétel nélkül szisztematikusak**: a szoftver nem kopik, nem öregszik, nem romlik el véletlenszerűen. Ezért a szoftver biztonsági integritását nem lehet meghibásodási rátával igazolni; helyette a fejlesztési folyamat szigorúságát (módszerek, verifikáció, független értékelés) kell a megkövetelt integritáshoz igazítani (15. fejezet). Ugyanez igaz a hardver tervezési hibáira is.

2.6.2. Biztonságos és veszélyes, észlelt és rejtett meghibásodások

A védelmi rendszerek elemzésében a meghibásodásokat két szempont szerint osztályozzuk (2.5. ábra):

- **biztonságos meghibásodás** (*safe failure*): nem akadályozza a biztonsági funkciót, sőt gyakran kiváltja (például téves leállítás); **veszélyes meghibásodás** (*dangerous failure*): megakadályozza, hogy a biztonsági funkció igény esetén működjön;
- **észlelt meghibásodás** (*detected*): a beépített diagnosztika (önteszt, plauzibilitásvizsgálat, figyelőáramkör) felfedi; **rejtett meghibásodás** (*undetected*): csak igénykor vagy a proof teszt során derül ki.

A négy részráta közül a **veszélyes, rejtett** (λ_{DU}) a legkritikusabb: ez az a meghibásodás, amely csendben megfosztja a rendszert a védelemtől. A **diagnosztikai lefedettség** (*diagnostic coverage*, DC) azt fejezi ki, hogy a veszélyes meghibásodások mekkora hányadát fedi fel a diagnosztika:



2.5. ábra. A meghibásodási ráta felosztása biztonságos/veszélyes és észlelt/rejtett részekre. A veszélyes, rejtett meghibásodások (λ_{DU}) a biztonsági funkció integritásának fő meghatározói.

$$DC = \frac{\lambda_{DD}}{\lambda_{DD} + \lambda_{DU}}. \tag{2.4}$$

Ezek a részráták az FMEDA (6. fejezet) eredményei, és a 13. fejezet PFD-számításának bemenő adatai.

2.6.3. Független, közös okú és kaszkád meghibásodások

A megbízhatósági modellek legtöbbje feltételezi, hogy az elemek egymástól függetlenül hibásodnak meg. A valóságban ez a feltevés gyakran sérül:

- **Közös okú meghibásodás** (*common cause failure*, CCF): két vagy több elem meghibásodása, amelyek egyetlen közös okra vezethetők vissza (közös tápellátás, azonos tervezési hiba, közös környezeti hatás, például hő, nedvesség vagy rezgés, közös karbantartási hiba).
- **Közös módusú meghibásodás** (*common mode failure*): több elem ugyanazon a módon hibásodik meg. Gyakran a közös ok következménye, de nem feltétlenül.
- **Kaszkád meghibásodás** (*cascading failure*): egy elem meghibásodása más elemek meghibásodását idézi elő (például egy rövidzár kiold egy közös biztosítékot).
- **Kétpontos meghibásodás** (*dual-point failure*, ISO 26262): két független hiba együttese sért meg egy biztonsági célt, tipikusan egy biztonsági elem és az azt védő biztonsági mechanizmus egyidejű meghibásodása.

A közös okú meghibásodás a redundáns rendszerek legfontosabb korlátja. Két azonos elem párhuzamos kapcsolása elméletben a meghibásodási valószínűség négyzetére csökkenti a kockázatot, a gyakorlatban azonban a közös okú rész gyakran nagyságrendekkel nagyobb, mint a független rész. A 13. fejezet számpéldájában a ReaktorVéd két redundáns szelepének PFD-jéből a közös okú tag nagyjából 89%-ot tesz ki.

2.6.4. Időbeli lefolyás és mérték szerint

A meghibásodások az időbeli lefolyásuk szerint lehetnek **tartósak** (*permanent*), **átmenetiek** (*transient*, például egy sugárzás okozta bitátbillenés) és **időszakosak** (*intermittent*, például egy laza csatlakozó). Az átmeneti hibaok rövid ideig hat, de a következménye tartós lehet: egy átbillent bit a memóriában hibás vezérlőjelhez vezethet. A működőképesség elvesztésének mértéke szerint degradációs, részleges, teljes és katasztrofális meghibásodást különböztetünk meg; a fokozatosan kialakuló (degradációs) meghibásodások előrejelzése a prediktív karbantartás tárgya.

2.7. A hibák kezelésének négy eszköze

Avižienis és munkatársai a hibák elleni védekezés négy alapvető eszközét különböztetik meg (Avižienis és mtsai. 2004):

1. **Hibamegelőzés** (*fault prevention*): a hibaokok keletkezésének megakadályozása gondos specifikációval, bevált tervezési elvekkel, minőségirányítással, képzéssel.
2. **Hibatűrés** (*fault tolerance*): a szolgáltatás fenntartása hibaok jelenlétében is, redundanciával, diagnosztikával, hibaelszigeteléssel és biztonságos állapotba vezérléssel.
3. **Hibaeltávolítás** (*fault removal*): a hibaokok felderítése és megszüntetése verifikációval (felülvizsgálat, analízis, tesztelés) a fejlesztés során, illetve karbantartással az üzemeltetés során.
4. **Hibaelőrejelzés** (*fault forecasting*): a jelen lévő hibaokok számának, a jövőbeli előfordulásuknak és következményeiknek becslése megbízhatósági modellekkel és elemzésekkel.

Egyik eszköz sem tökéletes önmagában. A hibamegelőzés nem garantálja, hogy nem marad hiba a rendszerben; a hibatűrés a közös okú hibák ellen korlátozottan hatékony; a tesztelés csak a hibák jelenlétét mutathatja ki, a hiányukat nem; az előrejelzés pedig csak annyira jó, amennyire a modell és az adatok. A biztonságkritikus rendszerek fejlesztése ezért mindig a négy eszköz összehangolt alkalmazása. A szabványok követelményrendszere is ezt tükrözi: a véletlenszerű hardvermeghibásodások ellen elsősorban a hibatűrés és az előrejelzés (kvantitatív célértékek), a szisztematikus meghibásodások ellen a hibamegelőzés és a hibaeltávolítás (folyamatkövetelmények) véd.

2.8. A futó esetek fogalmi térképe

ReaktorVéd: veszély, veszélyes esemény, kár

- **Irányított berendezés:** a 6 m³-es félszakaszos keverős reaktor a hűtőköpennyel, az adagolórendszerrel és a hűtővízkörrel.
- **Veszély:** az exoterm reakció hőtermelése és a reaktorban felhalmozódó, még el nem reagált reaktáns.
- **Veszélyes helyzet:** a reakciós fázisban a reaktor hűtése csak a hűtővízkörön múlik, és a reaktor közelében kezelők dolgoznak.
- **Veszélyes esemény:** a hűtés kiesése után a hőmérséklet megfut, a nyomás a tervezési nyomás fölé nő, és a reaktor felhasad.
- **Kár:** a közelben tartózkodók halálos sérülése, mérgező anyag kibocsátása, a berendezés megsemmisülése.
- **Biztonsági funkció:** „Ha bármelyik hőmérséklet-távadó a 95 °C-os határértéket meghaladja, a biztonsági rendszer 10 másodpercen belül zárja az adagoló vezeték mindkét gyorszáró szelepét.”

VészFék: egy funkció, két veszély

A vészfékező asszisztens kétféleképpen okozhat kárt. Ha **indokolatlanul fékez** (például egy híd árnyékát akadálynak érzékeli), a hátulról érkező jármű ráfuthat. Ha **nem fékez, amikor kellene**, a frontális ütközés elmarad megakadályozásra. Az első tipikusan meghibásodás vagy téves érzékelés következménye, a második gyakran az érzékelés teljesítőképességének korlátja (például ködben). A két veszély más-más szabvány fogalmaival kezelhető: az előbbi főként az ISO 26262 (funkcionális biztonság), az utóbbi az ISO 21448 (a rendeltetésszerű funkció biztonsága) tárgya (18. fejezet).

KoboCella: a veszély az együttműködésből fakad

A kollaboratív robotcellában a fő veszély a robotkar mozgási energiája és szorítóereje. A veszélyes helyzet maga a munka célja: a szerelő és a robot közös munkatérben dolgozik. A veszélyes esemény az ütközés vagy a becsípődés, a kár a szerelő sérülése. A cella biztonsági funkciói (sebesség- és távolságfelügyelet, erő- és teljesítménykorlátozás) nem megszüntetik a veszélyes helyzetet, hanem a kár súlyosságát és valószínűségét tartják elfogadható szinten.

Összefoglalás

- A veszély a kár forrása; a kockázat a kár valószínűségének és súlyosságának együttese; a biztonság a nem tolerálható kockázattól való mentesség.
- A kockázatcsökkentés három pontja: a veszély elkerülése, a veszélyes esemény megelőzése, a következmények csökkentése.
- A funkcionális biztonság az E/E/PE biztonsági rendszerek helyes működésén múló biztonság; a biztonsági funkciót meghatározott integritással kell megvalósítani.
- Az üzembiztonság tulajdonságai a rendelkezésre állás, a megbízhatóság, a biztonság, az integritás és a karbantarthatóság.
- A hibaok–hiba–meghibásodás lánc rekurzív: az alsóbb szint meghibásodása a felsőbb szint hibaoka.
- A véletlenszerű hardvermeghibásodások valószínűségi modellekkel, a szisztematikus meghibásodások (köztük minden szoftverhiba) folyamatkövetelményekkel kezelhetők.
- A veszélyes, rejtett meghibásodás (λ_{DU}) és a közös okú meghibásodás a védelmi rendszerek legfontosabb gyengeségei.

Önellenőrzés

1. Egy autópálya-alagútban a szellőztetés kiesése esetén a levegő szén-monoxid-koncentrációja megnőhet. Nevezd meg a veszélyt, a veszélyes helyzetet, a veszélyes eseményt és a kárt!
2. Miért nem lehet a szoftver biztonsági integritását meghibásodási rátával igazolni?
3. Egy rendszer meghibásodásai közötti átlagos idő 2000 óra, az átlagos javítási idő 10 óra. Mekkora a hosszú távú rendelkezésre állás? Mit mond ez a rendszer megbízhatóságáról 5000 órás időtartamra?
4. Egy nyomástávadó veszélyes meghibásodási rátája $\lambda_D = 2 \cdot 10^{-6}\,\mathrm{h}^{-1}$, amelyből a diagnosztika $1{,}2 \cdot 10^{-6}\,\mathrm{h}^{-1}$-et fed fel. Mekkora a diagnosztikai lefedettség és a λ_{DU}?

Megoldás.

1. Veszély: a mérgező szén-monoxid (a járművek kipufogógáza). Veszélyes helyzet: az alagútban járművek és utasok tartózkodnak, a szellőztetés kiesett. Veszélyes esemény: a koncentráció az egészségre ártalmas szint fölé nő, miközben az emberek nem tudják elhagyni az alagutat (például torlódás miatt). Kár: mérgezés, haláleset.
2. Mert a szoftverhibák szisztematikusak: nem véletlen időpontban, egymástól függetlenül következnek be, hanem determinisztikusan, meghatározott bemenetek és állapotok mellett, minden példányban azonosan. A meghibásodási ráta fogalma egy populáció véletlenszerű meghibásodásaira értelmes; a szoftver integritását a fejlesztési folyamat szigorúságával kell biztosítani.
3. $A = 2000/(2000 + 10) \approx 0{,}995$. Ez a rendelkezésre állás nagy, de nem mond semmit arról, hogy az első meghibásodás előtt a rendszer kibír-e 5000 órát. Exponenciális élettartam esetén $R(5000) = e^{-5000/2000} \approx 0{,}082$, vagyis a rendszer 92%-os valószínűséggel meghibásodik ennyi idő alatt, csak gyorsan javítják.

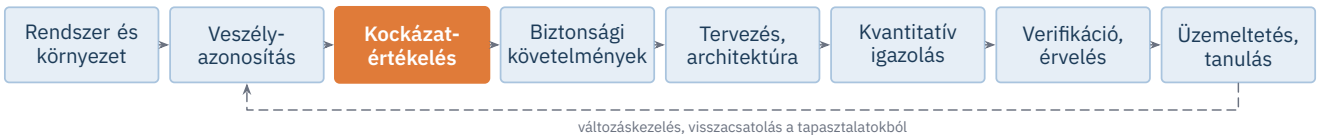
4. $DC = 1{,}2/2{,}0 = 60\%$, $\lambda_{DU} = 2 \cdot 10^{-6} - 1{,}2 \cdot 10^{-6} = 8 \cdot 10^{-7}\,\mathrm{h}^{-1}$.

Továbbolvasás

- Avižienis, A. et al. (2004): Basic concepts and taxonomy of dependable and secure computing. *IEEE TDSC* 1(1), 11–33 – az üzembiztonsági fogalmak alapműve (Avižienis és mtsai. 2004).
- ISO/IEC Guide 51:2014 – a biztonsági alapfogalmak forrása a szabványírók számára (ISO/IEC 2014).
- Rausand, M. (2014): *Reliability of Safety-Critical Systems.* Wiley, 2–3. fejezet – a meghibásodások osztályozása a védelmi rendszerek nézőpontjából (Rausand 2014).

3.

Kockázatértékelés és a kockázat elfogadhatósága



A veszélyek azonosítása után két kérdésre kell válaszolni: mekkora a kockázat, és elfogadható-e. Az első kérdés mérnöki becslés, a második társadalmi, jogi és szervezeti értékítélet. Ez a fejezet a kockázatkezelés folyamatával kezdődik, majd bemutatja a kockázat elfogadhatóságának nemzetközileg használt kritériumait (ALARP, GAMAB, MEM, F–N görbe). Részletesen tárgyalja a gyakorlatban legelterjedtebb, de sokszor rosszul használt eszközt, a kockázati mátrixot. Végül azt mutatja meg, hogyan lesz a kockázatértékelés eredményéből a biztonsági funkciókra kioszthatő, számszerű kockázatcsökkentési követelmény.

Tanulási eredmények

A fejezet végére képes leszel

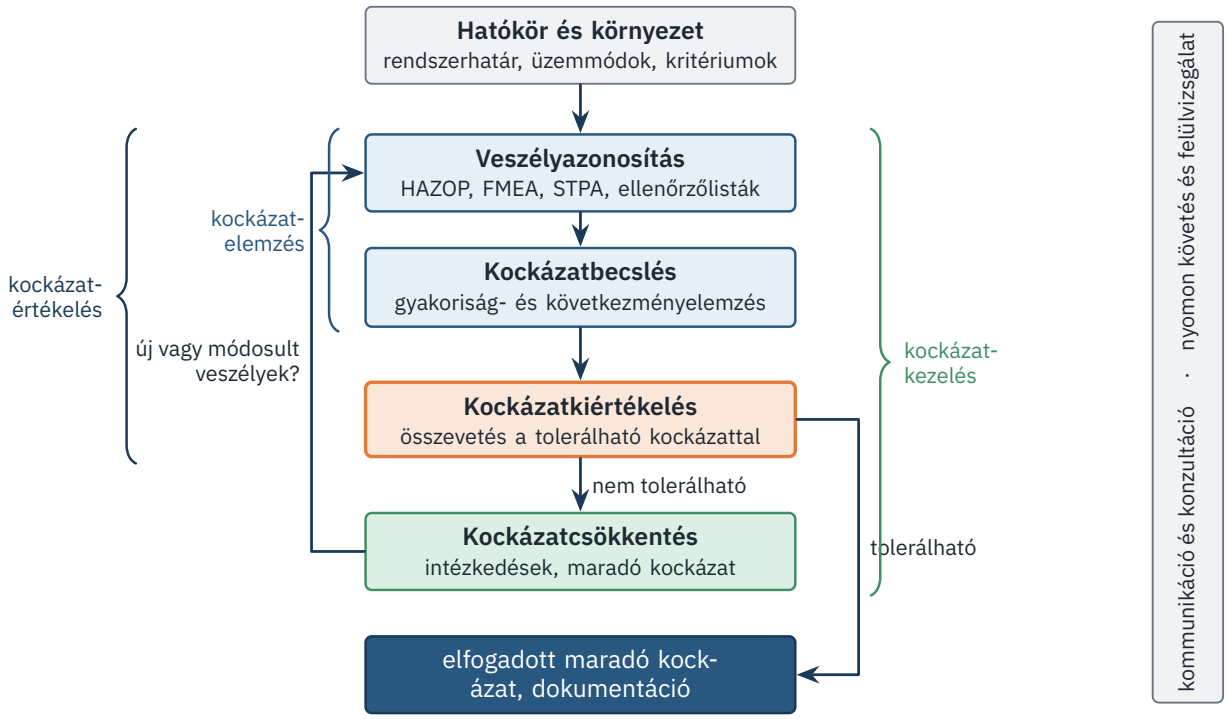
- leírni a kockázatkezelés folyamatát és fogalmait (kockázatelemzés, -kiértékelés, -értékelés, -kezelés);
- alkalmazni az ALARP-elvet, és megkülönböztetni a nem tolerálható, a tolerálható és az általánosan elfogadható kockázati tartományt;
- egyéni és társadalmi kockázatot számolni, és F–N görbét értelmezni;
- következetes kockázati mátrixot tervezni és kalibrálni, valamint felismerni a mátrixok tipikus hibáit;
- a tolerálható kockázatból meghatározni a szükséges kockázatcsökkentést.

3.1. A kockázatkezelés folyamata

A kockázatkezelés (*risk management*) a kockázatok azonosítására, elemzésére, kiértékelésére és kezelésére irányuló tevékenységek összehangolt rendszere. Általános kerete az ISO 31000 (ISO 2018b), a módszerek tárháza az IEC 31010 (IEC 2019a), a biztonsági szabványok pedig ugyanezt a folyamatot követik a saját fogalmaikkal (3.1. ábra).

A kockázatkezelés fogalmai

- **Kockázatelemzés** (*risk analysis*): a rendelkezésre álló információ módszeres felhasználása a veszélyek azonosítására és a kockázat becslésére.



3.1. ábra. A kockázatkezelés folyamata. A kockázatelemzés és a kockázatkiértékelés együtt alkotja a kockázatértékelést; ha a kockázat nem tolerálható, a csökkentő intézkedések után az elemzést meg kell ismételni.

- **Kockázatbecslés** (*risk estimation*): a kockázat mértékének meghatározása gyakoriság- és következményelemzéssel.
- **Kockázatkiértékelés** (*risk evaluation*): a becsült kockázat összevetése a kockázati kritériumokkal, annak eldöntésére, hogy a kockázat tolerálható-e.
- **Kockázatértékelés** (*risk assessment*): a kockázatelemzés és a kockázatkiértékelés együttese.
- **Kockázatcsökkentés, kockázatkezelés** (*risk reduction, risk treatment*): intézkedések kiválasztása és megvalósítása a kockázat csökkentésére, majd a maradó kockázat újbóli értékelése.

A folyamat nem egyszeri, hanem iteratív. Egy kockázatcsökkentő intézkedés (például egy új védelmi funkció) maga is új veszélyeket hozhat (téves leállítás, karbantartási hibák), ezért az elemzést meg kell ismételni. A folyamatos kockázatkezelés (*continuous risk management*, CRM) a projektmenedzsmentben is elterjedt formában hat lépés ismétlődő ciklusaként írja le ezt: azonosítás, elemzés, tervezés, nyomon követés, beavatkozás, és minden lépést kísérő kommunikáció és dokumentálás (Goldberg és mtsai. 1994). A dokumentálás nem adminisztratív teher: a kockázatértékelés eredménye a biztonsági érvelés (17. fejezet) egyik legfontosabb bizonyítéka.

3.1.1. Kvalitatív, félkvantitatív és kvantitatív értékelés

A kockázatbecslés pontossága és ráfordítása széles skálán mozog:

- a **kvalitatív** értékelés szóbeli kategóriákkal dolgozik (például „ritka”, „súlyos”), gyors és átfogó, de szubjektív;

- a **félkvantitatív** értékelés a kategóriákhoz számtartományokat (például gyakoriság-nagyságrendeket) rendel, így következetesebb és ellenőrizhetőbb. Ilyen a kockázati mátrix kalibrált formája és a LOPA (11. fejezet);
- a **kvantitatív kockázatelemzés** (*quantitative risk assessment*, QRA) a gyakoriságokat és a következményeket modellekkel (hibafa, eseményfa, terjedési és hatásmodellek) számolja, és egyéni, illetve társadalmi kockázati mutatókat ad.

A jó gyakorlat a fokozatosság: kvalitatív szűréssel kezdünk, és a nagy vagy bizonytalan kockázatú forgatókönyveket vizsgáljuk egyre részletesebben.

3.2. Mikor elfogadható a kockázat?

A tolerálható kockázat meghatározása nem tisztán mérnöki feladat. A társadalom a kockázatot nem csak a várható kár nagysága szerint ítéli meg: más a viszonya az önként vállalt (autóvezetés) és a kényszerűen elszenvedett (egy veszélyes üzem közelében lakás), az ismert és az új, a megszokott és a katasztrofális kockázathoz. 2012-ben Magyarországon 541-en haltak meg közúti balesetben, mégis naponta milliók közlekednek, vagyis a társadalom ezt a kockázatot a gyakorlatban elfogadja. Ugyanez a szám egy ipari létesítmény esetén elképzelhetetlen volna. A mérnöki gyakorlat ezért néhány jól definiált, nemzetközileg elterjedt kritériumrendszert használ.

3.2.1. Az ALARP-elv

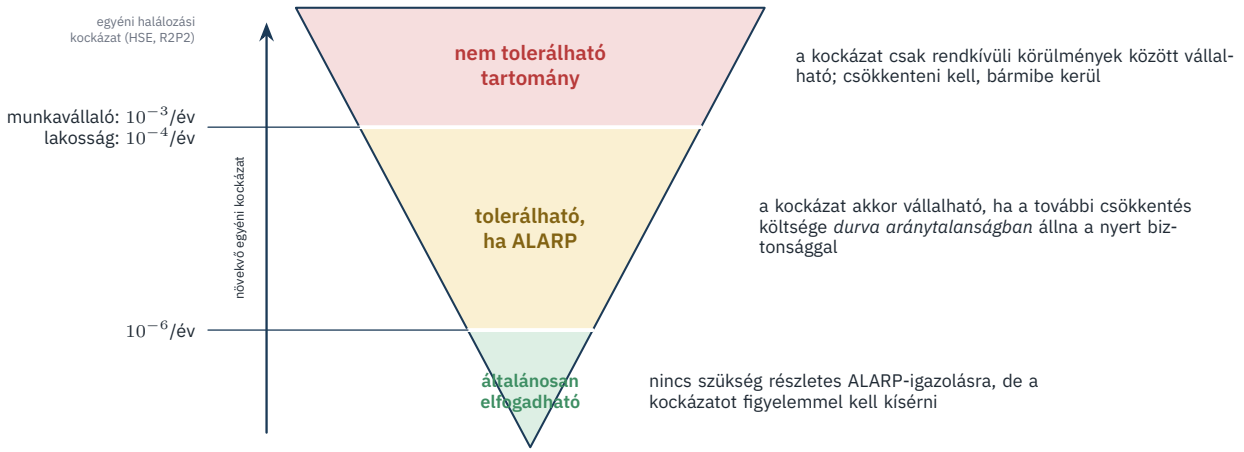
Az egyesült királysági munkavédelmi hatóság (HSE) *Reducing Risks, Protecting People* (R2P2) című kiadványa fogalmazta meg a ma legszélesebb körben használt keretrendszert, a kockázat tolerálhatóságának modelljét (Health and Safety Executive 2001). A kockázati teret három tartományra osztja (3.2. ábra):

- **Nem tolerálható tartomány:** a kockázat csak rendkívüli körülmények között vállalható, mértékétől függetlenül csökkenteni kell.
- **Tolerálható tartomány, ha ALARP:** a kockázat akkor vállalható, ha „ésszerűen megvalósítható módon a lehető legkisebb” (*As Low As Reasonably Practicable*, ALARP). Ez azt jelenti, hogy a további kockázatcsökkentés költsége (pénzben, időben, erőfeszítésben) **durva aránytalanságban** (*gross disproportion*) állna az elérhető biztonsági előnnyel.
- **Általánosan elfogadható tartomány:** a kockázat olyan kicsi, hogy nincs szükség részletes ALARP-igazolásra, de a kockázatot továbbra is figyelemmel kell kísérni.

A HSE irányértékei az egyéni halálozási kockázatra: a nem tolerálható és a tolerálható tartomány határa munkavállalókra évi 10^{-3}, a lakosságra évi 10^{-4}; az általánosan elfogadható tartomány felső határa mindkét csoportra évi 10^{-6}. Ezek irányértékek, nem jogszabályi határok, de a nemzetközi gyakorlatban (és sok vállalati kockázati kritériumrendszerben) referenciaként szolgálnak.

> **ALARP: „Reasonably Practicable”, nem „Possible”**
>
> A rövidítést gyakran hibásan „As Low As Reasonably Possible” formában oldják fel; ez a hiba a könyv 2014-es elődjében is szerepelt. A különbség nem szőrszálhasogatás. A *practicable* (megvalósítható) azt jelenti, hogy a kockázatcsökkentés költségét és hasznát mérlegelni kell, és az intézkedés csak akkor hagyható el, ha a költség *durván aránytalan* a haszonhoz képest. A bizonyítási teher a kockázatot létrehozón van: neki kell igazolnia, hogy minden ésszerűen megvalósítható intézkedést megtett. A rokon brit jogi fogalom a SFAIRP (*So Far As Is Reasonably Practicable*).

Az ALARP-igazolás gyakran költség–haszon elemzéssel történik. Ha egy intézkedés ΔR kockázatcsökkentést (például évi várható halálesetszám-csökkenést) eredményez T év élettartam alatt, és költsége K, akkor az intézkedés kötelező, ha



3.2. ábra. A kockázat tolerálhatóságának modellje (HSE, R2P2) az egyéni halálozási kockázat irányértékeivel.

$$K < d \cdot \Delta R \cdot T \cdot V_{\mathrm{VPF}}, \tag{3.1}$$

ahol V_{VPF} egy statisztikai haláleset megelőzésének értéke (*value of preventing a fatality*), $d \geq 1$ pedig az aránytalansági tényező. Ez utóbbi a kockázat nagyságával nő: a nem tolerálható határhoz közel 10 is lehet. Az összefüggés a döntést segíti, de nem helyettesíti a mérnöki és etikai mérlegelést.

3.2.2. Más kritériumrendszerek: GAMAB és MEM

A vasúti szabványok (EN 50126 (CENELEC 2017)) két további elvet is ismernek:

- **GAMAB** (*Globalement Au Moins Aussi Bon*, francia eredetű): egy új rendszer összességében legalább olyan biztonságos legyen, mint az általa kiváltott, már elfogadott rendszer. Az elv az összehasonlítás alapját egy ismert referenciában keresi.
- **MEM** (*Minimale Endogene Mortalität*, német eredetű): a legalacsonyabb természetes halálozási arány (a fejlett országokban a fiatal korosztálynál nagyjából évi $2 \cdot 10^{-4}$) ne nőjön meg érzékelhetően egy technikai rendszer miatt. Egyetlen rendszer egyéni kockázata tipikusan ennek legfeljebb 5%-a, azaz évi 10^{-5} lehet.

A három elv más-más kérdésre ad választ (mennyire csökkentsük, legyen legalább olyan jó, ne növelje érdemben az alapkockázatot), de a gyakorlatban hasonló nagyságrendű kritériumokhoz vezetnek.

3.2.3. Egyéni kockázat

Az egyéni kockázat (*individual risk per annum*, IRPA) egy meghatározott személy évi halálozási valószínűsége a vizsgált tevékenység miatt. Ha az i-edik veszélyes forgatókönyv gyakorisága F_i, a halálos kimenetel feltételes valószínűsége $P_{\mathrm{h},i}$, a személy jelenlétének valószínűsége pedig $P_{\mathrm{j},i}$, akkor

$$\mathrm{IRPA} = \sum_i F_i \, P_{\mathrm{h},i} \, P_{\mathrm{j},i}. \tag{3.2}$$

Egy kezelő egyéni kockázata

Egy kezelő a műszakja alatt a ReaktorVéd reaktorcsarnokában az idő 25%-át tölti, a többit a vezérlőteremben. Három forgatókönyvet vizsgálnak:

3.1. táblázat. Egy kezelő egyéni kockázatának összetevői

Forgatókönyv	F_i [1/év]	$P_{\mathrm{h},i}$	$P_{\mathrm{j},i}$	hozzájárulás [1/év]
reaktor felhasadása megfutó reakció miatt	$6{,}8 \cdot 10^{-6}$	0,9	0,25	$1{,}5 \cdot 10^{-6}$
mérgező gőz kis szivárgása	10^{-2}	10^{-4}	0,25	$2{,}5 \cdot 10^{-7}$
tűz a betöltőnyílásnál	10^{-4}	0,1	0,25	$2{,}5 \cdot 10^{-6}$
összesen				$\mathbf{4{,}3 \cdot 10^{-6}}$

Az eredmény a tolerálható tartomány alsó részébe esik, így ALARP-igazolás szükséges. A legnagyobb hozzájárulás a tűz forgatókönyvé, tehát a kockázatcsökkentést ott érdemes kezdeni (például zárt betöltéssel), nem a látványosabb, de ritkább reaktorfelhasadásnál. A reaktorfelhasadás gyakorisága már a biztonsági funkcióval együtt értendő; ezt az értéket a 11. fejezet eseményfája adja.

3.2.4. Társadalmi kockázat és az F–N görbe

A társadalmi kockázatot az **F–N görbe** ábrázolja (3.3. ábra): a vízszintes tengelyen az áldozatok száma (N), a függőlegesen annak a gyakorisága (F), hogy egy évben N vagy annál több áldozatot követelő baleset történik. Mindkét tengely logaritmikus. A kritériumvonal általános alakja

$$F(N) = \frac{C}{N^{\alpha}}, \tag{3.3}$$

ahol $\alpha = 1$ kockázatsemleges, $\alpha > 1$ kockázatkerülő hozzáállást fejez ki: a sok áldozatot követelő balesetek gyakoriságát aránytalanul jobban korlátozza. A holland külső biztonsági szabályozás például $C = 10^{-3}$ és $\alpha = 2$ értékkel dolgozik (tíz vagy több áldozat esetén).

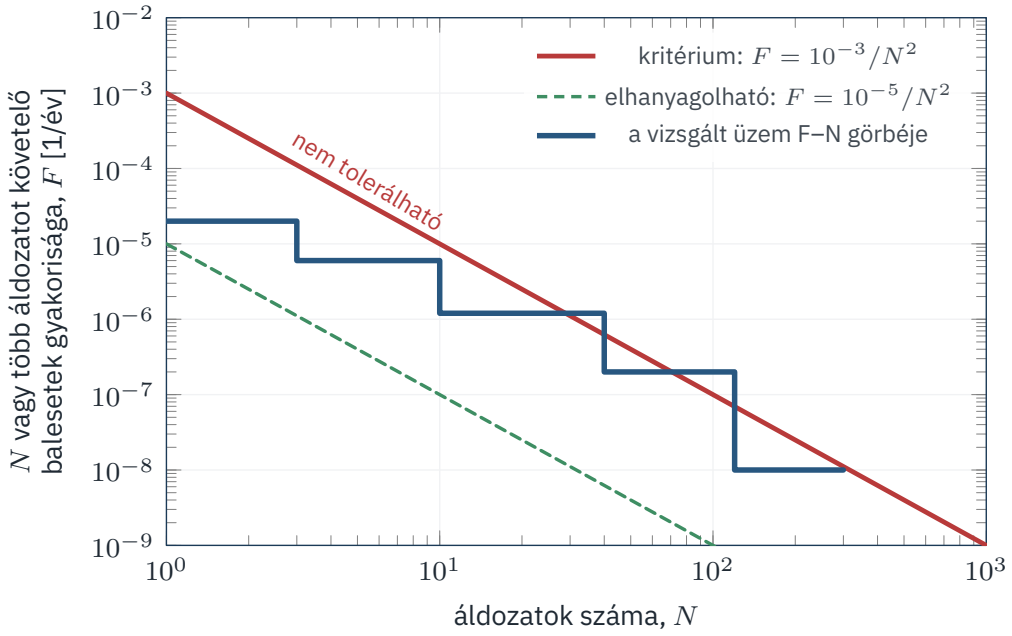
A kockázatkerülés ($\alpha > 1$) a mérnöki megfontolásokban is megjelenik. Az iso-kockázati görbe általánosított alakja $F \cdot C^{\alpha} =$ állandó. Ha egy csaknem elhanyagolható súlyosságú eseménynél (C_0) a tolerálható gyakoriság F_0, akkor egy C súlyosságú eseménynél

$$F_{\mathrm{tol}}(C) = F_0 \left(\frac{C_0}{C} \right)^{\alpha}. \tag{3.4}$$

Ez az összefüggés a kockázati mátrix celláinak kalibrálásához ad elvi alapot.

3.3. A kockázati mátrix

A kockázati mátrix (*risk matrix*) a legelterjedtebb kvalitatív és félkvantitatív kockázatértékelési eszköz. A következmény súlyosságát és a bekövetkezés gyakoriságát kategóriákba sorolja, és a két kategória kombinációjához kockázati szintet rendel. Népszerűségét egyszerűségének és szemléletességének



3.3. ábra. F–N görbe. A vizsgált üzem lépcsős görbéje sehol nem lépheti át a kritériumvonalat.

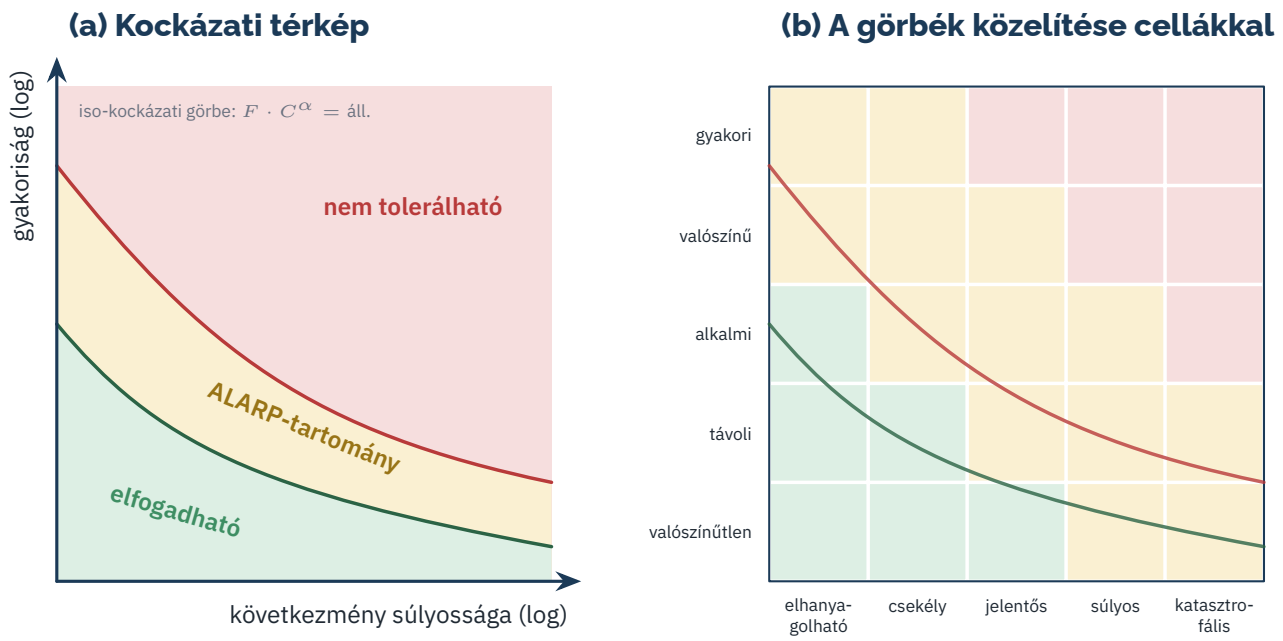
köszönheti; a NASA mérnöki eszköztára is az előzetes kockázatelemzés alapeszközeként mutatja be (Goldberg és mtsai. 1994). Ugyanakkor a kockázati mátrix az egyik leggyakrabban rosszul használt eszköz is.

3.3.1. A mátrix felépítése

A mátrix az iso-kockázati görbék diszkrét közelítése (3.4. ábra). A felépítés lépései:

1. **A gyakorisági kategóriák meghatározása.** Például a MIL-STD-882E (U.S. Department of Defense 2012) szerinti fokozatok: gyakori, valószínű, alkalmi, távoli, valószínűtlen, kizárt. A félkvantitatív változatban minden kategóriához nagyságrendi tartomány tartozik.
2. **A súlyossági kategóriák meghatározása.** Például: katasztrofális, kritikus, jelentős, elhanyagolható. Minden kategóriát érdemes több dimenzióban (emberi sérülés, környezeti kár, anyagi kár, hírnév) definiálni.
3. **A kockázati szintek hozzárendelése a cellákhoz.** A cellák az iso-kockázati görbéket közelítik. Fontos, hogy az elfogadhatósági határokat nem az elemző, hanem a kockázatkezelésért felelős vezetés határozza meg.
4. **Kalibráció.** Néhány ismert, elfogadottnak tekintett veszélyhelyzetet elhelyezünk a mátrixban, és ellenőrizzük, hogy a besorolásuk megfelel-e a szervezet értékítéletének. Ezek a kalibrációs pontok a későbbi értékelések viszonyítási alapjai.

A félkvantitatív mátrix kategóriáit célszerű logaritmikusan, egy-egy nagyságrendnyi lépésekkel definiálni (3.2. táblázat). Így a cellák közötti átlós vonalak valóban iso-kockázati vonalak: egy lépés lefelé a gyakoriságban és egy lépés felfelé a súlyosságban azonos kockázatot jelent ($\alpha = 1$ esetén).



3.4. ábra. A kockázati mátrix mint az iso-kockázati görbék közelítése: (a) folytonos kockázati térkép, (b) a görbék közelítése cellákkal.

3.2. táblázat. Egy félkvantitatív kockázati mátrix kategóriái (példa)

Gyakoriság	Tartomány [1/év]	Súlyosság	Emberi következmény	Anyagi kár
gyakori	$> 10^{-1}$	katasztrofális	több haláleset	$> 10^{10}$ Ft
valószínű	$10^{-2} – 10^{-1}$	súlyos	egy haláleset vagy maradandó egészségkárosodás	$10^{9} – 10^{10}$ Ft
alkalmi	$10^{-3} – 10^{-2}$	jelentős	súlyos, visszafordítható sérülés	$10^{8} – 10^{9}$ Ft
távoli	$10^{-4} – 10^{-3}$	csekély	könnyű sérülés, elsősegély	$10^{7} – 10^{8}$ Ft
valószínűtlen	$< 10^{-4}$	elhanyagolható	nincs sérülés	$< 10^{7}$ Ft

3.3.2. Tervezési szabályok

A 2014-es jegyzet a NASA eszköztára nyomán néhány fontos tervezési szabályt fogalmazott meg, amelyek ma is érvényesek, és amelyeket a későbbi kutatások megerősítettek:

- **Kevés kategória.** Az értékelés szubjektív, ezért a túl finom felbontás (például 12×8-as mátrix) nem növeli a pontosságot, csak a következetlenséget. A 4×4-es vagy 5×5-ös mátrix általában elég.
- **Kevés kockázati zóna.** Három zóna (nem tolerálható, tolerálható ALARP mellett, általánosan elfogadható) a legtöbb célra megfelel, és közvetlenül illeszkedik az ALARP-modellhez.

- **Folytonos, monoton zónák.** Egy cellából a szomszédos cellába lépve legfeljebb egy kockázati szintet lehessen ugrani, és a kockázat ne csökkenhessen, ha a gyakoriság vagy a súlyosság nő.
- **Kalibrált és dokumentált.** Minden kategória definíciója legyen írásban rögzítve, mérhető tartományokkal, és a mátrixot kalibrálni kell ismert esetekkel.

3.3.3. A kockázati mátrixok korlátai

Louis Anthony Cox 2008-ban megjelent, nagy hatású tanulmánya (Cox 2008) matematikailag megmutatta, hogy a kockázati mátrixoknak súlyos korlátai vannak:

- **Gyenge felbontás:** a mátrix kvantitatívan nagyon különböző kockázatokat sorolhat azonos cellába, és alig különbözőket eltérő cellákba.
- **Rangsorhibák:** negatívan korreláló gyakoriság és súlyosság esetén (gyakori, de enyhe, illetve ritka, de súlyos események) a mátrix a kisebb kockázathoz rendelhet magasabb szintet. Ez rossz erőforrás-elosztáshoz vezethet.
- **Szubjektív bemenetek:** a kategóriákba sorolás értelmezésfüggő, és a bizonytalanság a mátrixban nem látszik.
- **Az összegzés hiánya:** a mátrix az egyes forgatókönyveket külön-külön értékeli, de a rendszer teljes kockázatát nem összegzi. Sok, egyenként „zöld” kockázat együtt nem tolerálható lehet.

Cox következtetése nem az, hogy a mátrixot el kell vetni, hanem hogy gondosan, a korlátainak tudatában kell használni: elsősorban szűrésre és kommunikációra, nem pedig erőforrás-allokációra vagy kvantitatív célok igazolására. A nagy kockázatú forgatókönyveket a III. rész kvantitatív módszereivel kell tovább vizsgálni.

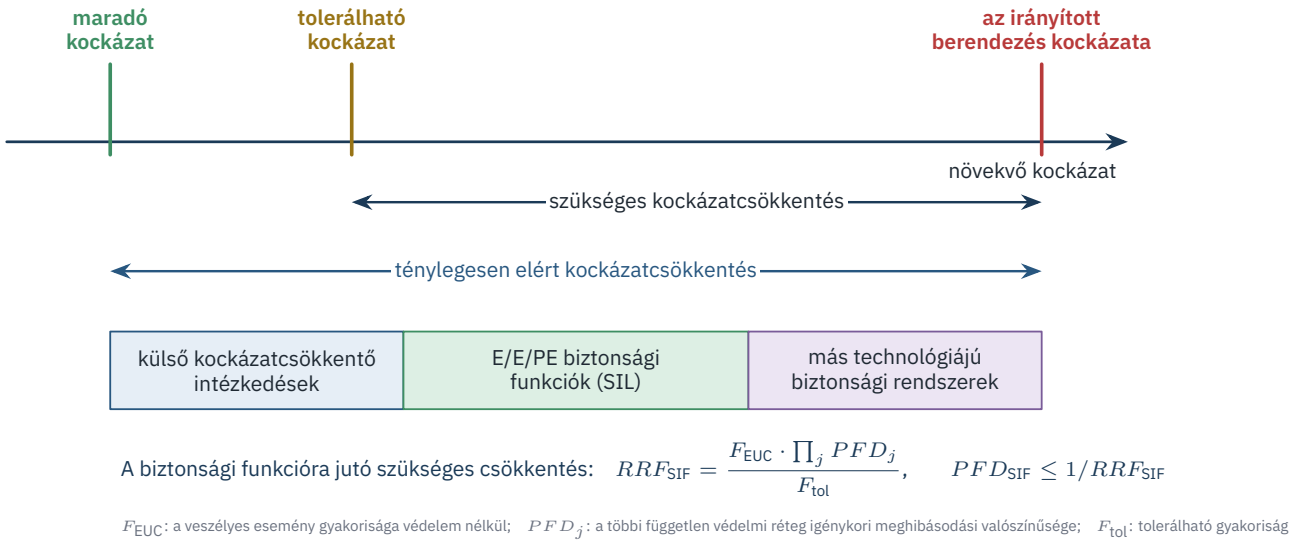
> **Aktív és passzív biztonsági megoldások a mátrixban**
>
> A kockázati mátrix jól szemlélteti a kockázatcsökkentés két irányát. A **passzív biztonsági elemek** (például a gyűrődőzóna, a védőburkolat vagy a kármentő) elsősorban a következmény súlyosságát csökkentik, így a pontot balra mozdítják. Az **aktív biztonsági megoldások** (például a vészfékező asszisztens vagy a reaktor biztonsági leállítása) elsősorban a bekövetkezés gyakoriságát csökkentik, így a pontot lefelé mozdítják.

3.4. A szükséges kockázatcsökkentés meghatározása

A kockázatértékelés célja végső soron egy számszerű követelmény: mennyivel kell csökkenteni a kockázatot, és ebből mennyit kell a biztonsági funkcióknak teljesíteniük. Az IEC 61508 ezt a 3.5. ábrán látható módon írja le.

Az irányított berendezés kockázata (védelem nélkül) egy adott veszélyes eseményre F_{EUC} gyakoriságú. Ha a tolerálható gyakoriság F_{tol}, akkor a szükséges teljes kockázatcsökkentési tényező $F_{\text{EUC}}/F_{\text{tol}}$. Ezt a különböző védelmi rétegek együtt teljesítik: külső kockázatcsökkentő intézkedések (például egy mechanikus biztonsági szelep), más technológiájú biztonsági rendszerek és az E/E/PE biztonsági funkciók. Ha a többi réteg igénykori meghibásodási valószínűsége PFD_j, akkor a biztonsági funkcióra jutó kockázatcsökkentési tényező (*risk reduction factor*, RRF) és a megengedhető legnagyobb igénykori meghibásodási valószínűség (PFD):

$$RRF_{\text{SIF}} = \frac{F_{\text{EUC}} \cdot \prod_j PFD_j}{F_{\text{tol}}}, \qquad PFD_{\text{SIF}} \leq \frac{1}{RRF_{\text{SIF}}}. \tag{3.5}$$



3.5. ábra. A szükséges kockázatcsökkentés felosztása az IEC 61508 szemléletében: a biztonsági funkcióknak a többi védelmi réteg által nem fedezett részt kell teljesíteniük.

Az IEC 61508 ennek kapcsán három alapelvet rögzít, amelyeket a 2014-es jegyzet is kiemelt: kockázatmentes állapot soha nem érhető el; a biztonságot már a tervezési folyamat elején figyelembe kell venni; a nem tolerálható kockázatot mindenképpen csökkenteni kell.

ReaktorVéd: mekkora kockázatcsökkentés kell?

A vállalat kockázati kritériuma szerint egy olyan forgatókönyv tolerálható gyakorisága, amely egy munkavállaló halálát okozhatja, $F_{\text{tol}} = 10^{-5}$/év (ez a HSE-tartomány közepére esik, kellő tartalékot hagyva más forgatókönyvekre). A hűtés kiesése évente $0{,}1$ gyakorisággal fordul elő, és az esetek felében $(0{,}5)$ a reaktor éppen az exoterm fázisban van. A magas hőmérséklet riasztása és a kezelő beavatkozása független védelmi rétegként $PFD = 0{,}1$ valószínűséggel hibázik. Így a biztonsági funkció nélkül a veszélyes esemény gyakorisága

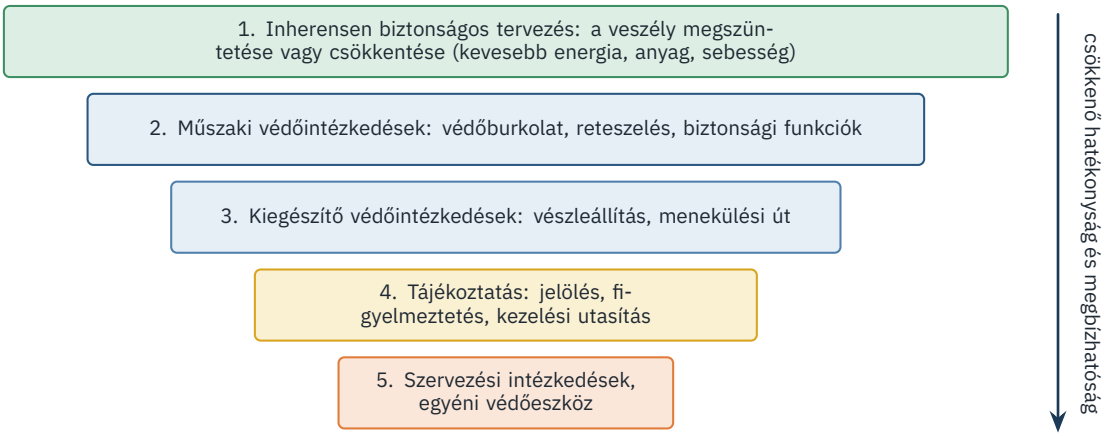
$$F = 0{,}1 \cdot 0{,}5 \cdot 0{,}1 = 5 \cdot 10^{-3}/\text{év},$$

a biztonsági funkcióra jutó szükséges kockázatcsökkentés $RRF \geq 5 \cdot 10^{-3}/10^{-5} = 500$, vagyis $PFD_{\text{SIF}} \leq 2 \cdot 10^{-3}$. Ez a 4. fejezetben bevezetett SIL 2 tartományba esik ($10^{-3} \leq PFD < 10^{-2}$), annak is a szigorúbb felébe. A gondolatmenetet a 11. fejezet a LOPA módszerével formalizálja, a 13. fejezet pedig igazolja, hogy a tervezett biztonsági funkció teljesíti a követelményt.

3.4.1. Az intézkedések hierarchiája

A kockázatcsökkentő intézkedések nem egyenértékűek. A gépbiztonsági alapszabvány, az ISO 12100 (ISO 2010) **háromlépcsős módszere** és a munkavédelemben használt intézkedési hierarchia egyaránt azt mondja ki, hogy a hatékonyabb, megbízhatóbb intézkedésekkel kell kezdeni (3.6. ábra).

1. **Inherensen biztonságos tervezés:** a veszély megszüntetése vagy csökkentése magában a tervben. Trevor Kletz folyamatbiztonsági elvei szerint (Kletz és Amyotte 2010): csökkentsük a veszélyes anyag mennyiségét (*minimize*), cseréljük kevésbé veszélyesre (*substitute*), mérsékeljük a körülményeket (*moderate*: alacsonyabb nyomás, hőmérséklet), egyszerűsítsük a rendszert (*simplify*).



3.6. ábra. Az intézkedések hierarchiája. Az inherensen biztonságos tervezés a leghatékonyabb; a tájékoztatás és a szervezési intézkedések csak kiegészítői a műszaki megoldásoknak.

2. **Műszaki védőintézkedések:** védőburkolatok, reteszelések, biztonsági funkciók, amelyek a megmaradt veszélyek ellen védenek.
3. **Kiegészítő védőintézkedések:** vészleállító, menekülési lehetőség, mentési eszközök.
4. **Tájékoztatás:** jelölések, figyelmeztetések, kezelési utasítás a maradó kockázatokról.
5. **Szervezési intézkedések és egyéni védőeszközök:** eljárások, képzés, engedélyezési rendszer, védőfelszerelés.

A sorrend lényege, hogy minél lejjebb haladunk, annál inkább az emberi viselkedésen múlik a védelem, és annál kevésbé megbízható. Egy veszély, amelyet a tervezés megszüntetett, soha többé nem okozhat balesetet; egy figyelmeztető tábla hatása a kezelő figyelmén, képzettségén és pillanatnyi állapotán múlik.

> **KoboCella: a háromlépcsős módszer a gyakorlatban**
>
> A KoboCella tervezői először azt vizsgálják, csökkenthető-e magának a veszélynek a nagysága: kisebb tömegű robotkar, lekerekített élek, a sebesség korlátozása az ember közelében. Ez az inherensen biztonságos tervezés. A megmaradt veszélyek ellen műszaki védőintézkedések következnek: lézerszkenneres sebesség- és távolságfelügyelet, erő- és teljesítménykorlátozás, biztonsági reteszelés a cellaajtón. Kiegészítő védőintézkedés a vészleállító gomb. Végül a kezelési utasítás és a képzés írja le a maradó kockázatokat, például hogy hosszú haj vagy laza ruházat becsípődhet. A tervezők dokumentálják, miért nem lehetett a veszélyt egy magasabb lépcsőn kezelni. Ez az ALARP-igazolás gépbiztonsági megfelelője.

3.5. A bizonytalanság kezelése

Minden kockázatbecslés bizonytalan. A gyakoriságok ritka eseményekre vonatkoznak, amelyekről kevés adat van; a következménymodellek egyszerűsítenek; a szakértői becslések szubjektívek. A bizonytalanságnak két fajtáját érdemes megkülönböztetni:

- **aleatorikus** (véletlen) bizonytalanság: a jelenség természetes változékonysága (például hogy egy adott évben hányszor esik ki a hűtés). Ez több adattal sem szűnik meg;
- **episztemikus** (ismereti) bizonytalanság: a tudásunk hiányossága (például hogy mekkora valójában a szelepek meghibásodási rátája). Ez több adattal, jobb modellekkel csökkenthető.

A jó kockázatértékelés a bizonytalanságot nem rejti el, hanem kimutatja: érzékenységvizsgálattal (mely paraméterek befolyásolják leginkább az eredményt), tartománybecsléssel és valószínűségi (Monte Carlo) bizonytalanságelemzéssel. Ezekkel a módszerekkel a 10. és a 13. fejezet foglalkozik részletesen. Ha a bizonytalanság nagy és a lehetséges következmény súlyos, a döntéshozók gyakran az óvatosság elvét (*precautionary principle*) követik: a kockázat bizonyított kicsiségéig a kedvezőtlenebb feltevésekkel számolnak.

Gyakorlat

A gyakorlati útmutató 2. laborjában egy interaktív alkalmazással kockázati mátrixot tervezhetsz. Megadhatod a kategóriákat és a kritériumokat, elhelyezheted a forgatókönyveket, és megnézheted, mikor ad a mátrix eltérő rangsort, mint a kvantitatív kockázat (Cox rangsorhibája). A hozzá tartozó Colab-notebook egyéni kockázatot és F–N görbét számol egy kis létesítményre.

Összefoglalás

- A kockázatértékelés a kockázatelemzésből (veszélyazonosítás és kockázatbecslés) és a kockázatkiértékelésből áll; a folyamat iteratív.
- Az ALARP-elv szerint a tolerálható tartományban a kockázatot addig kell csökkenteni, amíg a további csökkentés költsége durva aránytalanságban nem állna a haszonnal. A rövidítés feloldása *As Low As Reasonably Practicable*.
- A HSE irányértékei az egyéni halálozási kockázatra: munkavállalóra 10^{-3}, lakosságra 10^{-4} (nem tolerálható határ), illetve 10^{-6} (általánosan elfogadható határ) évente.
- A társadalmi kockázatot F–N görbével ábrázoljuk; a kockázatkerülő kritériumok a sok áldozatot követelő eseményeket szigorúbban korlátozzák.
- A kockázati mátrix hasznos szűrő- és kommunikációs eszköz, de felbontása gyenge, rangsorhibákat okozhat, ezért kvantitatív döntésekre korlátozottan alkalmas.
- A tolerálható gyakoriságból és a többi védelmi réteg hatásából meghatározható a biztonsági funkcióra jutó szükséges kockázatcsökkentés (RRF) és a megengedett PFD.
- Az intézkedések hierarchiájában az inherensen biztonságos tervezés áll az első helyen.

Önellenőrzés

1. Mi a különbség a kockázatelemzés és a kockázatértékelés között?
2. Egy új technológia egyéni kockázata egy lakosra évi $5 \cdot 10^{-5}$. Melyik tartományba esik a HSE-modell szerint, és mi a teendő?
3. Miért vezethet rangsorhibához a kockázati mátrix, ha egy szervezet forgatókönyveinél a gyakoriság és a súlyosság negatívan korrelál?
4. Egy veszélyes esemény gyakorisága védelem nélkül évi $0{,}2$. A tolerálható gyakoriság évi 10^{-4}. Egy mechanikus biztonsági szelep $PFD = 10^{-2}$ valószínűséggel hibázik. Mekkora PFD-t kell teljesítenie a biztonsági funkciónak?
5. Sorold be az intézkedési hierarchiába: (a) a robot sebességének korlátozása 250 mm/s-ra a gyártó által, (b) figyelmeztető sárga csík a padlón, (c) fényfüggöny, amely belépéskor megállítja a robotot.

Megoldás.

1. A kockázatelemzés a veszélyek azonosítását és a kockázat becslését jelenti (mekkora a kockázat?). A kockázatértékelés ezt kiegészíti a kockázatkiértékeléssel (tolerálható-e a kockázat?).
2. A lakosságra vonatkozó 10^{-4} és 10^{-6} közötti tolerálható tartományba esik. A kockázat csak akkor vállalható, ha ALARP-igazolással kimutatható, hogy minden ésszerűen megvalósítható csökkentő intézkedést megtettek.
3. Mert a mátrix cellái a kockázatot durván közelítik. Ha a ritka, de súlyos események és a gyakori, de enyhe események kvantitatív kockázata közel azonos, a mátrix a cellahatárok elhelyezkedése miatt az egyiket „vörösbe”, a másikat „sárgába” sorolhatja. A sorrend akár meg is fordulhat a valós kockázatokhoz képest, és az erőforrások rossz helyre kerülnek.
4. $RRF = 0{,}2 \cdot 10^{-2}/10^{-4} = 20$, tehát $PFD_{\text{SIF}} \le 0{,}05$. Ez a SIL 1 tartományba esik ($10^{-2} \le PFD < 10^{-1}$).

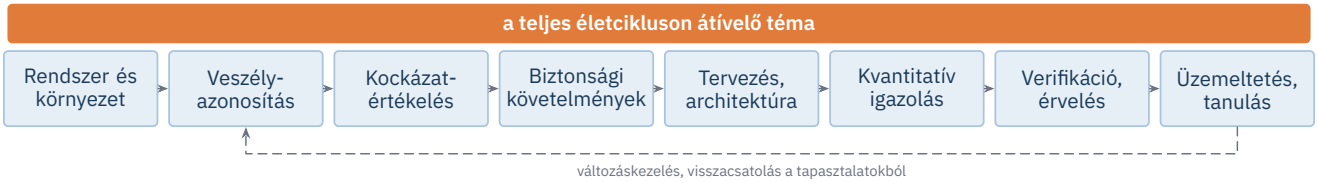
5. (a) inherensen biztonságos tervezés (a veszély nagyságát csökkenti), (c) műszaki védőintézkedés, (b) tájékoztatás. A sorrend egyben a hatékonyság sorrendje is.

Továbbolvasás

- Health and Safety Executive (2001): *Reducing Risks, Protecting People.* HSE Books – az ALARP-elv és a tolerálhatósági modell alapdokumentuma, ingyenesen elérhető (Health and Safety Executive 2001).
- Cox, L. A. (2008): What's wrong with risk matrices? *Risk Analysis* 28(2), 497–512 (Cox 2008).
- IEC 31010:2019 – a kockázatértékelési módszerek áttekintése alkalmazási útmutatóval (IEC 2019a).
- Kletz, T., Amyotte, P. (2010): *Process Plants: A Handbook for Inherently Safer Design.* CRC Press (Kletz és Amyotte 2010).

4.

A biztonsági életciklus és a szabványok rendszere



A biztonság nem egyetlen tervezési döntés eredménye, hanem egy teljes életcikluson át tartó, dokumentált tevékenységsorozaté, a koncepciótól a leszerelésig. Ez a fejezet azt mutatja be, hogyan írják le ezt az életciklust a funkcionális biztonsági szabványok. Először azt vizsgáljuk meg, miért van szükség életciklus-szemléletre. Ezután részletesen tárgyaljuk az IEC 61508 alapszabványt és a biztonsági integritási szint (SIL) fogalmát, majd áttekintjük az ágazati szabványokat (folyamatipar, közúti jármű, gépipar, vasút, repülés, orvostechnika) és integritási szintjeiket. A fejezet végén a funkcionális biztonság menedzsmentjéről és a jogszabályi háttérről lesz szó.

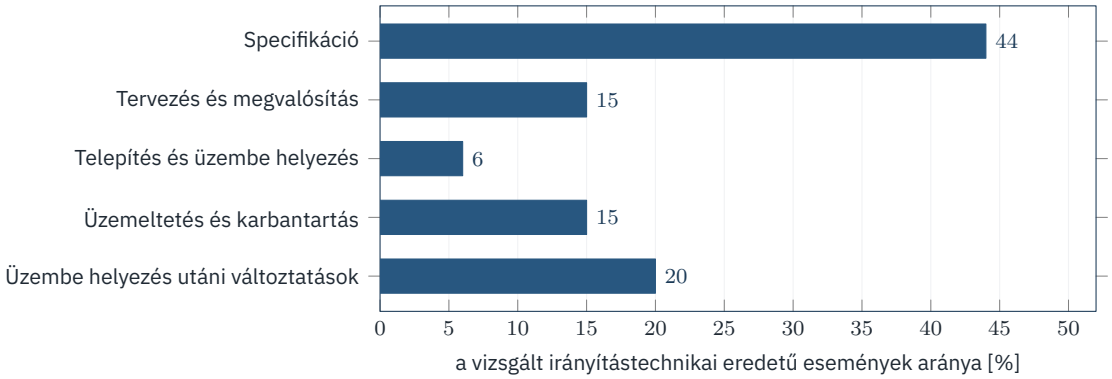
Tanulási eredmények

A fejezet végére képes leszel

- megindokolni az életciklus-szemlélet szükségességét a baleseti okok életciklus-fázisok szerinti megoszlásával;
- leírni az IEC 61508 felépítését és biztonsági életciklusának szakaszait;
- a SIL-t az alacsony és a nagy igénybevételű üzemmódban értelmezni, és ismertetni a SIL teljesítésének három pillérét;
- az ISO 26262 szerint ASIL-t meghatározni a súlyosság, a kitettség és az irányíthatóság alapján;
- az ISO 13849-1 kockázati gráfjával meghatározni a szükséges teljesítményszintet (PL_r);
- összevetni és helyesen értelmezni a különböző ágazati integritási szinteket (SIL, ASIL, PL, DAL).

4.1. Miért életciklus?

Az egyesült királysági munkavédelmi hatóság (HSE) az *Out of Control* című tanulmányában 34 olyan balesetet és súlyos eseményt elemzett, amelyekben irányítástechnikai rendszerek meghibásodása játszott szerepet (Health and Safety Executive 2003). Az eredmény a szakma számára meglepő volt (4.1. ábra): az okok közel fele (44%) a specifikációban keletkezett, további 15% a tervezésben és megvalósításban, és csak a kisebbik rész köthető az üzemeltetéshez. Feltűnő a „üzembe helyezés utáni változtatások” 20%-os aránya is.



4.1. ábra. A vizsgált irányítástechnikai eredetű balesetek és események okainak megoszlása az életciklus-fázisok szerint (HSE, *Out of Control* adatai alapján).

A tanulság kettős. Először: a balesetek okainak jelentős része már a tervezés során beépül a rendszerbe, ezért a biztonságot a fejlesztés legelején kell kezdeni. Másodszor: a rendszer biztonsága az üzemeltetés során, a karbantartásokkal és a módosításokkal romolhat, ezért a biztonsági tevékenységnek a leszerelésig tartania kell. Az életciklus-modell mindkét felismerést rendszerbe foglalja: előírja, hogy mely tevékenységeket kell elvégezni, milyen sorrendben, és milyen dokumentációval igazolni.

4.2. Az IEC 61508 alapszabvány

Az IEC 61508 *Villamos/elektronikus/programozható elektronikus biztonsági rendszerek funkcionális biztonsága* című szabványsorozat (IEC 2010a) a funkcionális biztonság alapszabványa (*basic safety publication*). Első kiadása 1998–2000-ben, második kiadása 2010-ben jelent meg. A harmadik kiadás 2026-ban a végső szavazási szakaszban van, a fő részek hivatalos megjelenése 2027-re várható. A szabvány két célt szolgál: közvetlenül alkalmazható olyan rendszerekre, amelyekre nincs ágazati szabvány, és keretet ad az ágazati szabványoknak (IEC 61511, IEC 62061, EN 50129, ISO 26262 stb.).

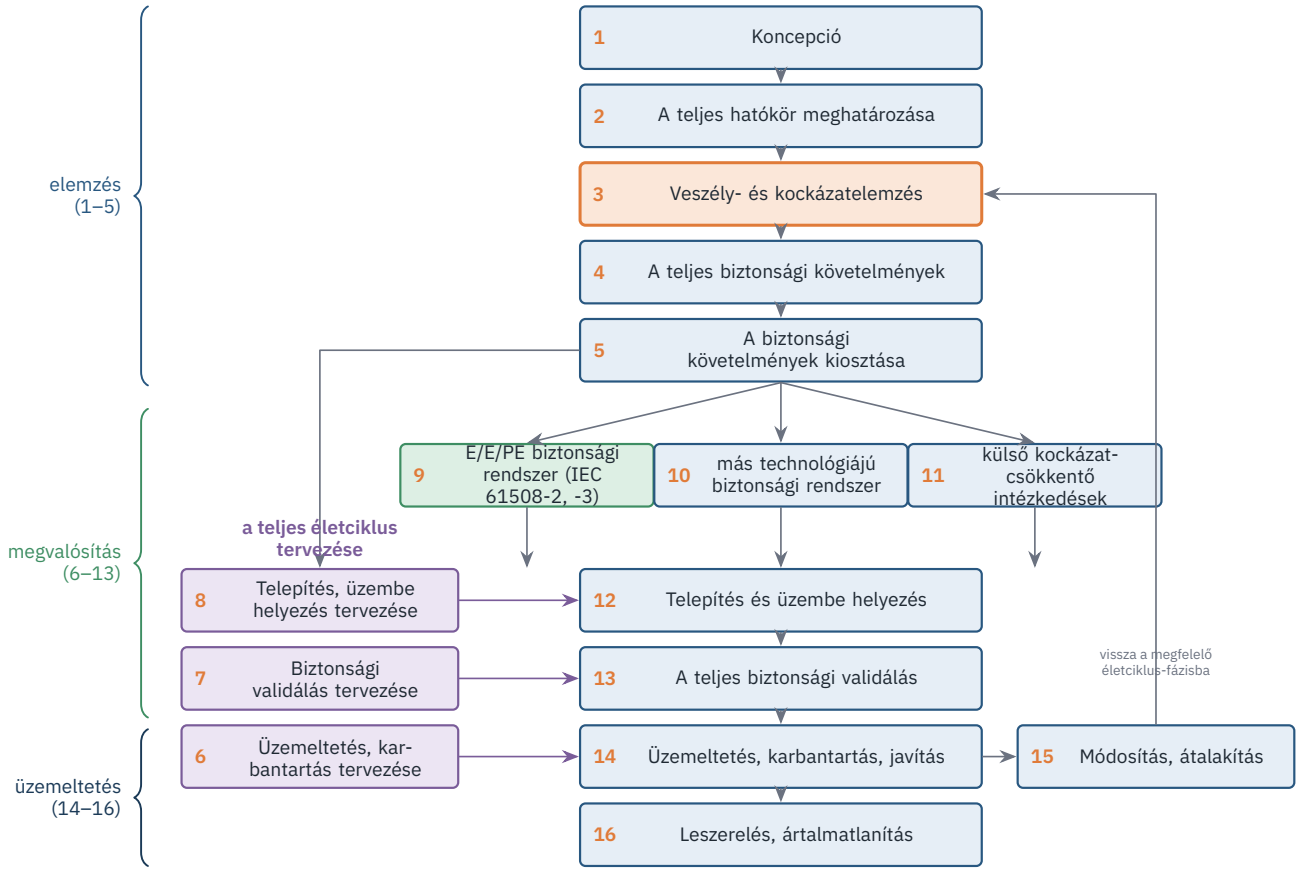
A szabvány hét részből áll:

4.1. táblázat. Az IEC 61508 részei

Rész	Tartalom	Jelleg
IEC 61508-1	Általános követelmények (életciklus, menedzsment, dokumentáció)	normatív
IEC 61508-2	Az E/E/PE biztonsági rendszerek hardverkövetelményei	normatív
IEC 61508-3	Szoftverkövetelmények	normatív
IEC 61508-4	Fogalommeghatározások és rövidítések	normatív
IEC 61508-5	Példák a SIL meghatározására	tájékoztató
IEC 61508-6	Útmutató a 2. és 3. rész alkalmazásához (többek között PFD- és PFH-képletek)	tájékoztató
IEC 61508-7	Technikák és intézkedések áttekintése	tájékoztató

4.2.1. A biztonsági életciklus

A szabvány központi eleme a 16 fázisú teljes biztonsági életciklus (4.2. ábra). Minden fázisnak meghatározott célja, bemenete, kimenete és ellenőrzési követelménye van.



4.2. ábra. Az IEC 61508 teljes biztonsági életciklusa egyszerűsített formában. A lila fázisok az életciklus későbbi szakaszainak tervezését jelentik, és párhuzamosan futnak a megvalósítással.

A fázisok három csoportba sorolhatók.

Elemzés (1–5. fázis). Az első két fázis az irányított berendezés és környezetének megértése, valamint az elemzés hatókörének rögzítése. A harmadik fázis, a veszély- és kockázatelemzés, a módszertan kulcsa: itt kell feltárni azokat a meghibásodási lehetőségeket, amelyek balesethez vezethetnek, a normál üzemben, az indításnál, a leállításnál, a karbantartásnál és az előrelátható rendellenes használatnál is. A negyedik fázis a teljes biztonsági követelményeket fogalmazza meg (milyen biztonsági funkciókra van szükség), az ötödik ezeket kiosztja az egyes védelmi rétegekre, és meghatározza a biztonsági funkciók szükséges integritási szintjét. A ReaktorVéd esetében ez az a pont, ahol a hűtés kiesésének veszélyéhez a „gyorszáró szelepek zárása” biztonsági funkció és a SIL 2 követelmény rendelődik (3.4. szakasz).

Megvalósítás (6–13. fázis). A 6–8. fázis az életciklus későbbi szakaszainak tervezését jelenti: hogyan fogják a rendszert üzemeltetni és karbantartani (milyen gyakran kell a proof tesztet elvégezni), hogyan validálják a biztonsági funkciókat, és hogyan telepítik és helyezik üzembe a rendszert. Ezek a tervek a megvalósítással párhuzamosan készülnek, mert a tervezési döntéseket befolyásolják. A 9. fázis az E/E/PE biztonsági rendszer megvalósítása; erre vonatkozik a szabvány 2. (hardver) és 3. (szoftver) része, saját, részletes életciklussal. A 10. fázis a más technológiájú (például mechanikus

vagy hidraulikus) biztonsági rendszereket, a 11. a külső kockázatcsökkentő intézkedéseket (például a gyorsasági korlátozást, a védőtöltést vagy a mentési tervet) fedi le. A 12–13. fázis a telepítés, az üzembe helyezés és a teljes biztonsági validálás.

Üzemeltetés (14–16. fázis). A 14. fázis az üzemeltetés, a karbantartás és a javítás; ide tartoznak a proof tesztek és a meghibásodások nyilvántartása. A 15. fázis a módosításokat kezeli: minden változtatás hatását elemezni kell, és vissza kell térni ahhoz az életciklus-fázishoz, amelyet a módosítás érint. A 16. fázis a leszerelés és az ártalmatlanítás, amelynek saját veszélyei vannak (maradék anyagok, akkumulátorok, nyomás alatti részek).

Az életciklus-séma legfontosabb szerepe az, hogy **útmutatóként és ellenőrző listaként** szolgáljon. Felhívja a figyelmet a kockázatkezeléshez kötődő feladatokra, és biztosítja, hogy egy elemzési vagy tervezési lépés csak indokolt és dokumentált esetben maradhasson el.

4.3. A biztonsági integritási szint (SIL)

A biztonsági integritás annak valószínűsége, hogy egy biztonsági rendszer a megkövetelt biztonsági funkciót meghatározott körülmények között, meghatározott időtartam alatt kielégítően végrehajtja. A biztonsági integritási szint (*safety integrity level*, SIL) ennek négy diszkrét fokozata, a SIL 1-től (legkisebb) a SIL 4-ig (legnagyobb).

4.3.1. Üzemmódok: alacsony és nagy igénybevétel

A SIL számszerű célértéke attól függ, milyen gyakran kell a biztonsági funkciónak működnie.

Alacsony igénybevételű és nagy igénybevételű vagy folyamatos üzemmód

- **Alacsony igénybevételű üzemmód** (*low demand mode*): a biztonsági funkciót csak igény esetén kell végrehajtani, és az igények gyakorisága nem nagyobb évi egynél. A mérőszám a **működési igénykori átlagos meghibásodási valószínűség** (*average probability of dangerous failure on demand*, $\mathrm{PFD}_{\mathrm{avg}}$).
- **Nagy igénybevételű üzemmód** (*high demand mode*): az igények gyakorisága évi egynél nagyobb. **Folyamatos üzemmód** (*continuous mode*): a biztonsági funkció a normál működés része. Mindkét esetben a mérőszám a **veszélyes meghibásodás átlagos gyakorisága** óránként (*average frequency of dangerous failure per hour*, PFH).

Az üzemmód régi meghatározása

Az IEC 61508 első kiadása az alacsony igénybevételű üzemmódot úgy definiálta, hogy az igénygyakoriság nem nagyobb évi egynél *és* nem nagyobb a proof tesztek gyakoriságának kétszeresénél. Ezt a megfogalmazást a 2014-es jegyzet is átvette. A 2010-es második kiadás a proof tesztre vonatkozó feltételt elhagyta; ma egyedül az évi egy igény a határ.

4.2. táblázat. A SIL célértékei az IEC 61508 szerint

SIL	$\mathrm{PFD}_{\mathrm{avg}}$ (alacsony igénybevétel)	RRF	PFH [h^{-1}] (nagy igénybevétel, folyamatos)
4	$10^{-5} \leq PFD < 10^{-4}$	$10\,000 < RRF \leq 100\,000$	$10^{-9} \leq PFH < 10^{-8}$
3	$10^{-4} \leq PFD < 10^{-3}$	$1\,000 < RRF \leq 10\,000$	$10^{-8} \leq PFH < 10^{-7}$

SIL	PFD_{avg} (alacsony igénybevétel)	RRF	PFH [h^{-1}] (nagy igénybevétel, folyamatos)
2	$10^{-3} \leq PFD < 10^{-2}$	$100 < RRF \leq 1\,000$	$10^{-7} \leq PFH < 10^{-6}$
1	$10^{-2} \leq PFD < 10^{-1}$	$10 < RRF \leq 100$	$10^{-6} \leq PFH < 10^{-5}$

A két mérőszám dimenziója különbözik: a PFD_{avg} valószínűség (dimenzió nélküli), a PFH gyakoriság (1/óra). Az alacsony igénybevételű funkció által hagyott maradó kockázat gyakorisága az igénygyakoriság és a PFD szorzata ($F = F_{\text{igény}} \cdot PFD_{avg}$), a nagy igénybevételű funkcióé közvetlenül a PFH. A közlekedésben és az energiatermelésben a súlyos következményű veszélyekre jellemzően SIL 3 vagy SIL 4 követelményt támasztanak.

4.3.2. A SIL teljesítésének három pillére

Egy biztonsági funkció akkor teljesíti az előírt SIL-t, ha három, egymástól független feltétel egyszerre teljesül:

1. **Kvantitatív célérték a véletlenszerű hardvermeghibásodásokra.** A funkció PFD_{avg}- vagy PFH-értéke a megfelelő tartományba esik. Ennek számítását a 13. fejezet tárgyalja.
2. **Architekturális korlátok.** A hardver hibatűrése (*hardware fault tolerance*, HFT: hány meghibásodást visel el a funkció elvesztése nélkül) és a biztonságos meghibásodási hányad (*safe failure fraction*, SFF) együtt korlátozza az elérhető legnagyobb SIL-t. A korlátok célja, hogy a SIL ne csak kedvező (gyakran optimista) adatokon nyugvó számításból következzen.
3. **Szisztematikus képesség** (*systematic capability*, SC 1–4). A szisztematikus hibák elkerülésére és kezelésére alkalmazott fejlesztési módszerek, technikák és ellenőrzések szigorúsága. A szabvány fejlesztési, verifikációs és dokumentációs követelményei SIL-szintenként szigorodnak.

A harmadik pillér különösen fontos, mert a szoftverre csak ez alkalmazható. A szisztematikus hibák nem számszerűsíthetők meghibásodási rátával, ezért ellenük a fejlesztési folyamat minőségével kell védekezni (15. fejezet).

4.3.3. A SIL meghatározásának módszerei

A szükséges SIL-t a kockázatértékelésből kell levezetni. Az IEC 61508-5 és az IEC 61511-3 több módszert mutat be:

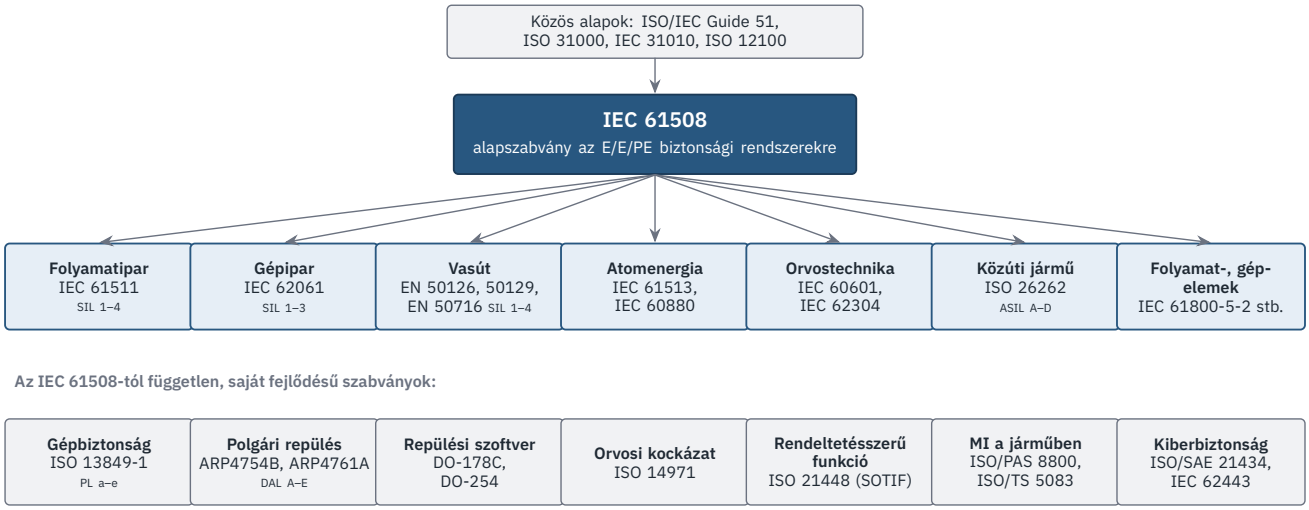
- **kvantitatív módszer:** a tolerálható gyakoriságból és a többi védelmi réteg hatásából számított szükséges PFD (3.5. egyenlet);
- **rétegzett védelem elemzése (LOPA):** félkvantitatív, nagyságrendi becsléseken alapuló módszer, a folyamatiparban a legelterjedtebb (11. fejezet);
- **kockázati gráf:** kvalitatív döntési fa a következmény súlyossága, a kitettség, az elkerülés lehetősége és az igény valószínűsége alapján. Gyors, de kalibrálni kell, és hajlamos a túl- vagy alulbecslésre;
- **kalibrált kockázati mátrix:** a 3.3. szakaszban tárgyalt mátrix, a cellákhoz SIL-t rendelve.

> **A SIL nem a rendszer, hanem a funkció tulajdonsága**
>
> Gyakori, de pontatlan szóhasználat: „SIL 2-es érzékelő”, „SIL 3-as PLC”. A SIL egy *biztonsági funkcióhoz* rendelt követelmény. Egy eszközről csak annyi mondható, hogy egy adott SIL-ig *alkalmas* (*SIL capable*)

egy funkcióban való felhasználásra, meghatározott feltételekkel (architektúra, proof teszt, felhasználási előírások). Egy SIL 3-ig tanúsított logikai egységből egycsatornás, rosszul karbantartott érzékelőkkel csak alacsonyabb integritású funkció építhető.

4.4. Ágazati szabványok

Az IEC 61508 általános keretét az egyes iparágak a saját sajátosságaikhoz igazították (4.3. ábra). Néhány fontos terület saját, az IEC 61508-tól független hagyománnyal rendelkezik, de fogalmaik egymásba fordíthatók.



4.3. ábra. A funkcionális biztonsági szabványok családja. A felső sor az IEC 61508-ból származtatott ágazati szabványokat, az alsó a független fejlődésű, de rokon szabványokat mutatja.

4.4.1. Folyamatipar: IEC 61511

Az IEC 61511 (IEC 2016b) a folyamatipari biztonsági műszerezett rendszerekre (SIS) vonatkozik. A szabvány a végfelhasználónak és a rendszerintegrátornak szól, nem az eszközgyártónak: az eszközöknek vagy az IEC 61508 szerint tanúsítottnak, vagy „használatban bizonyítottnak” (*prior use*) kell lenniük. A szabvány saját életciklust ír elő, és a hardver hibatűrésére egyszerűsített követelményt ad: SIL 1-hez és alacsony igénybevételű SIL 2-höz HFT = 0, nagy igénybevételű SIL 2-höz és SIL 3-hoz HFT = 1, SIL 4-hez HFT = 2. A SIL 4 a folyamatiparban rendkívül ritka; a szabvány azt javasolja, hogy ilyen igény esetén inkább a folyamatot kell újratervezni.

ReaktorVéd: az IEC 61511 életciklusa

A ReaktorVéd biztonsági funkciója az IEC 61511 hatálya alá tartozik. A projekt dokumentumai: a HAZOP-jelentés (5. fejezet), a LOPA-jelentés a SIL-kiosztással (11. fejezet), a biztonsági követelményspecifikáció (*safety requirements specification*, SRS), a SIL-igazoló számítás (13. fejezet), a validálási terv, valamint az üzemeltetési és karbantartási terv a proof tesztek gyakoriságával és módjával. A specifikáció a funkció minden lényeges jellemzőjét rögzíti: a kioldási határértéket (95 °C), a válaszidőt (10 s), a biztonságos állapotot (mindkét gyorszáró szelep zárva), a SIL-t (2), a proof teszt gyakoriságát (évente), és a téves leállítások megengedett gyakoriságát.

4.4.2. Közúti járművek: ISO 26262

Az ISO 26262 (ISO 2018a) a sorozatgyártású közúti járművek E/E rendszereinek funkcionális biztonsági szabványa. Első kiadása 2011-ben, második, a teherautókra, buszokra és motorkerékpárokra is kiterjesztett kiadása 2018-ban jelent meg. A harmadik kiadás (2026-ban DIS, azaz nemzetközi szabványtervezet szakaszban) a gépi tanulás, az agilis fejlesztés és a hibatűrő (*fail-operational*) architektúrák kérdéseivel bővül. Az IEC 61508-hoz képest a legfontosabb különbségek:

- **Tömeggyártás:** a biztonságot a fejlesztés során kell igazolni, a példányonkénti validálás nem lehetséges. Az IEC 61508 inkább egyedi vagy kis sorozatú rendszerekre (például egy vegyi üzem védelmi rendszerére) gondol, ahol a telepített rendszert validálják.
- **Biztonságosra tervezett termék:** a járműben a szabályozó és a biztonsági funkciók gyakran nem választhatók szét (a fékrendszer egyszerre végzi a normál fékezést és a biztonsági funkciót), szemben az IEC 61508 „irányított berendezés + független védelmi rendszer” modelljével.
- **A közlekedési helyzet egésze:** a kockázatértékelés a vezetőt, a többi közlekedőt és a helyzet irányíthatóságát is figyelembe veszi.
- **Biztonsági kultúra:** a szabvány kiemelten foglalkozik a biztonsági menedzsmenttel és -kultúrával, mert a nagy komplexitású rendszerekben a balesetek inkább a szervezet működéséből fakadnak.

Az ASIL meghatározása. A veszélyelemzés és kockázatbecslés (*hazard analysis and risk assessment*, HARA) során minden veszélyes eseményt (egy meghibásodás és egy működési helyzet kombinációját) három paraméterrel jellemeznek (4.3. táblázat): a sérülés **súlyossága** (S0–S3), a működési helyzetnek való **kitettség** valószínűsége (E0–E4) és a vezető vagy más érintettek számára a helyzet **irányíthatósága** (C0–C3).

4.3. táblázat. Az ISO 26262 HARA-paraméterei (rövidített megfogalmazás)

Paraméter	0	1	2	3	4
S – súlyosság	nincs sérülés	könnyű, közepes sérülés	súlyos, életveszélyes (túlélés valószínű)	életveszélyes (túlélés bizonytalan), halálos	–
E – kitettség	elképzelhetetlen	nagyon kis valószínűség	kis valószínűség	közepes valószínűség	nagy valószínűség
C – irányíthatóság	általában irányítható	egyszerűen irányítható	általában irányítható	nehezen vagy nem irányítható	–

Az autóipari biztonsági integritási szint (*automotive safety integrity level*, ASIL) a három paraméterből egy táblázat alapján adódik (4.4. táblázat). Ha bármelyik paraméter 0 értékű, nincs ASIL-besorolás; az eredmény QM (*quality management*) is lehet, ami azt jelenti, hogy elég a szokásos minőségirányítási folyamat.

4.4. táblázat. Az ASIL meghatározása az ISO 26262-3 szerint

		C1	C2	C3
S1	E1	QM	QM	QM
	E2	QM	QM	QM
	E3	QM	QM	A
	E4	QM	A	B
S2	E1	QM	QM	QM

		C1	C2	C3
	E2	QM	QM	A
	E3	QM	A	B
	E4	A	B	C
S3	E1	QM	QM	A
	E2	QM	A	B
	E3	A	B	C
	E4	B	C	D

A táblázatnak egyszerű szerkezete van: ha a paraméterek indexét összeadjuk ($s + e + c$), akkor 10 esetén ASIL D, 9 esetén C, 8 esetén B, 7 esetén A, ennél kisebb összegnél QM adódik. Ez az összefüggés jól megjegyezhető, de a szabvány a táblázatot tekinti mérvadónak.

A HARA eredménye a **biztonsági célok** (*safety goals*) listája. Minden biztonsági cél egy veszélyes esemény megelőzésére vagy hatásának csökkentésére vonatkozó legfelső szintű követelmény, a hozzá tartozó ASIL-lel. A biztonsági célokból vezetik le a funkcionális, majd a technikai biztonsági koncepciót.

Az ISO 26262 lehetővé teszi az **ASIL-dekompozíciót**: egy magas ASIL-ű követelmény két, egymástól kellően független elemre bontható, alacsonyabb ASIL-lel (például ASIL D = ASIL C(D) + ASIL A(D), vagy ASIL B(D) + ASIL B(D)). A zárójeles betű jelzi az eredeti ASIL-t, amelyhez a függetlenség igazolása kötődik. A dekompozíció csak a szisztematikus képességre vonatkozó követelményeket enyhíti; a hardvermetrikákat továbbra is az eredeti ASIL szerint kell teljesíteni.

Hardvermetrikák. A véletlenszerű hardvermeghibásodásokra az ISO 26262-5 három mérőszámot ír elő (4.5. táblázat): az egypontos hibák metrikáját (*single-point fault metric*, SPFM), a latens hibák metrikáját (*latent fault metric*, LFM) és a véletlenszerű hardvermeghibásodásokból eredő biztonsági cél megsértésének valószínűségi metrikáját (*probabilistic metric for random hardware failures*, PMHF).

4.5. táblázat. Az ISO 26262-5 hardvermetrikáinak célértékei

	ASIL B	ASIL C	ASIL D
SPFM	$\geq 90\%$	$\geq 97\%$	$\geq 99\%$
LFM	$\geq 60\%$	$\geq 80\%$	$\geq 90\%$
PMHF [h^{-1}]	$< 10^{-7}$	$< 10^{-7}$	$< 10^{-8}$

Az ASIL nem valószínűségi kategória

A 2014-es jegyzet egy ábrája minden ASIL-szinthez óránkénti meghibásodási valószínűséget rendelt (ASIL A-hoz is), és egyértelmű SIL–ASIL-megfeleltetést sugallt. Mindkettő félrevezető. Az ASIL a HARA kvalitatív paramétereiből adódik, és elsősorban a fejlesztési folyamat szigorúságát határozza meg. Valószínűségi célérték (PMHF) csak az ASIL B, C és D szinthez tartozik, és az is csak a véletlenszerű hardvermeghibásodásokra. A SIL és az ASIL közötti megfeleltetést egyik szabvány sem definiálja; a szakirodalomban közölt összevetések tájékoztató jellegűek.

VészFék: egy HARA-sor

A VészFék egyik veszélyes eseménye: *indokolatlan teljes erejű fékezés autópályán, nagy sebességnél, követő jármű jelenlétében.* A csapat besorolása és indoklása:

- **S3:** a hátulról ráfutó jármű nagy sebességkülönbségnél életveszélyes sérülést okozhat;
- **E4:** az autópályás haladás követő járművel a jármű üzemidejének jelentős része;
- **C3:** a követő jármű vezetője számára a hirtelen, váratlan teljes fékezés nehezen kezelhető, a VészFék vezetője pedig nem tudja „visszavenni” a fékezést.

Az eredmény ASIL D. A biztonsági cél: „A VészFék nem kezdeményezhet indokolatlan, X m/s²-nél nagyobb lassulású fékezést $v > 60$ km/h sebességnél.” A konkrét határértékeket járműteszteken alapuló irányíthatósági vizsgálatok adják. Egy másik veszélyes esemény, a *kismértékű, rövid indokolatlan fékezés városi forgalomban*, jóval alacsonyabb besorolást kap (például S1, E4, C1 → QM), ami jól mutatja, hogy az ASIL a veszélyes eseményhez, nem a funkcióhoz kötődik.

4.4.3. Gépbiztonság: ISO 13849-1 és IEC 62061

A gépek biztonsági irányítórendszereire két szabvány alkalmazható. Az **IEC 62061** (IEC 2021) az IEC 61508 gépipari változata, SIL 1–3 szintekkel. Az **ISO 13849-1** (ISO 2023) saját hagyományú, és a teljesítményszint (*performance level*, PL) fogalmát használja, öt fokozattal (a–e). A PL a nagy igénybevételű üzemmód PFH-tartományaival definiált (4.6. táblázat), és öt tényező együtteséből adódik: a **kategória** (B, 1, 2, 3, 4, amely az architektúrát és a hibákkal szembeni viselkedést írja le), a csatornánkénti veszélyes meghibásodásig eltelt átlagos idő ($\mathrm{MTTF_D}$), az átlagos diagnosztikai lefedettség ($\mathrm{DC_{avg}}$), a közös okú meghibásodások elleni intézkedések és a szisztematikus képesség.

4.6. táblázat. Az ISO 13849-1 teljesítményszintjei

PL	PFH [$\mathrm{h^{-1}}$]	megközelítőleg megfelelő SIL (IEC 62061)
a	$10^{-5} \le PFH < 10^{-4}$	–
b	$3 \cdot 10^{-6} \le PFH < 10^{-5}$	1
c	$10^{-6} \le PFH < 3 \cdot 10^{-6}$	1
d	$10^{-7} \le PFH < 10^{-6}$	2
e	$10^{-8} \le PFH < 10^{-7}$	3

A szükséges teljesítményszintet ($\mathrm{PL_r}$) az ISO 13849-1 kockázati gráfja adja (4.4. ábra). A gráf három paramétere: a sérülés **súlyossága** (S1: könnyű, S2: súlyos), a veszélynek való **kitettség gyakorisága és időtartama** (F1: ritka, F2: gyakori), valamint az **elkerülés lehetősége**, illetve a veszélyes esemény bekövetkezésének valószínűsége (P1: lehetséges, P2: alig lehetséges).

KoboCella: a sebesség- és távolságfelügyelet PL~r~-je

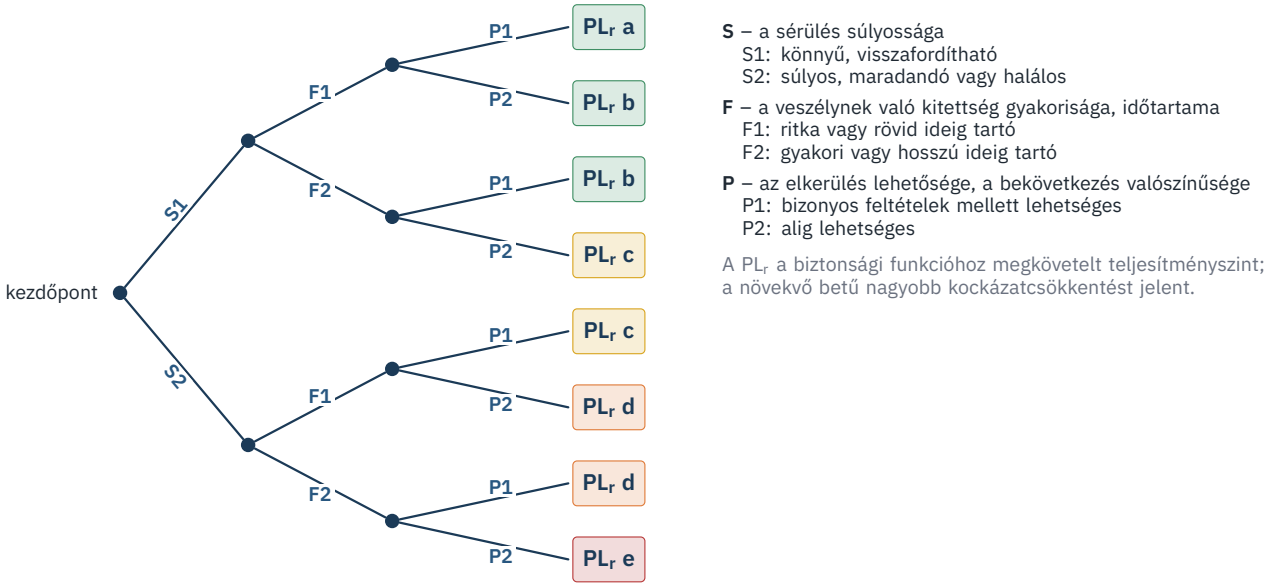
A KoboCella egyik biztonsági funkciója: „Ha a lézerszkenner a robot 1,2 m-es körzetében személyt érzékel, a robot sebessége legfeljebb 250 mm/s lehet; 0,5 m-en belül a robot biztonságosan megáll.” A funkció meghibásodása esetén a robot teljes sebességgel ütközhet a szerelővel.

- **S2:** a teljes sebességű ütközés fejmagasságban súlyos sérülést okozhat;
- **F2:** a szerelő a műszak nagy részében a robot munkaterében dolgozik;
- **P2:** a robot gyors mozgása miatt a szerelő nem tud kitérni.

Az eredmény $\mathbf{PL_r}$ = **e**. Ez nagyon szigorú követelmény (3-as vagy 4-es kategóriájú architektúra, nagy diagnosztikai lefedettség). A tervezők ezért megvizsgálják, hogy a robot maximális sebességének és tömegének csökkentésével (inherensen biztonságos tervezés) elérhető-e, hogy a teljes sebességű ütközés se okozzon súlyos sérülést (S1). Ebben az esetben a követelmény $\mathrm{PL_r}$ = c-re csökkenne. A 7. fejezetben ugyanezt a funkciót rendszerelméleti szemlélettel is megvizsgáljuk.

4.4.4. Vasút, repülés, orvostechnika

Vasút. A vasúti alkalmazásokra a CENELEC-szabványok vonatkoznak: az EN 50126 (CENELEC 2017) a megbízhatóság, rendelkezésre állás, karbantarthatóság és biztonság (RAMS) általános folyamatát,



4.4. ábra. Az ISO 13849-1 kockázati gráfja a szükséges teljesítményszint meghatározására.

az EN 50129 (CENELEC 2018) a biztonsági elektronikus jelzőrendszereket, az EN 50716 (CENELEC 2023) pedig 2023 óta a vasúti szoftverfejlesztést szabályozza; ez utóbbi az EN 50128 és az EN 50657 helyébe lépett. A vasút négy SIL-szintet használ, amelyekhez a tolerálható veszélyességi ráta (*tolerable hazard rate*, THR) tartományai tartoznak. A RAMS-szemlélet erőssége, hogy a biztonság mellett a rendelkezésre állást és a karbantarthatóságot is egységes keretben kezeli.

Polgári repülés. A repülésben az ARP4754B (a repülőgépek és rendszereik fejlesztése) és az ARP4761A (a biztonságértékelési folyamat) a meghatározó, mindkettő 2023 decemberi kiadású (SAE International 2023a, 2023b). A biztonságértékelés a funkcionális veszélyelemzéssel (*functional hazard assessment*, FHA) kezdődik, amely a meghibásodási állapotokat következményük szerint sorolja be (katasztrofális, veszélyes, jelentős, csekély, nincs hatás). Ezekhez a megengedett valószínűségek tartoznak repült óránként: katasztrofális állapotra 10^{-9}, veszélyesre 10^{-7}, jelentősre 10^{-5}. Ezt követi az előzetes és a végleges rendszerbiztonsági értékelés (PSSA, SSA). A fejlesztési biztosítottsági szint (*development assurance level*, DAL A–E) a szoftverre (DO-178C (RTCA 2011a)) és az elektronikus hardverre (DO-254) vonatkozó fejlesztési szigort határozza meg, hasonlóan a SIL szisztematikus pilléréhez.

Orvostechnika. Az orvostechnikai eszközök kockázatkezelésének szabványa az ISO 14971 (ISO 2019), a szoftver életciklus-folyamatoké az IEC 62304 (IEC 2006c), amely a szoftvert a lehetséges kár súlyossága szerint A, B és C biztonsági osztályba sorolja.

4.4.5. Az integritási szintek összevetése

A különböző ágazatok integritási szintjei hasonló logikát követnek, de nem egyenértékűek (4.7. táblázat). A meghatározásuk módja (kvantitatív célérték, kvalitatív kockázati paraméterek, következményosztály), az általuk szabályozott tevékenységek köre és a mögöttük álló feltevések eltérnek.

4.7. táblázat. Az ágazati integritási szintek összevetése. A szintek nem cserélhetők fel egymással.

Ágazat	Szabvány	Integritási szint	Meghatározás alapja	Valószínűségi célérték
általános, folyamatipar	IEC 61508, 61511	SIL 1–4	szükséges kockázatcsökkentés	PFD_{avg} vagy PFH tartomány
közúti jármű	ISO 26262	ASIL A–D	S, E, C (HARA)	PMHF csak B–D-re
gépipar	ISO 13849-1	PL a–e	S, F, P kockázati gráf	PFH tartomány
vasút	EN 50126, 50129	SIL 1–4	THR	THR tartomány
repülés	ARP4754B, DO-178C	DAL A–E	FHA következményosztály	a meghibásodási állapotra, repült óránként
orvostechnika	IEC 62304	A, B, C osztály	a lehetséges kár súlyossága	nincs

Egy komponens tanúsítása az egyik ágazat szintjére nem jelenti automatikusan a megfelelést egy másik ágazatban. Az összevetés hasznos a gondolkodásmódok közötti átjáráshoz és a beszállítói lánc kommunikációjához, de a megfelelést mindig az adott ágazat szabványa szerint kell igazolni.

4.5. A funkcionális biztonság menedzsmentje

A szabványok nem csak a műszaki megoldásra, hanem a szervezetre és a folyamatokra is követelményeket támasztanak. A **funkcionális biztonság menedzsmentje** (*functional safety management*, FSM) magában foglalja:

- a **felelősségek** és a **kompetenciák** rögzítését: ki felel az egyes életciklus-fázisokért, és milyen képzettséggel, tapasztalattal rendelkezik;
- a **biztonsági tervet**, amely leírja az elvégzendő tevékenységeket, a módszereket, a dokumentációt és az ütemezést;
- a **megerősítő intézkedéseket** (*confirmation measures*): a verifikációt, a funkcionális biztonsági felülvizsgálatot (*audit*) és a funkcionális biztonsági értékelést (*assessment*), amelyeket az integritási szinttel növekvő függetlenségű személynek, részlegnek vagy szervezetnek kell végeznie;
- a **konfigurációkezelést** és a **változáskezelést** (*management of change*, MoC), amely biztosítja, hogy minden módosítás hatását elemezzék;
- a **tapasztalatok gyűjtését**: a meghibásodások, a téves leállítások és a majdnem balesetek nyilvántartását és elemzését.

A biztonsági kultúra nem mérhető közvetlenül, de a menedzsment-követelmények sokat elárulnak róla. A 4.8. táblázat az ISO 26262 nyomán összefoglalja a jó és a nem elfogadható biztonsági kultúra jellemzőit.

4.8. táblázat. A biztonsági kultúra jellemzői az ISO 26262 nyomán

Nem elfogadható biztonsági kultúra	Megfelelő biztonsági kultúra
A felelősség nem követhető nyomon.	A folyamat biztosítja a biztonsági döntések nyomon követhetőségét.
A költség és a határidő elsőbbséget élvez a biztonsággal és a minőséggel szemben.	A funkcionális biztonság a legmagasabb prioritású szempont.

Nem elfogadható biztonsági kultúra	Megfelelő biztonsági kultúra
Reaktív hozzáállás: széles körű tesztelés a fejlesztés végén, a vezetés csak probléma esetén avatkozik be. Nincsenek rendszeres fejlesztési ciklusok. A biztonsági aggályt felvető munkatársat problémásnak tekintik.	Megelőző hozzáállás: a biztonsági és minőségi kérdéseket a lehető legkorábban felismerik és megoldják. A folyamatos fejlesztés minden folyamat része. A biztonsági aggályok felvetését ösztönzik és jutalmazzák.

4.6. A jogszabályi háttér

A szabványok önkéntesek, de a jogszabályok gyakran rájuk hivatkoznak: az uniós termékszabályozásban a harmonizált szabványok alkalmazása a megfelelőség vélelmét adja. A biztonságkritikus rendszerek szempontjából a legfontosabb uniós jogszabályok 2026 végén:

- **Gépekről szóló rendelet**, (EU) 2023/1230 (Európai Parlament és Tanács 2023): a gépi irányelvet váltja fel, 2027. január 20-tól alkalmazandó. Kiemelten foglalkozik az önfejlődő (gépi tanulást alkalmazó) biztonsági funkciókkal és a kiberbiztonsági szempontokkal.
- **Mesterséges intelligenciáról szóló rendelet**, (EU) 2024/1689 (Európai Parlament és Tanács 2024a): a nagy kockázatú MI-rendszerekre vonatkozó kötelezettségeket a 2026-ban elfogadott módosítás (*digital omnibus*) elhalasztotta. Az önálló, III. melléklet szerinti nagy kockázatú rendszerekre 2027. december 2-től, a szabályozott termékekbe (például gépekbe, járművekbe, orvostechnikai eszközökbe) beépített MI-rendszerekre 2028. augusztus 2-től alkalmazandók.
- **Kiberrezilienciáról szóló rendelet**, (EU) 2024/2847 (Európai Parlament és Tanács 2024b): a digitális elemeket tartalmazó termékek kiberbiztonsági követelményei; fő kötelezettségei 2027 végétől alkalmazandók.
- **Seveso III. irányelv**, 2012/18/EU (Európai Parlament és Tanács 2012): a veszélyes anyagokkal kapcsolatos súlyos balesetek megelőzése, a folyamatipari létesítmények biztonsági jelentésének és kockázatértékelésének kerete.
- A közúti járműveknél az ENSZ EGB előírásai (például a fékrendszerekre vonatkozó 13. számú, a kiberbiztonsági 155. számú (UNECE 2021a) és az automatizált sávtartó rendszerekről szóló 157. számú (UNECE 2021b)) a típusjóváhagyás feltételei.

Összefoglalás

- A baleseti okok jelentős része a specifikációban és a tervezésben keletkezik, egy másik része az üzemeltetés és a módosítások során; ezért a biztonsági tevékenységnek a teljes életciklust le kell fednie.
- Az IEC 61508 16 fázisú biztonsági életciklusa elemzés, megvalósítás és üzemeltetés szakaszokra tagolódik, és ellenőrző listaként is szolgál.
- A SIL egy biztonsági funkció követelménye: alacsony igénybevételnél PFD_{avg}-, nagy igénybevételnél PFH-tartomány. Teljesítéséhez a kvantitatív célérték, az architekturális korlátok és a szisztematikus képesség együtt szükséges.
- Az ISO 26262-ben az ASIL a súlyosság, a kitettség és az irányíthatóság alapján adódik; nem valószínűségi kategória, bár B–D szinten PMHF-célérték tartozik hozzá.
- Az ISO 13849-1 PL-szintjei PFH-tartományokkal definiáltak, a szükséges szintet kockázati gráf adja.
- Az ágazati integritási szintek hasonló logikájúak, de nem egyenértékűek.
- A funkcionális biztonság menedzsmentje (felelősség, kompetencia, független értékelés, változáskezelés) a műszaki megoldással egyenrangú követelmény.

Önellenőrzés

1. Egy biztonsági funkció igénygyakorisága évi 3. Melyik üzemmódba tartozik, és milyen mérőszámmal kell jellemezni az integritását?
2. Mi a SIL teljesítésének három pillére? Melyik alkalmazható a szoftverre, és miért csak az?
3. Határozd meg az ASIL-t: S2, E3, C3! És S3, E2, C2 esetén?
4. A KoboCella egy másik funkciójánál S1, F2, P1 adódik. Mekkora a $\mathrm{PL_r}$?
5. Miért nem lehet egy ISO 26262 szerinti ASIL D-s komponenst automatikusan SIL 3-as funkcióba beépíteni egy folyamatipari védelmi rendszerben?

Megoldás.

1. Nagy igénybevételű üzemmódba (évi egynél több igény), mérőszáma a PFH (veszélyes meghibásodás átlagos gyakorisága óránként).
2. A kvantitatív célérték a véletlenszerű hardvermeghibásodásokra, az architekturális korlátok (HFT, SFF) és a szisztematikus képesség. A szoftverre csak a szisztematikus képesség alkalmazható, mert a szoftverhibák szisztematikusak: nincs véletlenszerű meghibásodási rátájuk, ezért sem PFD-vel, sem hardvermetrikákkal nem jellemezhetők.
3. S2, E3, C3: a táblázat szerint ASIL B (az indexösszeg $2 + 3 + 3 = 8$). S3, E2, C2: ASIL A ($3 + 2 + 2 = 7$).
4. A gráfban S1 → F2 → P1 ágon $\mathrm{PL_r}$ = b.
5. Mert a két szabvány más feltevésekre épül (tömeggyártás, a járműre jellemző környezet és üzemmód, eltérő hardvermetrikák és architektúra-követelmények), és nincs közöttük hivatalos megfeleltetés. A komponens felhasználhatóságát az IEC 61511/61508 szerint kell igazolni (például a szükséges meghibásodási adatokkal, a proof teszt módjával és a szisztematikus képesség igazolásával).

Továbbolvasás

- Smith, D. J., Simpson, K. G. L. (2020): *The Safety Critical Systems Handbook.* 5. kiadás, Butterworth-Heinemann – az IEC 61508 és az ágazati szabványok gyakorlati értelmezése (Smith és Simpson 2020).
- Health and Safety Executive (2003): *Out of Control.* HSG238 – az irányítástechnikai eredetű balesetek okainak elemzése (Health and Safety Executive 2003).
- Rausand, M. (2014): *Reliability of Safety-Critical Systems.* Wiley, 1–3. fejezet (Rausand 2014).

II. rész – Veszélyazonosítás



Pannon Egyetem
University of Pannonia

5.

Veszélyazonosítás: a veszélylistáktól a HAZOP-ig



A kockázatértékelés csak annyira lehet jó, amennyire a veszélyazonosítás teljes. Egy fel nem tárt veszély semmilyen kockázati számításban nem jelenik meg, és ellene nem épül védelem. A veszélyazonosítás ezért a biztonsági életciklus legkritikusabb, egyben legkevésbé „algoritmizálható" lépése: módszeresség, csapatmunka és szakmai tapasztalat kell hozzá. Ez a fejezet a veszélyazonosítás általános elveivel kezdődik, majd bemutatja az előzetes módszereket (veszélylisták, előzetes veszélyelemzés, „Mi van, ha?" elemzés), részletesen tárgyalja a folyamatipar alapmódszerét, a HAZOP-ot, végül a funkcióalapú módszereket (FHA, HARA). Az FMEA és az STPA külön fejezetet kap.

Tanulási eredmények

A fejezet végére képes leszel

- megindokolni, miért nem bizonyítható a veszélyazonosítás teljessége, és mivel növelhető;
- előzetes veszélyelemzést készíteni ellenőrzőlisták és energiaforrás-szemlélet alapján;
- megtervezni és levezetni egy HAZOP-ülést: csomópontok, tervezési szándék, vezérszavak, eltérések, okok, következmények, védelmek, javaslatok;
- funkcionális veszélyelemzést (FHA) és ISO 26262 szerinti HARA-t végezni egy egyszerű járműfunkcióra;
- az életciklus fázisához és a rendszer jellegéhez illő módszert választani.

5.1. A veszélyazonosítás elvei

A veszélyazonosítás célja a rendszer összes ésszerűen előrelátható veszélyének, veszélyes helyzetének és veszélyes eseményének feltárása, a normál üzemben, az indításnál és leállításnál, a karbantartásnál, a meghibásodások után, valamint az előrelátható helytelen használat esetén is. Néhány alapelv:

- **A teljesség nem bizonyítható, csak valószínűsíthető.** Nincs olyan módszer, amely garantálná, hogy minden veszélyt megtaláltunk. A teljesség valószínűségét több, eltérő nézőpontú módszer együttes alkalmazása növeli: egy alkatrész-orientált (FMEA), egy folyamat-orientált (HAZOP) és egy funkció- vagy irányítás-orientált (FHA, STPA) módszer más-más veszélyeket talál meg.

- **Csapatmunka.** A veszélyazonosítás multidiszciplináris tevékenység. A tervező ismeri a szándékot, az üzemeltető a valóságos működést, a karbantartó a tipikus meghibásodásokat, a biztonsági szakember a módszert. Az egyedül dolgozó elemző a saját vakfoltjait reprodukálja.
- **Strukturált kreativitás.** A jó módszer egyszerre ad keretet (minden csomópontot, minden paramétert megvizsgálunk) és ösztönöz a „mi lenne, ha” típusú gondolkodásra.
- **Dokumentáltság.** A veszélyazonosítás eredménye a **veszélynapló** (*hazard log*): a feltárt veszélyek, okaik, következményeik, a védelmek és a javaslatok élő nyilvántartása, amelyet a teljes életcikluson át vezetni kell.
- **Időzítés.** A veszélyazonosítást a lehető legkorábban el kell kezdeni, amikor a tervet még olcsón lehet változtatni, és a tervezés előrehaladtával egyre részletesebb módszerekkel meg kell ismételni.

5.2. Előzetes módszerek

5.2.1. Veszélylisták és energiaforrás-szemlélet

A legegyszerűbb, mégis nélkülözhetetlen eszköz az **ellenőrzőlista**: a korábbi tapasztalatokat összegző lista a lehetséges veszélyfajtákról. Az ISO 12100 B melléklete például a gépek jellemző veszélyeit sorolja fel (mechanikai, villamos, termikus, zaj, rezgés, sugárzás, anyagok, ergonómia, környezet) (ISO 2010). A Haddon-féle energia–gát szemlélet (1.4.2. szakasz) szerint minden veszély mögött valamilyen energiaforrás áll. Ezért jó kiindulópont a rendszerben tárolt vagy átalakított energiák számbavétele: helyzeti, mozgási, nyomás-, hő-, vegyi, villamos és sugárzási energia.

Az ellenőrzőlista ereje a gyorsaságban és abban rejlik, hogy a korábbi tapasztalatok nem vesznek el. Gyengesége, hogy csak azt találja meg, ami a listán szerepel: új technológiánál, szokatlan kölcsönhatásoknál vak.

5.2.2. Előzetes veszélyelemzés (PHA)

Az előzetes veszélyelemzés (*preliminary hazard analysis*, PHA) a koncepciós fázis módszere, amikor a tervnek még csak a fő elemei ismertek. A MIL-STD-882E (U.S. Department of Defense 2012) rendszerbiztonsági szabvány a veszélylista (*preliminary hazard list*, PHL) összeállításával kezdi, majd minden veszélyhez meghatározza a lehetséges okokat, a következményeket, egy előzetes kockázati besorolást (általában kockázati mátrixszal), és a szükséges intézkedéseket vagy további elemzéseket. A PHA eredménye mutatja meg, hol kell később részletes elemzést (HAZOP, FMEA, hibafa) végezni.

KoboCella: előzetes veszélyelemzés (részlet)

5.1. táblázat. A KoboCella előzetes veszélyelemzésének részlete

Veszély	Veszélyes helyzet	Lehetséges következmény	Előzetes kockázat	Intézkedés, további elemzés
a robotkar mozgási energiája	a szerelő a közös munkatérben	ütközés, fejsérülés	nem tolerálható	sebesség- és távolságfelügyelet (PL_r meghatározás), STPA

szorítóerő a megfogónál	kéz a megfogó közelében	ujj becsípődése	nem tolerálható	erő- és teljesítménykorlátozás, ISO/TS 15066 határértékek
leeső munkadarab	a robot a szerelő fölött mozgat	ütés, zúzódás	tolerálható ALARP mellett	pályatervezési korlátozás, megfogófigyelés
sűrített levegő	tömlőszakadás	szem- és bőrsérülés	tolerálható ALARP mellett	tömlőrögzítés, nyomáscsökkentés
szoftverfrissítés	a frissítés után más a robot viselkedése	váratlan mozgás	nem tolerálható	változáskezelés, validálás frissítés után
ismétlődő mozdulatok	a szerelő a robot ütemét követi	mozgásszervi panasz	tolerálható ALARP mellett	ergonómiai vizsgálat, ütemezés

A táblázat utolsó előtti sora jól mutatja, hogy a veszélyazonosítás nem csak a hardverre vonatkozik: a szoftver módosítása, a szervezeti folyamatok és az ember–gép együttműködés is veszélyforrás.

5.2.3. „Mi van, ha?” elemzés

A „Mi van, ha?” elemzés (*what-if analysis*) strukturált ötletgyűjtés: a csapat a rendszer egyes részeire vagy műveleteire vonatkozóan „Mi van, ha…?” kérdéseket tesz fel („Mi van, ha a hűtővízszivattyú leáll?”, „Mi van, ha a kezelő rossz adagot tölt be?”), és megvizsgálja a következményeket és a meglévő védelmeket. Strukturáltabb változata a SWIFT (*structured what-if technique*), amely előre definiált kérdéskategóriákat (emberi hiba, berendezés-meghibásodás, közművek kiesése, külső hatások, időzítés stb.) használ. A módszer rugalmas és gyors, ezért jól használható változtatások (*management of change*) hatásának előzetes vizsgálatára, de alapossága nagyban függ a csapat tapasztalatától.

5.3. HAZOP: veszély- és üzemeltethetőségi vizsgálat

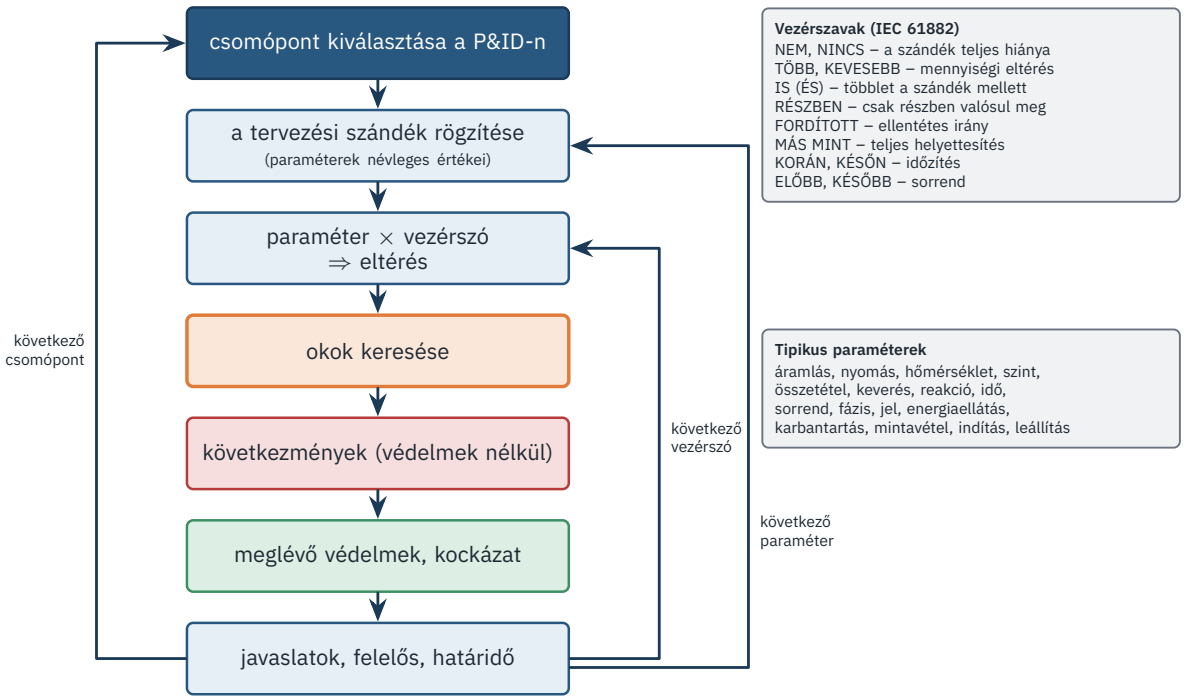
A HAZOP (*hazard and operability study*) módszert az 1960-as években az Imperial Chemical Industries (ICI) vegyipari vállalatnál fejlesztették ki, és mára a folyamatipar legelterjedtebb veszélyazonosítási módszere lett. Alkalmazási útmutatója az IEC 61882 (IEC 2016c). A név a módszer kettős célját mutatja: a biztonsági veszélyek mellett az üzemeltethetőségi problémákat (amelyek nem veszélyesek, de rontják a hatékonyságot vagy a minőséget) is feltárja.

5.3.1. Az alapgondolat

A HAZOP alapgondolata, hogy a veszélyek a **tervezési szándéktól való eltérésekből** fakadnak. A módszer ezért a terv minden részét (csomópontját) végigveszi, rögzíti a tervezési szándékot (milyen áramlást, nyomást, hőmérsékletet vár a tervező), majd **vezérszavak** (*guide words*) segítségével rendszeresen előállítja a lehetséges eltéréseket. A vezérszó és a folyamatparaméter kombinációja egy eltérés: „NINCS + áramlás” = nincs áramlás; „TÖBB + hőmérséklet” = túl magas hőmérséklet; „FORDÍTOTT + áramlás” = visszaáramlás. Minden értelmes eltérésnél a csapat megkeresi a lehetséges okokat és a következményeket, rögzíti a meglévő védelmeket, és ha szükséges, javaslatot tesz.

5.3.2. A vizsgálat menete

A HAZOP-vizsgálat lépéseit a 5.1. ábra mutatja.



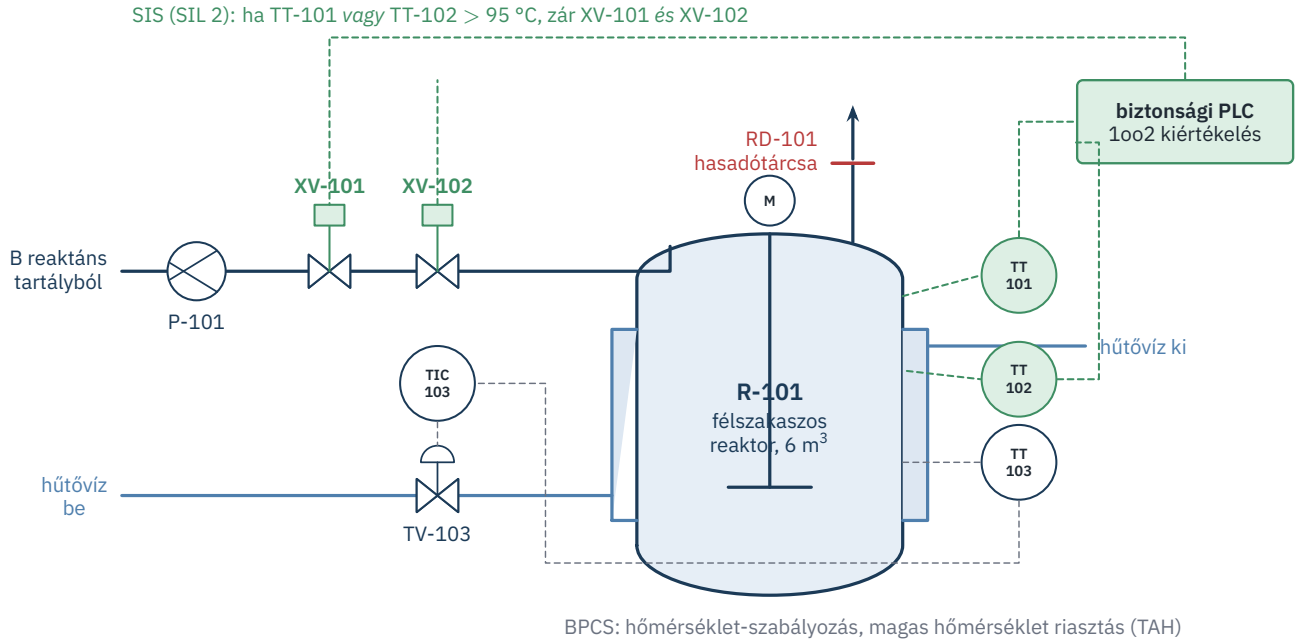
5.1. ábra. A HAZOP-vizsgálat menete és a szabványos vezérszavak.

1. **Előkészítés.** A vizsgálat hatókörének és céljainak rögzítése, a csapat összeállítása, a dokumentáció (csővezeték- és műszerezési rajz, azaz P&ID, folyamatleírás, üzemeltetési utasítások, anyagbiztonsági adatlapok) összegyűjtése, a csomópontok kijelölése.
2. **Csomópontonkénti vizsgálat.** Minden csomópontra (például egy vezetékszakasz vagy egy tartály a hozzá tartozó berendezésekkel) a tervezési szándék rögzítése, majd a paraméterek és a vezérszavak végigvétele.
3. **Okok és következmények.** Minden értelmes eltérés okainak felsorolása (berendezés-meghibásodás, emberi hiba, külső hatás), majd a következmények vizsgálata *a védelmek figyelmen kívül hagyásával*. Ez fontos: a következményt először úgy kell felmérni, mintha a védelmek nem működnének, mert csak így látszik, mennyit kell a védelmeknek teljesíteniük.
4. **Védelmek és kockázat.** A meglévő védelmek rögzítése (szabályozás, riasztás, biztonsági funkció, biztonsági szelep), és sok vállalatnál egy előzetes kockázati besorolás (kockázati mátrixszal).
5. **Javaslatok.** Ha a kockázat nem tolerálható vagy bizonytalan, javaslat további védelemre, tervmódosításra vagy részletesebb elemzésre (például LOPA-ra), felelőssel és határidővel.

A **csapat** jellemzően 5–8 főből áll: független, a módszerben jártas vezető, jegyzőkönyvvezető, folyamatmérnök, irányítástechnikus, üzemeltetési szakember (lehetőleg tapasztalt kezelő), karbantartási szakember és szükség szerint vegyész, gépész vagy biztonsági szakember. A vezető feladata, hogy a csapat a módszert következetesen alkalmazza, de ne veszítse el a kreativitását, és hogy a megbeszélés ne csússzon át tervezési vitába. A HAZOP a problémák *azonosítására* szolgál; a megoldásukat a javaslatokban rögzített felelősök utólag dolgozzák ki.

5.3.3. Kidolgozott HAZOP a ReaktorVéd esetre

A ReaktorVéd egyszerűsített folyamatábráját a 5.2. ábra mutatja. A B reaktánst a P-101 szivattyú adagolja a reaktorba két sorba kötött gyorszáró szelepen (XV-101, XV-102) át. A reakcióhőt a köpenyben áramló hűtővíz vonja el; a hőmérsékletet a TIC-103 szabályozó tartja az alapjelen a TV-103 szelep állításával. A független biztonsági rendszer két hőmérséklet-távadóval (TT-101, TT-102) figyeli a reaktort, és bármelyik határérték-túllépésekor mindkét gyorszáró szelepet zárja. A reaktoron hasadótárcsa (RD-101) is van.



5.2. ábra. A ReaktorVéd egyszerűsített folyamat- és műszerezési ábrája. A zöld elemek a független biztonsági rendszerhez (SIS) tartoznak.

A 5.2. táblázat a „reaktor hűtőköre” csomópont HAZOP-jegyzőkönyvének részlete. A tervezési szándék: a hűtővíz 15–25 °C-os belépő hőmérséklettel, legalább 12 m³/h térfogatárammal áramlik a köpenyen át, és a reaktor hőmérsékletét az exoterm fázisban 70 ± 3 °C-on tartja.

5.2. táblázat. A ReaktorVéd HAZOP-jegyzőkönyvének részlete: a reaktor hűtőköre

Eltérés	Lehetséges okok	Következmények (védelmek nélkül)	Meglévő védelmek	Javaslat
NINCS hűtővízáramlás	hűtővízszivattyú leállása; áramszünet; TV-103 zárt helyzetben beragad; elzárt kézi szelep	a hőmérséklet megfut, a nyomás a tervezési érték fölé nő, a reaktor felhasadhat, mérgező kibocsátás, halálos sérülés	TAH-103 riasztás és kezelői beavatkozás; SIS: TT-101/102, XV-101/102; RD-101	LOPA a forgatókönyvre (SIL-meghatározás); a hasadótárcsa méretezésének ellenőrzése kétfázisú áramlásra

Eltérés	Lehetséges okok	Következmények (védelmek nélkül)	Meglévő védelmek	Javaslat
KEVESEBB hűtővízáramlás	szennyezett hőcserélő felület; részben zárt szelep; szivattyúkopás	lassabb hőmérséklet-emelkedés, a fenti következmény késleltetve	TIC-103 szabályozás; TAH-103	a köpeny tisztítási gyakoriságának felülvizsgálata; áramlásmérés a hűtőkörbe
TÖBB hűtővízáramlás	TV-103 nyitott helyzetben beragad	a reakció lelassul, a reaktáns felhalmozódik; később hirtelen beinduló reakció	TAL alacsony hőmérséklet riasztás	**reteszelés: az adagolás tiltása, ha a hőmérséklet 60 °C alatt van** (a felhalmozódás megakadályozása)
MAGASABB belépő hőmérséklet	a hűtőtorony túlterhelt, nyári csúcs	csökkenő hűtési kapacitás, hőmérséklet-emelkedés	TIC-103; TAH-103	a hűtési kapacitás igazolása 30 °C-os belépő hőmérsékletre
FORDÍTOTT áramlás	csőcsere utáni hibás bekötés	a köpeny nem hűt, lásd NINCS áramlás	üzembe helyezési ellenőrzés	üzembe helyezési ellenőrzőlista kiegészítése
MÁS MINT víz a köpenyben	termékszivárgás a köpenybe (repedés)	a reakcióelegy a hűtővízbe jut, környezetszennyezés	–	vezetőképesség-mérés a visszatérő hűtővízben

A jegyzőkönyv két fontos tanulságot hordoz. Az első sor az a súlyos forgatókönyv, amelyre a biztonsági funkciót tervezik; a javaslat szerint a szükséges integritást LOPA-val kell meghatározni (11. fejezet). A harmadik sor kevésbé nyilvánvaló, de klasszikus vegyipari veszélyre mutat rá: ha a túlhűtés miatt a reakció nem indul be, a folyamatosan adagolt reaktáns felhalmozódik, és később egyszerre, hatalmas hőfejlődéssel reagálhat el. Ilyen „rejtett” veszélyek feltárásában a HAZOP vezérszavas rendszeressége különösen erős.

> **A következményt a védelmek figyelembevételével becsülik**
>
> Gyakori hiba, hogy a HAZOP-csapat a következmény oszlopba azt írja, ami a védelmek *működése* mellett történik („a SIS leállítja az adagolást, nincs következmény”). Így a következmény súlyossága nem látszik, és nem derül ki, mennyit kell a védelmeknek teljesíteniük. A helyes eljárás: a következményt a védelmek nélkül kell felmérni, a védelmeket külön oszlopban rögzíteni, és a kockázatot ezek együttes figyelembevételével értékelni.

5.3.4. HAZOP-változatok

A HAZOP gondolata, vagyis a szándéktól való eltérések rendszeres előállítása vezérszavakkal, a folyamatiparon túl is alkalmazható:

- **Eljárási HAZOP:** üzemeltetési eljárások (indítás, leállítás, tisztítás) lépéseinek vizsgálata. A paraméterek itt a műveletek, a vezérszavak például KIMARAD, KORÁN, KÉSŐN, MÁS SORRENDBEN.
- **CHAZOP** (*computer HAZOP*) és **szoftver-HAZOP:** irányítórendszerek és szoftverek vizsgálata. A paraméterek az adatfolyamok, jelek és állapotok, a vezérszavak például NINCS, TÖBB, KEVESEBB, KORÁN, KÉSŐN, ROSSZ ÉRTÉK.

- **Ember–gép HAZOP:** a kezelő és a rendszer közötti információcserék és beavatkozások vizsgálata.

5.3.5. A HAZOP erősségei és korlátai

Erősségei: rendszeres, a csomópontok és vezérszavak teljes körű bejárása kevés veszélyt hagy rejtve; a multidiszciplináris csapat sokféle tudást mozgósít; az üzemeltethetőségi problémákat is feltárja; jól dokumentálható, és közvetlenül illeszkedik a LOPA-hoz.

Korlátai: időigényes (egy közepes üzem HAZOP-ja több hét lehet); erősen függ a vezető és a csapat tapasztalatától; elsősorban egyes eltérésekre és azok közvetlen hatásaira koncentrál, a több egyidejű hiba kombinációit és a csomópontok közötti távoli kölcsönhatásokat kevésbé tárja fel; a komplex irányítási és szoftveres kölcsönhatásokban kevésbé erős, mint az STPA. A HAZOP a P&ID-k elkészülte után végezhető érdemben, tehát a koncepciós fázisban még nem alkalmazható.

5.4. Funkcióalapú veszélyelemzés

5.4.1. Funkcionális veszélyelemzés (FHA)

A funkcionális veszélyelemzés (*functional hazard assessment*, FHA) a repülésben kialakult módszer, az ARP4761A biztonságértékelési folyamatának első lépése (SAE International 2023b). Kiindulópontja nem a fizikai terv, hanem a **funkciók listája**, ezért már a koncepciós fázisban elvégezhető. Minden funkcióra megvizsgálja a lehetséges **meghibásodási állapotokat** (*failure conditions*): a funkció elvesztését, a téves vagy indokolatlan működését, a részleges vagy késleltetett működését. Mindegyiket a következménye szerint osztályozza (katasztrofális, veszélyes, jelentős, csekély, nincs hatás), és meghatározza a megengedett valószínűséget és a fejlesztési biztosítottsági szintet.

VészFék: funkcionális veszélyelemzés

5.3. táblázat. A VészFék funkcionális veszélyelemzésének részlete

Funkció	Meghibásodási állapot	Működési helyzet	Hatás	Besorolás
akadály észlelése és vészfékezés	elmarad a fékezés valós akadály esetén	városi forgalom, gyalogos az úttesten	gyalogos elütése, ha a vezető sem reagál	súlyos (a vezető a felelős, a funkció kiegészítő)
akadály észlelése és vészfékezés	indokolatlan teljes fékezés	autópálya, követő jármű	ráfutásos baleset nagy sebességkülönbséggel	katasztrofális
akadály észlelése és vészfékezés	késve induló fékezés	bármely	csökkenő ütközéscsökkentő hatás	jelentős
figyelmeztetés a vezetőnek	téves figyelmeztetés	bármely	a vezető megzavarása, bizalomvesztés, a funkció kikapcsolása	csekély–jelentős

> Az elemzés legfontosabb tanulsága, hogy ugyanannak a funkciónak két ellentétes meghibásodási állapota (elmaradó és indokolatlan működés) más-más súlyosságú. Ezért a tervezésnek más-más eszközökkel kell kezelnie őket: az indokolatlan fékezés ellen a funkcionális biztonság eszközei (redundáns érzékelés, plauzibilitásvizsgálat), az elmaradó fékezés ellen a teljesítőképesség növelése és a SOTIF szemlélete (18. fejezet) segít.

5.4.2. Veszélyelemzés és kockázatbecslés (HARA)

Az ISO 26262 szerinti HARA a jármű funkciói (az *item*) szintjén dolgozik (ISO 2018a). Lépései:

1. **Az item definíciója:** a funkció, határai, interfészei, a feltételezett működési környezet.
2. **Működési helyzetek elemzése:** a releváns közlekedési helyzetek, üzemmódok és környezeti feltételek katalógusa (autópálya, város, parkolás, csúszós út, éjszaka stb.).
3. **Veszélyazonosítás:** a funkció meghibásodási viselkedéseinek (elvesztés, indokolatlan működés, hibás működés) és a működési helyzeteknek a kombinációi adják a veszélyes eseményeket. Az azonosításhoz FHA, HAZOP-szerű vezérszavak és tapasztalati listák használhatók.
4. **Osztályozás:** minden veszélyes eseményre S, E és C besorolás és az ASIL (4.4.2. szakasz).
5. **Biztonsági célok:** az azonos ASIL-ű vagy összefüggő veszélyes eseményekre biztonsági célok megfogalmazása.

A HARA-ban különös figyelmet érdemelnek azok a meghibásodások, amelyek egyszerre több funkciót érintenek vagy sokféle veszélyes helyzetet okozhatnak (közös okú, kaszkád hibák, például a tápellátás hibája). Az egyes veszélyes eseményeket egymástól függetlenül kell értékelni; a „legrosszabb eset” egyetlen besorolása elfedheti a különbségeket.

5.5. A módszerek kiválasztása

A 5.4. táblázat összefoglalja a veszélyazonosítási módszerek jellemzőit és tipikus alkalmazási helyüket az életciklusban.

5.4. táblázat. A veszélyazonosítási módszerek összevetése

Módszer	Életciklus-fázis	Nézőpont	Fő erőssége	Fő korlátja
ellenőrzőlista	minden fázis	tapasztalat	gyors, a tudás nem vész el	csak az ismert veszélyeket találja meg
PHA	koncepció	veszélyforrás	korai, irányt mutat	durva felbontás
„Mi van, ha?”, SWIFT	koncepció, változtatások	ötletgyűjtés	rugalmas, gyors	a csapattól erősen függ
HAZOP	részletes tervezés (P&ID)	folyamateltérés	rendszeres, teljes körű	időigényes, egyhibás szemlélet
FMEA, FMECA	részletes tervezés	alkatrész-meghibásodás	alulról felfelé teljes	a kölcsönhatásokat és a kombinációkat gyengén kezeli

Módszer	Életciklus-fázis	Nézőpont	Fő erőssége	Fő korlátja
FHA, HARA	koncepció	funkció	korai, független a megvalósítástól	a megvalósítás részleteit nem látja
STPA	koncepciótól	irányítás, kölcsönhatás	specifikációs és ember–automatizálás hibák	új, kevésbé szabványosított

A gyakorlatban a módszerek egymásra épülnek: a koncepciós fázis PHA- vagy FHA-elemzése kijelöli a kritikus területeket, ahol a részletes tervezés során HAZOP és FMEA következik; a komplex irányítási kölcsönhatásokat STPA tárja fel; a súlyos forgatókönyveket pedig a III. rész kvantitatív módszerei számszerűsítik.

Nagy nyelvi modell mint HAZOP-segítő

A nagy nyelvi modellek hasznosak lehetnek a vezérszavas ötletgyűjtés kiegészítésére, de nem helyettesítik a szakértői csapatot, és állításaikat mindig ellenőrizni kell. Egy lehetséges prompt:
„Egy félszakaszos keverős reaktor hűtőkörének HAZOP-vizsgálatát végezzük. A tervezési szándék: hűtővíz 15–25 °C-on, legalább 12 m³/h, a reaktor 70 ± 3 °C-on tart egy exoterm reakciót. Az IEC 61882 vezérszavai közül vedd sorra a NINCS, TÖBB, KEVESEBB, FORDÍTOTT és MÁS MINT szót az áramlás és a hőmérséklet paraméterre. Minden eltérésnél sorolj fel lehetséges okokat (berendezés, ember, közmű, külső hatás) és következményeket a védelmek figyelmen kívül hagyásával. Jelöld meg, mely állításaid bizonytalanok vagy ellenőrzést igényelnek, és ne találj ki konkrét számadatokat.”
A modell válaszát a csapat a saját jegyzőkönyvével veti össze: a modell gyakran felvet a csapat által nem gondolt okokat, de téves vagy irreleváns javaslatokat is tesz.

Gyakorlat

A gyakorlati útmutató 3. laborjában HAZOP-jegyzőkönyvet készítesz a ReaktorVéd adagolóköréhez (a B reaktáns csomópontjához), majd a súlyos forgatókönyveket átadod a 9. labor LOPA-alkalmazásának.

Összefoglalás

- A veszélyazonosítás teljessége nem bizonyítható; a valószínűségét eltérő nézőpontú módszerek, multidiszciplináris csapat és korai, ismételt elemzés növeli.
- Az ellenőrzőlisták és az energiaforrás-szemlélet gyors kiindulópontot adnak; a PHA a koncepciós fázis módszere.
- A HAZOP a tervezési szándéktól való eltéréseket vezérszavak és paraméterek kombinációival állítja elő; a következményeket a védelmek nélkül kell felmérni.
- Az FHA és a HARA funkcióalapú módszerek, ezért a koncepciós fázisban is alkalmazhatók; ugyanannak a funkciónak az elmaradó és az indokolatlan működése eltérő súlyosságú lehet.
- A módszerek kiegészítik egymást; eredményük a teljes életcikluson át vezetett veszélynapló.

Önellenőrzés

1. Miért kell a HAZOP-ban a következményeket a védelmek figyelmen kívül hagyásával felmérni?
2. Állíts elő eltéréseket a „KÉSŐN” és a „MÁS SORRENDBEN” vezérszóval egy olyan eljárási lépésre, amely szerint „a kezelő a hűtés bekapcsolása után kezdi az adagolást”! Melyik a veszélyes?
3. Miért alkalmazható az FHA a koncepciós fázisban, a HAZOP pedig nem?
4. A ReaktorVéd HAZOP-jában a TÖBB hűtővízáramlás eltérés veszélyes következményt is okozhat. Mi ez, és milyen védelmet javasolt a csapat?

Megoldás.

1. Mert csak így derül ki a forgatókönyv valódi súlyossága és az, hogy mekkora kockázatcsökkentést kell a védelmeknek teljesíteniük. Ha a következményt a védelmek működésével becsüljük, a védelmek integritását nem lehet megalapozottan meghatározni, és egy hibás védelem rejtve marad.
2. KÉSŐN: a kezelő késve kezdi az adagolást; ez legfeljebb termelési veszteség. MÁS SORRENDBEN: a kezelő a hűtés bekapcsolása *előtt* kezdi az adagolást; ez veszélyes, mert a reakció hűtés nélkül indulhat be. Javaslat: reteszelés, amely csak igazolt hűtővízáramlás mellett engedi az adagolást.
3. Az FHA a funkciók listájából indul, amely már a koncepcióban ismert. A HAZOP a részletes tervből (P&ID, csomópontok, tervezési paraméterek) dolgozik, amely csak a részletes tervezésben áll rendelkezésre.
4. A túlhűtés miatt a reakció nem indul be, a folyamatosan adagolt reaktáns felhalmozódik, és később egyszerre, nagy hőfejlődéssel reagálhat el. A javasolt védelem az adagolás reteszelése, ha a reaktor hőmérséklete egy alsó határ (60 °C) alatt van.

Továbbolvasás

- IEC 61882:2016 – a HAZOP alkalmazási útmutatója, példákkal (IEC 2016c).
- Kletz, T. A. *HAZOP and HAZAN* (több kiadás, IChemE) – a HAZOP klasszikus gyakorlati bevezetője.
- SAE ARP4761A (2023) – az FHA és a teljes repülési biztonságértékelési folyamat (SAE International 2023b).
- MIL-STD-882E (2012) – a rendszerbiztonsági feladatok (PHL, PHA, SSHA, SHA) leírása (U.S. Department of Defense 2012).

6.

FMEA, FMECA és FMEDA



A hibamód- és hatáselemzés (*failure mode and effects analysis*, FMEA) a megbízhatósági és biztonsági mérnökség talán legismertebb módszere. Az 1940-es évek végén az amerikai hadiiparban született, az Apollo-programban terjedt el, az 1970-es évektől az autóiparban vált alapmódszerré, ma pedig szinte minden iparágban használják. Az FMEA alulról felfelé halad: minden alkatrészről vagy folyamatlépésről megkérdezi, hogyan hibásodhat meg, és mi lesz ennek a hatása. Ebben a fejezetben megismerjük a módszer mai, harmonizált formáját (AIAG–VDA, 2019), a kockázat-prioritási szám (RPN) korlátait és a helyébe lépő cselekvési prioritást (AP), a kritikusságelemzést (FMECA), végül a biztonsági funkciók számára kulcsfontosságú változatot, az FMEDA-t, amely a meghibásodási rátákat biztonságos és veszélyes, észlelt és rejtett részekre bontja.

Tanulási eredmények

A fejezet végére képes leszel

- elmagyarázni az FMEA alapgondolatát, típusait és helyét az életciklusban;
- végigvezetni egy FMEA-t az AIAG–VDA hét lépésén, a meghibásodási lánc (hatás–mód–ok) helyes megfogalmazásával;
- értékelni a súlyosságot, az előfordulást és az észlelést, és kritikusan értelmezni az RPN-t;
- megindokolni, miért lépett az RPN helyébe a cselekvési prioritás;
- FMECA-kritikusságot számolni;
- FMEDA-t készíteni egy egyszerű eszközre, és kiszámolni a diagnosztikai lefedettséget és a biztonságos meghibásodási hányadot.

6.1. Az FMEA alapgondolata

Az FMEA tervező, fejlesztő és dokumentációs módszer, amellyel módszeresen azonosíthatók az egyes alkatrészek vagy folyamatlépések meghibásodási módjainak következményei, és amellyel a termékek és folyamatok minősége és megbízhatósága folyamatosan fejleszthető. A módszer **induktív**: a „Mi történik, ha ez az elem így hibásodik meg?” kérdésből indul ki. Jellemzői:

- **alulról felfelé halad:** az alkatrész vagy folyamatlépés meghibásodási módjából kiindulva követi a hatásokat a magasabb rendszerszintekig (szemben a hibafával, amely a csúcseseménytől halad lefelé, 10. fejezet);

- **egyszeres hibákat vizsgál:** minden meghibásodási módot a többitől függetlennek tekint; a hibák kombinációit nem elemzi;
- **elvi lehetőségeket vizsgál:** érdektelen, hogy a meghibásodás előfordult-e már, vagy csak elvileg lehetséges;
- **megelőző szemléletű:** a hangsúly a meghibásodások megelőzésén és a terv vagy folyamat javításán van, nem a már bekövetkezett hibák utólagos elemzésén.

Az FMEA sikerének egyik legfontosabb feltétele, hogy **soha véget nem érő folyamat** legyen: élő dokumentum, amelyet a tervezés előrehaladtával, a változtatásokkal, az új tapasztalatokkal folyamatosan frissíteni kell.

Az FMEA-ra több szabvány és kézikönyv vonatkozik: az általános IEC 60812:2018 (IEC 2018), az autóipari AIAG–VDA *FMEA Handbook* (2019) (AIAG és VDA 2019), amely az amerikai AIAG és a német VDA korábban eltérő módszertanát egyesítette, valamint a SAE J1739 (2021).

6.1.1. Az FMEA típusai

- **Tervezési FMEA** (*design FMEA*, DFMEA): a termék konstrukciójának elemzése, a tervezésből eredő hibalehetőségek feltárása, még a gyártás megkezdése előtt.
- **Folyamat-FMEA** (*process FMEA*, PFMEA): a gyártási és szerelési folyamat elemzése, a gyártás során fellépő hibalehetőségek feltárása. A tervezési FMEA-ra épül, és figyelembe veszi a technológiát, az ellenőrzési lépéseket és a logisztikát.
- **Felügyeleti és rendszerválasz-FMEA** (*FMEA for monitoring and system response*, FMEA-MSR): az AIAG–VDA kézikönyv kiegészítő módszere, amely azt vizsgálja, hogy az üzem közbeni diagnosztika észleli-e a meghibásodást, és a rendszer biztonságos állapotba kerül-e. Közvetlen kapcsolatot teremt a funkcionális biztonsággal.

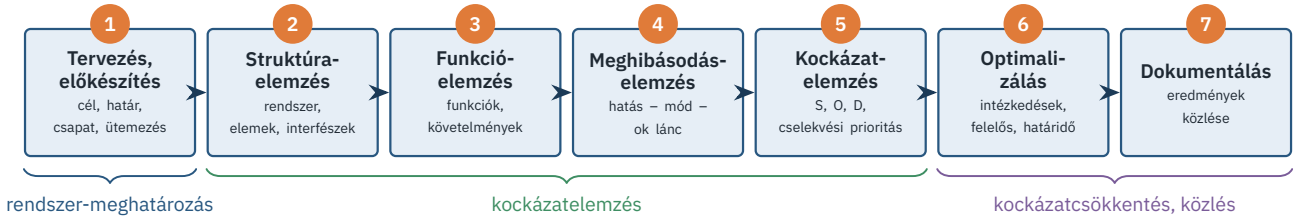
Az elemzés szintje szerint **rendszer-**, **alrendszer-** és **alkatrész-FMEA**-t különböztetünk meg. A határok meghúzása a csapat döntése; a lényeg, hogy a különböző szintek elemzései összekapcsolhatók legyenek: az alrendszer meghibásodási hatása a rendszer szintjén meghibásodási mód.

6.1.2. Mikor kell FMEA-t készíteni vagy frissíteni?

- **Új termék, technológia vagy gyártási folyamat:** az FMEA a teljes termékre vagy folyamatra kiterjed.
- **Meglévő termék vagy folyamat módosítása:** a korábbi FMEA-t a változás hatására kell kiterjeszteni.
- **Meglévő termék vagy folyamat új alkalmazása vagy környezete:** a megváltozott felhasználásra, helyszínre vagy környezetre kell összpontosítani.
- **Terepen jelentkező problémák, reklamációk:** az új meghibásodási módokat és okokat be kell építeni.

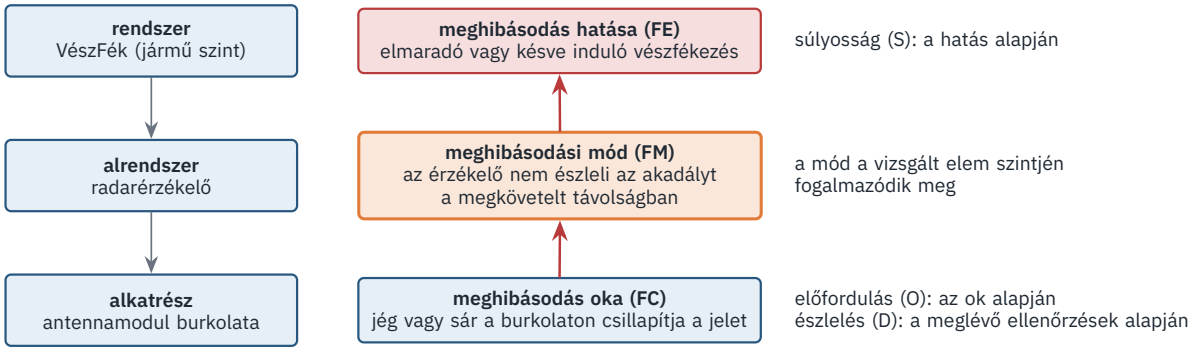
6.2. Az FMEA hét lépése

Az AIAG–VDA kézikönyv az FMEA-t hét lépésre bontja (6.1. ábra). A lépések a korábbi gyakorlat legjobb elemeit egyesítik, és a struktúra- és funkcióelemzés előtérbe helyezésével sokat javítanak az elemzés következetességén.



6.1. ábra. Az FMEA hét lépése az AIAG–VDA kézikönyv szerint.

1. **Tervezés és előkészítés.** Az elemzés céljának, hatókörének és határainak rögzítése, a csapat összeállítása, az ütemezés és az eszközök kiválasztása. A csapat multidiszciplináris: tervező, gyártástechnológus, minőségügyi, megbízhatósági és szükség szerint biztonsági szakember. Ezen a tudásigényen múlik az FMEA értéke: a módszer forgatókönyvszerű segítséget ad, de nem helyettesíti a szakmai ismereteket.
2. **Struktúraelemzés.** A rendszer felbontása hierarchikus szerkezetre (rendszer, alrendszer, alkatrész vagy folyamat, folyamatlépés, munkaelem) és az interfészek azonosítása.
3. **Funkcióelemzés.** Minden szerkezeti elemhez hozzárendeljük a funkcióit és a velük szemben támasztott követelményeket, lehetőleg mérhető formában (hőmérséklet-tartomány, nyomás, élettartam). Ha egy elemnek több funkciója van, mindegyiket fel kell sorolni.
4. **Meghibásodáselemzés.** Minden funkcióhoz megfogalmazzuk a **meghibásodási láncot**: a meghibásodási módot (*failure mode*, FM), annak hatását (*failure effect*, FE) a magasabb szinten és okát (*failure cause*, FC) az alacsonyabb szinten (6.2. ábra).
5. **Kockázatelemzés.** A meglévő megelőző és észlelő intézkedések rögzítése, a súlyosság (S), az előfordulás (O) és az észlelés (D) értékelése, majd a cselekvési prioritás (AP) meghatározása.
6. **Optimalizálás.** Intézkedések meghatározása a kockázat csökkentésére, felelőssel és határidővel; a bevezetés után az értékelés megismétlése.
7. **Dokumentálás.** Az eredmények összefoglalása és közlése az érintettekkel.



6.2. ábra. A meghibásodási lánc: a hatás, a mód és az ok három szomszédos szerkezeti szinthez kötődik.

A meghibásodási lánc helyes megfogalmazása az FMEA minőségének kulcsa. Néhány szabály:

- A **meghibásodási mód** fizikai, műszaki fogalmakkal írja le, *hogyan* nem teljesíti az elem a funkcióját („a tömítés nem zár 3 bar fölött”), nem pedig a felhasználó által érzékelt tünetet.
- A **hatást** úgy kell megfogalmazni, ahogyan a vevő (akár belső vevő, akár végfelhasználó) érzékeli. Ha a meghibásodás a biztonságot vagy a jogszabályi megfelelést érinti, azt egyértelműen jelezni kell.

- Az **ok** a meghibásodási mód kiváltója: tervezési gyengeség (rossz anyagválasztás, alulméretezés), gyártási eltérés vagy környezeti hatás. Minden módhoz az összes ésszerűen lehetséges okot fel kell sorolni, mert a hatékony intézkedések az okokra irányulnak.
- A különleges körülmények között fellépő módokat (alacsony vagy magas hőmérséklet, por, nedvesség, gyakoribb használat, nehéz terep) is figyelembe kell venni.

6.3. Az értékelés: súlyosság, előfordulás, észlelés

Az FMEA kockázatértékelése három, egyenként 1-től 10-ig terjedő skálán történik:

- **Súlyosság** (*severity*, S): a legsúlyosabb hatás súlyossága. Csak a terv megváltoztatásával csökkenthető.
- **Előfordulás** (*occurrence*, O): annak valószínűsége, hogy az ok bekövetkezik és meghibásodást okoz, a meglévő megelőző intézkedések figyelembevételével.
- **Észlelés** (*detection*, D): annak a valószínűsége, hogy a meglévő észlelő intézkedések (tesztek, vizsgálatok, ellenőrzések) *nem* szűrik ki az okot vagy a meghibásodási módot, mielőtt a termék a vevőhöz kerül. A magas érték rossz észlelést jelent.

A 6.1. táblázat egy általános, a 2014-es jegyzet táblázatait továbbfejlesztő skálát mutat. Az iparági kézikönyvek részletesebb, iparágspecifikus skálákat adnak; a lényeg, hogy a csapat egy skálát következetesen alkalmazzon az egész elemzésben.

6.1. táblázat. Általános S–O–D értékelési skála

Érték	Súlyosság (S)	Előfordulás (O)	Észlelés (D)
9–10	a felhasználóra veszélyes, sérti a jogszabályokat; 10: figyelmeztetés nélkül	a meghibásodás szinte biztos; nincs megelőzés	nincs ellenőrzés, vagy az nem képes észlelni
7–8	a termék nem működik, vagy alapfunkciója sérül	gyakori; a megelőzés gyenge	gyenge esélyű ellenőrzés (szemrevételezés, kis minta)
4–6	észrevehető, zavaró, de a termék alapvetően működik	alkalmi; a megelőzés részben hatékony	közepes esélyű, például statisztikai ellenőrzés
2–3	a felhasználó valószínűleg érzékeli, de alig zavarja	ritka; hatékony megelőzés	jó esélyű, bevált vizsgálati módszer
1	a felhasználó nem érzékeli	a megelőzés kizárja (például hibabiztos tervezés)	az ok vagy a mód biztosan észlelhető, illetve fizikailag kizárt

Az ellenőrzési intézkedéseket két oszlopban érdemes rögzíteni: a **megelőző** intézkedések (például tervezési irányelvek, méretezési számítások) az előfordulást, az **észlelő** intézkedések (tesztek, mérések) az észlelést befolyásolják. Mindig a megelőzés az előnyösebb megoldás.

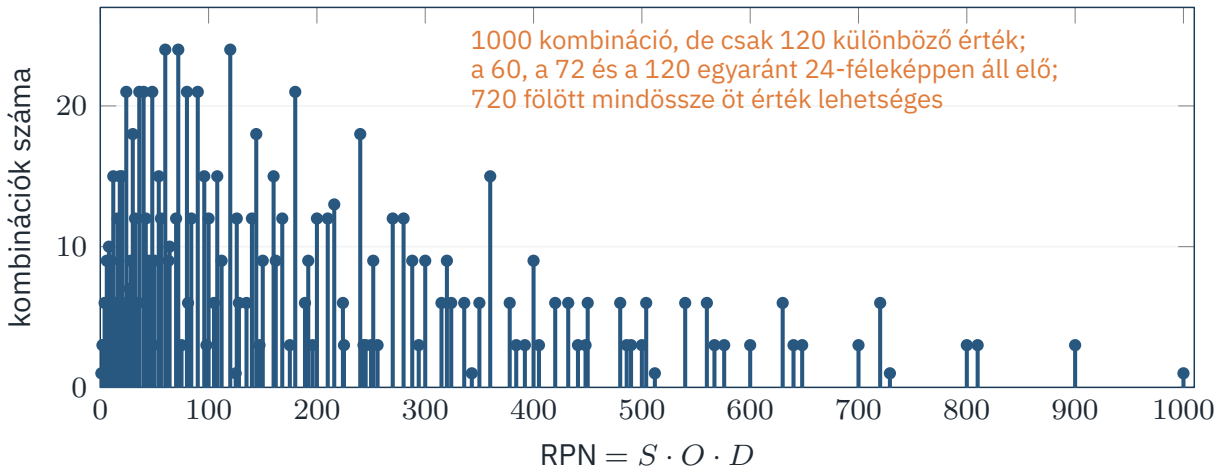
6.3.1. A kockázat-prioritási szám és korlátai

A hagyományos FMEA a három pontszám szorzatát, a kockázat-prioritási számot (*risk priority number*, RPN) használta a meghibásodási módok rangsorolására:

$$RPN = S \cdot O \cdot D, \qquad 1 \le RPN \le 1000. \tag{6.1}$$

Az RPN egyszerű és népszerű, de módszertanilag súlyos hibái vannak:

1. **Sorrendi skálák szorzása.** Az S, O és D sorrendi (ordinális) skálák: a 8-as súlyosság nem „kétszer olyan súlyos”, mint a 4-es. Az ilyen számok szorzatának nincs mennyiségi jelentése.
2. **Egyenlő súlyozás.** Az RPN a súlyosságot, az előfordulást és az észlelést egyenrangúnak tekinti. Egy halálos következményű, ritka meghibásodás ($S = 10, O = 2, D = 2$; $RPN = 40$) így alacsonyabb prioritást kap, mint egy zavaró, de ártalmatlan, gyakoribb hiba ($S = 4, O = 5, D = 5$; $RPN = 100$).
3. **Egyenetlen és nem egyértelmű eloszlás.** Az 1000 lehetséges kombinációból csak 120 különböző RPN-érték adódik (6.3. ábra). Ugyanaz az érték nagyon különböző kockázatokat takarhat: a 120-as érték például 24-féle (S, O, D) kombinációból áll elő.
4. **Önkényes küszöbök.** A gyakorlatban elterjedt küszöbértékek (például „$RPN < 40$ elfogadható, $RPN > 100$ beavatkozás szükséges”, ahogy a 2014-es jegyzet is közölte) semmilyen szabványban nem szerepelnek, és a fenti okok miatt nem is lehet őket megalapozni. Ráadásul arra ösztönöznek, hogy a csapat a pontszámokat „lefelé kerekítse”.



6.3. ábra. Az RPN lehetséges értékeinek eloszlása. A 1000 kombinációból csak 120 különböző érték adódik, és a skála felső része nagyon ritkán lakott.

6.3.2. A cselekvési prioritás

Az AIAG–VDA kézikönyv ezért az RPN helyett a **cselekvési prioritást** (*action priority*, AP) vezette be. Az AP egy táblázat, amely minden (S, O, D) kombinációhoz három prioritási szint egyikét rendeli: **magas** (H, intézkedés szükséges, vagy dokumentáltan meg kell indokolni, miért elegendők a meglévők), **közepes** (M, intézkedést kell mérlegelni) vagy **alacsony** (L, intézkedés lehetséges). Az AP alapelve, hogy a súlyosság a legfontosabb, utána az előfordulás következik, és csak végül az észlelés. Egy nagy

súlyosságú meghibásodásnál tehát jó észlelés mellett is magas lehet a prioritás, mert a megelőzés fontosabb a kiszűrésnél.

A hivatalos AP-táblázat a kézikönyvben található. A 6.2. táblázat egy saját, egyszerűsített szabályt mutat, amely csak a táblázat *logikáját* szemlélteti.

6.2. táblázat. A cselekvési prioritás logikája egyszerűsített szabállyal. Ez nem az AIAG–VDA kézikönyv táblázata.

Prioritás	Egyszerűsített szabály (szemléltetés)
magas (H)	$S \geq 9$ és ($O \geq 4$ vagy $D \geq 7$); vagy $7 \leq S \leq 8$ és $O \geq 6$
közepes (M)	$S \geq 7$, ha nem magas; vagy $4 \leq S \leq 6$ és $O \geq 6$
alacsony (L)	minden egyéb eset

A szakmai gyakorlat ettől függetlenül is egyértelmű: ha egy meghibásodás súlyossága 9 vagy 10 (a felhasználóra veszélyes vagy jogszabálysértő), akkor az RPN-től és az AP-től függetlenül célzott intézkedést kell meghatározni a következmény elkerülésére, az ok megszüntetésével, csökkentésével vagy ellenőrzésével. A javító intézkedések sorrendje: először a súlyosság (csak tervmódosítással), majd az előfordulás, végül az észlelés javítása.

VészFék: tervezési FMEA a radarérzékelőre (részlet)

6.3. táblázat. A VészFék radarérzékelőjének tervezési FMEA-részlete. *Az AP a 6.2. táblázat egyszerűsített szabálya szerint.

Funkció	Meghibásodási mód	Hatás (jármű)	S	Ok	Megelőzés / észlelés	O	D	RPN	AP*
akadály észlelése 0,5–150 m-en	nem észleli az akadályt a megkövetelt távolságban	elmaradó vészfékezés	8	jég vagy sár a burkolaton	fűtött burkolat / akulásdiagnosztika a jelszint alapján	4	4	128	M
akadály észlelése	téves (szellem-) célpontot jelez	indokolatlan erős fékezés	10	többutas visszaverődés fém szerkezetekről	célpontkövetési algoritmus / kiolulációs teszt kevés esettel	4	6	240	H
akadály távolságának mérése	a mért távolság eltolódik	késve induló fékezés	7	az oszcillátor hőmérsékletfüggő driftje	hőkompenzáció / hőkamrás teszt	3	3	63	M
adatküldés a fékvezérlőnek	az adat késve érkezik	késve induló fékezés	7	CAN-busz túlterhelés	ütemezésanalízis / időzítésfigyelés	2	2	28	M

A második sor magas prioritású: a téves célpont indokolatlan fékezést okozhat, amelynek súlyossága a HARA szerint a legmagasabb (4.4.2. szakasz). A csapat intézkedései: a radar és a kamera jeleinek összevetése a fékezés előtt (szenzorfúzió plauzibilitásvizsgálattal), valamint a valós forgalmi felvételeken alapuló szimulációs tesztkészlet bővítése. Az intézkedések után $O = 2$ és $D = 3$ (RPN = 60). A súlyosság nem változik, mert az csak a funkció koncepciójának módosításával csökkenthető.
Érdemes megfigyelni, hogy az első sor RPN-je (128) nagyobb, mint a negyediké (28), de kisebb, mint a másodiké (240). Ez a sorrend itt egybeesik az AP-vel; a fenti „halálos, de ritka” példa viszont mutatja, hogy ez nem mindig van így.

6.4. Kritikusságelemzés (FMECA)

A hibamód-, hatás- és kritikusságelemzés (*failure mode, effects and criticality analysis*, FMECA) az FMEA kiterjesztése, amely a meghibásodási módokat **kritikusságuk** szerint rangsorolja: aszerint, hogy mekkora valószínűséggel és milyen súlyos kárt okozhatnak. A klasszikus, a MIL-STD-1629A szabványból eredő mennyiségi változatban egy meghibásodási mód kritikussága:

$$C_m = \lambda_p \cdot \alpha \cdot \beta \cdot t, \tag{6.2}$$

ahol λ_p az alkatrész meghibásodási rátája, α a meghibásodási módarány (a meghibásodások mekkora hányada ebben a módban következik be), β annak a feltételes valószínűsége, hogy a mód a vizsgált hatást okozza, t pedig a működési idő. Egy alkatrész kritikussága a súlyos hatású módjainak kritikusságösszege. Az eredmény egy **kritikussági mátrixban** ábrázolható (súlyossági kategória a kritikusság függvényében), amely a kockázati mátrix rokona.

Egy gyorszáró szelep kritikussága

A ReaktorVéd XV-101 szelepének veszélyes meghibásodási rátája $\lambda_p = 2{,}5 \cdot 10^{-6}\,\mathrm{h}^{-1}$. A veszélyes meghibásodások 70%-a „nem zár” (beragadás), 30%-a „lassan zár” módban következik be. Ha nem zár, a biztonsági funkció biztosan elvész ($\beta = 1$); ha lassan zár, a válaszidő-követelmény az esetek felében sérül ($\beta = 0{,}5$). Egy év ($t = 8760\,\mathrm{h}$) működésre:

$$C_{\text{nem zár}} = 2{,}5 \cdot 10^{-6} \cdot 0{,}7 \cdot 1 \cdot 8760 = 1{,}53 \cdot 10^{-2}, \qquad C_{\text{lassan zár}} = 2{,}5 \cdot 10^{-6} \cdot 0{,}3 \cdot 0{,}5 \cdot 8760 = 3{,}3 \cdot 10^{-3}.$$

A beragadás kritikussága közel ötszöröse a lassú zárásénak, így a megelőző intézkedéseknek (például a részleges löketvizsgálatnak, 13. fejezet) elsősorban erre kell irányulniuk.

A kritikusságelemzéssel megtalálhatók azok a rendszerelemek, amelyekre a tervezés és az üzemeltetés során külön figyelmet kell fordítani. Az FMEA és az FMECA eljárását az IEC 60812 részletesen bemutatja.

6.5. FMEDA: a biztonsági funkciók FMEA-ja

A hibamód-, hatás- és diagnosztikaelemzés (*failure modes, effects and diagnostic analysis*, FMEDA) az FMEA kvantitatív változata, amelyet a biztonsági funkciókban felhasznált eszközökre dolgoztak ki. Célja, hogy az eszköz meghibásodási rátáját a 2.6.2. szakaszban bevezetett négy részre bontsa (λ_{SD}, λ_{SU}, λ_{DD}, λ_{DU}). Ezek az értékek a PFD-számítás (13. fejezet) és az architekturális korlátok közvetlen bemenetei. Az eszközgyártók tanúsítványaiban szereplő adatok tipikusan FMEDA-ból származnak.

Az FMEDA lépései:

1. az eszköz felbontása alkatrészekre vagy funkcionális blokkokra;
2. minden alkatrész meghibásodási rátájának és meghibásodási módjainak meghatározása (alkatrész-adatbázisokból, például az IEC 61709 vagy a SN 29500 szerint, vagy terepi adatokból);
3. minden módról annak eldöntése, hogy a biztonsági funkció szempontjából **biztonságos** vagy **veszélyes**;
4. annak eldöntése, hogy a beépített **diagnosztika** (önteszt, tartománytúllépés-figyelés, visszaolvasás, figyelőáramkör) észleli-e, és milyen lefedettséggel;
5. a részráták összegzése és a mérőszámok számítása.

A két legfontosabb mérőszám a **diagnosztikai lefedettség** és a **biztonságos meghibásodási hányad** (*safe failure fraction*, SFF):

$$DC = \frac{\sum \lambda_{DD}}{\sum \lambda_{DD} + \sum \lambda_{DU}}, \qquad SFF = \frac{\sum \lambda_S + \sum \lambda_{DD}}{\sum \lambda_S + \sum \lambda_D}. \tag{6.3}$$

Az SFF azt fejezi ki, hogy a meghibásodások mekkora hányada *nem* veszélyes és rejtett: vagy biztonságos, vagy a diagnosztika észleli. Az IEC 61508 architekturális korlátai (13. fejezet) az SFF-től és a hardver hibatűrésétől függően korlátozzák az elérhető SIL-t.

ReaktorVéd: a TT-101 hőmérséklet-távadó FMEDA-ja

A TT-101 egy ellenállás-hőmérős (RTD) távadó 4–20 mA-es kimenettel. A biztonsági funkció szempontjából a *magas* hőmérséklet jelzése a lényeg: az a meghibásodás veszélyes, amely miatt a távadó a valósnál *alacsonyabb* értéket mutat, és így a kioldás elmarad. Az a meghibásodás, amely magasabb értéket vagy tartományon kívüli jelet ad, kioldást okoz, tehát biztonságos. A logika a NAMUR NE 43 ajánlás szerinti tartományon kívüli jelet (3,6 mA alatt vagy 21 mA fölött) hibaként kezeli, és kiold.

6.4. táblázat. A TT-101 távadó egyszerűsített FMEDA-ja (1 FIT $= 10^{-9}\ \mathrm{h}^{-1}$)

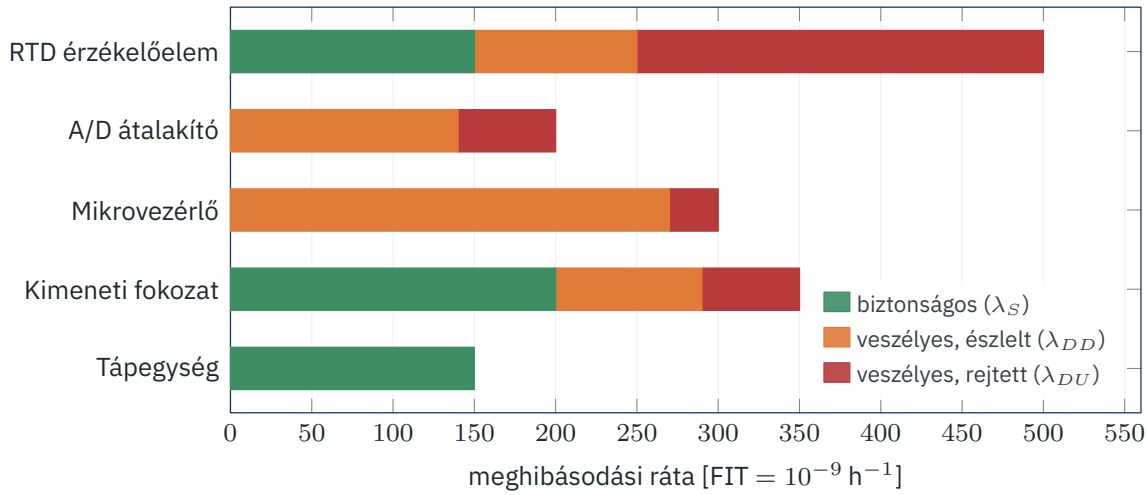
Blokk	Meghibásodási mód	λ [FIT]	Következmény	Diagnosztika	λ_S	λ_{DD}	λ_{DU}
RTD érzékelőelem	szakadás	150	felső tartományon kívüli jel	szakadásfigyelés	150	0	0
	rövidzár	100	alacsony érték	ellenálláshatár-figyelés	0	100	0
	lassú drift lefelé	250	alacsony érték	nincs	0	0	250
A/D átalakító	befagyott vagy hibás érték	200	hibás érték	referencia-összevetés (70%)	0	140	60
mikrovezérlő	programfutási hiba	300	befagyott kimenet	figyelőáramkör, RAM/ROM-teszt (90%)	0	270	30
kimeneti fokozat	áram a felső határon	200	kioldás	–	200	0	0
	áram beragad közbenső értéken	150	befagyott érték	áram-visszaolvasás (60%)	0	90	60

tápegység	kiesés	150	0 mA, a logika tartományon kívüli jel	kiold	150	0	0
összesen		**1500**			**500**	**600**	**400**

Az eredmények (6.4. ábra):

$$\lambda_D = 600{+}400 = 1000\,\text{FIT} = 1{,}0{\cdot}10^{-6}\,\text{h}^{-1}, \qquad DC = \frac{600}{1000} = 60\%, \qquad SFF = \frac{500+600}{1500} = 73{,}3\%.$$

A $\lambda_{DU} = 4 \cdot 10^{-7}\,\text{h}^{-1}$ és a $\lambda_{DD} = 6 \cdot 10^{-7}\,\text{h}^{-1}$ érték a 13. fejezet PFD-számításának bemenete. A legnagyobb rejtett veszélyes hozzájárulás az érzékelőelem lassú driftje (250 FIT). Ezt a diagnosztika nem fedi fel, de a két távadó jelének összevetése (eltérésfigyelés a biztonsági logikában) felfedné. Ez jó példája annak, hogyan javíthatja egy architektúraszintű döntés az eszközszintű mérőszámokat.



6.4. ábra. A TT-101 távadó meghibásodási rátájának megoszlása blokkok és kategóriák szerint.

Az FMEDA eredménye a felhasználástól is függ

Ugyanaz a meghibásodási mód az egyik alkalmazásban biztonságos, a másikban veszélyes lehet. A TT-101 felső tartományon kívüli jele magas hőmérsékletű kioldásnál biztonságos, de egy *alacsony* hőmérsékletű védelemnél (például fagyásvédelem) veszélyes. Ugyanígy a „0 mA” jel csak akkor biztonságos, ha a logika valóban kiold rá, és nem csak riaszt. Egy tanúsítvány FMEDA-adatai tehát mindig a tanúsítványban rögzített feltételek (*safety manual*) mellett érvényesek.

6.6. Az FMEA a gyakorlatban

6.6.1. Utógondozás

A javító intézkedések bevezetésének és nyomon követésének fontosságát nem lehet túlhangsúlyozni. A bármilyen alaposan elkészített FMEA is csak kis értéket képvisel, ha nem követik intézkedések. Az FMEA-ért felelős személy feladata, hogy meggyőződjön az intézkedések bevezetéséről és hatásosságáról, például a műszaki rajzok, folyamatleírások és munkautasítások átnézésével, a gyártásellenőrzési terv (*control plan*) felülvizsgálatával. Az FMEA élő dokumentum: a tervezési

koncepció idején el kell kezdeni, a fejlesztés során folyamatosan frissíteni kell, és a gyártási eszközök kiadása előtt teljessé kell tenni.

6.6.2. Szoftver-FMEA

Az FMEA szoftverre is alkalmazható, de másképp, mint a hardverre. A szoftverelemek (függvények, modulok, adatok) meghibásodási módjai: hibás érték, hiányzó vagy késve érkező kimenet, váratlan végrehajtás, hibás állapotátmenet, adatsérülés. A szoftver-FMEA nem a szoftver megbízhatóságát számszerűsíti (erre nem alkalmas, mert a szoftverhibák szisztematikusak), hanem azt vizsgálja, milyen hatása van egy szoftverhibának a rendszerre, és vannak-e ellene védelmek. Különösen hasznos olyan rendszerekben, ahol a hardvervédelem minimális, és a szoftver feladata a hardverhibák kezelése is (15. fejezet).

6.6.3. Erősségek és korlátok

Előnyök: következetes módszer; „válságmenedzsment” helyett kockázatkezelés; számszerűsített (bár sorrendi) kockázat; dokumentált tapasztalatok; célzott ok–hatás elemzés; a gyenge pontok és a javító intézkedések rendszeres feltárása.

Hátrányok: nagy időigény; szubjektív kockázatbecslés; a költségek és a haszon nehezen becsülhetők; jelentős utógondozást igényel; a többszörös meghibásodásokat, a közös okú hibákat és a komponensek közötti dinamikus kölcsönhatásokat nem kezeli. Az FMEA ott vált be leginkább, ahol a veszély mechanikai berendezésekből vagy villamos meghibásodásokból ered, és kevésbé a folyamatok dinamikájából. Ez utóbbira a HAZOP, a komplex irányítási kölcsönhatásokra az STPA alkalmasabb.

Az FMEA és a hibafa egymás kiegészítői: az FMEA alulról felfelé feltárja az egyes meghibásodási módokat, a hibafa felülről lefelé a kombinációikat. A gyakorlatban az FMEA eredménye gyakran a hibafa alapeseményeinek forrása.

Gyakorlat

A gyakorlati útmutató 4. laborjában egy böngészős FMEA-munkalappal dolgozol, amely ugyanarra a táblázatra egyszerre számol RPN-t és cselekvési prioritást, és megjelöli azokat a sorokat, ahol a két rangsor ellentmond egymásnak. A hozzá tartozó notebook az FMEDA-mérőszámokat számolja, és megmutatja, hogyan változik a DC és az SFF a diagnosztika fejlesztésével.

Összefoglalás

- Az FMEA induktív, alulról felfelé haladó módszer, amely az egyes meghibásodási módok hatásait tárja fel; egyszeres, egymástól független hibákat vizsgál.
- Az AIAG–VDA hét lépése: tervezés, struktúraelemzés, funkcióelemzés, meghibásodáselemzés (hatás–mód–ok lánc), kockázatelemzés, optimalizálás, dokumentálás.
- Az RPN sorrendi skálák szorzata; egyenlő súlyozása, nem egyértelmű eloszlása és önkényes küszöbei miatt a rangsorolásra kevéssé alkalmas. Helyét a súlyosságot előtérbe helyező cselekvési prioritás vette át.
- A 9–10-es súlyosságú meghibásodásokra a prioritási számtól függetlenül célzott intézkedés szükséges.
- Az FMECA a meghibásodási módokat kritikusságuk ($C_m = \lambda_p \alpha \beta t$) szerint rangsorolja.
- Az FMEDA a meghibásodási rátát biztonságos és veszélyes, észlelt és rejtett részekre bontja; eredménye a DC és az SFF, a PFD-számítás és az architekturális korlátok bemenete.

Önellenőrzés

1. Mi a különbség a meghibásodási mód és a meghibásodás hatása között? Adj példát egy villamos motor tekercsszigetelésére!
2. Két meghibásodási mód: (a) $S = 9, O = 3, D = 3$; (b) $S = 3, O = 6, D = 6$. Számold ki az RPN-t! Melyiket rangsorolja előbbre az RPN, és melyiket kellene?
3. Egy eszköz meghibásodási rátái: $\lambda_S = 300$ FIT, $\lambda_{DD} = 450$ FIT, $\lambda_{DU} = 250$ FIT. Mekkora a DC és az SFF?
4. Miért függ az FMEDA eredménye attól, hogy milyen biztonsági funkcióban használjuk az eszközt?

Megoldás.

1. A meghibásodási mód az, *ahogyan* az elem nem teljesíti a funkcióját (például a tekercsszigetelés átüt, testzárlat alakul ki). A hatás az, amit a magasabb szinten vagy a felhasználó érzékel (a motor leáll, kiold a védelem, a hajtott berendezés megáll). Az ok a módot kiváltó jelenség (például túlmelegedés miatti szigetelésöregedés).
2. (a) $RPN = 81$, (b) $RPN = 108$. Az RPN a (b)-t rangsorolja előbbre, holott az (a) súlyossága 9 (a felhasználóra veszélyes), így annak kellene magasabb prioritást kapnia; a cselekvési prioritás logikája szerint is az (a) a fontosabb.
3. $DC = 450/(450 + 250) = 64{,}3\%$; $SFF = (300 + 450)/(300 + 450 + 250) = 75\%$.
4. Mert a meghibásodási mód biztonságos vagy veszélyes jellege a biztonsági funkció irányától függ (magas vagy alacsony határérték, energiával vagy energiamentesítéssel működő kioldás), és a diagnosztika hatása is attól függ, hogy a logika hogyan reagál a hibajelzésre (kiold vagy csak riaszt).

Továbbolvasás

- AIAG, VDA (2019): *Failure Mode and Effects Analysis – FMEA Handbook* – a harmonizált autóipari módszertan, az AP-táblázatokkal (AIAG és VDA 2019).
- IEC 60812:2018 – az FMEA és FMECA általános szabványa (IEC 2018).
- Rausand, M. (2014): *Reliability of Safety-Critical Systems*, 7–8. fejezet – az FMEDA és a meghibásodási ráták felosztása (Rausand 2014).

7.

Rendszerelméleti veszélyelemzés: STAMP és STPA



Az FMEA és a hibafa alapfeltevése, hogy a balesetek alkatrészhibákból vagy azok kombinációiból fakadnak. A szoftverintenzív, automatizált, emberrel együttműködő rendszerekben azonban egyre gyakoribbak azok a balesetek, amelyekben egyetlen alkatrész sem hibásodik meg. A baleset a komponensek nem biztonságos kölcsönhatásából, hibás követelményekből vagy az ember és az automatizálás félreértéséből fakad. Az 1. fejezetben bemutatott Mars Polar Lander, a Boeing 737 MAX vagy az Uber-baleset is ilyen. Erre a kihívásra adott választ Nancy Leveson STAMP-modellje és az erre épülő veszélyelemzési módszer, az STPA. Ez a fejezet a módszert a KoboCella kollaboratív robotcelláján vezeti végig.

Tanulási eredmények

A fejezet végére képes leszel

- elmagyarázni, miért nem elegendőek az alkatrészhiba-alapú módszerek a szoftverintenzív és ember–automatizálás rendszerekben;
- leírni a STAMP fogalmait: biztonsági korlát, hierarchikus irányítási struktúra, folyamatmodell, visszacsatolás;
- végigvezetni az STPA négy lépését: veszteségek és veszélyek meghatározása, irányítási struktúra modellezése, nem biztonságos irányítási akciók azonosítása, veszteségforgatókönyvek feltárása;
- az STPA eredményeiből biztonsági követelményeket levezetni;
- összevetni az STPA-t a hagyományos módszerekkel, és tudni, mikor melyiket érdemes alkalmazni.

7.1. Miért kell új módszer?

A hagyományos veszélyelemzési módszerek (FMEA, hibafa, HAZOP) az 1950–1970-es évek elektromechanikus rendszereire születtek. Ezekben a baleset tipikusan valóban egy vagy több alkatrész meghibásodásából fakadt, és a komponensek közötti kölcsönhatások viszonylag egyszerűek, átláthatók voltak. A mai rendszerekben három dolog változott meg (Leveson 2011):

1. **A szoftver nem hibásodik meg, mégis okozhat balesetet.** A szoftver pontosan azt teszi, amit a kód előír. Ha a követelmény hibás vagy hiányos (nem gondolt egy működési helyzetre), a szoftver

„helyesen” fog veszélyes dolgot tenni. A szoftverhiba-arány fogalmának nincs értelme, és a „meghibásodás” helyett a „nem biztonságos viselkedés” a helyes kategória.
2. **A komponensek kölcsönhatása bonyolulttá vált.** Egy modern jármű vagy gyártócella tucatnyi irányító egységből áll, amelyek hálózaton kommunikálnak. A veszély nem egy komponensben, hanem a kölcsönhatásban rejlik: két, külön-külön helyesen működő funkció együtt nem biztonságos.
3. **Az ember szerepe megváltozott.** Az ember már nem közvetlen kezelő, hanem felügyelő, aki az automatizálás állapotát értelmezi, és szükség esetén beavatkozik. A tipikus hiba nem a „rossz gomb megnyomása”, hanem az, hogy az ember másnak hiszi a rendszer állapotát, mint amilyen (üzemmód-tévesztés), vagy nem érti, mit csinál az automatizálás.

Az alkatrészhiba-alapú módszerek ezeket a jelenségeket rosszul kezelik. Az FMEA-ban a szoftvernek nincs meghibásodási módja abban az értelemben, ahogy egy szelepnek; a hibafa alapeseményei között nem szerepel az, hogy „a követelmény nem vette figyelembe a lábak kinyitásakor keletkező rezgést”.

7.2. A STAMP modell

A STAMP (*Systems-Theoretic Accident Model and Processes*) a rendszerelméletre és az irányításelméletre épül (Leveson 2011). Három alapfogalma van.

Biztonsági korlátok. A biztonság a rendszer emergens tulajdonsága: nem az egyes komponensekben, hanem a komponensek viselkedését korlátozó szabályokban jelenik meg. A biztonsági korlát (*safety constraint*) olyan feltétel, amelynek teljesülnie kell ahhoz, hogy a rendszer ne kerüljön veszélyes állapotba („a robot sebessége 250 mm/s alatt legyen, ha ember van a közelében”).

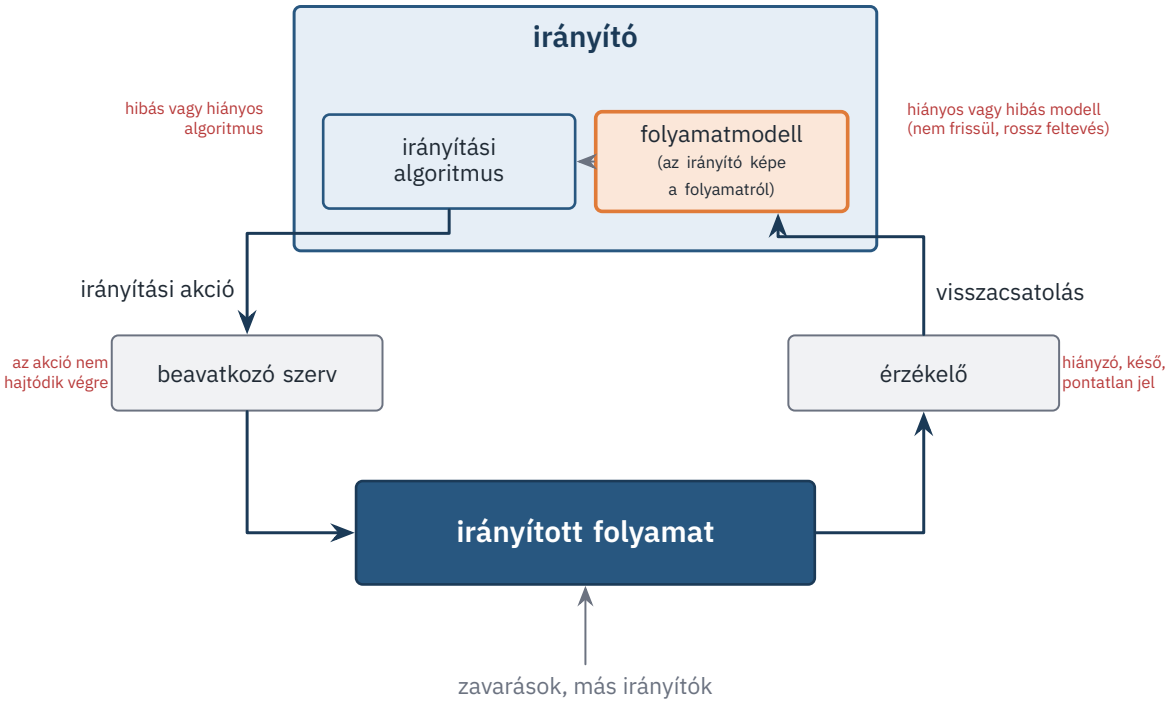
Hierarchikus irányítási struktúra. A biztonsági korlátokat irányítók érvényesítik, amelyek hierarchiába rendeződnek: a jogalkotótól és a hatóságtól a vállalatvezetésen, a mérnöki szervezeten és a kezelőn át az automatikus irányítókig. Minden szint irányítja az alatta lévőt (követelményekkel, utasításokkal, irányító parancsokkal), és visszajelzést kap tőle (jelentésekkel, mérésekkel).

Folyamatmodell. Minden irányító, ember és gép egyaránt, a döntéseit a saját **folyamatmodellje** alapján hozza: az irányított folyamat állapotáról alkotott képe alapján (7.1. ábra). Egy emberi kezelő folyamatmodellje a mentális modellje, egy szoftveré a belső változói. A balesetek jelentős része abból fakad, hogy a folyamatmodell eltér a valóságtól: a Therac-25 szoftvere más beállítást „hitt”, mint ami valójában volt; a 737 MAX MCAS-funkciója a hibás érzékelőjel miatt túl nagynak „hitte” az állásszöget; az Uber szoftvere a gyalogost minden átsorolásnál új objektumnak „hitte”.

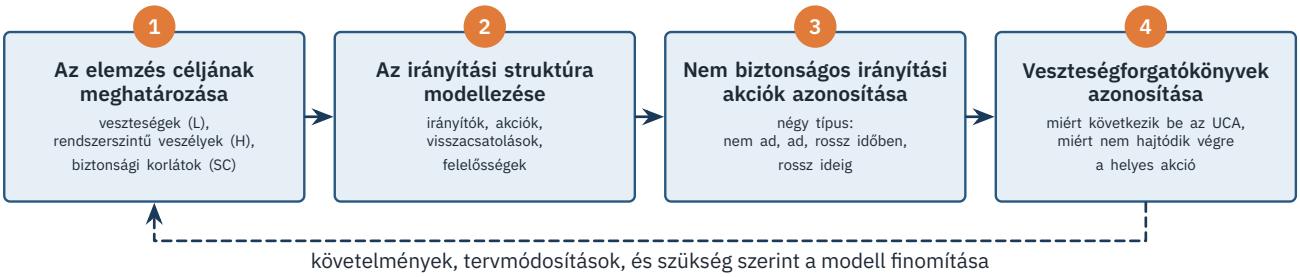
A STAMP szerint a baleset oka mindig az, hogy egy biztonsági korlát nem érvényesült. Ennek lehetséges okai az irányítási hurok elemeihez köthetők: az irányító nem ad ki szükséges parancsot, vagy nem biztonságos parancsot ad; a parancsot rosszkor vagy rossz ideig adja; a parancs nem hajtódik végre; a visszacsatolás hiányzik, késik vagy hibás, és ezért a folyamatmodell téves.

7.3. Az STPA lépései

Az STPA (*System-Theoretic Process Analysis*) a STAMP-ra épülő veszélyelemzési módszer. Részletes leírása az MIT kutatóinak *STPA Handbook* című, ingyenesen elérhető kézikönyvében található (Leveson és Thomas 2018). A módszer négy lépésből áll (7.2. ábra).



7.1. ábra. Az általános irányítási hurok a STAMP szerint, a lehetséges hibaforrásokkal. Az irányító döntéseit a folyamatmodellje alapján hozza, amely a visszacsatolásból frissül.



7.2. ábra. Az STPA négy lépése. Az eredmény a biztonsági követelmények és tervmódosítások listája.

7.3.1. 1. lépés: az elemzés céljának meghatározása

Az első lépés meghatározza, mit akarunk megelőzni.

- **Veszteségek** (*losses*, L): az érintettek számára elfogadhatatlan kimenetelek, például emberi sérülés vagy halál, környezeti kár, a küldetés vagy a termelés elvesztése, a hírnév sérülése.
- **Rendszerszintű veszélyek** (*system-level hazards*, H): olyan rendszerállapotok vagy feltételek, amelyek a környezet legrosszabb esetű körülményeivel együtt veszteséghez vezetnek. Fontos, hogy a veszély a *rendszer* állapota legyen (amit a tervező befolyásolni tud), ne a környezeté, és ne egy alkatrész meghibásodása. A „robotkar meghibásodik" nem rendszerszintű veszély, a „a robotkar a szerelőhöz közel a biztonságos határnál nagyobb sebességgel mozog" az.
- **Rendszerszintű biztonsági korlátok** (*system-level constraints*, SC): a veszélyek fordítottjai, vagyis azok a feltételek, amelyeknek teljesülniük kell a veszélyek megelőzéséhez, illetve a veszteség csökkentéséhez, ha a veszély mégis bekövetkezik.

Minden veszélyhez hozzárendeljük, mely veszteség(ek)hez vezethet, így a nyomon követhetőség a teljes elemzésen át biztosított.

7.3.2. 2. lépés: az irányítási struktúra modellezése

A második lépés a rendszer **irányítási struktúrájának** felrajzolása: az irányítók (emberek, szervezetek, automatikus egységek), az általuk kiadott irányítási akciók és az általuk kapott visszacsatolások. A modell nem fizikai rajz és nem adatfolyam-diagram, hanem funkcionális: azt mutatja meg, *ki mit irányít, és mit tud* az irányított folyamatról. Minden irányítóhoz rögzítjük a felelősségeit (melyik biztonsági korlát érvényesítéséért felel) és a folyamatmodelljének fontos változóit.

7.3.3. 3. lépés: nem biztonságos irányítási akciók

A harmadik lépés minden irányítási akcióra megvizsgálja, milyen körülmények között lehet **nem biztonságos** (*unsafe control action*, UCA). Az STPA négy típust különböztet meg:

1. az akció **elmaradása** veszélyhez vezet (*not providing causes hazard*);
2. az akció **kiadása** veszélyhez vezet (*providing causes hazard*);
3. az akció **túl korán, túl későn vagy rossz sorrendben** érkezik (*too early, too late, out of order*);
4. az akció **túl korán áll le vagy túl sokáig tart** (folyamatos akcióknál; *stopped too soon, applied too long*).

Minden UCA-t a **kontextusával** együtt kell megfogalmazni: egy akció önmagában ritkán veszélyes, csak bizonyos helyzetben. A jó UCA-megfogalmazás öt eleme: az irányító, az akció típusa, maga az akció, a kontextus és a hivatkozás a veszélyre. Például: *„A biztonsági vezérlő nem* aktiválja *a sebességkorlátozást*, amikor *személy tartózkodik a figyelt zónában* [H1]."

7.3.4. 4. lépés: veszteségforgatókönyvek

A negyedik lépés azt vizsgálja, *miért* következhet be egy UCA, illetve miért nem hajtódhat végre egy helyes akció. Két forgatókönyv-típus van:

- **(a) Miért adja ki az irányító a nem biztonságos akciót?** Az irányítási algoritmus hibás vagy hiányos; a folyamatmodell téves (az irányító rosszul ismeri a folyamat állapotát), mert a visszacsatolás hiányzik, késik, pontatlan, vagy rosszul értelmezik; az irányító más irányítóktól ellentmondó parancsokat kap.
- **(b) Miért nem hajtódik végre a helyes akció, vagy miért nem éri el a hatását?** A beavatkozó szerv hibás vagy lassú; az irányított folyamat másképp viselkedik, mint feltételezték; zavarás vagy más irányító akciója hatástalanítja.

A forgatókönyvekből közvetlenül adódnak a **követelmények és tervezési döntések**: redundáns vagy diverz visszacsatolás, a folyamatmodell érvényességének ellenőrzése, üzemmódjelzés, reteszelés, a kezelői felület módosítása, szervezeti eljárások.

7.4. Kidolgozott STPA a KoboCella esetre

7.4.1. Veszteségek, veszélyek, korlátok

A KoboCella STPA-jának első lépése a 7.1. táblázatban látható.

7.1. táblázat. A KoboCella STPA-jának veszteségei, rendszerszintű veszélyei és biztonsági korlátai

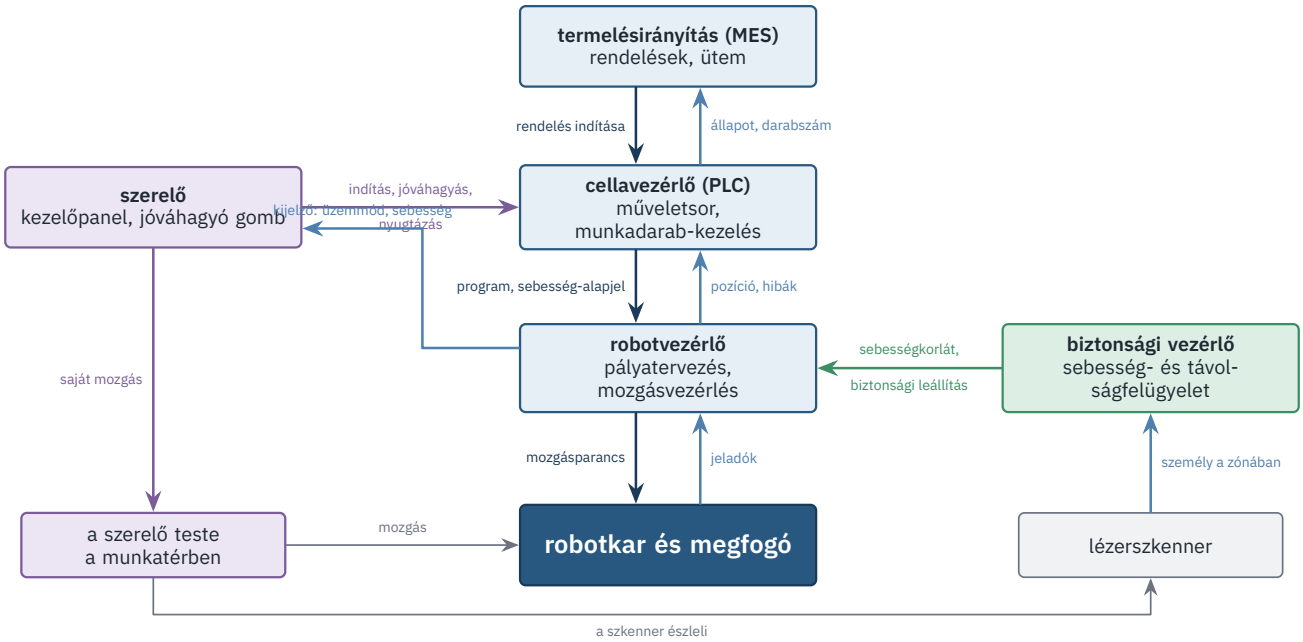
Azonosító	Megfogalmazás	Kapcsolat
L1	a szerelő sérülése	–
L2	a berendezés vagy a munkadarab károsodása	–
L3	a termelés kiesése	–
H1	a robotkar a szerelőhöz közel a biztonságos határnál nagyobb sebességgel mozog	L1
H2	a robot a szerelő testére a biomechanikai határértéknél nagyobb erőt vagy nyomást fejt ki	L1
H3	a robot a megfogott munkadarabot ellenőrizetlenül elengedi a szerelő fölött	L1, L2
H4	a cella indokolatlanul leáll, vagy nem indítható újra	L3
SC1	a robotkar sebessége nem haladhatja meg a szerelőtől mért távolsághoz tartozó biztonságos határt	H1
SC2	a robot által kifejtett erő és nyomás nem haladhatja meg az ISO/TS 15066 szerinti határértékeket	H2
SC3	a megfogó csak akkor nyithat, ha a munkadarab a lerakóhelyen van	H3

Azonosító	Megfogalmazás	Kapcsolat
SC4	a biztonsági funkciók nem okozhatnak a termelést aránytalanul zavaró téves leállást	H4

Érdemes észrevenni a H4–SC4 párt. Az STPA nem csak a biztonságot vizsgálja, hanem a termelékenységgel való konfliktust is láthatóvá teszi. A túl gyakori téves leállás a 1.4.5. szakaszban leírt sodródás klasszikus kiváltója: a kezelők elkezdik áthidalni a „kényelmetlen” védelmet.

7.4.2. Az irányítási struktúra

A KoboCella irányítási struktúráját a 7.3. ábra mutatja. A termelésirányító rendszer (MES) adja ki a rendeléseket a cellavezérlőnek; a cellavezérlő (PLC) futtatja a műveletsort, és sebesség-alapjelet ad a robotvezérlőnek; a robotvezérlő tervezi a pályát és vezérli a robotkart. A biztonsági vezérlő a lézerszkenner jelei alapján felügyeli a szerelő és a robot távolságát, és sebességkorlátozást vagy biztonsági leállítást kényszerít a robotvezérlőre. A szerelő a kezelőpanelen indítja, jóváhagyja és nyugtázza a műveleteket, és saját testével is része az irányított folyamatnak.



7.3. ábra. A KoboCella irányítási struktúrája. Lefelé az irányítási akciók, felfelé (kék) a visszacsatolások; a zöld elemek a biztonsági funkcióhoz, a lila elemek a szerelőhöz tartoznak.

A modell egy fontos részletét már a felrajzolás során észrevette a csapat: a szerelő az üzemmódot (kollaboratív vagy nem kollaboratív) és a pillanatnyi sebességkorlátot csak a kezelőpanel egy kis kijelzőjén látja, amelyre munka közben nem néz rá. A folyamatmodellje ezért elavulhat.

7.4.3. Nem biztonságos irányítási akciók

A 7.2. táblázat két irányítási akcióra mutatja az UCA-elemzést: a biztonsági vezérlő „sebességkorlátozás aktiválása” és a szerelő „a következő lépés jóváhagyása” akciójára.

7.2. táblázat. A KoboCella két irányítási akciójának UCA-elemzése

Akció	Nem adja ki	Kiadja	Rossz időben vagy sorrendben	Rossz ideig
biztonsági vezérlő: sebességkorlátozás	UCA-1: nem korlátoz, amikor személy van a figyelt zónában [H1]	UCA-2: korlátoz, amikor nincs személy a zónában (téves leállás) [H4]	UCA-3: a korlátozás azután lép életbe, hogy a szerelő már a robot pályáján van [H1]	UCA-4: a korlátozást feloldja, mielőtt a személy elhagyta a zónát [H1]
szerelő: jóváhagyás	(nem veszélyes; a cella vár)	UCA-5: jóváhagyja a következő lépést, amikor a keze még a megfogó közelében van [H2]	UCA-6: a lerakás előtt hagyja jóvá a megfogó nyitását [H3]	–

7.4.4. Veszteségforgatókönyvek és követelmények

Az UCA-1-hez (a sebességkorlátozás elmarad, pedig személy van a zónában) a csapat a következő forgatókönyveket tárta fel:

- **F-1 (téves folyamatmodell, takarás):** a szerelő a nagy munkadarab mögött áll, a lézerszkenner nem látja, a biztonsági vezérlő ezért úgy „tudja”, hogy a zóna üres. → *Követelmény:* a szkenner elhelyezését úgy kell megtervezni, hogy a munkadarab semmilyen pozícióban ne takarja a szerelő lehetséges helyzeteit; ennek igazolására takarási elemzés (szimuláció) szükséges.
- **F-2 (téves folyamatmodell, konfiguráció):** a karbantartás során a szkenner zónakonfigurációját módosították (a figyelt zónát kisebbre állították a téves leállások miatt), és ezt senki nem ellenőrizte. → *Követelmény:* a biztonsági konfiguráció módosítása csak jelszóval, naplózva és ellenőrzés után lehetséges; a konfiguráció ellenőrzőösszegét a biztonsági vezérlő minden indításkor ellenőrzi.
- **F-3 (a beavatkozás nem hat):** a biztonsági vezérlő kiadja a korlátozást, de a robotvezérlő egy szoftverfrissítés után a korlátozást csak a következő mozgásszakasztól alkalmazza. → *Követelmény:* a sebességkorlátozás végrehajtását a biztonsági vezérlőnek független sebességméréssel kell ellenőriznie, és eltérés esetén biztonságos leállítást kell kezdeményeznie; minden robotvezérlő-frissítés után a biztonsági funkciót validálni kell.
- **F-4 (visszacsatolás késése):** a szkenner válaszideje és a robot fékútja együtt nagyobb, mint amennyit a zónaméret megenged. → *Követelmény:* a védőtávolságot a robot legrosszabb esetű fékútjával és a teljes válaszidővel kell méretezni (lásd a gyakorlati útmutató feladatát).

Az UCA-5-höz (a szerelő a keze veszélyes helyzetében hagyja jóvá a következő lépést) a forgatókönyv egészen más jellegű: a szerelő folyamatmodellje téves (azt hiszi, hogy a robot kollaboratív üzemmódban, erőkorlátozással mozog, holott egy speciális műveletnél nem kollaboratív üzemmódban van). → *Követelmény:* az üzemmódot a munkaterületen jól látható módon kell jelezni (például színes fénysávval a roboton); a nem kollaboratív üzemmódban a jóváhagyás csak kétkezes vezérlővel lehetséges.

Ezek a forgatókönyvek jól mutatják az STPA erejét. Az F-2 (karbantartási konfigurációmódosítás), az F-3 (szoftverfrissítés utáni viselkedésváltozás) és az UCA-5 (üzemmód-tévesztés) egyike sem alkatrész-meghibásodás, ezért egy hagyományos FMEA-ban vagy hibafában nehezen jelenne meg. Mindhárom a valós ipari balesetek tipikus mintázata.

VészFék: az STPA gyors alkalmazása

A VészFék fékezési parancsának UCA-elemzése:

- *nem adja ki:* nem kezdeményez fékezést, amikor akadály van a pályán, és az ütközés elkerülhetetlen [H: frontális ütközés];
- *kiadja:* fékezést kezdeményez, amikor nincs valós akadály [H: ráfutásos baleset];
- *túl későn:* a fékezés akkor kezdődik, amikor az ütközés már nem csökkenthető számottevően;
- *túl korán áll le:* a fékezés megszakad, amikor a szoftver az akadályt átsorolja (például gyalogosból „ismeretlen objektummá”).

Az utolsó UCA egyik forgatókönyve közvetlenül megfelel az Uber-baleset jelentésében leírt mechanizmusnak: az osztályozás ingadozása miatt a rendszer mozgáselőzménye minden átsorolásnál elveszett (1.2. szakasz). A levezetett követelmény: az objektum követését és a fékezési döntést az osztályozástól függetlenül, a pálya és a mozgás alapján is meg kell hozni.

7.5. Az STPA helye a módszerek között

Az STPA nem váltja ki a hagyományos módszereket, hanem kiegészíti őket. A 7.3. táblázat a fő különbségeket foglalja össze.

7.3. táblázat. Az STPA és a hagyományos módszerek összevetése

Szempont	FMEA, hibafa	HAZOP	STPA
baleseti modell	alkatrészhiba, hibalánc	eltérés a tervezési szándéktól	nem érvényesülő biztonsági korlát
kiindulópont	alkatrészek vagy csúcsesemény	P&ID csomópontok	veszteségek, irányítási struktúra
szoftver, követelményhibák	gyengén	közepesen (CHAZOP)	erősen
ember és automatizálás	gyengén	közepesen	erősen (folyamatmodell)
kvantitatív eredmény	igen (valószínűség, gyakoriság)	nem (félkvantitatív kiegészítéssel)	nem
alkalmazhatóság	részletes tervezés	részletes tervezés	a koncepciótól kezdve

Az STPA különösen akkor ajánlott, ha a rendszer szoftverintenzív, sok irányítóból áll, ember és automatizálás együttműködésén alapul, vagy új, kevéssé ismert technológiát alkalmaz. Mivel a koncepciós fázisban is elvégezhető (a részletes terv ismerete nélkül), korán befolyásolhatja az architektúrát. A kvantitatív igazoláshoz (PFD, PMHF) viszont továbbra is a III. rész módszerei kellenek: az STPA megmondja, *mit* kell biztosítani, a hibafa és a Markov-modell pedig azt, hogy *mennyire megbízhatóan* sikerült.

A módszernek több változata és kiterjesztése is van. A **CAST** (*Causal Analysis based on STAMP*) a STAMP-ra épülő balesetvizsgálati módszer; az **STPA-Sec** a kiberbiztonsági veszélyek elemzésére alkalmazza ugyanezt a szemléletet. Az autóiparban a SAE J3187 ajánlott gyakorlat (2022) írja le az STPA alkalmazását, és a rendeltetésszerű funkció biztonságát szabályozó ISO 21448 is a lehetséges módszerek között említi.

Az STPA mint ellenőrzőlista

Az STPA-t néha táblázatkitöltési gyakorlatként alkalmazzák: minden akcióhoz mechanikusan beírnak négy UCA-t, és a forgatókönyveket elhagyják. Így a módszer legértékesebb része, a negyedik lépés vész el, ahol a valódi okok és a tervezési követelmények születnek. Az UCA-k csak a „mit”, a forgatókönyvek a „miért” kérdésére felelnek, és csak a „miért”-ből lesz tervmódosítás.

7.5.1. Erősségek és korlátok

Erősségek: a specifikációs és követelményhibákat, a komponensek közötti kölcsönhatásokat, a szoftver és az ember szerepét egységes keretben kezeli; a koncepciós fázisban is alkalmazható; közvetlenül tervezési követelményeket ad; a szervezeti és szabályozási szinteket is elemzi.

Korlátok: a módszer viszonylag új, a szabványosítása és az iparági gyakorlata még formálódik; nagy rendszereknél az irányítási struktúra és a forgatókönyvek száma gyorsan nő; az eredmény minősége, akárcsak a HAZOP-nál, a csapat tudásán múlik; nem ad számszerű kockázati mérőszámot.

Gyakorlat

A gyakorlati útmutató 5. laborjában egy böngészős STPA-munkalappal dolgozol: felrajzolod egy egyszerű rendszer irányítási struktúráját, kitöltöd az UCA-táblázatot, és minden UCA-hoz legalább két forgatókönyvet és egy követelményt rendelsz. A feladat a VészFék és a KoboCella közül választható.

Összefoglalás

- A szoftverintenzív és ember–automatizálás rendszerekben a balesetek gyakran meghibásodás nélkül, nem biztonságos kölcsönhatásokból és hibás követelményekből fakadnak.
- A STAMP szerint a biztonság emergens tulajdonság, amelyet a hierarchikus irányítási struktúra a biztonsági korlátok érvényesítésével tart fenn; a balesetek oka a nem érvényesülő korlát.
- Az irányítók a folyamatmodelljük alapján döntenek; a téves folyamatmodell a balesetek gyakori oka.
- Az STPA négy lépése: veszteségek, veszélyek és korlátok; irányítási struktúra; nem biztonságos irányítási akciók (négy típus, kontextussal); veszteségforgatókönyvek és követelmények.
- Az STPA kiegészíti a hagyományos módszereket; korán alkalmazható, de nem ad kvantitatív eredményt.

Önellenőrzés

1. Miért nem rendszerszintű veszély az, hogy „a lézerszkenner meghibásodik”? Fogalmazd meg helyesen a mögötte álló veszélyt!
2. Sorold fel az UCA négy típusát, és adj mindegyikre példát egy lift ajtajának „ajtó nyitása” akciójára!
3. Mi a folyamatmodell, és miért fontos a STAMP-ban? Mutasd be egy példán, hogyan vezethet baleshez a téves folyamatmodell!
4. Melyik STPA-lépés adja a tervezési követelményeket, és miért?

Megoldás.

1. Mert alkatrészhiba, nem a rendszer állapota; ráadásul a szkenner meghibásodása nem feltétlenül vezet veszélyhez (lehet biztonságos meghibásodás is). A helyes veszély: „a robotkar a szerelőhöz közel a biztonságos határnál nagyobb sebességgel mozog”. A szkenner meghibásodása ennek egy lehetséges oka a negyedik lépés forgatókönyveiben.
2. *Nem adja ki:* nem nyitja az ajtót, amikor a fülke a szinten áll, és tűz miatt menekülni kell. *Kiadja:* kinyitja az ajtót, amikor a fülke nincs a szinten. *Rossz időben:* túl korán nyit, amikor a fülke még mozog. *Rossz ideig:* túl korán zárja, amikor valaki még az ajtóban áll.
3. A folyamatmodell az irányító képe az irányított folyamat állapotáról (embernél a mentális modell, szoftvernél a belső változók). Az irányító ez alapján dönt, ezért ha a modell eltér a valóságtól, helyes algoritmus mellett is

nem biztonságos döntést hoz. Példa: a 737 MAX MCAS-funkciója egyetlen hibás állásszög-érzékelő jele alapján túl nagynak „hitte” az állásszöget, és ismételten lefelé trimmelte a gépet.
4. A negyedik lépés (veszteségforgatókönyvek), mert ez tárja fel, *miért* következhet be egy nem biztonságos akció, illetve miért nem hajtódik végre a helyes akció. A konkrét okokra (takarás, konfigurációmódosítás, késleltetés, üzemmód-tévesztés) adható tervezési válasz.

Továbbolvasás

- Leveson, N. G. (2011): *Engineering a Safer World.* MIT Press – a STAMP és az STPA elméleti alapjai, nyílt hozzáféréssel (Leveson 2011).
- Leveson, N. G., Thomas, J. P. (2018): *STPA Handbook* – lépésenkénti útmutató sok példával, ingyenesen letölthető (Leveson és Thomas 2018).
- ISO/TS 15066:2016 – a kollaboratív robotok biomechanikai határértékei (ISO 2016).

III. rész – Kvantitatív megbízhatóság és biztonság



Pannon Egyetem
University of Pannonia

8.

Megbízhatóság-elmélet és élettartam-adatok



A III. rész a biztonság számszerűsítésének eszközeit tárgyalja. Ennek alapja a megbízhatóság-elmélet: annak matematikai leírása, hogyan hibásodnak meg az alkatrészek az idő függvényében. Ebben a fejezetben az élettartamot valószínűségi változóként kezeljük, levezetjük a megbízhatósági függvényt, a meghibásodási rátát és az átlagos élettartamot, majd megismerjük a gyakorlatban legfontosabb eloszlásokat (exponenciális, Weibull, lognormális). A fejezet második fele azt mutatja meg, hogyan becsülhetők ezek a jellemzők valós, gyakran hiányos (cenzúrázott) adatokból, és milyen bizonytalansággal. A következő fejezetek (blokkdiagram, hibafa, Markov-modell, PFD-számítás) mind erre az alapra épülnek. A valószínűségszámítási előismereteket az A. függelék foglalja össze.

Tanulási eredmények

A fejezet végére képes leszel

- definiálni és egymásból levezetni az $F(t)$, $R(t)$, $f(t)$, $h(t)$ és $H(t)$ függvényeket, valamint az MTTF-et;
- értelmezni a kádgörbét és a meghibásodási ráta időbeli alakjának következményeit a karbantartásra;
- alkalmazni az exponenciális, a Weibull- és a lognormális eloszlást, és ismerni a feltevéseiket;
- megkülönböztetni az MTTF, MTBF, MTTR és MTTFF fogalmát, és kiszámolni a rendelkezésre állást;
- meghibásodási rátát becsülni terepi adatokból konfidenciakorláttal, és Weibull-eloszlást illeszteni cenzúrázott adatokra.

8.1. Az élettartam mint valószínűségi változó

Tekintsünk egy nem helyreállítható elemet, amely a $t = 0$ időpontban kezd működni, és az első meghibásodásáig működik. A meghibásodásig eltelt idő, a **hibamentes működési idő** τ egy nemnegatív, folytonos valószínűségi változó. Előre nem tudjuk megmondani, egy adott példány mikor hibásodik meg, de a sok azonos példányból álló populáció viselkedése statisztikailag jól leírható.

Az élettartam jellemző függvényei

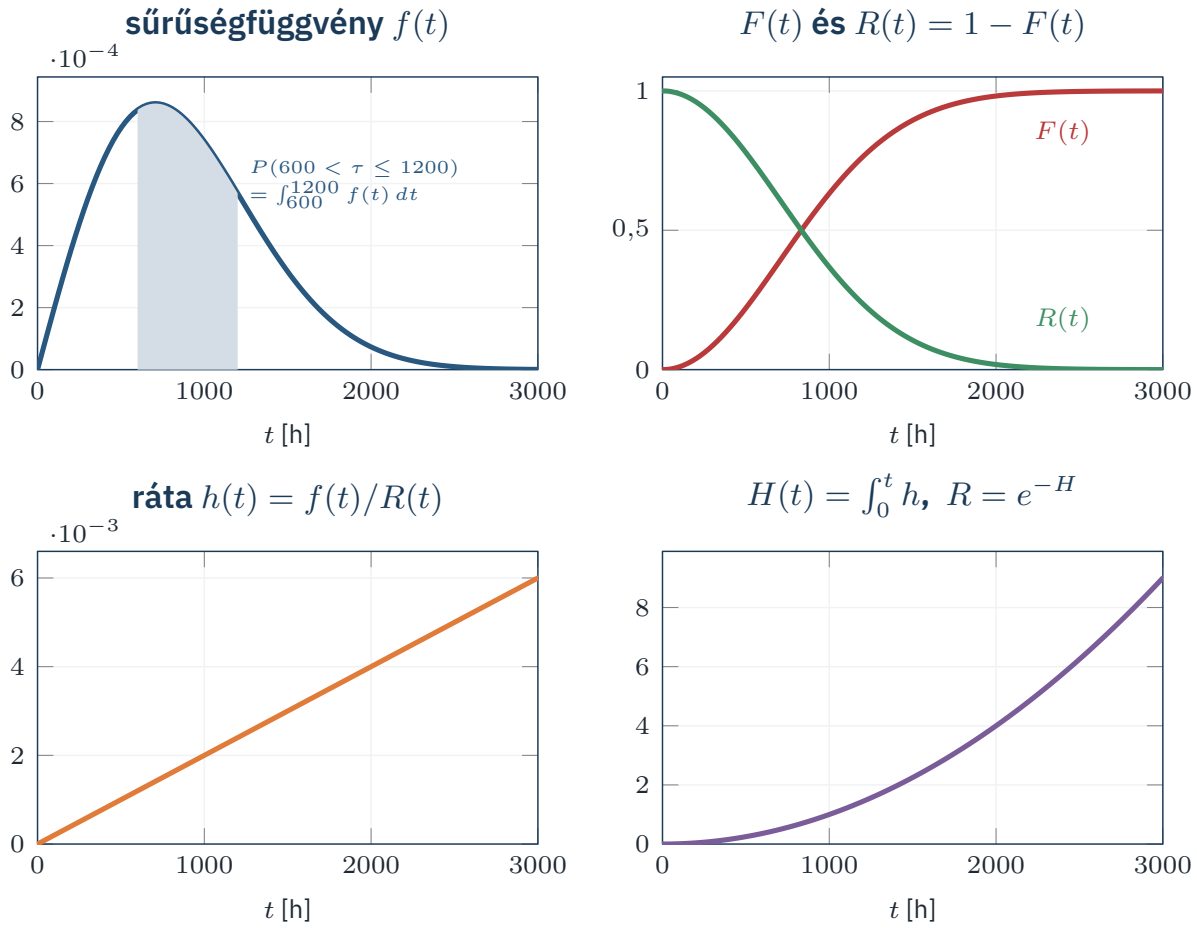
- **Eloszlásfüggvény** (meghibásodási valószínűség): $F(t) = P(\tau \leq t)$, annak valószínűsége, hogy az elem t időpontig meghibásodik.

- **Megbízhatósági függvény** (túlélési függvény): $R(t) = P(\tau > t) = 1 - F(t)$, annak valószínűsége, hogy az elem t időpontig hibamentesen működik.
- **Sűrűségfüggvény:** $f(t) = \dfrac{\mathrm{d}F(t)}{\mathrm{d}t} = -\dfrac{\mathrm{d}R(t)}{\mathrm{d}t}$.

Az IEC 61508 a meghibásodási valószínűséget PF-fel (*probability of failure*) jelöli, a magyar szabványok $Q(t)$-vel; ebben a könyvben az $F(t)$ jelölést használjuk. A sűrűségfüggvény segítségével bármely időintervallumba eső meghibásodás valószínűsége kiszámítható:

$$P(a < \tau \leq b) = \int_a^b f(t)\,\mathrm{d}t = F(b) - F(a) = R(a) - R(b). \tag{8.1}$$

A függvények lefutását a 8.1. ábra mutatja egy Weibull-eloszlású elem példáján. Az $R(t)$ 1-ről indul és monoton csökken nullához; az $F(t)$ ennek tükörképe.



8.1. ábra. Az élettartam jellemző függvényei egy Weibull-eloszlású ($\beta = 2$, $\eta = 1000$ h) elemre.

8.1.1. Az átlagos élettartam

A hibamentes működési idő várható értéke az **átlagos élettartam** (*mean time to failure*, MTTF):

$$\text{MTTF} = E[\tau] = \int_0^\infty t\, f(t)\, \mathrm{d}t. \tag{8.2}$$

Mivel $f(t) = -R'(t)$, parciális integrálással egy sokszor kényelmesebb alakot kapunk:

$$\text{MTTF} = \int_0^\infty t\,(-R'(t))\, \mathrm{d}t = \left[-t\,R(t)\right]_0^\infty + \int_0^\infty R(t)\, \mathrm{d}t = \int_0^\infty R(t)\, \mathrm{d}t, \tag{8.3}$$

mivel véges várható érték esetén $t\,R(t) \to 0$, ha $t \to \infty$. Az MTTF tehát a megbízhatósági görbe alatti terület.

8.1.2. A meghibásodási ráta

A megbízhatóság-elmélet talán legfontosabb fogalma a **meghibásodási ráta** (*failure rate, hazard rate*). Kérdés: ha egy elem t időpontig hibamentesen működött, mekkora a valószínűsége, hogy a következő Δt időben meghibásodik? A feltételes valószínűség definíciója szerint, ha A a $(t, t+\Delta t]$ intervallumban bekövetkező meghibásodás, B pedig a t időpontig tartó hibamentes működés eseménye:

$$P(A \mid B) = \frac{P(A \cap B)}{P(B)} = \frac{P(t < \tau \le t+\Delta t)}{P(\tau > t)} = \frac{R(t) - R(t+\Delta t)}{R(t)}. \tag{8.4}$$

Ezt az időtartammal osztva és $\Delta t \to 0$ határátmenetet képezve kapjuk a meghibásodási rátát:

$$h(t) = \lim_{\Delta t \to 0} \frac{R(t) - R(t+\Delta t)}{\Delta t\, R(t)} = \frac{-R'(t)}{R(t)} = \frac{f(t)}{R(t)}. \tag{8.5}$$

A $h(t)\,\Delta t$ szorzat tehát közelítőleg annak a valószínűsége, hogy egy *addig működő* elem a következő rövid Δt időben meghibásodik. A sűrűségfüggvénytől az különbözteti meg, hogy feltételes: $f(t)\,\Delta t$ az eredeti populációhoz, $h(t)\,\Delta t$ a még működő példányokhoz viszonyít.

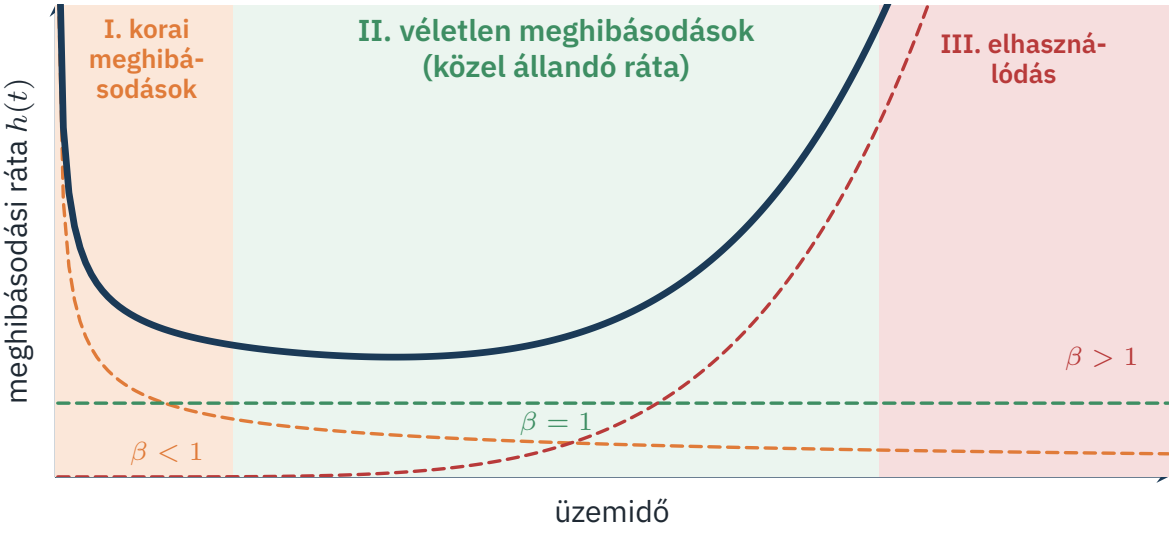
A 8.5. egyenlet egy egyszerű differenciálegyenlet $R(t)$-re: $R'(t) = -h(t)\,R(t)$, $R(0) = 1$. Megoldása

$$R(t) = \exp\left(-\int_0^t h(u)\, \mathrm{d}u\right) = e^{-H(t)}, \tag{8.6}$$

ahol $H(t)$ a **kumulált meghibásodási ráta**. Ez az összefüggés azt mutatja, hogy a meghibásodási ráta egyértelműen meghatározza az élettartam eloszlását: elegendő $h(t)$ alakját ismerni.

8.2. A kádgörbe

Sok alkatrész-populáció meghibásodási rátája az élettartam során a 8.2. ábrán látható „kádgörbe” alakot követi.



8.2. ábra. A kádgörbe: a meghibásodási ráta tipikus időbeli alakja három szakasszal.

- **I. Korai meghibásodások szakasza** (csökkenő ráta). A gyártási hibás, gyenge példányok gyorsan kiesnek, így a ráta csökken. Tipikus okok: nem megfelelő minőségszabályozás vagy gyártási eljárás, gyenge minőségű anyag, szerelési hibák, nem megfelelő hibakeresés, rossz kezelés vagy csomagolás. Ellenszer: beégetés (*burn-in*), környezeti igénybevételi szűrés, szigorú átvételi vizsgálat.
- **II. Véletlen meghibásodások szakasza** (közel állandó ráta). A meghibásodások véletlenszerű, külső igénybevételek (túlterhelés, feszültségtüske, ütés) következményei, amelyek az elem korától függetlenek. Ez a hasznos élettartam szakasza; a biztonsági számítások többsége erre vonatkozik.
- **III. Elhasználódás szakasza** (növekvő ráta). Visszafordíthatatlan fizikai és kémiai folyamatok (kopás, fáradás, korrózió, öregedés) miatt a ráta nő. Tipikus okok: elégtelen karbantartás, súrlódás okozta kopás, anyagfáradás, korrózió, rossz felülvizsgálati gyakorlat. Ellenszer: megelőző csere a ráta meredek növekedése előtt.

A kádgörbe szemléletes modell, de nem minden alkatrészre érvényes. Az elektronikus alkatrészek többségénél a III. szakasz jóval a hasznos élettartam után kezdődne, a II. szakasz dominál. A mechanikus alkatrészeknél (csapágyak, tömítések, szelepek) viszont gyakran nincs valódi állandó szakasz, az elhasználódás már korán megjelenik. A szoftverre a kádgörbe egyáltalán nem értelmezhető, mert a szoftver nem öregszik (15. fejezet).

A meghibásodási ráta alakja meghatározza a karbantartási stratégiát. **Megelőző csere csak növekvő meghibásodási ráta esetén** csökkenti a meghibásodások számát. Állandó ráta mellett a csere hatástalan (az új elem ugyanakkora eséllyel hibásodik meg, mint a régi), csökkenő ráta mellett pedig kifejezetten káros, mert friss, nagyobb kockázatú példányt tesz a jól bejáratott helyére.

8.3. Élettartam-eloszlások

8.3.1. Az exponenciális eloszlás

Ha a meghibásodási ráta állandó, $h(t) = \lambda$, akkor a 8.6. egyenletből

$$R(t) = e^{-\lambda t}, \qquad F(t) = 1 - e^{-\lambda t}, \qquad f(t) = \lambda e^{-\lambda t}, \qquad t \geq 0. \tag{8.7}$$

Az átlagos élettartam és a szórásnégyzet:

$$\mathrm{MTTF} = \int_0^\infty e^{-\lambda t}\,\mathrm{d}t = \frac{1}{\lambda}, \qquad D^2(\tau) = \frac{1}{\lambda^2}. \tag{8.8}$$

A meghibásodási ráta tehát az átlagos élettartam reciproka. Érdemes megjegyezni, hogy $R(\mathrm{MTTF}) = e^{-1} \approx 0{,}368$: az elemek csaknem kétharmada meghibásodik az átlagos élettartam eléréséig.

Az exponenciális eloszlás legfontosabb tulajdonsága az **emlékezetnélküliség**:

$$P(\tau > t + s \mid \tau > t) = \frac{e^{-\lambda(t+s)}}{e^{-\lambda t}} = e^{-\lambda s} = P(\tau > s). \tag{8.9}$$

Egy t ideig hibátlanul működött elem további élettartama ugyanolyan eloszlású, mint egy újé: az elem „nem öregszik”. Ez a II. szakasz véletlen meghibásodásainak értelmezése. A tulajdonság kényelmes (a számítások egyszerűvé válnak, és a Markov-modellek erre épülnek), de alkalmazása mindig feltevés, amelyet ellenőrizni kell.

Az exponenciális eloszlás az elektronikai iparban szokásos FIT-egységgel is kifejezhető: 1 FIT egy meghibásodás 10^9 óránként. A TT-101 távadó $\lambda_D = 1000$ FIT veszélyes meghibásodási rátája (6.5. szakasz) tehát $10^{-6}\,\mathrm{h}^{-1}$, ami átlagosan 114 évenként egy veszélyes meghibásodást jelent egy példányra. Ez nem jelenti azt, hogy egy távadó 114 évig működik (az elhasználódás jóval korábban jelentkezik), hanem azt, hogy egy nagy populációban a hasznos élettartam alatt ilyen gyakorisággal jelentkeznek véletlen meghibásodások.

8.3.2. A Weibull-eloszlás

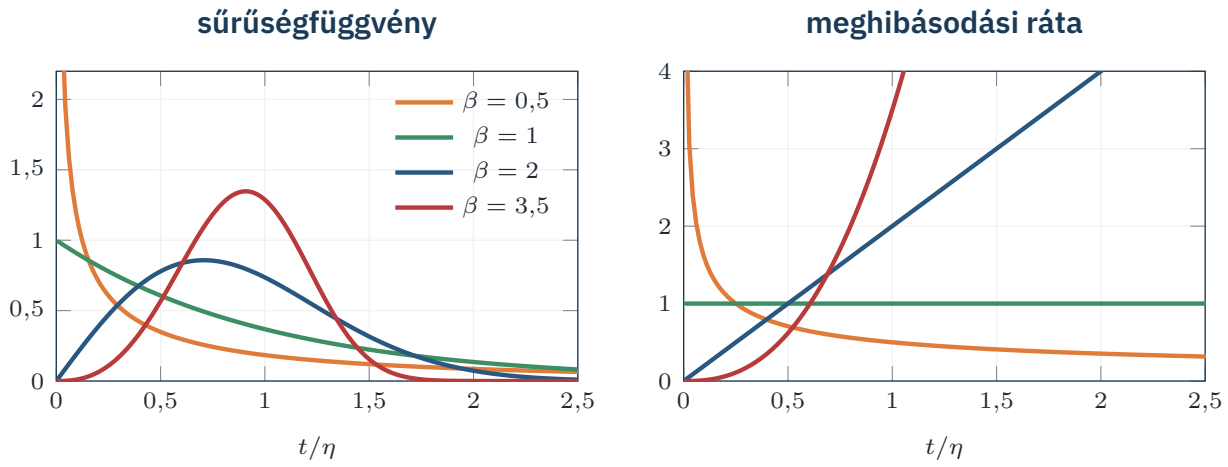
A megbízhatósági gyakorlatban, különösen a mechanikus alkatrészek és a karbantartás-tervezés terén, a leggyakrabban alkalmazott eloszlás a Waloddi Weibull svéd mérnökről elnevezett eloszlás (Weibull 1951):

$$F(t) = 1 - \exp\left[-\left(\frac{t}{\eta}\right)^{\beta}\right], \qquad h(t) = \frac{\beta}{\eta}\left(\frac{t}{\eta}\right)^{\beta-1}, \qquad t \geq 0, \tag{8.10}$$

ahol $\eta > 0$ a **skálaparaméter** (karakterisztikus élettartam: $F(\eta) = 1 - e^{-1} \approx 63{,}2\%$), $\beta > 0$ az **alakparaméter**. A 2014-es jegyzet az $F(t) = 1 - e^{-at^b}$ alakot használta; a kettő megfeleltetése $b = \beta$, $a = \eta^{-\beta}$.

A meghibásodási ráta hatványfüggvény (8.3. ábra): $\beta < 1$ esetén monoton csökkenő (korai meghibásodások), $\beta = 1$ esetén állandó (exponenciális eloszlás), $\beta > 1$ esetén monoton növekvő

(elhasználódás). A Weibull-eloszlás így a kádgörbe mindhárom szakaszát leírja; $\beta \approx 3{,}5$ körül alakja a normális eloszláshoz hasonló.



8.3. ábra. A Weibull-eloszlás sűrűségfüggvénye és meghibásodási rátája különböző alakparaméterek mellett.

Az átlagos élettartam és a szórásnégyzet az Euler-féle gammafüggvénnyel (Γ) fejezhető ki:

$$\mathrm{MTTF} = \eta\,\Gamma\left(1+\frac{1}{\beta}\right), \qquad D^2(\tau) = \eta^2\left[\Gamma\left(1+\frac{2}{\beta}\right) - \Gamma^2\left(1+\frac{1}{\beta}\right)\right]. \tag{8.11}$$

A gyakorlatban hasznos a B_p élettartam: az az idő, amely alatt a populáció p százaléka meghibásodik. Weibull-eloszlásnál $B_p = \eta\,[-\ln(1-p/100)]^{1/\beta}$. A csapágyiparban például a B_{10} élettartam a szokásos méretezési jellemző.

8.3.3. További eloszlások

- A **lognormális eloszlás** ($\ln\tau$ normális eloszlású) jól írja le a javítási időket és bizonyos fáradásos, repedésterjedéses meghibásodásokat. Meghibásodási rátája először nő, majd csökken.
- A **normális eloszlás** a kifejezetten elhasználódásos meghibásodásokra (például kopó alkatrészek) alkalmazható, ha a szórás kicsi az átlaghoz képest.
- A **gammaeloszlás** egy olyan rendszer élettartamát írja le, amely k egymást követő, exponenciális eloszlású „sokk” után hibásodik meg (például hidegtartalékos rendszerek, 9. fejezet).

8.1. táblázat. A fontosabb élettartam-eloszlások

Eloszlás	$h(t)$ alakja	Tipikus alkalmazás
exponenciális	állandó	elektronika a hasznos élettartamban; biztonsági számítások alapfeltevése
Weibull, $\beta < 1$	csökkenő	korai meghibásodások, gyártási hibák
Weibull, $\beta > 1$	növekvő	kopás, fáradás, öregedés; karbantartás-tervezés
lognormális	nő, majd csökken	javítási idők, repedésterjedés

Eloszlás	$h(t)$ alakja	Tipikus alkalmazás
normális	gyorsan növekvő	erősen elhasználódásos alkatrészek
gamma	$k > 1$ esetén növekvő	hidegtartalékos rendszerek

8.4. Javítható rendszerek jellemzői

A javítható (helyreállítható) rendszerek meghibásodás után javítással újra üzembe állnak. Élettörténetük működési és javítási szakaszok váltakozása. Ennek jellemzésére a következő, gyakran összekevert mennyiségek szolgálnak:

- **MTTF** (*mean time to failure*): az átlagos hibamentes működési idő (javítható rendszernél: két meghibásodás között, ha a javítás „újszerű” állapotot állít helyre);
- **MTTFF** (*mean time to first failure*): az első meghibásodásig eltelt átlagos idő; redundáns rendszereknél eltérhet az MTTF-től;
- **MTTR** (*mean time to restoration/repair*): az átlagos helyreállítási idő, amely magában foglalja a hiba észlelését, a hibakeresést, az alkatrész beszerzését, a javítást és az újraindítást;
- **MTBF** (*mean time between failures*): két egymást követő meghibásodás közötti átlagos idő, a javítással együtt: $\mathrm{MTBF} = \mathrm{MTTF} + \mathrm{MTTR}$.

A gyakorlatban, különösen az alkatrész-adatlapokon, az MTBF-et gyakran az MTTF szinonimájaként használják, mert a javítási idő elhanyagolható az üzemidőhöz képest. Mindig érdemes ellenőrizni, melyik definíciót alkalmazzák.

A meghibásodási és helyreállítási lépésekből álló folyamat egyik alapvető jellemzője a **készenléti tényező** vagy **rendelkezésre állás**: annak valószínűsége, hogy a rendszer egy tetszőleges, kellően késői időpontban működik. Ha a működési idők átlaga $T_1 = \mathrm{MTTF}$, a helyreállítási idők átlaga $T_2 = \mathrm{MTTR}$, akkor a stacionárius rendelkezésre állás

$$A = \frac{\mathrm{MTTF}}{\mathrm{MTTF} + \mathrm{MTTR}}. \tag{8.12}$$

Ezt a 12. fejezetben Markov-modellből is levezetjük. Fontos ismét hangsúlyozni: a megbízhatóság a t időpontig *folyamatosan* helyes működést, a rendelkezésre állás a t *időpontban* helyes működést fejezi ki.

8.5. Megbízhatósági adatok és becslésük

8.5.1. Adatforrások

A megbízhatósági számítások bemenő adatai három fő forrásból származnak:

- **Alkatrész-adatbázisok és -kézikönyvek:** például a Siemens SN 29500, az IEC 61709 (az alkatrészek referenciakörülményei és a terhelésfüggés modelljei), a francia FIDES-módszertan, a folyamatipari OREDA-kézikönyv (offshore berendezések terepi adatai) vagy a SINTEF PDS adatkézikönyve. A régi amerikai katonai kézikönyvet (MIL-HDBK-217F) ma is idézik, de elavultsága miatt erősen kritizálják: a modern alkatrészekre gyakran nagyságrendekkel pesszimistább becslést ad.
- **Gyártói adatok:** tanúsítványok, FMEDA-jelentések, gyorsított élettartam-vizsgálatok eredményei.

- **Terepi adatok:** a saját üzem karbantartási és meghibásodási nyilvántartása. Ezek a legrelevánsabbak, de gyakran hiányosak (nem rögzítik a meghibásodási módot, az üzemidőt vagy a környezetet), és a populáció kicsi.

Az adatok bizonytalansága nagy: ugyanarra az alkatrésztípusra a különböző források gyakran egy nagyságrendnyi eltérést adnak. Ezért a jó gyakorlat az adatforrás dokumentálása, a konzervatív (felső korlát) értékek használata és az érzékenységvizsgálat.

8.5.2. A meghibásodási ráta becslése terepi adatokból

Állandó meghibásodási rátát feltételezve, ha egy populáció összesen T üzemórát teljesített, és ezalatt r meghibásodás történt, a meghibásodási ráta maximum likelihood (ML) becslése

$$\hat{\lambda} = \frac{r}{T}. \tag{8.13}$$

A pontbecslés önmagában félrevezető lehet, ha kevés a meghibásodás. Időre csonkított megfigyelés esetén (a megfigyelés egy előre rögzített időpontig tart) a λ egyoldalú, $1-\alpha$ szintű felső konfidenciakorlátja a khi-négyzet eloszlással fejezhető ki (Rausand és mtsai. 2020):

$$\lambda_U = \frac{\chi^2_{1-\alpha}(2r+2)}{2T}, \tag{8.14}$$

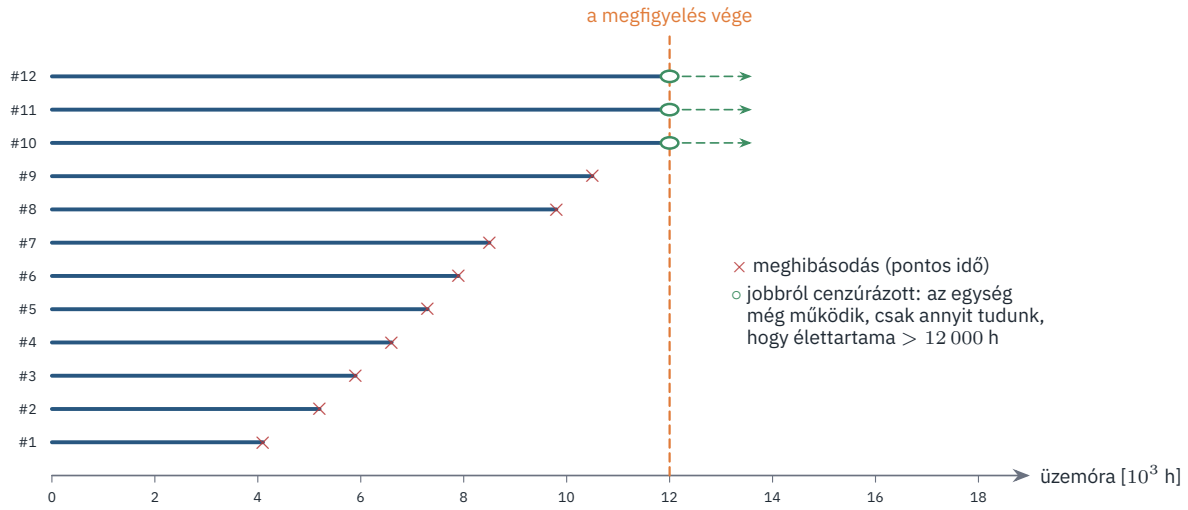
ahol $\chi^2_{1-\alpha}(\nu)$ a ν szabadsági fokú khi-négyzet eloszlás $1-\alpha$ kvantilise. Ha egyáltalán nem volt meghibásodás ($r=0$), a felső korlát $\lambda_U = -\ln\alpha/T$; 95%-os szinten ez $\approx 3/T$ („a hármas szabály”).

> **ReaktorVéd: a távadók terepi adatai**
>
> Az üzemeltető vállalat több telephelyén 250 azonos típusú hőmérséklet-távadó működik biztonsági funkcióban. Négy év alatt a proof tesztek három veszélyes, rejtett meghibásodást tártak fel (mindhárom az érzékelőelem lefelé irányuló driftje volt). Az összesített üzemidő $T = 250 \cdot 4 \cdot 8760 = 8{,}76 \cdot 10^6$ óra.
>
> $$\hat{\lambda}_{DU} = \frac{3}{8{,}76 \cdot 10^6} = 3{,}4 \cdot 10^{-7}\,\mathrm{h}^{-1}.$$
>
> A 70%-os felső konfidenciakorlát (az IEC 61508 a terepi adatokon alapuló, úgynevezett 2H útvonalon legalább 70%-os konfidenciaszintet ír elő) $\chi^2_{0{,}7}(8) = 9{,}52$ felhasználásával
>
> $$\lambda_{DU,U} = \frac{9{,}52}{2 \cdot 8{,}76 \cdot 10^6} = 5{,}4 \cdot 10^{-7}\,\mathrm{h}^{-1},$$
>
> a 90%-os felső korlát $7{,}6 \cdot 10^{-7}\,\mathrm{h}^{-1}$. A gyártói FMEDA $4 \cdot 10^{-7}\,\mathrm{h}^{-1}$ értéke (6.5. szakasz) a terepi pontbecslés és a 70%-os korlát közé esik, tehát a terepi tapasztalat alátámasztja. Ha a terepi adat jóval kedvezőtlenebb lett volna, a SIL-igazolást a terepi értékkel kellett volna újraszámolni.

8.5.3. Cenzúrázott adatok

Az élettartam-vizsgálatok és a terepi megfigyelések ritkán érnek véget úgy, hogy minden példány meghibásodott. Gyakori, hogy néhány egység a megfigyelés végén még működik, vagy más okból (például megelőző csere miatt) kikerül a megfigyelésből. Ezeknek az egységeknek nem ismerjük az

élettartamát, csak azt, hogy *nagyobb* egy bizonyos értéknél. Az ilyen adatot **jobbról cenzúrázottnak** nevezzük (8.4. ábra).



8.4. ábra. Cenzúrázott élettartam-adatok: kilenc egység meghibásodott, három a megfigyelés végén még működik.

A cenzúrázott adatok elhagyása vagy meghibásodásként kezelése egyaránt súlyos torzítást okoz: az előbbi a legjobb egységeket hagyja ki (az élettartamot alulbecsüli), az utóbbi meghibásodást tételez fel ott, ahol nem volt. A helyes kezeléshez két módszercsalád áll rendelkezésre:

- **Nemparaméteres becslés:** a Kaplan–Meier-becslő (Kaplan és Meier 1958) a megbízhatósági függvényt lépcsős függvénnyel becsli, minden meghibásodási időpontban a még megfigyelés alatt álló egységek számával korrigálva: $\hat{R}(t) = \prod_{t_i \le t} (1 - d_i/n_i)$, ahol d_i a t_i időpontbeli meghibásodások, n_i az ekkor még kockázatnak kitett egységek száma.
- **Paraméteres becslés:** egy feltételezett eloszlás (például Weibull) paramétereit a maximum likelihood módszerrel becsüljük, amelyben a meghibásodott egységek a sűrűségfüggvénnyel, a cenzúrázottak a megbízhatósági függvénnyel járulnak hozzá a likelihoodhoz:

$$L(\beta, \eta) = \prod_{i \in \text{meghib.}} f(t_i; \beta, \eta) \prod_{j \in \text{cenz.}} R(t_j; \beta, \eta). \tag{8.15}$$

KoboCella: a robothajtások csapágyainak élettartama

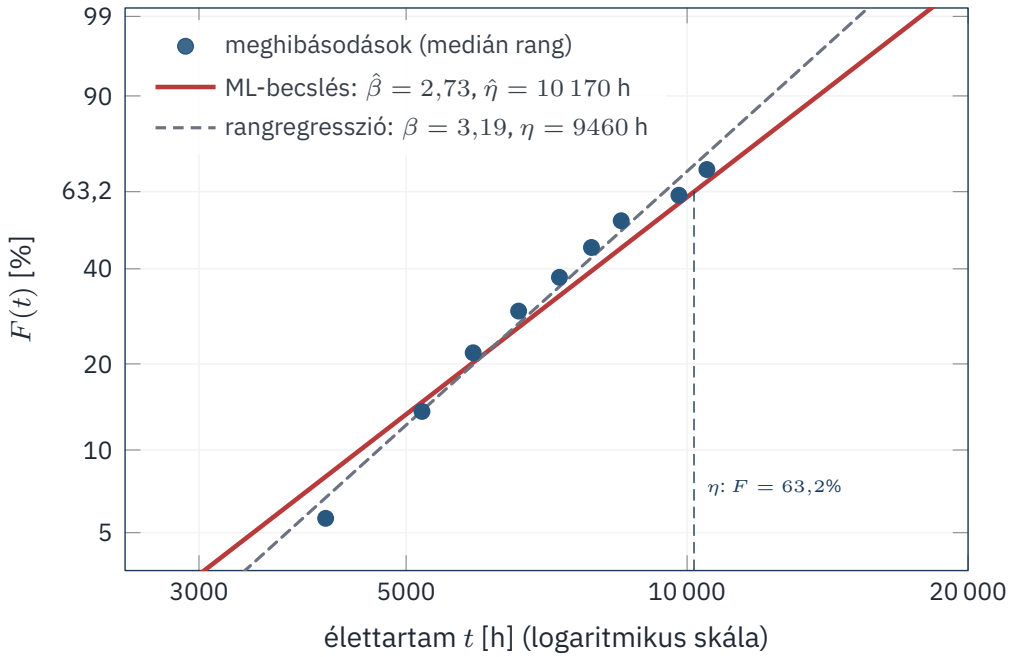
A KoboCella típusú robotok egyik hajtásának csapágyairól egy flottafigyelő rendszer 12 adatot gyűjtött (8.4. ábra): kilenc csapágy 4100 és 10 500 üzemóra között meghibásodott, három 12 000 óránál még működött. A Weibull-likelihood numerikus maximalizálása (lásd a gyakorlati útmutató notebookját) a következő becslést adja:

$$\hat{\beta} = 2{,}73, \qquad \hat{\eta} = 10\,170\ \text{h}, \qquad \widehat{\text{MTTF}} = \hat{\eta}\,\Gamma(1 + 1/\hat{\beta}) \approx 9050\ \text{h}.$$

Az alakparaméter egyértelműen 1 fölött van, tehát elhasználódásos meghibásodásról van szó, és a megelőző csere értelmes stratégia. A B_{10} élettartam $\hat{\eta}\,(-\ln 0{,}9)^{1/\hat{\beta}} \approx 4460$ óra: ennyi idő alatt a csapágyak 10%-a hibásodik meg. Egy 6000 órát már teljesített csapágy további 2000 órás túlélési valószínűsége

$$R(8000 \mid 6000) = \frac{R(8000)}{R(6000)} \approx 0{,}75,$$

szemben egy új csapágy $R(2000) \approx 0{,}99$ értékével. Az exponenciális eloszlás emlékezetnélkülisége itt egyértelműen nem teljesül.
A becslés grafikus ellenőrzésére a **Weibull-papír** szolgál (8.5. ábra): az $x = \ln t$, $y = \ln[-\ln(1-F)]$ koordinátarendszerben a Weibull-eloszlás egyenes, amelynek meredeksége β. A pontok a medián rangok (Benard-közelítés: $F_i \approx (i-0{,}3)/(n+0{,}4)$) alapján kerülnek az ábrára. A rangregresszió ($\beta = 3{,}19$) és az ML-becslés ($\beta = 2{,}73$) eltérése jól mutatja, hogy kis mintánál a módszerválasztás érdemben befolyásolja az eredményt. Cenzúrázott adatoknál a likelihood módszer az elméletileg megalapozottabb (Meeker és Escobar 1998; Abernethy 2006).



8.5. ábra. A csapágyadatok Weibull-papíron. A pontok a meghibásodott egységek medián rangjai; a vonalak a két becslési módszer eredményei.

Az MTBF mint garantált élettartam

Gyakori félreértés, hogy egy „100 000 órás MTBF-ű” eszköz 100 000 órát (11 évet) működik. Az MTBF egy *populáció* átlagos jellemzője a hasznos élettartam alatt, állandó meghibásodási rátát feltételezve. Exponenciális eloszlásnál az eszközök 63%-a meghibásodik az MTBF eltelte előtt, és az elhasználódás (amelyet az MTBF nem tartalmaz) a valóságban jóval korábban jelentkezhet. Egy 5 éves tervezett élettartamú eszköznél az MTBF csak arról tájékoztat, hogy e 5 év alatt egy nagy populációban mekkora gyakorisággal várhatók véletlen meghibásodások.

Gyakorlat

A gyakorlati útmutató 6. laborjában egy interaktív alkalmazással vizsgálhatod az exponenciális, a Weibull- és a lognormális eloszlás függvényeit, és csúszkákkal figyelheted meg, hogyan változik a kádgörbe. A hozzá tartozó Colab-notebook a fejezet két esetét (terepi λ-becslés, Weibull-illesztés cenzúrázott adatokra) számolja végig, és bootstrap-módszerrel konfidenciaintervallumot is ad a Weibull-paraméterekre.

Összefoglalás

- Az élettartam valószínűségi változó; jellemző függvényei az $F(t)$, $R(t) = 1 - F(t)$, $f(t)$, valamint a feltételes meghibásodási ráta $h(t) = f(t)/R(t)$. Az $R(t) = \exp(-\int_0^t h)$ összefüggés miatt a ráta egyértelműen meghatározza az eloszlást.
- Az átlagos élettartam a megbízhatósági görbe alatti terület: $\mathrm{MTTF} = \int_0^\infty R(t)\,\mathrm{d}t$.
- A kádgörbe három szakasza: korai (csökkenő), véletlen (állandó) és elhasználódásos (növekvő) meghibásodások. Megelőző csere csak növekvő ráta esetén hasznos.
- Az exponenciális eloszlás emlékezetnélküli, $\mathrm{MTTF} = 1/\lambda$; a Weibull-eloszlás alakparamétere a kádgörbe mindhárom szakaszát leírja.
- Javítható rendszereknél $A = \mathrm{MTTF}/(\mathrm{MTTF} + \mathrm{MTTR})$; az MTBF nem garantált élettartam.
- A terepi adatokból becsült ráta mellé konfidenciakorlátot kell adni; a cenzúrázott adatokat a likelihoodban a megbízhatósági függvénnyel kell figyelembe venni.

Önellenőrzés

1. Mutasd meg, hogy exponenciális eloszlásnál a meghibásodási ráta állandó!
2. Egy elem élettartama exponenciális, $\lambda = 2 \cdot 10^{-5}\,\mathrm{h}^{-1}$. Mekkora a valószínűsége, hogy egy év (8760 h) alatt meghibásodik? Mekkora az MTTF?
3. Egy Weibull-eloszlású alkatrésznél $\beta = 0{,}7$. Érdemes-e megelőzően cserélni? Miért?
4. Egy 40 eszközből álló populáció 5 év alatt egyetlen meghibásodást sem produkált. Adj 95%-os felső korlátot a meghibásodási rátára!
5. Miért torzít, ha a cenzúrázott egységeket egyszerűen elhagyjuk az elemzésből?

Megoldás.

1. $h(t) = f(t)/R(t) = \lambda e^{-\lambda t}/e^{-\lambda t} = \lambda$.
2. $F(8760) = 1 - e^{-2\cdot 10^{-5}\cdot 8760} = 1 - e^{-0{,}1752} \approx 0{,}161$, azaz 16,1%. $\mathrm{MTTF} = 1/\lambda = 50\,000$ óra.
3. Nem érdemes. $\beta < 1$ csökkenő meghibásodási rátát jelent (korai meghibásodások): az új alkatrész nagyobb eséllyel hibásodik meg, mint a már bejáratott. A megelőző csere növelné a meghibásodások számát; helyette a beégetés vagy a gyártási minőség javítása segít.
4. $T = 40 \cdot 5 \cdot 8760 = 1{,}752 \cdot 10^6$ óra, $\lambda_U = -\ln 0{,}05/T \approx 3{,}0/1{,}752 \cdot 10^6 \approx 1{,}7 \cdot 10^{-6}\,\mathrm{h}^{-1}$.
5. Mert a cenzúrázott egységek éppen a hosszabb élettartamúak (a megfigyelés végéig kitartottak). Elhagyásuk a mintát a rövidebb élettartamok felé torzítja, így az átlagos élettartamot alulbecsüljük, a meghibásodási rátát túlbecsüljük.

Továbbolvasás

- Rausand, M., Barros, A., Høyland, A. (2020): *System Reliability Theory.* 3. kiadás, Wiley – a megbízhatóság-elmélet átfogó tankönyve (Rausand és mtsai. 2020).
- Meeker, W. Q., Escobar, L. A. (1998): *Statistical Methods for Reliability Data.* Wiley – az élettartam-adatok statisztikai elemzésének alapműve (Meeker és Escobar 1998).
- Modarres, M., Kaminskiy, M., Krivtsov, V. (2016): *Reliability Engineering and Risk Analysis.* 3. kiadás, CRC Press (Modarres és mtsai. 2016).

9.

Rendszermegbízhatóság: blokkdiagramok és k-ból-n rendszerek



Az előző fejezet egyetlen elem élettartamát írta le. A valódi rendszerek sok elemből állnak, és a rendszer megbízhatósága attól függ, hogyan kapcsolódnak egymáshoz az elemek *a működés logikája szerint*. A megbízhatósági blokkdiagram (*reliability block diagram*, RBD) ennek a logikának a legszemléletesebb leírása. Ebben a fejezetben megismerjük a blokkdiagram és a struktúrafüggvény fogalmát, a soros és párhuzamos rendszerek összefüggéseit, a redundáns rendszerek legfontosabb családját, a k-ból-n (koon) rendszereket, a nem soros-párhuzamos rendszerek kezelését, a hideg és meleg tartalékolást, végül a bizonytalan komponensadatok hatását a rendszer megbízhatóságára.

Tanulási eredmények

A fejezet végére képes leszel

- megbízhatósági blokkdiagramot és struktúrafüggvényt felírni egy rendszerre;
- kiszámolni soros, párhuzamos és ezekből összetett rendszerek megbízhatóságát és MTTF-jét;
- a k-ból-n rendszerek megbízhatóságát binomiális modellel számolni, és értelmezni a biztonsági rendszerek 1oo2, 2oo2 és 2oo3 architektúráit;
- nem soros-párhuzamos rendszereket állapotfelsorolással, dekompozícióval és minimális utakkal kezelni;
- összevetni a meleg és a hideg tartalékolást;
- a komponensadatok bizonytalanságát intervallumbecsléssel és Monte Carlo-szimulációval a rendszer szintjére vinni.

9.1. A megbízhatósági blokkdiagram

A megbízhatósági blokkdiagram a rendszer *helyes működésének* grafikus, logikai modellje (IEC 61078 (IEC 2016a)). Minden blokk egy elemet vagy funkciót jelöl, amely vagy működik, vagy nem. A diagramnak egy bemenete és egy kimenete van; a rendszer akkor működik, ha a bemenettől a kimenetig van olyan útvonal, amelynek minden blokkja működik.

Az RBD szemléletes, de három fontos feltevésre épül:

1. **Két állapot:** minden elem vagy működik, vagy meghibásodott; a részleges működés vagy a degradáció nem ábrázolható.
2. **Függetlenség:** az elemek meghibásodásai egymástól függetlenek, és egy elem meghibásodása nem változtatja meg a többi elem megbízhatósági jellemzőit. A közös okú meghibásodásokat külön kell modellezni (10. és 13. fejezet).
3. **Statikus logika:** a meghibásodások sorrendje nem számít. A sorrendfüggő viselkedés (például a tartalék csak a fő elem kiesése után indul) Markov-modellt igényel (12. fejezet).

> **A blokkdiagram nem kapcsolási rajz**
>
> A megbízhatósági blokkdiagram a *logikai* kapcsolatot ábrázolja, nem a fizikait. Két, villamosan párhuzamosan kötött ellenállás megbízhatósági szempontból soros lehet (ha bármelyik szakadása a funkció elvesztését okozza), és két sorba kötött szelep megbízhatósági szempontból párhuzamos lehet (ha bármelyik zárása elég az áramlás megszüntetéséhez). Ráadásul ugyanannak a rendszernek más-más blokkdiagramja lehet különböző funkciókra vagy meghibásodási módokra: a sorba kötött szelepek a „zárás” funkcióra párhuzamosak, a „nyitás” funkcióra sorosak.

9.1.1. A struktúrafüggvény

A blokkdiagram matematikai megfelelője a **struktúrafüggvény**. Jelölje x_i az i-edik elem állapotát ($x_i = 1$, ha működik, $x_i = 0$, ha nem), és $\phi(\mathbf{x})$ a rendszer állapotát. Soros és párhuzamos rendszerre:

$$\phi_{\text{soros}}(\mathbf{x}) = \prod_{i=1}^{n} x_i, \qquad \phi_{\text{párh.}}(\mathbf{x}) = 1 - \prod_{i=1}^{n} (1 - x_i). \tag{9.1}$$

Ha az elemek függetlenek, és az i-edik elem megbízhatósága $p_i = P(x_i = 1)$, akkor a rendszer megbízhatósága a struktúrafüggvény várható értéke: $R = E[\phi(\mathbf{x})]$. A gyakorlatban értelmes rendszerek **koherensek**: minden elem számít (van olyan állapot, amelyben az elem állapota dönti el a rendszerét), és egy elem javulása nem ronthatja a rendszert.

9.2. Soros és párhuzamos rendszerek

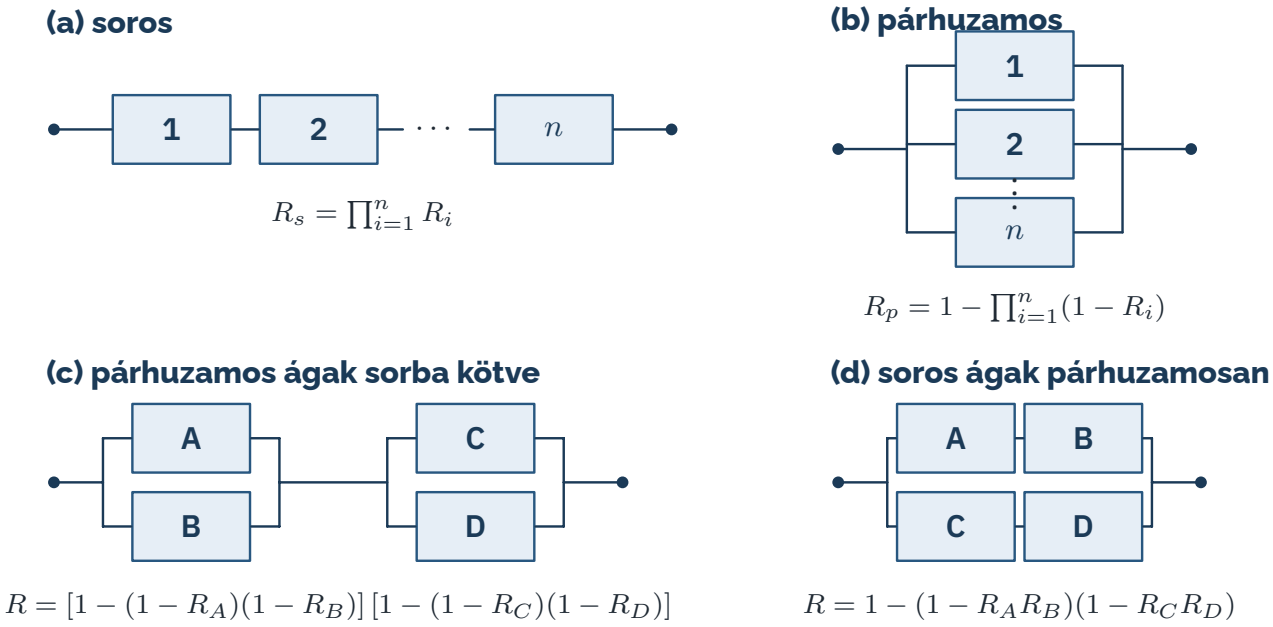
A 9.1. ábra az alapvető struktúrákat és megbízhatóságukat mutatja.

A **soros rendszer** akkor működik, ha minden eleme működik; megbízhatósága az elemek megbízhatóságának szorzata, ezért az elemek számának növelésével csökken. A **párhuzamos rendszer** akkor hibásodik meg, ha minden eleme meghibásodott; meghibásodási valószínűsége az elemek meghibásodási valószínűségeinek szorzata, ezért az elemek számával nő a megbízhatósága.

9.2.1. Exponenciális elemekből álló rendszerek

Ha az elemek élettartama exponenciális, a soros rendszer élettartama is exponenciális, a ráták összegével:

$$R_s(t) = \prod_{i=1}^{n} e^{-\lambda_i t} = e^{-(\sum_i \lambda_i) t}, \qquad \text{MTTF}_s = \frac{1}{\sum_i \lambda_i}. \tag{9.2}$$



9.1. ábra. Az alapvető blokkdiagram-struktúrák és megbízhatóságuk független elemek esetén.

Ez a „ráták összeadása” az alkatrészszámláláson alapuló megbízhatósági előrejelzés (*parts count*) alapja: egy elektronikai modul meghibásodási rátája közelítőleg az alkatrészei rátáinak összege.

A párhuzamos rendszer élettartama már nem exponenciális. n azonos, λ rátájú elem esetén

$$R_p(t) = 1 - \left(1 - e^{-\lambda t}\right)^n, \qquad \mathrm{MTTF}_p = \frac{1}{\lambda}\sum_{k=1}^{n}\frac{1}{k}. \tag{9.3}$$

Két elemre $\mathrm{MTTF}_p = 1{,}5/\lambda$, háromra $1{,}83/\lambda$. A redundancia tehát az MTTF-et csak mérsékelten növeli; a valódi előnye a megbízhatóság növekedése a küldetés elején, amikor $\lambda t \ll 1$: ekkor $1 - R_p \approx (\lambda t)^n$, vagyis a meghibásodási valószínűség az egycsatornás érték n-edik hatványára csökken.

9.2.2. Összetett rendszerek

A soros és párhuzamos részekből összetett rendszert lépésenként egyszerűsítjük: kiszámoljuk a legegyszerűbb soros vagy párhuzamos részek megbízhatóságát, ezeket egy-egy blokkal helyettesítjük, és addig folytatjuk, amíg egyetlen blokk marad.

Soros-párhuzamos rendszer

Egy rendszer akkor működik, ha az $R_3 = 0{,}95$ megbízhatóságú elem mellett az $R_1 = 0{,}8$ és az $R_2 = 0{,}9$ megbízhatóságú elemek közül legalább az egyik működik. A párhuzamos rész megbízhatósága $1 - 0{,}2 \cdot 0{,}1 = 0{,}98$, a rendszeré

$$R = 0{,}95 \cdot 0{,}98 = 0{,}931.$$

Ugyanez az eredmény az **állapotfelsorolással** (igazságtáblával) is megkapható: a $2^3 = 8$ lehetséges állapotból azokat választjuk ki, amelyekben a rendszer működik, és összegezzük a valószínűségüket.

9.1. táblázat. Állapotfelsorolás: a rendszer működő állapotai

x_1	x_2	x_3	rendszer	állapot valószínűsége
1	0	1	működik	$0{,}8 \cdot 0{,}1 \cdot 0{,}95 = 0{,}076$
0	1	1	működik	$0{,}2 \cdot 0{,}9 \cdot 0{,}95 = 0{,}171$
1	1	1	működik	$0{,}8 \cdot 0{,}9 \cdot 0{,}95 = 0{,}684$
			összesen	**0,931**

Az állapotfelsorolás minden kétállapotú, független elemekből álló rendszerre alkalmazható, de n elemnél 2^n állapotot kell megvizsgálni, ezért nagy rendszerekre csak számítógéppel használható.

9.3. k-ból-n rendszerek

A redundáns rendszerek legfontosabb családja a **k-ból-n rendszer** (*k-out-of-n*, koon): n azonos funkciójú csatornából legalább k-nak működnie kell a rendszer működéséhez. A soros rendszer az n-ből-n, a párhuzamos az 1-ből-n speciális esete. Ha a csatornák függetlenek és azonos p megbízhatóságúak, a működő csatornák száma binomiális eloszlású, és

$$R_{koon} = \sum_{j=k}^{n} \binom{n}{j} p^j (1-p)^{n-j}. \tag{9.4}$$

A leggyakoribb esetek:

$$R_{1\text{oo}2} = 2p - p^2, \qquad R_{2\text{oo}2} = p^2, \qquad R_{2\text{oo}3} = 3p^2 - 2p^3. \tag{9.5}$$

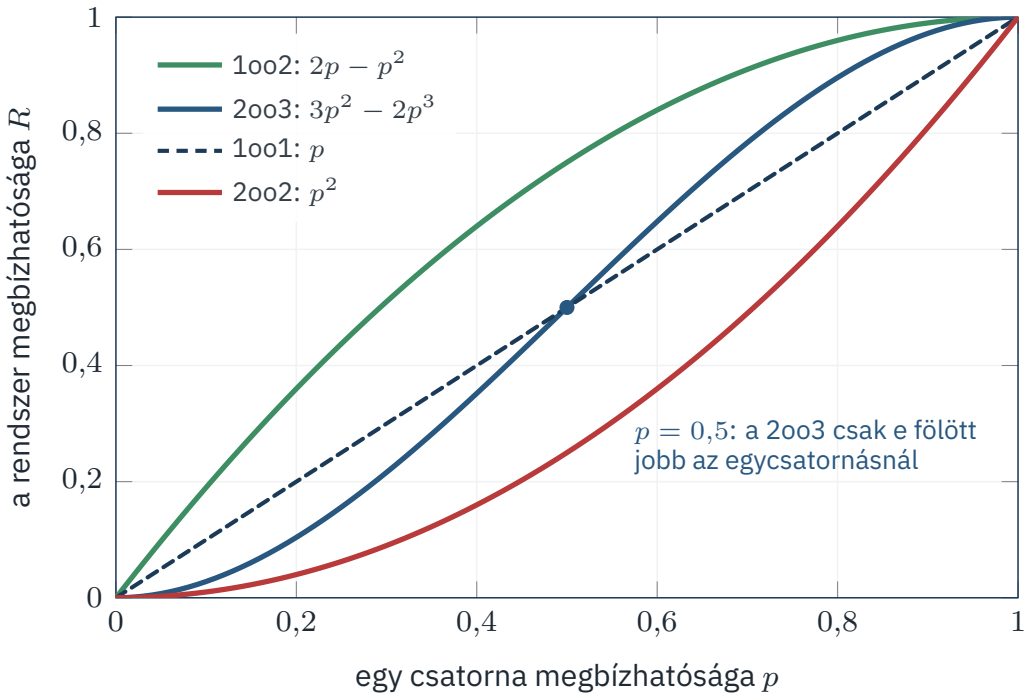
A 9.2. ábra ezeket hasonlítja össze. Egy meglepő tanulság: a 2oo3 rendszer csak akkor jobb az egycsatornásnál, ha $p > 0{,}5$. Hosszú időre nézve (amikor p kicsi) rosszabb is lehet. Exponenciális csatornáknál a 2oo3 rendszer átlagos élettartama $\mathrm{MTTF}_{2\text{oo}3} = 5/(6\lambda)$, *kisebb*, mint egyetlen csatornáé. A 2oo3 előnye tehát nem a hosszú élettartam, hanem az, hogy rövid (például a proof tesztek közötti) időtartamra nagy megbízhatóságot ad, és közben a téves működésekkel szemben is tűrőképes.

Mit jelent a „hibakombinációk száma”?

A 2014-es jegyzet a redundáns rendszereket a lehetséges hibakombinációk leszámlálásával tárgyalta (hányféleképpen hibásodhat meg egy 1oo3 rendszer úgy, hogy még működik). A leszámlálás jó kombinatorikai gyakorlat, de önmagában nem ad megbízhatóságot: a kombinációk valószínűsége különböző, és a rendszer megbízhatóságához éppen a valószínűségekkel súlyozott összeg kell. Ezt adja a binomiális modell (9.4. egyenlet), a kombinációk száma pedig a binomiális együtthatókban jelenik meg.

9.3.1. Szavazó architektúrák a biztonsági rendszerekben

A biztonsági rendszerekben a koon jelölés jelentése sajátos: az **MooN** architektúra azt jelenti, hogy N csatornából legalább M-nek kell *jeleznie* (kioldást kérnie) ahhoz, hogy a biztonsági funkció végrehajtódjon. Egy biztonsági rendszer kétféleképpen hibázhat (2.6.2. szakasz): **veszélyesen** (igénykor nem old ki) és **biztonságosan** (igény nélkül kiold, azaz téves leállást okoz). A szavazási logika a kettő között egyensúlyoz (9.2. táblázat).



9.2. ábra. A k-ból-n rendszerek megbízhatósága a csatornamegbízhatóság függvényében.

9.2. táblázat. A szavazó architektúrák tulajdonságai a biztonsági rendszerekben

Architektúra	A funkció végrehajtásához	Ennyi veszélyes hibát tűr (HFT)	Ennyi téves kioldást tűr	Jellemző
1oo1	az egyetlen csatorna jelez	0	0	egyszerű, nincs redundancia
1oo2	bármelyik csatorna jelez	1	0	biztonságos, de kétszeres téves kioldási gyakoriság
2oo2	mindkét csatorna jelez	0	1	kevés téves kioldás, de rosszabb biztonság
2oo3	legalább két csatorna jelez	1	1	egyensúly a biztonság és a rendelkezésre állás között
1oo3	bármelyik csatorna jelez	2	0	nagyon nagy biztonság, sok téves kioldás

A folyamatiparban az érzékelőknél és a beavatkozóknál az 1oo2 és a 2oo3 a leggyakoribb. Az 1oo2 a biztonság szempontjából kiváló, de mivel bármelyik csatorna téves jele leállítást okoz, a téves leállások gyakorisága kétszeres. Ahol a téves leállás költséges vagy maga is veszélyes (például egy nagy erőmű vagy egy kényes vegyi folyamat leállítása), ott a 2oo3 terjedt el.

ReaktorVéd: a biztonsági funkció blokkdiagramja

A ReaktorVéd biztonsági funkciója akkor sikeres, ha igénykor a betáplálás leáll (9.3. ábra). Ehhez legalább egy hőmérséklet-távadónak jeleznie kell (1oo2), a biztonsági PLC-nek működnie kell (1oo1), és legalább az egyik gyorszáró szelepnek zárnia kell (1oo2, mert a két szelep sorba van kötve a vezetéken).



9.3. ábra. A ReaktorVéd biztonsági funkciójának megbízhatósági blokkdiagramja a „betáplálás leállítása igénykor” funkcióra.

Ha a blokkokat az igénykori meghibásodási valószínűségükkel ($q = 1 - p$) jellemezzük, a funkció meghibásodási valószínűsége független elemekkel

$$q_{\text{SIF}} \approx q_{TT}^2 + q_L + q_V^2,$$

ahol a kis tagok szorzatait elhanyagoltuk. A távadókra $q_{TT} = 1{,}76 \cdot 10^{-3}$, a logikára $q_L = 5 \cdot 10^{-5}$, a szelepekre $q_V = 1{,}10 \cdot 10^{-2}$ (ezek az egycsatornás átlagos PFD-értékek, 13. fejezet) adódik:

$$q_{\text{SIF}} \approx 3{,}1 \cdot 10^{-6} + 5 \cdot 10^{-5} + 1{,}2 \cdot 10^{-4} = 1{,}7 \cdot 10^{-4}.$$

Ez az eredmény *túlságosan optimista*, két okból. A blokkdiagram feltételezi, hogy a két szelep és a két távadó egymástól függetlenül hibásodik meg, holott közös okú meghibásodásaik is lehetnek. Továbbá az átlagos meghibásodási valószínűségek szorzata nem azonos a szorzat időátlagával, ha a csatornákat egyszerre tesztelik. A 13. fejezet mindkét hatást figyelembe veszi, és nagyjából nyolcszor nagyobb értéket kap ($1{,}37 \cdot 10^{-3}$). A blokkdiagram tehát jó a struktúra megértésére, de a biztonsági funkciók igazolásához önmagában nem elegendő.

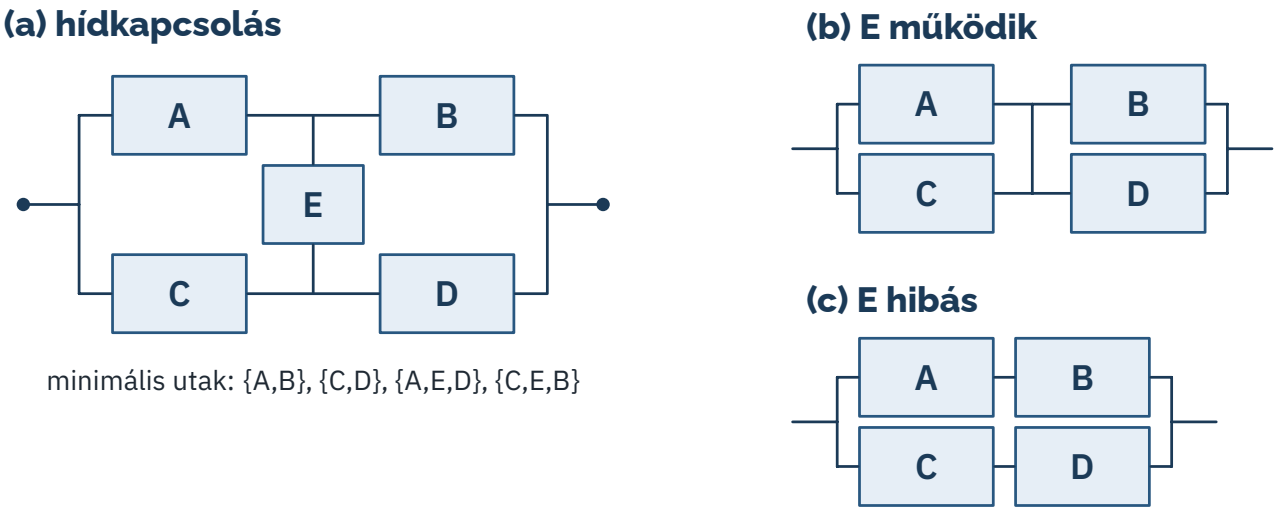
9.4. Nem soros-párhuzamos rendszerek

Nem minden rendszer bontható soros és párhuzamos részekre. A klasszikus példa a **hídkapcsolás** (9.4. ábra): az E elem összeköti a két ágat, így az A–E–D és a C–E–B útvonal is lehetséges. Az E meghibásodása nem teszi lehetetlenné a működést, de megváltoztatja a lehetséges utakat. A 2014-es jegyzet egy hasonló, „nem valódi párhuzamos” elrendezést mutatott be. Az ilyen rendszerek három módszerrel kezelhetők.

Állapotfelsorolás. Mind a $2^5 = 32$ állapotot megvizsgáljuk; egyszerű, de nagy rendszereknél kezelhetetlen.

Dekompozíció (feltételes valószínűség). Kiválasztunk egy kulcselemet (itt E-t), és a teljes valószínűség tételét alkalmazzuk:

$$R = p_E \cdot R(\text{rendszer} \mid E \text{ működik}) + (1 - p_E) \cdot R(\text{rendszer} \mid E \text{ hibás}). \tag{9.6}$$



9.4. ábra. A hídkapcsolás és dekompozíciója az E elem szerint.

Ha E működik, a rendszer két párhuzamos pár soros kapcsolása: $[1-(1-p_A)(1-p_C)]\,[1-(1-p_B)(1-p_D)]$. Ha E hibás, két soros ág párhuzamos kapcsolása: $1-(1-p_Ap_B)(1-p_Cp_D)$. Ezt a lépést Shannon-kifejtésnek is nevezik; a hibafák bináris döntési diagramjai (10. fejezet) ugyanerre épülnek.

A hídkapcsolás megbízhatósága

Ha minden elem megbízhatósága $p = 0{,}9$:

$$R(\text{E működik}) = (1-0{,}1^2)^2 = 0{,}9801, \qquad R(\text{E hibás}) = 1-(1-0{,}81)^2 = 0{,}9639,$$

$$R = 0{,}9 \cdot 0{,}9801 + 0{,}1 \cdot 0{,}9639 = 0{,}97848.$$

Azonos elemekre a zárt alak $R = 2p^2 + 2p^3 - 5p^4 + 2p^5$, amely ugyanezt adja.

Minimális utak és vágatok. Egy **út** (*path set*) olyan elemhalmaz, amelynek működése biztosítja a rendszer működését; **minimális**, ha egyetlen eleme sem hagyható el. Egy **vágat** (*cut set*) olyan elemhalmaz, amelynek meghibásodása a rendszer meghibásodását okozza; **minimális**, ha egyetlen eleme sem hagyható el. A híd minimális útjai: {A,B}, {C,D}, {A,E,D}, {C,E,B}; minimális vágatai: {A,C}, {B,D}, {A,E,D}, {C,E,B}. A rendszer megbízhatósága a minimális utak uniójának valószínűsége, amely a szita-formulával (inklúzió–exklúzió) pontosan, korlátokkal közelítőleg számolható. A minimális vágatok a hibafa-elemzés központi fogalmai; a 10. fejezet részletesen tárgyalja őket.

9.5. Meleg és hideg tartalékolás

A párhuzamos redundancia kétféleképpen valósítható meg:

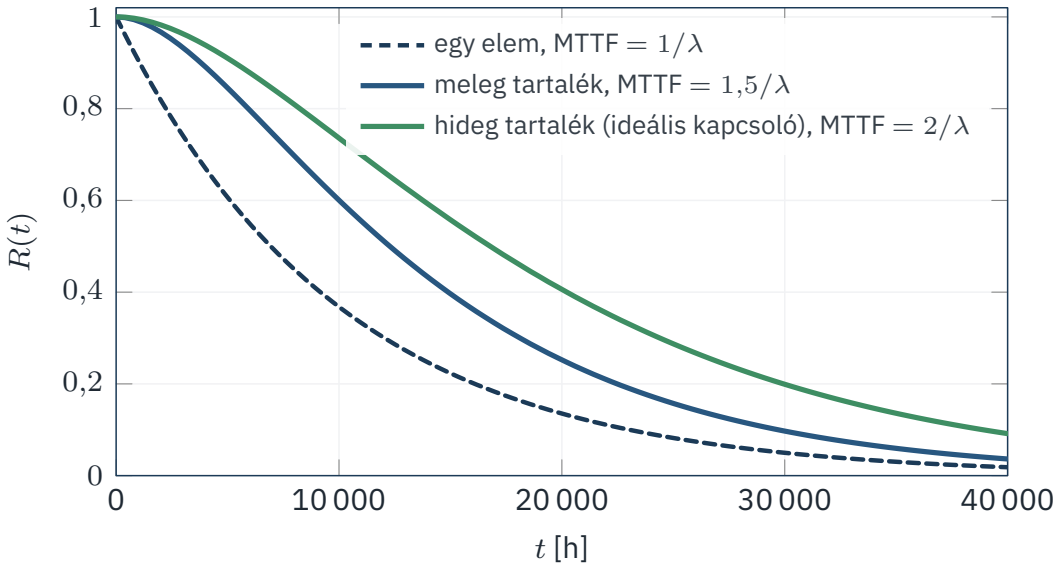
- **Meleg (aktív) tartalék:** a tartalék elem folyamatosan működik a fő elemmel együtt, így ugyanúgy öregszik és hibásodhat meg. Ez a párhuzamos rendszer.

- **Hideg (passzív) tartalék:** a tartalék elem kikapcsolt állapotban vár, és csak a fő elem kiesésekor kapcsolódik be. Közben (ideális esetben) nem hibásodik meg.

Két azonos, λ rátájú elemből álló hidegtartalékos rendszer ideális (hibátlan) átkapcsolóval addig működik, amíg mindkét elem egymás után meg nem hibásodik; élettartama a két exponenciális élettartam összege, azaz gamma- (Erlang-) eloszlású:

$$R_{\text{hideg}}(t) = e^{-\lambda t}(1+\lambda t), \qquad \text{MTTF}_{\text{hideg}} = \frac{2}{\lambda}. \tag{9.7}$$

Ez jobb, mint a meleg tartalék $1{,}5/\lambda$ értéke (9.5. ábra). A valóságban azonban az átkapcsoló maga is meghibásodhat, a hideg tartalék a várakozás alatt is öregedhet (például a szigetelés vagy a kenőanyag), és a bekapcsoláskor nem indul el (*failure to start*). Ezek a hatások a hideg tartalék előnyét gyakran eltüntetik. Pontos modellezésükre a Markov-modell alkalmas (12. fejezet).



9.5. ábra. Egy elem, a meleg tartalékos és az ideális hideg tartalékos rendszer megbízhatósága ($\lambda = 10^{-4}\,\text{h}^{-1}$).

KoboCella: kétcsatornás biztonsági funkció

A KoboCella sebességfelügyeletét a PL_r = e követelmény miatt kétcsatornás architektúrával kell megvalósítani (az ISO 13849-1 szerinti 4-es kategória): két független sebességmérés és két független lekapcsolási út, amelyek bármelyike megállítja a robotot (1oo2). Blokkdiagram-szempontból ez a „biztonságos megállítás” funkcióra párhuzamos, a „termelés folytatása” funkcióra viszont soros. Bármelyik csatorna téves jelzése leállítja a cellát, ezért a tervezőknek a téves leállások gyakoriságát is ellenőrizniük kell (H4 veszély, 7.4. szakasz).

9.6. Bizonytalan komponensadatok

A komponensek megbízhatósága ritkán ismert pontosan. A 2014-es jegyzet egy szemléletes példán mutatta be, hogyan vihető át a bizonytalanság a rendszer szintjére **intervallumbecsléssel**: minden elemre alsó és felső korlátot adunk, és mivel a koherens rendszerek megbízhatósága minden

elemben monoton nő, a rendszer alsó korlátját az alsó, a felső korlátját a felső komponenskorlátokból számoljuk.

Intervallumbecslés és Monte Carlo-szimuláció

Egy rendszer két alrendszerből áll. Az S_2 az S_1 tartaléka (a kettő párhuzamos). S_1-ben három komponens (A, B, C) közül legalább egynek működnie kell (párhuzamos), S_2-ben mindhárom komponensnek (D, E, F) működnie kell (soros). A komponensek megbízhatósága tízéves terepi adatokból tartományként ismert:

9.3. táblázat. A komponensek megbízhatósági tartományai

Komponens	A	B	C	D	E	F
alsó korlát	0,70	0,80	0,60	0,98	0,96	0,98
felső korlát	0,72	0,84	0,62	0,99	0,97	0,99

Alsó korlátok: $S_1 = 1 - 0{,}30 \cdot 0{,}20 \cdot 0{,}40 = 0{,}976$, $S_2 = 0{,}98 \cdot 0{,}96 \cdot 0{,}98 = 0{,}922$, a rendszer $1 - (1 - 0{,}976)(1 - 0{,}922) = 0{,}9981$. Felső korlátok: $S_1 = 0{,}983$, $S_2 = 0{,}951$, a rendszer $0{,}9992$. A rendszer megbízhatósága tehát a $[0{,}998;\ 0{,}999]$ intervallumba esik.
Az intervallumbecslés konzervatív, mert feltételezi, hogy minden komponens egyszerre a legrosszabb (vagy legjobb) értéket veszi fel. A **Monte Carlo-szimuláció** reálisabb képet ad: minden komponens megbízhatóságát a tartományán belül véletlenszerűen (például egyenletes eloszlással) választjuk, kiszámoljuk a rendszer megbízhatóságát, és ezt sokszor megismételjük. 200 000 ismétlés után a rendszer megbízhatóságának átlaga 0,99870, az 5%-os és 95%-os kvantilise 0,99848 és 0,99891. Ez az intervallum jóval szűkebb, mint a konzervatív korlátok által adott, mert a szélsőértékek egyidejű bekövetkezése valószínűtlen.

A Monte Carlo-módszer előnye, hogy tetszőleges bonyolultságú rendszerre és tetszőleges bemeneti eloszlásra alkalmazható; hátránya, hogy a bemeneti eloszlások megválasztása maga is bizonytalan, és a ritka események vizsgálatához sok ismétlés kell.

9.7. Erősségek és korlátok

Előnyök: a megbízhatósági blokkdiagram lehetővé teszi a tervezési koncepciók korai összevetését; könnyebben áttekinthető, mint a legtöbb más logikai modell; az elemek elrendezése tükrözheti a rendszerbeli szerepüket; könnyen átalakítható hibafává és eseményfává (11. fejezet).

Korlátok: a rendszert részeire kell bontani, és minden rész megbízhatóságát meg kell becsülni, ami nagy rendszereknél erőforrásigényes; a komponensadatok gyakran hiányosak, szubjektívek és eltérő konfidenciájúak; nem minden rendszer modellezhető soros-párhuzamos struktúrával; a függetlenségi és a kétállapotú feltevés sokszor sérül; a sorrendfüggő és javítható viselkedés nem ábrázolható.

Gyakorlat

A gyakorlati útmutató 7. laborjában egy böngészős blokkdiagram-szerkesztővel építhetsz soros, párhuzamos és k-ból-n struktúrákat, és megnézheted, hogyan változik a rendszer megbízhatósága és MTTF-je az idő függvényében. A hozzá tartozó Colab-notebook a hídkapcsolást állapotfelsorolással, dekompozícióval és Monte Carlo-szimulációval is kiszámolja, és megmutatja a közös okú meghibásodások hatását az 1oo2 struktúrára.

Összefoglalás

- A blokkdiagram a helyes működés logikai modellje; feltevései a kétállapotú elemek, a függetlenség és a statikus logika. A blokkdiagram nem kapcsolási rajz.

- Soros rendszerben a megbízhatóságok, párhuzamos rendszerben a meghibásodási valószínűségek szorzódnak; exponenciális elemeknél a soros rendszer rátája a ráták összege.
- A k-ból-n rendszer megbízhatósága binomiális összeg; a 2oo3 csak $p > 0{,}5$ fölött jobb az egycsatornásnál, és MTTF-je kisebb.
- A biztonsági rendszerek szavazó architektúrái a veszélyes és a biztonságos meghibásodásokkal szembeni tűrés között egyensúlyoznak.
- A nem soros-párhuzamos rendszerek állapotfelsorolással, dekompozícióval vagy minimális utakkal és vágatokkal kezelhetők.
- A hideg tartalék elvben jobb a melegnél, de az átkapcsoló és az indulási hibák ezt az előnyt csökkentik.
- A komponensadatok bizonytalansága intervallumbecsléssel (konzervatív) vagy Monte Carlo-szimulációval (reálisabb) vihető át a rendszer szintjére.

Önellenőrzés

1. Egy soros rendszer 50 elemből áll, mindegyik megbízhatósága 0,999. Mekkora a rendszer megbízhatósága? Mennyi kellene egy elemé ahhoz, hogy a rendszeré 0,99 legyen?
2. Két azonos, $\lambda = 10^{-4}\,\mathrm{h}^{-1}$ rátájú elem párhuzamosan van kötve. Mekkora a rendszer MTTF-je, és mekkora a megbízhatósága 1000 óra után?
3. Számold ki egy 2oo3 rendszer megbízhatóságát $p = 0{,}95$ mellett! Hasonlítsd össze az 1oo2 és az 1oo1 rendszerrel!
4. Miért mondjuk, hogy a sorba kötött két gyorszáró szelep megbízhatósági szempontból párhuzamos? Mikor lenne soros?
5. Miért ad szűkebb intervallumot a Monte Carlo-szimuláció, mint a korlátokon alapuló intervallumbecslés?

Megoldás.

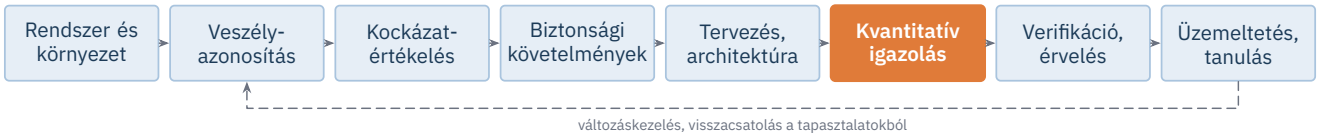
1. $R = 0{,}999^{50} \approx 0{,}951$. A célhoz $p^{50} = 0{,}99$, azaz $p = 0{,}99^{1/50} \approx 0{,}9998$.
2. MTTF $= 1{,}5/\lambda = 15\,000$ óra. $R(1000) = 1 - (1 - e^{-0{,}1})^2 = 1 - 0{,}0952^2 \approx 0{,}9909$.
3. $R_{2oo3} = 3 \cdot 0{,}9025 - 2 \cdot 0{,}857 = 0{,}99275$; $R_{1oo2} = 1 - 0{,}05^2 = 0{,}9975$; $R_{1oo1} = 0{,}95$. A 2oo3 sokkal jobb az egycsatornásnál, de valamivel rosszabb az 1oo2-nél (cserébe tűri a téves jelzéseket).
4. A „zárás” (az áramlás megszüntetése) funkcióra bármelyik szelep zárása elegendő, ezért párhuzamosak. A „nyitás” (az áramlás biztosítása) funkcióra mindkét szelepnek nyitva kell lennie, ezért erre a funkcióra sorosak; ez a téves leállások szempontjából fontos.
5. Mert a korlátokon alapuló becslés feltételezi, hogy minden komponens egyszerre a szélsőértékén van, ami nagyon valószínűtlen. A Monte Carlo-szimuláció a komponensek egyidejű, egymástól független ingadozását veszi figyelembe, így a rendszer megbízhatóságának valószínű tartományát adja.

Továbbolvasás

- Rausand, M., Barros, A., Høyland, A. (2020): *System Reliability Theory*, 4–6. fejezet – struktúrafüggvények, koherens rendszerek, k-ból-n rendszerek (Rausand és mtsai. 2020).
- IEC 61078:2016 – a megbízhatósági blokkdiagramok szabványa (IEC 2016a).
- Trivedi, K. S., Bobbio, A. (2017): *Reliability and Availability Engineering.* Cambridge University Press (Trivedi és Bobbio 2017).

10.

Hibafa-elemzés



A blokkdiagram a helyes működés útjait írja le. A biztonsági elemzésben gyakran a fordított kérdés a természetesebb: *milyen meghibásodás-kombinációk vezetnek egy adott nemkívánatos eseményhez?* A hibafa-elemzés (*fault tree analysis*, FTA) erre a kérdésre ad szisztematikus, grafikus és matematikailag pontos választ. A hibafa egy kiválasztott csúcseseményből indul ki, és logikai kapukon keresztül, lépésről lépésre bontja le azokat az okokat, amelyek együtt vagy külön-külön előidézhetik. Ebben a fejezetben megismerjük a hibafa szimbólumait és szerkesztési szabályait, a minimális vágatok meghatározását, a csúcsesemény valószínűségének kiszámítását, a fontossági mértékeket, a közös okú meghibásodások modellezését és a bizonytalanság kezelését. A ReaktorVéd példáján azt is látni fogjuk, hogy egy hibafa hogyan vezet közvetlenül jobb tervezési döntéshez.

Tanulási eredmények

A fejezet végére képes leszel

- egy nemkívánatos csúcseseményhez szabályosan felépíteni a hibafát;
- a minimális vágatokat a MOCUS-eljárással meghatározni, és értelmezni az egypontos hibákat;
- az alapesemények valószínűségét a meghibásodás jellegének megfelelő modellel megadni;
- a csúcsesemény valószínűségét ritkaesemény-közelítéssel, felső korláttal és bináris döntési diagrammal kiszámolni;
- a Birnbaum-, a Fussell–Vesely-, a kockázatnövelési és a kockázatcsökkentési mértéket kiszámolni és a tervezésben felhasználni;
- a közös okú meghibásodásokat β-tényezős modellel beépíteni a hibafába;
- az eredmény bizonytalanságát Monte Carlo-szimulációval becsülni, és felismerni a hibafa korlátait.

10.1. A hibafa lényege és helye a biztonsági elemzésben

A hibafa-elemzést 1961–62-ben H. A. Watson dolgozta ki a Bell Telephone Laboratories-nál, az amerikai légierő Minuteman interkontinentális rakétáinak indítóirányító rendszeréhez. A módszert a Boeing fejlesztette tovább és vitte át a repülőgép-tervezésbe, a nukleáris ipar pedig az 1970-es évek valószínűségi biztonsági elemzéseiben tette általánossá (Ericson 1999). Az Egyesült Államok Nukleáris Szabályozó Hatóságának 1981-es *Fault Tree Handbook*-ja (NUREG-0492) máig a szakterület alapműve (Vesely és mtsai. 1981), amelyet a NASA 2002-ben űrtechnikai alkalmazásokkal bővített ki (Stamatelatos és mtsai. 2002). A módszer nemzetközi szabványa az IEC 61025 (IEC 2006a).

A hibafa **deduktív, felülről lefelé haladó** módszer: egy meghatározott rendszerszintű eseményből indul, és az okai felé halad. Ebben különbözik az FMEA-tól, amely **induktív**: az egyes elemek meghibásodási módjaiból indul, és azok következményeit követi a rendszer szintjéig (6. fejezet). A két módszer kiegészíti egymást. Az FMEA jó abban, hogy semmilyen meghibásodási mód ne maradjon ki, a hibafa pedig abban, hogy a *kombinációkat*, a többszörös meghibásodásokat és a közös okokat feltárja. Ezért a szabványok sokszor mindkettőt előírják. Az ISO 26262 például a hardvertervezés biztonsági elemzésére ASIL C és D esetén mind az induktív, mind a deduktív elemzést erősen ajánlja (ISO 2018a), a repülésben az ARP4761A a hibafát a rendszerbiztonsági értékelés egyik alapeszközeként kezeli (SAE International 2023b), a folyamatiparban pedig a biztonsági funkciók PFD-értékének ellenőrzésére is használják (13. fejezet).

A hibafa egyszerre három dolgot ad:

1. **Megértést:** a fa felépítése közben a csapat végiggondolja, hogyan működik és hogyan hibásodhat meg a rendszer. Sok gyenge pont már a szerkesztés közben kiderül, számítás nélkül.
2. **Kvalitatív eredményt:** a minimális vágatok megmutatják, mely meghibásodás-kombinációk okozzák a csúcseseményt, és melyek az egypontos hibák.
3. **Kvantitatív eredményt:** ha az alapesemények valószínűsége ismert, kiszámolható a csúcsesemény valószínűsége és az egyes események hozzájárulása.

> **Hibafa**
>
> A **hibafa** egy kiválasztott nemkívánatos rendszerállapot (a **csúcsesemény**) és az azt előidéző események közötti logikai kapcsolatok grafikus modellje. Az eseményeket logikai kapuk kötik össze; a fa levelei az **alapesemények**, amelyeket tovább már nem bontunk, és amelyekre valószínűségi adat áll rendelkezésre.

10.2. Szimbólumok és szerkesztési szabályok

A hibafa jelölésrendszere szabványos (10.1. ábra). A leggyakoribb kapu az **ÉS** (a kimeneti esemény akkor következik be, ha minden bemenet bekövetkezik) és a **VAGY** (elég egyetlen bemenet). A **k az n-ből** kapu a szavazó architektúrák tömör leírása. A **TILTÓ** kapu egy feltételhez köti a továbbterjedést (például: a szivárgásból csak akkor lesz tűz, ha gyújtóforrás is van). A **PRIORITÁSOS ÉS** a sorrendfüggő viselkedést jelöli, és már a dinamikus hibafákhoz tartozik (10.9. szakasz). A **házesemény** biztosan bekövetkező vagy biztosan be nem következő feltétel; segítségével egyetlen fában több üzemállapot vagy tervváltozat írható le úgy, hogy a megfelelő házeseményt 0-ra vagy 1-re állítjuk.

10.2.1. A szerkesztés lépései

A hibafa-elemzés folyamata hat lépésből áll:

1. **A rendszer és a határok meghatározása:** mi tartozik a vizsgált rendszerhez, milyen üzemállapotra és milyen külső feltételekre (például tápellátás, környezet, kezelői beavatkozás) vonatkozik az elemzés, és milyen részletezettségig bontunk.
2. **A csúcsesemény pontos megfogalmazása:** *mi* történik, *hol* és *mikor*. A „a védelem meghibásodik” túl tág; a „működési igény esetén a védelem nem állítja le a betáplálást 30 másodpercen belül” már elemezhető. A csúcseseményt rendszerint a veszélyelemzés (5. fejezet) jelöli ki.
3. **A fa felépítése:** a csúcseseménytől lefelé, minden szinten a *közvetlen, szükséges és elégséges okokat* keresve.
4. **Kvalitatív elemzés:** a minimális vágatok meghatározása és értelmezése.
5. **Kvantitatív elemzés:** a csúcsesemény valószínűsége, a fontossági mértékek és a bizonytalanság.
6. **Értelmezés és dokumentálás:** a következtetések, a feltevések és a javasolt intézkedések rögzítése.



10.1. ábra. A hibafa szabványos szimbólumai (IEC 61025 nyomán). Az ÉS- és a VAGY-kapu a statikus hibafa alapja; a prioritásos ÉS a dinamikus hibafák eszköze.

10.2.2. A szerkesztés szabályai

A NUREG-0492 néhány egyszerű szabályt fogalmazott meg, amelyek betartása a jó hibafák titka (Vesely és mtsai. 1981):

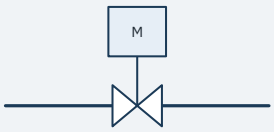
- **Az eseményt hibaként fogalmazzuk meg:** minden eseménydobozba azt írjuk, *mi* a hiba és *mikor* következik be („a szelep nem zár igénykor”, nem pedig „szelep”).
- **Elemhiba vagy rendszerhiba?** Ha az eseményt egyetlen elem meghibásodása okozhatja, **elemállapot-hiba**: alá rendszerint egy VAGY-kapu kerül a három lehetséges okkal (*primer hiba*: az elem a tervezési határain belül hibásodik meg; *szekunder hiba*: a határain kívüli igénybevétel, például túlfeszültség vagy túlmelegedés teszi tönkre; *vezérlési hiba*: az elem hibátlan, de rossz időben vagy rossz jelet kap). Ha nem, **rendszerállapot-hiba**, amelyet tovább kell bontani a közvetlen okai szerint.
- **Nincs csoda:** ha egy elem normális működése továbbviszi a hibát, feltételezzük, hogy normálisan működik. Nem számolunk azzal, hogy egy másik hiba „véletlenül” megállítja a folyamatot.
- **Teljes kapu:** egy kapu minden bemenetét definiáljuk, mielőtt bármelyiket tovább bontanánk. Így a fa szintenként épül, és nem vész el egyetlen ág sem.
- **Nincs kapu–kapu kapcsolat:** a kapuk bemenetei mindig megnevezett események, nem közvetlenül másik kapuk. A kapu alatti eseménydoboz kényszerít arra, hogy kimondjuk, mi történik azon a ponton.

Ezekhez a gyakorlat néhány tanácsot ad: gondolkodjunk *kis lépésekben* (minden szinten csak a közvetlen okot keressük, ne ugorjunk az alapeseményekhez); ugyanazt az eseményt mindenütt ugyanazzal a névvel és azonosítóval jelöljük, mert csak így ismerhetők fel az ismétlődő események; és rögzítsük a feltevéseket (például hogy a tápellátás a fa határán kívül van-e).

ReaktorVéd: a biztonsági funkció hibafája

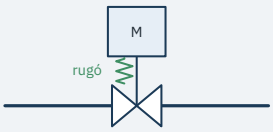
A ReaktorVéd biztonsági funkciójának csúcseseménye: *„működési igény esetén a védelem nem állítja le a folyamatot”*. Az első változatban a két gyorszáró szelep **energiával működtetett**: nyugalomban nyitva van, és a zárásához a mágnesszelepnek áramot kell kapnia (10.2. ábra, bal oldal). A szelepeket és a logikát az E1, a két hőmérséklet-távadót az E2 tápegység látja el.

energiával működtetett



nyugalomban nyitva; a záráshoz áram kell.
tápkiesés ⇒ igénykor nem zár: **veszélyes**
a tápellátás a veszélyes hibafa alapeseménye

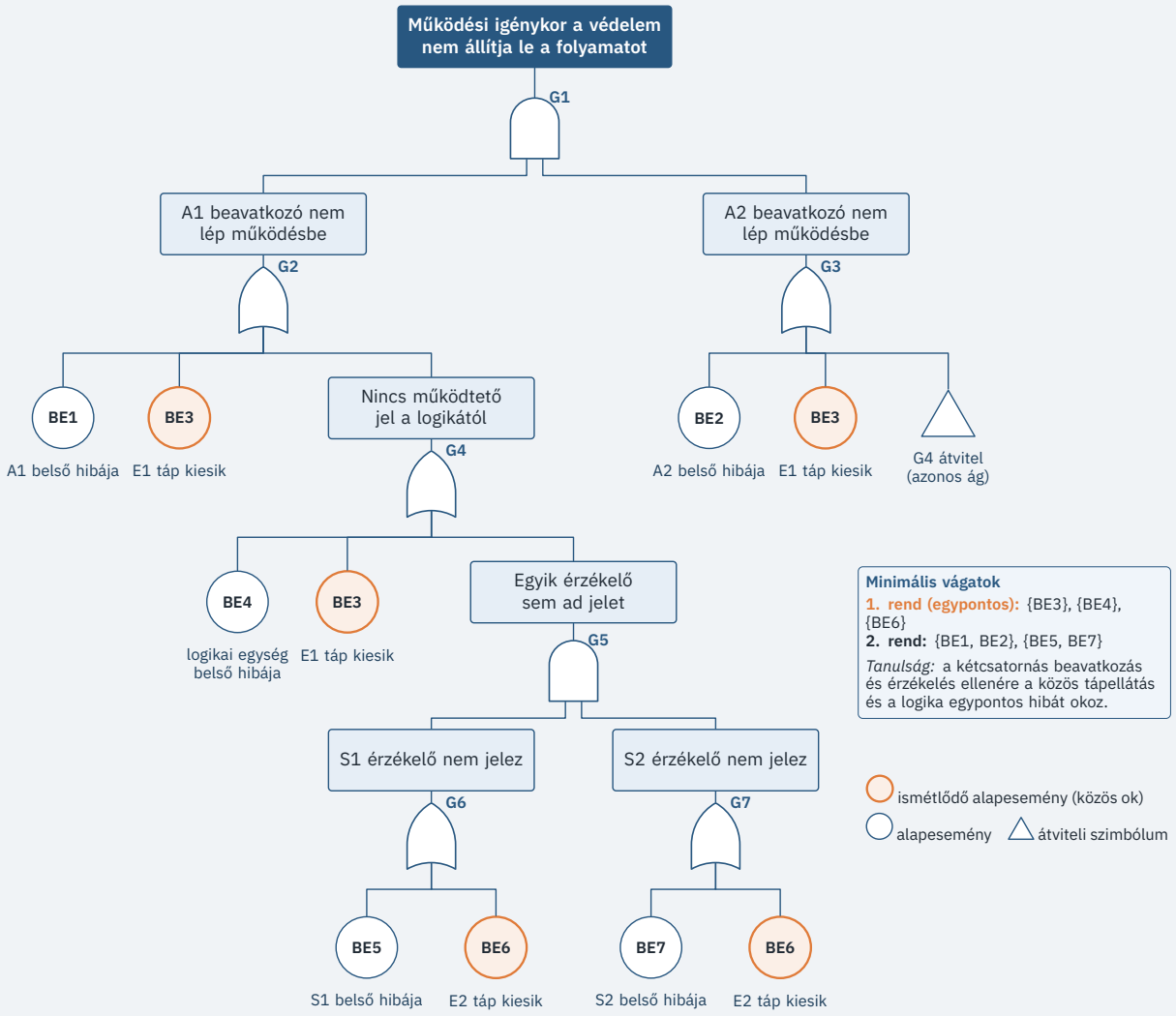
energiamentesítéssel működtetett



üzemben árammal nyitva tartva; a rugó zár.
tápkiesés ⇒ a szelep zár: **biztonságos**
a tápellátás csak a téves leál-
lások hibafájában szerepel

10.2. ábra. Az energiával és az energiamentesítéssel működtetett beavatkozó viselkedése tápkiesés esetén. A második változat hibabiztos: a tápkiesés a biztonságos állapotba viszi a folyamatot.

A hibafa (10.3. ábra) csúcsa alatt ÉS-kapu áll: a funkció csak akkor marad el, ha *mindkét* szelep nem zár (G1). Egy szelep nem zár, ha belső hibája van, ha nincs tápellátása, vagy ha nem kap működtető jelet a logikától (G2, G3; a G4 ágat a G3 alatt átviteli szimbólum jelöli, mert ugyanaz, mint a G2 alatt). A működtető jel elmarad, ha a logika hibás, ha nincs tápellátása, vagy ha egyik érzékelő sem jelez (G5). Egy érzékelő nem jelez, ha belső hibája van, vagy ha az E2 tápegység kiesik (G6, G7).



10.3. ábra. A ReaktorVéd energiával működtetett biztonsági funkciójának hibafája. A narancssárga körök ismétlődő alapeseményeket jelölnek: ugyanaz a tápegység több ágban is szerepel.

A fa első ránézésre megnyugtató: a csúcs alatt ÉS-kapu áll, és az érzékelőket is redundánsan kötöttük be. Az ismétlődő események azonban megtörik ezt a redundanciát. A BE3 (az E1 tápegység kiesése) egyszerre mindkét szelepet megbénítja, a BE6 (az E2 kiesése) egyszerre mindkét érzékelőt. A következő szakasz ezt pontosan is megmutatja.

10.3. Kvalitatív elemzés: minimális vágatok

Vágat és minimális vágat

Vágat az alapesemények olyan halmaza, amelyek együttes bekövetkezése előidézi a csúcseseményt. **Minimális vágat** az a vágat, amelyből egyetlen eseményt sem hagyhatunk el úgy, hogy vágat maradjon. A minimális vágat **rendje** az elemeinek száma. A duális fogalom az **út**: az alapesemények olyan halmaza, amelyek közül egyik sem következik be, és ez biztosítja, hogy a csúcsesemény se következzen be.

A minimális vágatok a hibafa Boole-algebrai kifejezésének **diszjunktív normálformáját** adják: a csúcsesemény a minimális vágatok VAGY-kapcsolata, minden vágat pedig az elemeinek ÉS-kapcsolata,

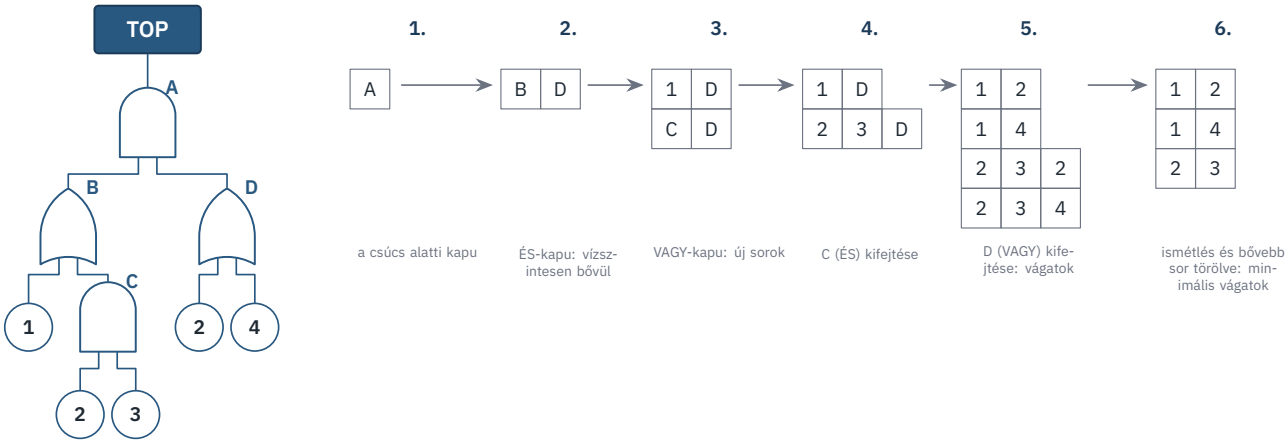
$$\text{TOP} = K_1 \cup K_2 \cup \cdots \cup K_m, \qquad K_j = \bigcap_{i \in K_j} B_i. \tag{10.1}$$

A minimális vágatok listája a hibafa legfontosabb kvalitatív eredménye. Az **elsőrendű** vágatok az **egypontos hibák**: egyetlen esemény elég a csúcseseményhez. Biztonsági rendszerben ezek a legsúlyosabb gyenge pontok. A magasabb rendű vágatoknál azt érdemes vizsgálni, hogy az elemeik **ki vannak-e téve közös oknak**: ha egy másodrendű vágat két eleme ugyanabban a helyiségben van, ugyanazt a tápot használja, ugyanaz a csapat tartja karban, vagy ugyanabból a gyártási sorozatból való, akkor a valóságban inkább elsőrendűként viselkedhet (10.6. szakasz).

10.3.1. A MOCUS-eljárás

Kis fáknál a minimális vágatok Boole-algebrai átalakítással is megkaphatók, nagyobb fáknál azonban szisztematikus eljárás kell. A klasszikus módszer a Fussell és Vesely által 1972-ben javasolt **MOCUS** (*method for obtaining cut sets*) (Fussell és Vesely 1972; Rausand és mtsai. 2020). Az eljárás egy mátrixot épít, és felülről lefelé helyettesíti a kapukat a bemeneteikkel (10.4. ábra):

1. Írjuk a csúcs alatti kapu jelét a mátrix első sorába.
2. Egy **ÉS-kaput** a bemeneteivel helyettesítünk *ugyanabban a sorban*, egymás mellett (a sor bővül).
3. Egy **VAGY-kaput** úgy helyettesítünk, hogy a sort annyiszor másoljuk, ahány bemenete van, és mindegyik másolatba egy-egy bemenet kerül (a mátrix új sorokkal bővül).
4. Addig folytatjuk, amíg a mátrixban csak alapesemények maradnak. Minden sor egy vágat.
5. Minden sorból töröljük az ismétlődő eseményeket ($A \cap A = A$), majd töröljük azokat a sorokat, amelyek egy másik sort tartalmaznak ($A \cup (A \cap B) = A$). A megmaradó sorok a minimális vágatok.



10.4. ábra. A MOCUS-eljárás lépései egy kis hibafán. Az ÉS-kapu vízszintesen, a VAGY-kapu függőlegesen bővíti a mátrixot; a végén az ismétlődéseket és a bővebb sorokat töröljük.

A példafa kifejezése $\text{TOP} = (x_1 + x_2 x_3)(x_2 + x_4)$; a MOCUS-eljárás négy logikai vágatot ad: $\{1,2\}$, $\{1,4\}$, $\{2,3\}$ és $\{2,3,4\}$. Az utolsó tartalmazza a $\{2,3\}$ vágatot, ezért nem minimális. A minimális vágatok tehát $\{1,2\}$, $\{1,4\}$ és $\{2,3\}$. Érdemes megfigyelni, hogy a 2-es esemény két ágban is szerepel.

Ha ezt nem vennénk észre, és a két ágat függetlennek tekintenénk, hibás eredményt kapnánk: az ismétlődő események miatt a hibafa kapunként, alulról felfelé nem számolható ki helyesen.

ReaktorVéd: a minimális vágatok

A 10.3. ábra hibafájára a MOCUS-eljárás öt minimális vágatot ad:

- **elsőrendű (egypontos) vágatok:** $\{\mathrm{BE3}\}$ (az E1 tápegység kiesése), $\{\mathrm{BE4}\}$ (a logika hibája), $\{\mathrm{BE6}\}$ (az E2 tápegység kiesése);
- **másodrendű vágatok:** $\{\mathrm{BE1}, \mathrm{BE2}\}$ (mindkét szelep belső hibája), $\{\mathrm{BE5}, \mathrm{BE7}\}$ (mindkét távadó belső hibája).

A kétcsatornás beavatkozás és érzékelés ellenére tehát három egypontos hiba van. A logika egycsatornás, ezt tudatosan vállaltuk (a biztonsági PLC belső diagnosztikája nagyon jó). A két tápegység viszont *rejtett* egypontos hiba: a blokkdiagramon (9.3. ábra) nem is látszott, mert a blokkdiagram csak a funkció elemeit ábrázolta, a közös támogató rendszereket nem. Ez a hibafa egyik legnagyobb értéke: a közös függőségeket (tápellátás, sűrített levegő, hűtés, kommunikáció, szoftver) a fa szerkesztése szinte kikényszeríti.

10.3.2. Bináris döntési diagramok

A minimális vágatok száma nagy fáknál robbanásszerűen nőhet: egy nukleáris erőmű valószínűségi biztonsági elemzésében milliónyi vágat is előfordul, ezért a gyakorlatban csak egy valószínűségi küszöb fölötti vágatokat tartják meg. A pontos megoldás modern eszköze a **bináris döntési diagram** (*binary decision diagram*, BDD), amelyet Bryant vezetett be a logikai áramkörök ellenőrzésére (Bryant 1986), és Rauzy alkalmazott először a hibafa-elemzésben (Rauzy 1993).

A BDD a Boole-függvény **Shannon-kifejtésén** alapul: egy kiválasztott x változó szerint

$$f = x \cdot f|_{x=1} + \bar{x} \cdot f|_{x=0}. \tag{10.2}$$

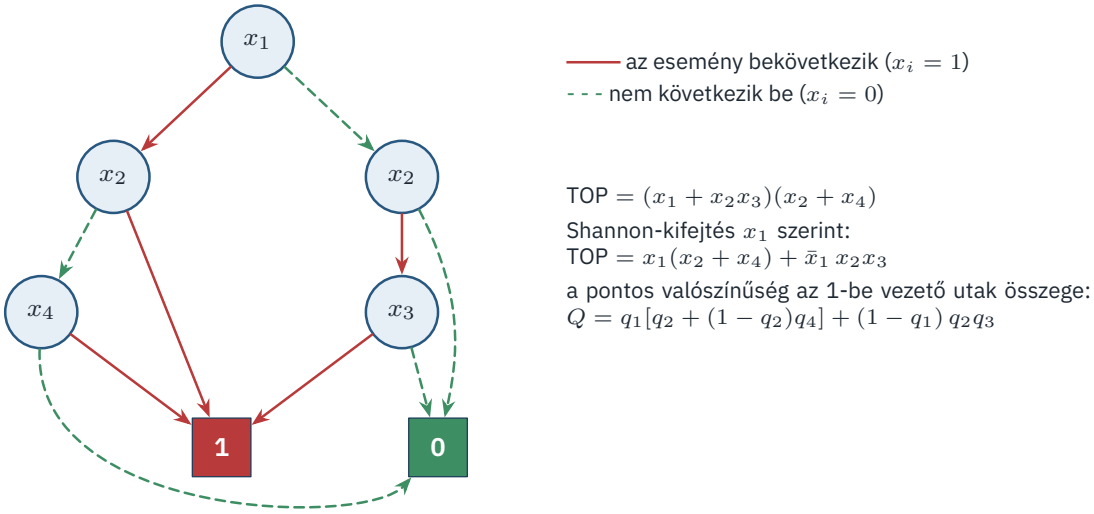
Ha a változókat rögzített sorrendben egymás után kifejtjük, és az azonos részfüggvényeket összevonjuk, egy irányított, körmentes gráfot kapunk, amelynek minden csúcsa egy változó, két kimenő éle pedig a változó 1, illetve 0 értékéhez tartozik (10.5. ábra). A gráf 1-es levelébe vezető utak *diszjunkt* eseményeket írnak le, ezért valószínűségeik egyszerűen összeadhatók, és a csúcsesemény **pontos** valószínűségét adják, közelítés nélkül.

A BDD mérete erősen függ a változók sorrendjétől; jó sorrenddel a legtöbb gyakorlati hibafa kezelhető méretű diagramot ad, rossz sorrenddel viszont a diagram exponenciálisan nagy lehet (Bryant 1986; Ruijters és Stoelinga 2015). A modern hibafa-eszközök ezért heurisztikákkal választják meg a sorrendet. A BDD-ből a minimális vágatok is előállíthatók, így a két megközelítés jól kiegészíti egymást.

10.4. Kvantitatív elemzés

10.4.1. Az alapesemények valószínűsége

A csúcsesemény valószínűségének kiszámításához minden alapeseményhez valószínűséget kell rendelni. Az, hogy melyik modell a helyes, a meghibásodás jellegétől és a karbantartási stratégiától függ (10.1. táblázat; a megbízhatósági alapfogalmakat a 8. fejezet tárgyalta).



10.5. ábra. A $(x_1+x_2x_3)(x_2+x_4)$ függvény bináris döntési diagramja $x_1 < x_2 < x_3 < x_4$ változósorrend mellett. Az 1-es levélbe vezető utak egymást kizáró eseményeket írnak le.

10.1. táblázat. Az alapesemények szokásos valószínűségi modelljei

Az alapesemény jellege	Modell	Közelítés kis értékekre
igénykori meghibásodás (például egy relé nem húz meg)	rögzített q	–
nem javítható elem, t üzemidő után	$q(t) = 1 - e^{-\lambda t}$	λt
azonnal feltárt, javítható meghibásodás	$Q = \dfrac{\lambda\,\mathrm{MTTR}}{1 + \lambda\,\mathrm{MTTR}}$	$\lambda\,\mathrm{MTTR}$
rejtett meghibásodás, T_1 időközű próbával	$\bar{q} = 1 - \dfrac{1 - e^{-\lambda T_1}}{\lambda T_1}$	$\lambda T_1/2$

A biztonsági rendszerekben a legfontosabb az utolsó sor: a nyugalmi állapotban lévő védelem meghibásodása rejtve marad, amíg egy próba vagy egy valódi igény fel nem fedi. Az átlagos meghibásodási valószínűség ilyenkor a próbaidőköz felével arányos. Ha a hibát a próba után még javítani is kell, az MTTR is hozzáadódik: $\bar{q} \approx \lambda(T_1/2 + \mathrm{MTTR})$.

ReaktorVéd: az alapesemények adatai

A próbaidőköz $T_1 = 8760$ óra (évente egy próba), a javítási idő $\mathrm{MTTR} = 8$ óra. A szelepeknek nincs diagnosztikájuk, a rejtett veszélyes meghibásodási ráta $\lambda_{DU} = 2{,}5 \cdot 10^{-6}\,\mathrm{h}^{-1}$, így $q_{\mathrm{BE1}} = q_{\mathrm{BE2}} = 2{,}5 \cdot 10^{-6} \cdot (4380 + 8) = 1{,}097 \cdot 10^{-2}$. A távadók diagnosztizált veszélyes hibáira a logika kiold, ezért csak a fel nem fedett rész számít: $q_{\mathrm{BE5}} = q_{\mathrm{BE7}} = 4 \cdot 10^{-7} \cdot 4380 = 1{,}752 \cdot 10^{-3}$. A logika átlagos meghibásodási valószínűsége $q_{\mathrm{BE4}} = 5 \cdot 10^{-5}$, a tápegységeké igénykor $q_{\mathrm{BE3}} = q_{\mathrm{BE6}} = 10^{-3}$.

10.4.2. A kapuk és a csúcsesemény valószínűsége

Független bemenetek esetén az ÉS-kapu kimenetének valószínűsége a bemenetek valószínűségeinek szorzata, a VAGY-kapué a komplementerek szorzatának komplementere:

$$Q_{\text{ÉS}} = \prod_{i=1}^{n} q_i, \qquad Q_{\text{VAGY}} = 1 - \prod_{i=1}^{n} (1 - q_i) \approx \sum_{i=1}^{n} q_i. \tag{10.3}$$

Ezek a képletek csak akkor alkalmazhatók kapuról kapura, alulról felfelé, ha a fában nincsenek ismétlődő események. Általános esetben a minimális vágatokból számolunk. A K_j vágat valószínűsége független alapeseményekkel $Q(K_j) = \prod_{i \in K_j} q_i$, a csúcsesemény valószínűségére pedig három közelítés terjedt el:

$$Q_{\text{ritka}} = \sum_{j=1}^{m} Q(K_j) \;\geq\; Q_{\text{MCUB}} = 1 - \prod_{j=1}^{m} \left(1 - Q(K_j)\right) \;\geq\; Q_{\text{pontos}}. \tag{10.4}$$

A **ritkaesemény-közelítés** egyszerűen összeadja a vágatok valószínűségét; koherens fára mindig felső korlát, és kis valószínűségeknél nagyon pontos. A **minimálisvágat-felsőkorlát** (*minimal cut set upper bound*, MCUB) úgy számol, mintha a vágatok függetlenek volnának; ez is felső korlát, és pontosabb. A **pontos** érték a szita-formulával (a vágatok összes metszetének előjeles összegével) vagy bináris döntési diagrammal adódik (Rausand és mtsai. 2020). A MOCUS-példánál ($q_1 = 0{,}01$, $q_2 = 0{,}02$, $q_3 = 0{,}05$, $q_4 = 0{,}1$) a három érték 0,00220, 0,00220 és 0,00217: a különbség kevesebb, mint 1,5%.

> **Gyakoriság és valószínűség nem keverhető**
>
> A hibafa alapeseményei *valószínűségek* (dimenzió nélküli számok). Ha a csúcsesemény gyakoriság (például „tűz évente”), akkor legfeljebb egyetlen bemenet lehet gyakoriság, a többinek feltételes valószínűségnek kell lennie: $f_{\text{TOP}} = f_{\text{kezdő}} \cdot q_1 \cdot q_2 \cdots$. Két gyakoriság szorzata (1/év²) értelmetlen. Ez a hiba meglepően gyakori a gyakorlatban; a kezdő esemény gyakoriságának és a feltételes valószínűségeknek a szétválasztása az eseményfa természetes logikája (11. fejezet).

> **ReaktorVéd: a csúcsesemény valószínűsége**
>
> A vágatok valószínűsége és hozzájárulása a 10.2. táblázatban látható.

10.2. táblázat. A ReaktorVéd energiával működtetett változatának minimális vágatai

Minimális vágat	Valószínűség	Hozzájárulás
$\{\text{BE3}\}$ – az E1 tápegység kiesése	$1{,}000 \cdot 10^{-3}$	46,0%
$\{\text{BE6}\}$ – az E2 tápegység kiesése	$1{,}000 \cdot 10^{-3}$	46,0%
$\{\text{BE1}, \text{BE2}\}$ – mindkét szelep	$1{,}203 \cdot 10^{-4}$	5,5%
$\{\text{BE4}\}$ – a logika	$5{,}000 \cdot 10^{-5}$	2,3%
$\{\text{BE5}, \text{BE7}\}$ – mindkét távadó	$3{,}070 \cdot 10^{-6}$	0,1%
csúcsesemény (ritkaesemény-közelítés)	$\mathbf{2{,}173 \cdot 10^{-3}}$	100%

A ritkaesemény-közelítés $2{,}1734 \cdot 10^{-3}$, a felső korlát és a pontos érték egyaránt $2{,}1721 \cdot 10^{-3}$. Az eredmény egyértelmű: a kockázat 92%-át a két tápegység adja, a gondosan megtervezett szelep- és távadó-redundancia szinte nem is számít. A biztonsági funkció így a SIL 2 tartományban marad ($10^{-3} \le \text{PFD} < 10^{-2}$), és nem teljesíti a LOPA által megkövetelt PFD $\le 2 \cdot 10^{-3}$ célt (3.4. szakasz).

Az átlagok szorzata nem a szorzat átlaga

A rejtett meghibásodású, együtt tesztelt redundáns elemeknél a kapuképletek csalóka eredményt adnak. Két, egyszerre tesztelt szelep együttes meghibásodási valószínűsége a próba után nulláról indul, és $(\lambda t)^2$ szerint nő. Ennek időátlaga $\lambda^2 T_1^2/3$, nem pedig az egyedi átlagok szorzata, $(\lambda T_1/2)^2 = \lambda^2 T_1^2/4$. Az eltérés 4/3-szoros: a ReaktorVéd szelepeire $1{,}60 \cdot 10^{-4}$ a helyes érték az $1{,}20 \cdot 10^{-4}$ helyett. A hibafa-eszközök többsége az alapeseményekhez átlagos valószínűséget rendel, ezért ezt a hatást külön kell kezelni. Az IEC 61508-6 képletei (13. fejezet) és a Markov-modellek (12. fejezet) helyesen veszik figyelembe (IEC 2010a; Rausand 2014).

10.5. Fontossági mértékek

A csúcsesemény valószínűsége egyetlen szám. A tervezőnek azonban arra van szüksége, hogy *hol* érdemes beavatkoznia. Erre szolgálnak a **fontossági mértékek**, amelyek az egyes alapesemények szerepét számszerűsítik (Borst és Schoonakker 2001; Rausand és mtsai. 2020). Jelölje Q a csúcsesemény valószínűségét, $Q(q_i = 1)$ azt az értéket, amikor az i-edik esemény biztosan bekövetkezik, $Q(q_i = 0)$ pedig azt, amikor biztosan nem.

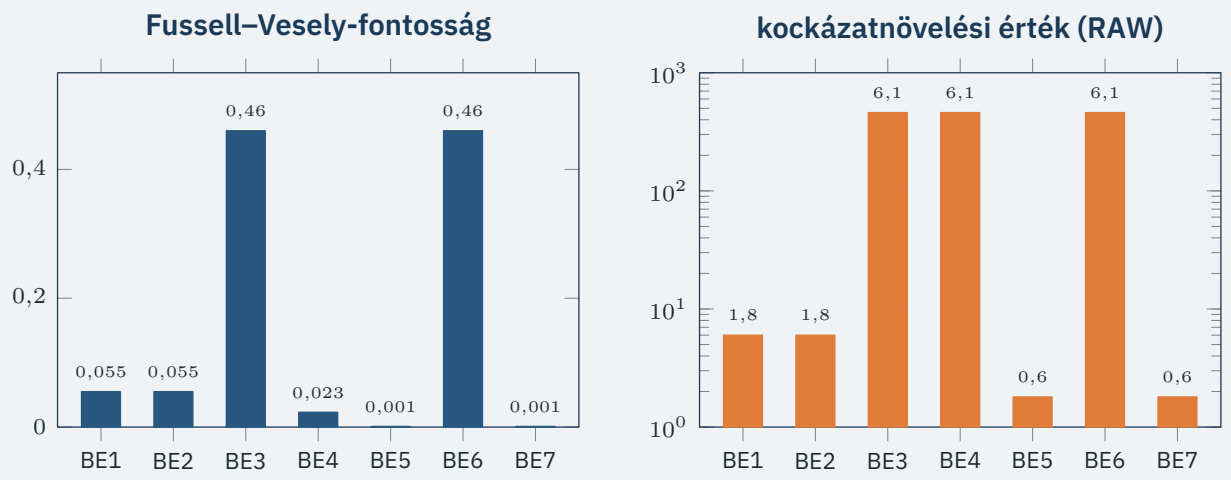
- **Birnbaum-fontosság** (Birnbaum 1969): $I_i^B = \partial Q / \partial q_i = Q(q_i = 1) - Q(q_i = 0)$. Annak a valószínűsége, hogy a rendszer olyan állapotban van, amelyben az i-edik esemény *kritikus*, vagyis az ő bekövetkezése dönti el a csúcseseményt. Az esemény saját valószínűségétől nem függ.
- **Fussell–Vesely-fontosság** (Fussell 1975): $I_i^{FV} = Q(\text{az } i\text{-t tartalmazó vágatok})/Q$. A csúcsesemény valószínűségének az a hányada, amelyhez az i-edik esemény hozzájárul. Ez a „hova folyik a kockázat” kérdésre felel.
- **Kockázatnövelési érték** (*risk achievement worth*, RAW): $\text{RAW}_i = Q(q_i = 1)/Q$. Hányszorosára nő a kockázat, ha az elem biztosan meghibásodott (például karbantartás miatt kiiktatták). A *megtartandó* teljesítmény mértéke.
- **Kockázatcsökkentési érték** (*risk reduction worth*, RRW): $\text{RRW}_i = Q/Q(q_i = 0)$. Hányadára csökkenne a kockázat, ha az elem tökéletes volna. A *fejlesztési* lehetőség felső korlátja.

ReaktorVéd: a fontossági mértékek

A 10.3. táblázat és a 10.6. ábra a ReaktorVéd energiával működtetett változatának fontossági mértékeit mutatja.

10.3. táblázat. A ReaktorVéd energiával működtetett változatának fontossági mértékei

Esemény	Birnbaum	Fussell–Vesely	RAW	RRW
BE1, BE2 – szelep belső hibája	$1{,}10 \cdot 10^{-2}$	0,055	6,0	1,06
BE3, BE6 – tápegység kiesése	0,999	0,460	460	1,85
BE4 – logika	0,998	0,023	460	1,02
BE5, BE7 – távadó belső hibája	$1{,}75 \cdot 10^{-3}$	0,001	1,8	1,00



10.6. ábra. A Fussell–Vesely-fontosság (a kockázat hányada, amelyhez az esemény hozzájárul) és a kockázatnövelési érték (logaritmikus skálán) a ReaktorVéd energiával működtetett változatában.

A számok három tanulságot hordoznak. Először: a tápegységek Fussell–Vesely-fontossága 0,46, RRW-je 1,85, vagyis egyetlen tápegység tökéletessé tétele majdnem felére csökkentené a kockázatot. Másodszor: a logika és a tápegységek RAW-értéke 460, vagyis ha bármelyikük kiesik, a funkció biztosan nem működik; ezeket tehát karbantartás alatt sem szabad pótlólagos védelem nélkül kiiktatni. Harmadszor: a szelep Birnbaum-fontossága pontosan a *másik* szelep meghibásodási valószínűsége, mert egy szelep csak akkor kritikus, ha a másik már nem zár. A szelepek fejlesztése ezért ebben a változatban alig hozna eredményt.

10.6. Közös okú meghibásodások a hibafában

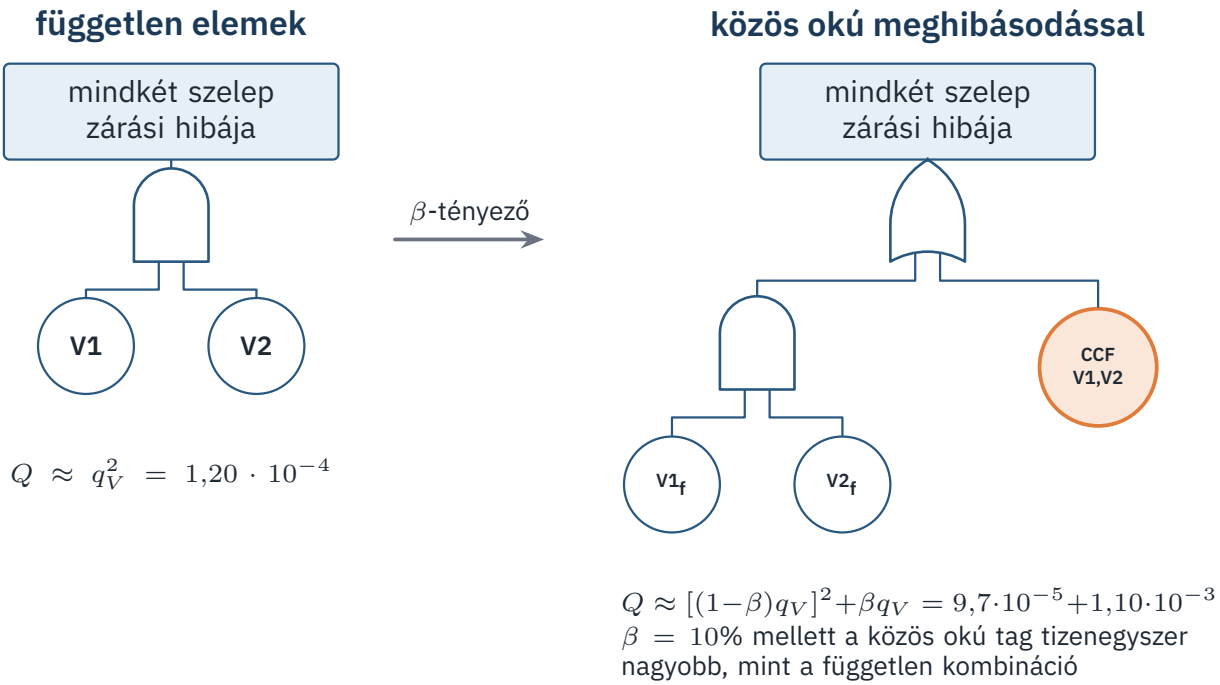
A redundancia csak akkor hatékony, ha a csatornák egymástól függetlenül hibásodnak meg. A valóságban a redundáns elemek gyakran ugyanazt a környezeti igénybevételt kapják, ugyanabból a gyártási sorozatból származnak, ugyanaz a csapat tartja karban őket, és ugyanaz a tervezési hiba lehet bennük. A **közös okú meghibásodás** (*common cause failure*, CCF) több elem egyidejű vagy közel egyidejű meghibásodása egyetlen közös ok miatt (Mosleh és mtsai. 1988). A ReaktorVéd tápegységei a CCF *explicit* esetét mutatták: a közös függőséget a fában külön eseményként ábrázoltuk. A rejtett, nem modellezett közös okokat paraméteres modellekkel kezeljük.

A legegyszerűbb a β**-tényezős modell**: az elem meghibásodási rátáját egy független és egy közös okú részre bontjuk,

$$\lambda = \lambda_{\mathrm{f}} + \lambda_{\mathrm{CCF}}, \qquad \lambda_{\mathrm{CCF}} = \beta\lambda, \qquad \lambda_{\mathrm{f}} = (1-\beta)\lambda, \tag{10.5}$$

ahol a közös okú rész a csoport *minden* elemét egyszerre éri. A hibafában ezt úgy építjük be, hogy minden elem eseményét egy VAGY-kapura cseréljük, amelynek egyik bemenete az elem független meghibásodása, a másik a csoport közös okú eseménye (10.7. ábra). A közös okú esemény így ismétlődő eseményként minden érintett ágban megjelenik, és a minimális vágatok között elsőrendű vágatként tűnik fel.

A β értékét az IEC 61508-6 D melléklete egy pontozásos eljárással adja meg, amely a szétválasztást, a diverzitást, a tervezési érettséget, az elemzéseket, a karbantartási eljárásokat, a képzést és a környezeti feltételeket értékeli; az eredmény a logikai egységekre 0,5–5%, az érzékelőkre és a beavatkozókra 1–10% (IEC 2010a). A finomabb modellek (az α-tényezős modell vagy a PDS-módszer C_{MooN}-tényezői)



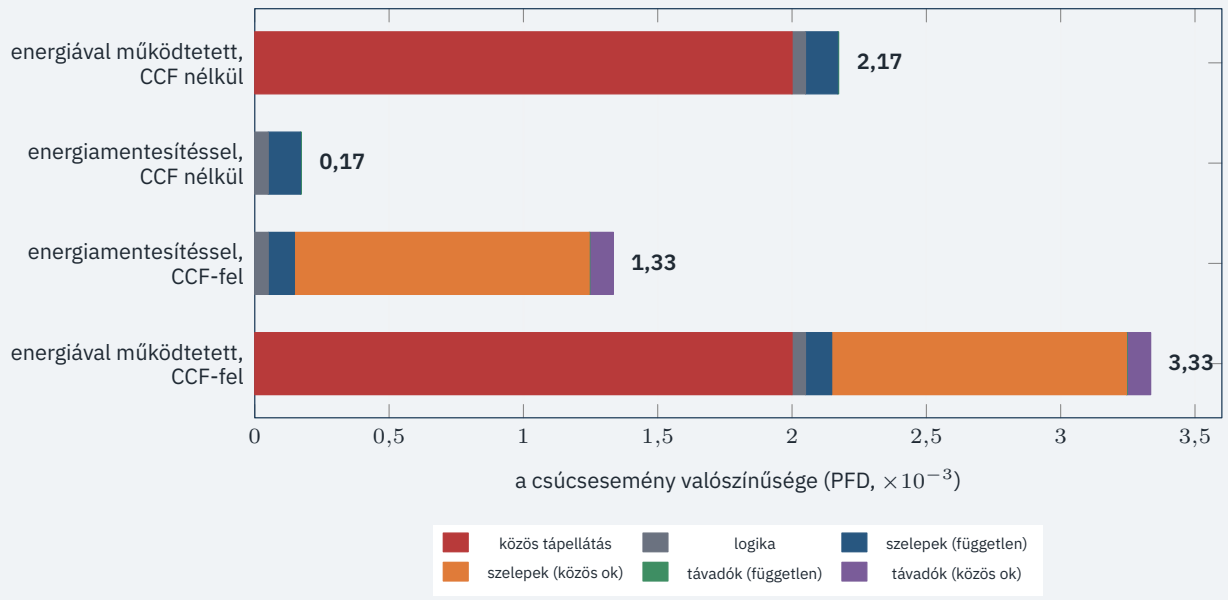
10.7. ábra. A közös okú meghibásodás beépítése a hibafába β-tényezős modellel. A közös okú esemény ismétlődő alapeseményként jelenik meg, és elsőrendű vágatot alkot.

a különböző rendű közös okú eseményeket is megkülönböztetik, ami 2oo3 és nagyobb redundanciánál fontos (Mosleh és mtsai. 1988; Hauge és mtsai. 2013).

ReaktorVéd: a hibabiztos változat

A fontossági mértékek alapján a tervezőcsapat két változtatást hajt végre: a szelepeket **energiamentesítéssel működtetett** (hibabiztos) kivitelre cseréli, amelyeket üzemben áram tart nyitva, és tápkiesésnél a rugó zárja (10.2. ábra, jobb oldal); a távadókat pedig áramhurkos, „élő nulla” jelzésű kivitelben köti be, így a táp- vagy vezetékhiba kioldást okoz. A tápegységek ezzel kikerülnek a veszélyes hibafából: kiesésük téves leállást okoz, és a téves leállások hibafájában jelennek meg.

Ha a közös okú meghibásodásokat figyelmen kívül hagyjuk, az új hibafa csúcseseménye $1{,}73 \cdot 10^{-4}$, ugyanannyi, mint a blokkdiagram eredménye (9. fejezet). Ez azonban túl szép, hogy igaz legyen. A szelepekre $\beta = 10\%$, a távadókra $\beta = 5\%$ értéket feltételezve a β-tényezős hibafa öt minimális vágatot ad: a szelepek közös okú hibája ($1{,}097 \cdot 10^{-3}$), a szelepek független hibáinak párja ($9{,}75 \cdot 10^{-5}$), a távadók közös okú hibája ($8{,}76 \cdot 10^{-5}$), a logika ($5{,}0 \cdot 10^{-5}$) és a távadók független hibáinak párja ($2{,}77 \cdot 10^{-6}$). A csúcsesemény $1{,}335 \cdot 10^{-3}$.



10.8. ábra. A ReaktorVéd négy hibafa-változatának összevetése. A hibabiztos kialakítás megszünteti a tápegységek egypontos hibáját, de a közös okú szelephibák ezután a kockázat 82%-át adják.

A 10.8. ábra négy változatot vet össze. Ha az eredeti, energiával működtetett változatban is figyelembe vesszük a közös okokat, a csúcsesemény $3{,}33 \cdot 10^{-3}$, vagyis a hibabiztos átalakítás valójában két és félszeres javulást hozott, nem tizenkétszerest. A következő lépés már nem a tápellátásról szól: a kockázatot most a szelepek közös okú meghibásodása uralja. Ezt a 13. fejezet részleges löketpróbával (*partial stroke testing*) csökkenti. A hibafa eredménye ($1{,}335 \cdot 10^{-3}$) jól egyezik az IEC 61508-6 képleteivel kapott $1{,}37 \cdot 10^{-3}$ értékkel; a kis eltérés a független páros meghibásodások időátlagolásából (a 4/3-os tényezőből) adódik.

10.7. Bizonytalanság és érzékenység

A csúcsesemény valószínűsége pontosan annyira megbízható, amennyire az alapesemények adatai. Az alapadatok forrásai (gyártói adatok, adatbázisok, üzemi tapasztalat, szakértői becslés) gyakran egy nagyságrendnyi vagy még nagyobb bizonytalanságot hordoznak (8. fejezet). A hibafa-elemzésnek ezért három kiegészítő vizsgálata van.

Érzékenységvizsgálat: egy paramétert (például a próbaidőközt) változtatunk, és figyeljük a csúcsesemény változását. A ReaktorVéd energiával működtetett változatában a próbaidőköz felezése ($T_1 = 4380$ óra) a csúcseseményt csak $2{,}173 \cdot 10^{-3}$-ról $2{,}081 \cdot 10^{-3}$-ra, alig 4%-kal csökkenti, mert a próba a tápegységek hibáin nem segít. A hibabiztos, közös okokat is figyelembe vevő változatban ugyanez a lépés $1{,}335 \cdot 10^{-3}$-ról $6{,}68 \cdot 10^{-4}$-re, a felére csökkenti a csúcseseményt. Ugyanaz az intézkedés tehát az egyik tervben hatástalan, a másikban hatékony; ezt csak a teljes modell mutatja meg.

Bizonytalanságterjedés: az alapesemények valószínűségét eloszlással írjuk le, és Monte Carlo-szimulációval kiszámoljuk a csúcsesemény eloszlását. A valószínűségi biztonsági elemzésekben elterjedt a **lognormális eloszlás**, amelyet a mediánnal és a **hibatényezővel** (*error factor*, EF) adunk meg: $\mathrm{EF} = q_{95}/q_{50} = q_{50}/q_{05}$, így a 90%-os intervallum $[q_{50}/\mathrm{EF},\ q_{50}\cdot\mathrm{EF}]$ (Modarres és mtsai. 2016). Ha a szakértő csak egy tartományt ad meg (például „0,01 és 0,1 között”), lognormális feltevés mellett a természetes középérték a mértani közép ($\sqrt{0{,}01 \cdot 0{,}1} = 0{,}0316$), nem a számtani (0,055).

A modell bizonytalansága: a legnehezebb kérdés az, hogy a fa teljes-e, és helyesek-e a feltevései (függetlenség, kétállapotú elemek, a határok). Erre nincs képlet; a független átvizsgálás, a tapasztalt elemzők és az üzemi tapasztalattal való összevetés segít.

> **Monte Carlo-bizonytalanságelemzés a ReaktorVéd hibafáján**
>
> Tegyük fel, hogy az energiával működtetett változat minden alapeseményének valószínűsége lognormális eloszlású, mediánja a pontbecslés, hibatényezője EF $= 3$ (azaz $\sigma = \ln 3/1{,}645 = 0{,}668$). Kétszázezer mintából a csúcsesemény eloszlása:
>
> - medián: $2{,}47 \cdot 10^{-3}$, átlag: $2{,}75 \cdot 10^{-3}$;
> - 90%-os intervallum: $1{,}20 \cdot 10^{-3}$ és $5{,}28 \cdot 10^{-3}$ között.
>
> Két dolog figyelemre méltó. Az átlag nagyobb a pontbecslésnél ($2{,}17 \cdot 10^{-3}$), mert a lognormális eloszlás jobbra ferde: ha a pontbecsléseket mediánként értelmezzük, a csúcsesemény várható értéke magasabb a pontbecslésből számolt értéknél. Másrészt már a medián is meghaladja a LOPA által megkövetelt $2 \cdot 10^{-3}$-os célt, a 95%-os percentilis pedig annak két és félszerese. A bizonytalanság figyelembevétele tehát még erősebben alátámasztja, hogy ez a változat nem fogadható el.

10.8. A sikerfa és a hibafa kapcsolata

Minden hibafának van **duális** párja, a sikerfa, amely a csúcsesemény *be nem következését* írja le. A sikerfát úgy kapjuk, hogy minden eseményt a komplementerére cserélünk, az ÉS-kapukat VAGY-kapura és fordítva (De Morgan-azonosságok). A sikerfa minimális vágatai a hibafa **minimális útjai**: azok az elemhalmazok, amelyek hibátlan működése garantálja a funkciót. A ReaktorVéd hibabiztos változatában (közös okok nélkül) egy minimális út például: {TT1 működik, logika működik, V1 zár}. A minimális utak pontosan a megbízhatósági blokkdiagram útjai, vagyis a hibafa, a sikerfa és a blokkdiagram ugyanannak a struktúrafüggvénynek három nézete (9.1.1. szakasz). A hibafa előnye, hogy a meghibásodások *okait* is ábrázolja (tápellátás, vezérlési hibák, közös okok), és nem csak a funkció elemeit.

10.9. Dinamikus hibafák

A klasszikus (statikus) hibafa nem tud különbséget tenni aközött, hogy két esemény milyen sorrendben következik be. Sok rendszerben azonban éppen a sorrend számít: egy hideg tartalék csak akkor kapcsolódik be, ha a fő egység már kiesett; egy átkapcsoló csak akkor okoz bajt, ha *előbb* hibásodik meg, mint a fő egység; egy közös hálózati kapcsoló kiesése funkcionálisan magával rántja a tőle függő elemeket. Ezek kezelésére Dugan és munkatársai 1992-ben vezették be a **dinamikus hibafát** új kapukkal (Dugan és mtsai. 1992): prioritásos ÉS (PAND), tartalék (SPARE, hideg, meleg és forró tartalékkal), funkcionális függőség (FDEP) és kényszerített sorrend (SEQ).

A dinamikus hibafa már nem oldható meg Boole-algebrával: a dinamikus részfákat Markov-láncra (12. fejezet), stochasztikus Petri-hálóra vagy szimulációra kell visszavezetni, a statikus részeket pedig továbbra is BDD-vel lehet számolni (Ruijters és Stoelinga 2015). A gyakorlatban érdemes a fát modulokra bontani, és csak ott használni dinamikus kapukat, ahol a sorrend valóban számít.

> **VészFék: két csúcsesemény, ellentétes követelmények**
>
> A vészfékező asszisztensnek legalább két hibafára van szüksége (5. fejezet). Az egyik csúcsesemény az **indokolatlan erős fékezés** (amely hátulról ütközést okozhat), a másik a **fékezés elmaradása** veszélyes helyzetben. A két fa ellentétes irányba húz. Ha az érzékelést két független csatornára (radar és kamera) építjük, és bármelyik csatorna jelzésére fékezünk (1oo2), a fékezés elmaradásának fája két független

csatorna egyidejű hibáját igényli, az indokolatlan fékezés fája viszont bármelyik csatorna téves jelzésétől bekövetkezik. Ha mindkét csatorna egyetértése kell (2oo2), a helyzet fordított. A két fa együttes elemzése mutatja meg, hogy a döntési logika és a csatornák diagnosztikája hogyan hangolható össze. Az ISO 26262 szerint ezek a hibafák a véletlen hardverhibák és a rendszerszintű hibák elemzésére valók; ha a fékezés azért marad el, mert a köd korlátozza az érzékelést, és semmi sem hibásodott meg, az a rendeltetésszerű funkció biztonságának (ISO 21448) tárgya (18. fejezet).

10.10. Erősségek és korlátok

Előnyök: a hibafa egy konkrét, súlyos nemkívánatos eseményre összpontosít; feltárja a többszörös meghibásodásokat, a közös függőségeket és az egypontos hibákat; ötvözi a hardvert, a szoftvert, az emberi hibát és a külső eseményeket; szemléletes, ezért a tervezők és a döntéshozók közötti kommunikációt is segíti; kvantitatív eredménye a biztonsági követelmények igazolására is alkalmas; a fontossági mértékek pedig közvetlenül megmutatják, hol a leghatékonyabb a beavatkozás.

Korlátok: a hibafa csak azokat az okokat tartalmazza, amelyekre az elemzők gondoltak, ezért teljessége sosem garantált; minden csúcseseményhez külön fa kell; a statikus fa nem ábrázolja a sorrendet és az időbeli viselkedést; a kétállapotú elemek feltevése a fokozatos leromlást nem írja le; a függetlenségi feltevés sérülését (közös okok) külön kell modellezni; a kvantitatív eredmény érzékeny a gyakran bizonytalan alapadatokra; nagy rendszereknél a fa mérete és a vágatok száma kezelhetetlenné válhat. A komplex, szoftverintenzív rendszerekben, ahol a balesetek gyakran hibátlan elemek nem szándékolt kölcsönhatásából fakadnak, a hibafát rendszerelméleti módszerekkel (STPA, 7. fejezet) érdemes kiegészíteni.

Gyakorlat

A gyakorlati útmutató 8. laborjában egy böngészős hibafa-szerkesztővel építheted fel a ReaktorVéd hibafáját; az alkalmazás kiszámolja a minimális vágatokat, a csúcsesemény valószínűségét és a fontossági mértékeket, és a házeseményekkel válthatsz az energiával és az energiamentesítéssel működtetett változat között. A hozzá tartozó Colab-notebook a MOCUS-eljárást és a bináris döntési diagramot is lépésenként megvalósítja, és Monte Carlo-szimulációval elvégzi a bizonytalanságelemzést.

Összefoglalás

- A hibafa deduktív módszer: egy pontosan megfogalmazott csúcseseményből indul, és logikai kapukon át bontja le az okait az alapeseményekig.
- A jó hibafa szabályai: hibaként megfogalmazott események, elem- és rendszerhibák megkülönböztetése, „nincs csoda”, teljes kapuk, nincs kapu–kapu kapcsolat; az ismétlődő események következetes jelölése.
- A minimális vágatok a csúcsesemény diszjunktív normálformáját adják; az elsőrendű vágatok az egypontos hibák. A MOCUS-eljárás és a bináris döntési diagram a két fő megoldási módszer.
- Az alapesemények modellje a meghibásodás jellegétől függ; rejtett, időszakosan tesztelt meghibásodásnál az átlagos valószínűség $\lambda T_1/2$.
- A csúcsesemény valószínűsége ritkaesemény-közelítéssel, felső korláttal vagy pontosan (BDD) számolható; koherens fára a két közelítés felső korlát.
- A fontossági mértékek közül a Fussell–Vesely a kockázat megoszlását, a RAW a megtartandó, az RRW a fejleszthető teljesítményt mutatja.
- A közös okú meghibásodások β-tényezős modellel elsőrendű vágatként épülnek be; a redundáns rendszerekben gyakran ezek uralják az eredményt.
- A ReaktorVéd példája szerint a hibabiztos kialakítás megszüntette a tápegységek egypontos hibáját, a kockázatot ezután a szelepek közös okú meghibásodása uralja.

Önellenőrzés

1. Miért nem jó csúcsesemény az, hogy „a hűtőrendszer meghibásodik”? Fogalmazd meg pontosabban!
2. Egy hibafa csúcsa alatt ÉS-kapu áll, két bemenete $A + B$ és $A + C$. Melyek a minimális vágatok? Mekkora a csúcsesemény valószínűsége, ha $q_A = 10^{-3}$, $q_B = q_C = 10^{-2}$, és mekkora lenne, ha az A ismétlődését figyelmen kívül hagynánk?
3. Egy alapesemény rejtett meghibásodás, $\lambda = 2 \cdot 10^{-6}\,\mathrm{h}^{-1}$, a próbaidőköz fél év. Mekkora az átlagos meghibásodási valószínűsége? Hogyan változik, ha a javítás átlagosan 24 óráig tart?
4. Miért lehet egy alapesemény Birnbaum-fontossága nagy, a Fussell–Vesely-fontossága mégis kicsi? Mondj rá példát a ReaktorVéd hibafájából!
5. Két redundáns szivattyú meghibásodási valószínűsége igénykor egyenként $q = 0{,}02$, és $\beta = 5\%$. Számold ki az együttes meghibásodás valószínűségét a közös okok figyelembevételével és anélkül!
6. Miért nő a ReaktorVéd energiával működtetett változatában alig a biztonság, ha a próbaidőközt felezzük?

Megoldás.

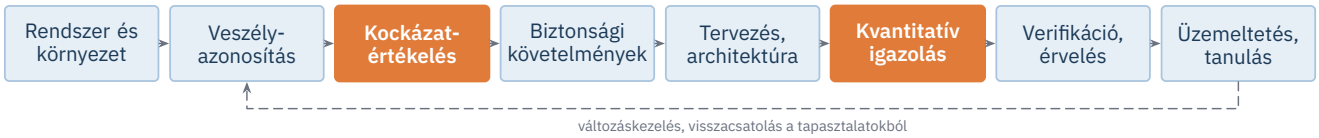
1. Nem mondja meg, mit jelent a meghibásodás, hol és mikor következik be. Pontosabban: „normál üzemben a reaktor hőelvonása 10 percen belül a tervezett érték 50%-a alá csökken”, vagy „igény esetén a tartalékhűtés 60 másodpercen belül nem indul el”.
2. TOP $= (A + B)(A + C) = A + AC + AB + BC = A + BC$. A minimális vágatok: $\{A\}$ és $\{B, C\}$. $Q \approx 10^{-3} + 10^{-4} = 1{,}1 \cdot 10^{-3}$. Ha az A ismétlődését figyelmen kívül hagynánk, és a két VAGY-kaput függetlennek tekintenénk: $(1{,}1 \cdot 10^{-2})^2 \approx 1{,}21 \cdot 10^{-4}$, ami majdnem egy nagyságrenddel alábecsülné a kockázatot.
3. $\bar{q} \approx \lambda T_1/2 = 2 \cdot 10^{-6} \cdot 4380/2 = 4{,}38 \cdot 10^{-3}$. A javítással $\bar{q} \approx \lambda(T_1/2 + \mathrm{MTTR}) = 2 \cdot 10^{-6} \cdot (2190 + 24) = 4{,}43 \cdot 10^{-3}$; a hatás csekély.
4. A Birnbaum-fontosság azt méri, mennyire kritikus az elem, *ha* meghibásodik; a saját valószínűségétől nem függ. A Fussell–Vesely-fontosság a tényleges hozzájárulást méri, amelyben az elem valószínűsége is benne van. A logika (BE4) Birnbaum-fontossága 0,998 (bekövetkezése szinte biztosan csúcseseményt okoz), de mivel nagyon megbízható ($q = 5 \cdot 10^{-5}$), Fussell–Vesely-fontossága csak 0,023.
5. Közös okok nélkül $q^2 = 4 \cdot 10^{-4}$. A β-tényezős modellel $[(1 - \beta)q]^2 + \beta q = (0{,}019)^2 + 0{,}001 = 3{,}61 \cdot 10^{-4} + 10^{-3} = 1{,}36 \cdot 10^{-3}$, vagyis több mint háromszoros.
6. Mert a kockázat 92%-át a tápegységek igénykori kiesése adja, amelyre a próbaidőköz nem hat. A próba csak a szelepek és a távadók rejtett hibáit fedi fel, azok pedig együtt is csak a kockázat kis részét adják. Előbb a domináns vágatokat kell megszüntetni.

Továbbolvasás

- Vesely, W. E. et al. (1981): *Fault Tree Handbook* (NUREG-0492). U.S. NRC – a szerkesztési szabályok és a klasszikus elmélet alapműve (Vesely és mtsai. 1981).
- Stamatelatos, M. et al. (2002): *Fault Tree Handbook with Aerospace Applications.* NASA – korszerű, gyakorlati kézikönyv, dinamikus kapukkal (Stamatelatos és mtsai. 2002).
- IEC 61025:2006 – a hibafa-elemzés nemzetközi szabványa (IEC 2006a).
- Rausand, M., Barros, A., Høyland, A. (2020): *System Reliability Theory*, 4. és 6. fejezet – hibafák, minimális vágatok, fontossági mértékek (Rausand és mtsai. 2020).
- Ruijters, E., Stoelinga, M. (2015): Fault tree analysis: A survey of the state-of-the-art. *Computer Science Review* – a statikus és dinamikus hibafák modern módszereinek áttekintése (Ruijters és Stoelinga 2015).
- Mosleh, A. et al. (1988): *Procedures for Treating Common Cause Failures in Safety and Reliability Studies* (NUREG/CR-4780) (Mosleh és mtsai. 1988).

11.

Eseményfa, rétegzett védelem és csokornyakkendő-modell



A hibafa egyetlen nemkívánatos esemény okait bontja le. A biztonsági elemzés másik fele a *következményeké*: ha a nemkívánatos esemény bekövetkezett, milyen kimenetekhez vezethet, és ezeket mely védelmek akadályozzák meg vagy enyhítik? Erre a kérdésre az eseményfa felel, amely egy kezdő eseményből kiindulva, a védelmek sikerén vagy kudarcán elágazva írja le a lehetséges eseménysorozatokat. A folyamatiparban az eseményfa egyszerűsített, félkvantitatív változata, a rétegzett védelem elemzése (LOPA) lett a biztonsági funkciók integritási követelményének meghatározására a legelterjedtebb módszer. A csokornyakkendő-modell pedig a hibafa és az eseményfa logikáját egyetlen, szemléletes képbe foglalja. A fejezet végén azt is áttekintjük, hogyan kapcsolódnak egymáshoz az eddig megismert modellek.

Tanulási eredmények

A fejezet végére képes leszel

- egy kezdő eseményhez eseményfát szerkeszteni, és kiszámolni a kimenetek gyakoriságát;
- felismerni és kezelni az eseményfa ágai közötti függőségeket, és az eseményfát hibafákkal összekapcsolni;
- LOPA-elemzést végezni: azonosítani a független védelmi rétegeket, és meghatározni egy biztonsági funkció szükséges PFD-jét és SIL-jét;
- csokornyakkendő-diagramot készíteni a megelőző és a következménycsökkentő gátakkal és az eszkalációs tényezőkkel;
- kiválasztani egy elemzési kérdéshez a megfelelő modellt (blokkdiagram, hibafa, eseményfa, csokornyakkendő, Markov-modell).

11.1. Az eseményfa

Az eseményfa-elemzés (*event tree analysis*, ETA) **induktív, előre haladó** módszer: egy **kezdő eseményből** (*initiating event*) indul, és időrendben végigköveti, hogy az egymást követő biztonsági funkciók, gátak vagy feltételek sikere vagy kudarca milyen **eseménysorozatokhoz** és **kimenetekhez** vezet. A módszer a nukleáris ipar 1975-ös reaktorbiztonsági tanulmányában (WASH-1400) vált a valószínűségi biztonsági elemzés alappillérévé, ahol a kezdő eseményekből kiinduló eseményfák ágaihoz a biztonsági

rendszerek hibafái kapcsolódtak (U.S. Nuclear Regulatory Commission 1975). Nemzetközi szabványa az IEC 62502 (IEC 2010b).

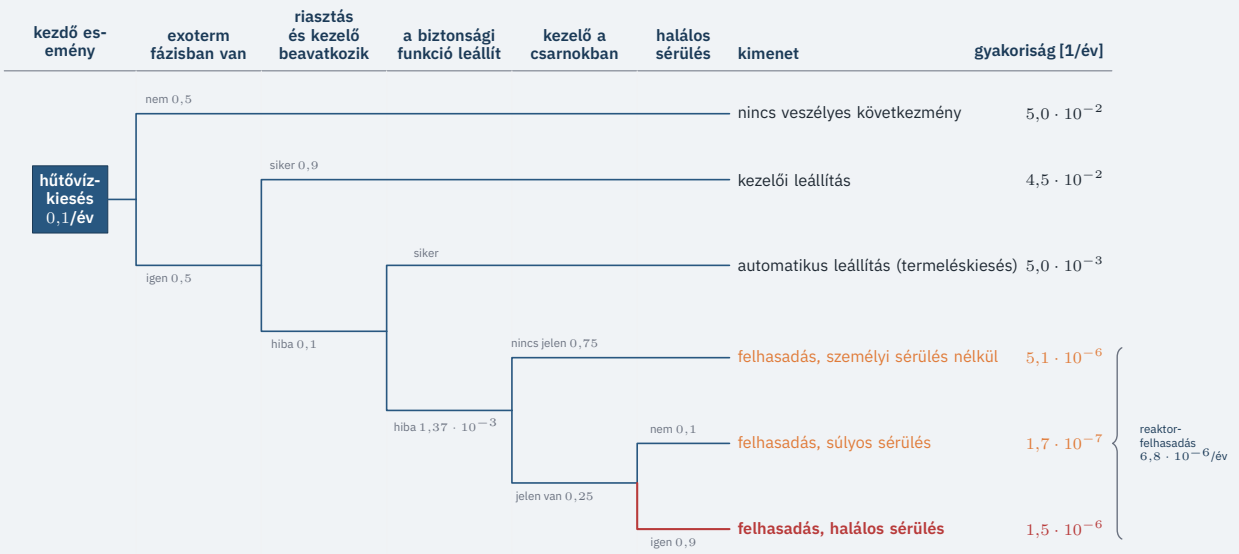
Az eseményfa szerkesztésének lépései (IEC 2010b; Rausand és mtsai. 2020):

1. **A kezdő esemény meghatározása:** egy olyan esemény, amely a védelmek közbelépése nélkül nemkívánatos következményhez vezetne (például hűtésvesztés, csőtörés, egy személy belépése a robot munkaterébe). A kezdő eseményeket a veszélyelemzés (HAZOP, FMEA) adja (5. fejezet); gyakoriságukat évenkénti előfordulásban adjuk meg.
2. **A biztonsági funkciók és feltételek azonosítása:** azok a gátak (automatikus védelmek, kezelői beavatkozás, passzív eszközök) és feltételek (például gyújtóforrás jelenléte, emberek jelenléte), amelyek befolyásolják a kimenetet. Ezek alkotják az eseményfa **fejléceit**, *időrendi sorrendben*.
3. **A fa felépítése:** minden fejlécnél két ág (siker és kudarc, vagy igen és nem); megállapodás szerint a felső ág a kedvező. Ha egy funkció kudarca után a további fejlécek már nem befolyásolják a kimenetet, az ág nem ágazik tovább.
4. **A kimenetek leírása és besorolása:** minden ág végén egy kimenet (például biztonságos leállás, termeléskiesés, kibocsátás, halálos sérülés).
5. **Számszerűsítés:** egy eseménysorozat gyakorisága a kezdő esemény gyakoriságának és az ágak *feltételes* valószínűségeinek szorzata,

$$f_{\text{sorozat}} = f_{\text{IE}} \cdot \prod_{k} P(\text{ág}_k \mid \text{az előző ágak}). \tag{11.1}$$

6. **Értékelés:** a kimenetek gyakoriságának összevetése a kockázati kritériumokkal (3. fejezet), az azonos kimenetek gyakoriságának összegzése, és a domináns sorozatok azonosítása.

> **ReaktorVéd: a hűtésvesztés eseményfája**
>
> A kezdő esemény a hűtővíz kiesése, évi $0{,}1$ gyakorisággal (11.1. ábra). Az első fejléc egy *feltétel*: az esetek felében a reaktor nem az exoterm fázisban van, ilyenkor a hűtés kiesése nem veszélyes. A második fejléc a magashőmérséklet-riasztás és a kezelői beavatkozás (a kezelő leállítja az adagolást és elindítja a tartalék hűtővízszivattyút), amely tíz esetből egyszer nem sikerül. A harmadik a biztonsági funkció, amelynek igénykori meghibásodási valószínűsége a 13. fejezet szerint $1{,}37 \cdot 10^{-3}$. A hasadótárcsát (RD-101) ebben a változatban nem számítjuk be védelemként, mert a HAZOP szerint méretezése kétfázisú áramlásra nem igazolt (5. fejezet). Ha a biztonsági funkció is kudarcot vall, a reaktor felhasadhat; a következmény attól függ, tartózkodik-e kezelő a reaktorcsarnokban (az idő 25%-ában), és ha igen, a sérülés az esetek 90%-ában halálos.



11.1. ábra. A ReaktorVéd hűtésvesztési forgatókönyvének eseményfája. A felső ág mindig a kedvező kimenet; a reaktorfelhasadás gyakorisága a három utolsó kimenet összege.

A reaktorfelhasadás gyakorisága $0{,}1 \cdot 0{,}5 \cdot 0{,}1 \cdot 1{,}37 \cdot 10^{-3} = 6{,}8 \cdot 10^{-6}$ évente, a halálos sérülésé $6{,}8 \cdot 10^{-6} \cdot 0{,}25 \cdot 0{,}9 = 1{,}5 \cdot 10^{-6}$ évente. Ezek az értékek szerepeltek a 3. fejezet egyéni kockázati számításában (3.1. táblázat). Az eseményfa a gyakoribb, de kevésbé súlyos kimeneteket is megmutatja: az automatikus leállítás évi $5 \cdot 10^{-3}$ gyakorisága például a termelési veszteség és a biztonsági funkció igénygyakoriságának becsléséhez kell.

11.1.1. Függőségek az ágak között

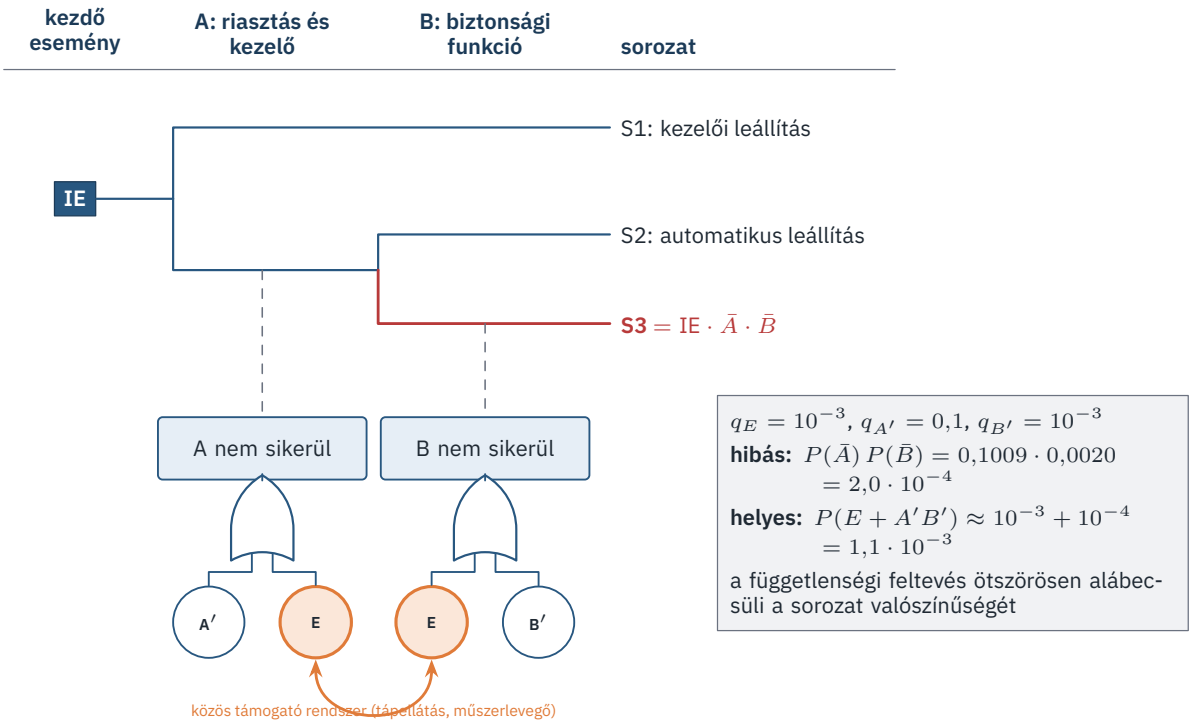
Az 11.1. egyenletben a valószínűségek *feltételesek*. Ha a fejlécek egymástól függetlenek, a feltételes valószínűség egyszerűen a funkció saját meghibásodási valószínűsége. A valóságban azonban a fejlécek gyakran közös támogató rendszerekre épülnek (tápellátás, műszerlevegő, hűtés, közös érzékelő vagy közös szoftver), és ilyenkor a függetlenségi feltevés súlyos alábecsléshez vezet.

A helyes megoldás az, hogy minden fejléchez hibafát készítünk, és az eseménysorozat valószínűségét az *összekapcsolt* hibafák minimális vágataiból számoljuk (11.2. ábra). A nukleáris gyakorlatban ezt a megközelítést „kis eseményfa, nagy hibafa” néven ismerik: az eseményfa csak a fő funkciókat tartalmazza, a függőségeket pedig a hibafák közös alapeseményei hordozzák (U.S. Nuclear Regulatory Commission 1975; Rausand és mtsai. 2020). A másik lehetőség, a „nagy eseményfa, kis hibafa”, a támogató rendszereket külön fejlécekként veszi fel, és a további ágakat ezek állapotától függően számolja.

A jelenség ugyanaz, amellyel a 10. fejezetben a ReaktorVéd tápegységeinél találkoztunk: a redundancia csak annyit ér, amennyire a rétegek függetlenek. Ezért a rétegzett védelem elemzésének egyik legfontosabb követelménye a rétegek függetlenségének igazolása.

KoboCella: eseményfa gyakori igény mellett

A kollaboratív robotcellában a kezdő esemény az, hogy a szerelő a robot gyors mozgása közben belép a közös munkatérbe. Ez műszakonként akár többször is megtörténik, ezért a kezdő esemény gyakorisága évente több száz vagy ezer is lehet. Az első fejléc a sebesség- és távolságfelügyelet (a lézerszkenner érzékeli a

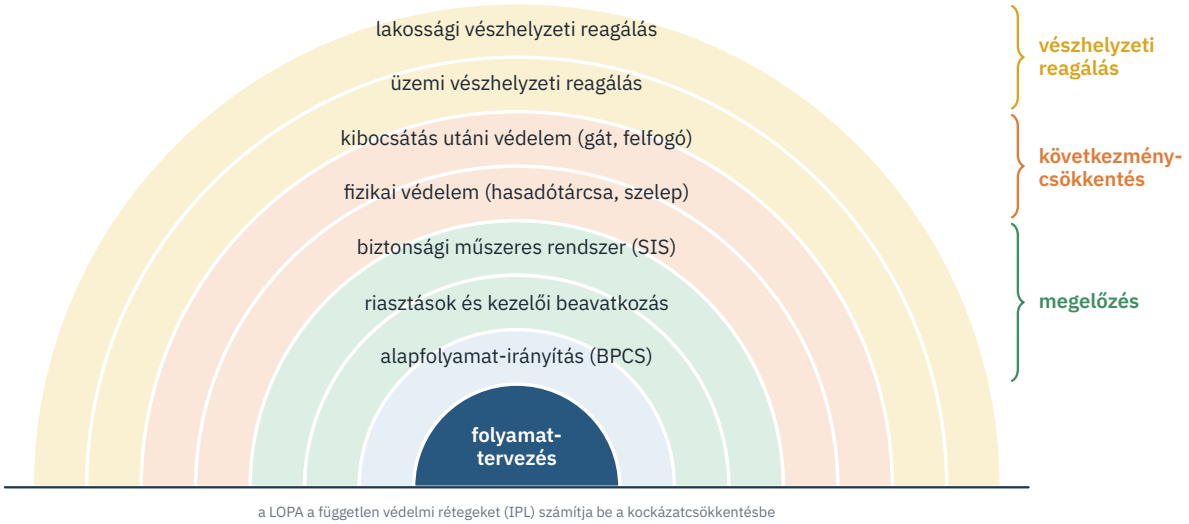


11.2. ábra. Az eseményfa fejléceihez kapcsolt hibafák. Ha két fejléc közös támogató rendszerre épül, az eseménysorozat valószínűségét az összekapcsolt hibafákból kell számolni; a függetlenségi feltevés ebben a példában ötszörösen alábecsüli a kockázatot.

szerelőt, és a robot lelassít), a második az erő- és teljesítménykorlátozás (ha mégis ütközés történik, az erő az ISO/TS 15066 szerinti biomechanikai határérték alatt marad) (ISO 2016). A kimenetek: nincs érintkezés; érintkezés megengedett erővel; érintkezés nagy sebességgel (sérülés).
Az eseményfa logikája ugyanaz, mint a ReaktorVédnél, de a számok természete más. Ha a kezdő esemény évente ezerszer fordul elő, egy 10^{-3}-as igénykori meghibásodási valószínűségű védelem évente egyszer hibázna. Ilyen gyakori igénynél a védelem **nagy igénybevételű** vagy **folyamatos üzemmódban** működik, és a mérvadó jellemző a veszélyes meghibásodás *gyakorisága* (PFH, óránként), nem az igénykori valószínűség (13. fejezet). A gépbiztonságban ezért a teljesítményszintet (PL) és a SIL-t is óránkénti meghibásodási gyakorisággal adják meg (ISO 2023; IEC 2021).

11.2. Rétegzett védelem elemzése (LOPA)

A folyamatiparban a veszélyes forgatókönyveket több, egymástól független védelmi réteg óvja (11.3. ábra). A legbelső réteg maga a folyamattervezés (az inherens biztonság), ezt követi az alapfolyamat-irányító rendszer (BPCS), a riasztások és a kezelői beavatkozás, a biztonsági műszeres rendszer (SIS), a fizikai védelmek (biztonsági szelep, hasadótárcsa), a kibocsátás utáni védelmek (gátak, felfogó medence), végül az üzemi és a lakossági vészhelyzeti reagálás (IEC 2016b; Center for Chemical Process Safety 2001). A rétegek a megelőzés, a következménycsökkentés és a vészhelyzeti reagálás három csoportjába sorolhatók, ami az energia–gát modell (1.4.2. szakasz) és a Reason-féle svájcisajt-modell (1. fejezet) mérnöki megfelelője.

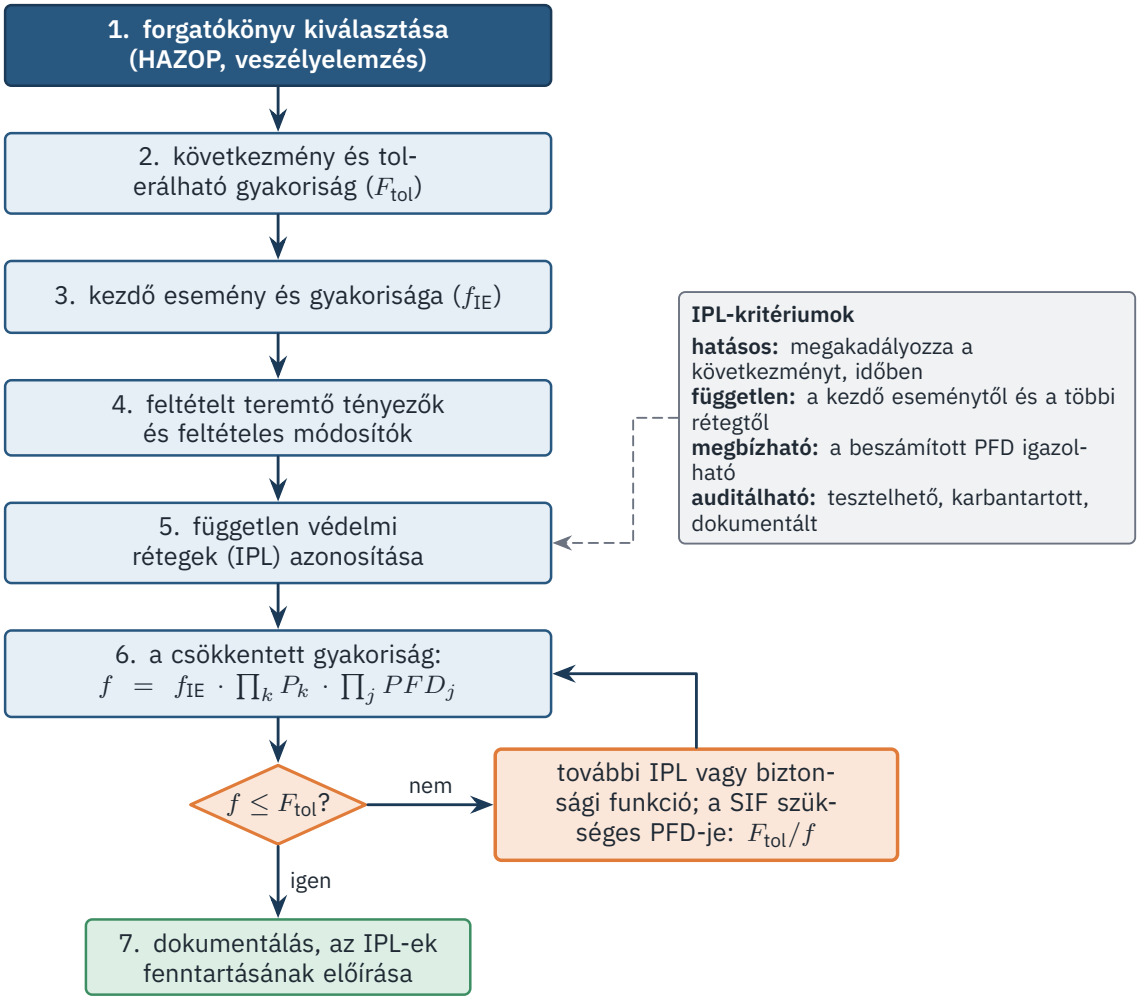


11.3. ábra. A folyamatipari létesítmények tipikus védelmi rétegei (az IEC 61511 és a CCPS nyomán). A LOPA csak azokat a rétegeket számítja be, amelyek a független védelmi réteg kritériumainak megfelelnek.

A **rétegzett védelem elemzése** (*layer of protection analysis*, LOPA) a CCPS 2001-es kézikönyvével vált általánossá (Center for Chemical Process Safety 2001), és az IEC 61511-3 F melléklete a SIL-meghatározás egyik elfogadott módszereként ismerteti (IEC 2016b). A LOPA lényegében egyetlen ágra szűkített, nagyságrendi becslésekkel dolgozó eseményfa: egy kezdő eseménytől egyetlen súlyos következményig követi a forgatókönyvet, és csak azt kérdezi, hogy a független védelmi rétegek együtt elegendő kockázatcsökkentést adnak-e. Félkvantitatív módszer: a számok nagyságrendi pontosságúak, de a következtetései ellenőrizhetők és következetesek, ami a kockázati mátrixokhoz képest nagy előny (3.3. szakasz).

A LOPA lépéseit a 11.4. ábra foglalja össze. A forgatókönyvet rendszerint a HAZOP adja. A tolerálható gyakoriságot a vállalat kockázati kritériumai határozzák meg a következmény súlyossága szerint.

A kezdő esemény gyakoriságát és a védelmi rétegek PFD-jét a vállalati vagy iparági táblázatokból veszik (11.1. táblázat). A **feltételt teremtő tényezők** (*enabling conditions*) annak a valószínűségét adják meg, hogy a rendszer a veszélyes állapotban van (például a reaktor exoterm fázisban), a **feltételes módosítók** (*conditional modifiers*) pedig a következmény bekövetkezésének feltételeit (gyújtás valószínűsége, emberek jelenléte, halálos kimenetel valószínűsége) (Center for Chemical Process Safety 2015).



11.4. ábra. A LOPA lépései. Ha a független védelmi rétegek nem csökkentik elégségesen a forgatókönyv gyakoriságát, a hiányzó kockázatcsökkentést egy biztonsági funkciónak kell biztosítania, és ebből adódik a szükséges SIL.

11.2.1. Független védelmi rétegek

A LOPA kulcsfogalma a **független védelmi réteg** (*independent protection layer*, IPL). Egy eszköz, rendszer vagy intézkedés csak akkor számít IPL-nek, ha négy feltételt teljesít (Center for Chemical Process Safety 2001, 2015):

- **hatásos:** önmagában képes megakadályozni a vizsgált következményt, és elég gyorsan működik ahhoz, hogy a folyamat még a veszélyes állapot előtt megálljon;

- **független:** működése nem függ a kezdő eseménytől és a forgatókönyvben beszámított többi IPL-től (nincs közös érzékelő, logika, beavatkozó, tápellátás vagy közös emberi hibaforrás);
- **megbízható:** a beszámított PFD a tervezés, a tesztelés és a tapasztalat alapján igazolható;
- **auditálható:** rendszeresen tesztelik és karbantartják, és ez dokumentált.

A kezelői beavatkozás csak akkor IPL, ha a riasztás független a kezdő eseménytől, a kezelőnek elegendő ideje van (jellemzően legalább tíz perc) a felismerésre, a döntésre és a beavatkozásra, és a teendőt írásos utasítás rögzíti, amelyet rendszeresen gyakorolnak (Center for Chemical Process Safety 2015). Az IEC 61511 azt is kiköti, hogy a BPCS-nek beszámított kockázatcsökkentés legfeljebb tízszeres lehet, ha a BPCS nincs a funkcionális biztonsági szabványok szerint tervezve (IEC 2016b). A puszta eljárások, a képzés és a felügyelet önmagukban nem IPL-ek; fontosak, de hatásuk nem számszerűsíthető úgy, mint egy fizikai rétegé.

11.1. táblázat. Jellemző LOPA-értékek a CCPS-irodalom nyomán; a vállalatok saját, igazolt értékeket használnak (Center for Chemical Process Safety 2001, 2015)

Kezdő esemény	Jellemző gyakoriság [1/év]
BPCS szabályozókör meghibásodása	10^{-1}
hűtővíz kiesése	10^{-1}
szivattyú tömítésének meghibásodása	10^{-1}–10^{-2}
kezelői hiba rutinfeladatnál (alkalmanként)	10^{-1}–10^{-3}
nyomástartó edény katasztrofális meghibásodása	10^{-5}–10^{-7}
Független védelmi réteg	**Jellemző PFD**
BPCS szabályozókör (ha nem a kezdő esemény része)	10^{-1}
riasztás és kezelői beavatkozás (legalább 10 perc)	10^{-1}
biztonsági szelep, hasadótárcsa (megfelelő méretezéssel)	10^{-2}
felfogó gát, felfogó medence	10^{-2}
biztonsági funkció (SIL 1 / SIL 2 / SIL 3)	10^{-1} / 10^{-2} / 10^{-3}

ReaktorVéd: LOPA és SIL-kiosztás

A 3. fejezet becslését most a LOPA formájában rögzítjük (11.2. táblázat).

11.2. táblázat. A ReaktorVéd hűtésvesztési forgatókönyvének LOPA-táblázata

LOPA-elem	Leírás	Érték
következmény	megfutó reakció, a reaktor felhasadása, halálos sérülés lehetősége	–
tolerálható gyakoriság	vállalati kritérium halálos kimenetelű forgatókönyvre	10^{-5}/év
kezdő esemény	a hűtővíz kiesése (szivattyúleállás, áramszünet, beragadt szelep)	0,1/év
feltételt teremtő tényező	a reaktor exoterm fázisban van	0,5
feltételes módosítók	nem alkalmazzuk (vállalati szabály)	1

IPL 1	TAH-103 riasztás és kezelői beavatkozás (a felfutás legalább 20 perc)	$0{,}1$
nem IPL	TIC-103 hőmérséklet-szabályozás: a kezdő esemény egyik oka	–
nem IPL	RD-101 hasadótárcsa: kétfázisú méretezése nem igazolt	–
csökkentett gyakoriság SIF nélkül	$0{,}1 \cdot 0{,}5 \cdot 0{,}1$	$5 \cdot 10^{-3}$/év
szükséges SIF	PFD $\leq 10^{-5}/(5 \cdot 10^{-3})$, RRF ≥ 500	**SIL 2**

A TAH-103 riasztás ugyanannak a BPCS-nek a része, mint a TIC-103 szabályozás. Beszámítható, mert a forgatókönyv kezdő eseményei között a riasztás eszközei (a TT-103 távadó és a kijelzés) nem szerepelnek; a BPCS vezérlőjének meghibásodását a csapat külön forgatókönyvként vizsgálja. Ez a mérlegelés jellemző a LOPA-ra: a függetlenséget minden forgatókönyvnél újra kell igazolni.
A táblázat egyben a biztonsági követelményspecifikáció kiindulópontja: a biztonsági funkciónak SIL 2 szinten, de a tartomány szigorúbb felében, PFD $\leq 2 \cdot 10^{-3}$ értékkel kell működnie. A 13. fejezet ellenőrzi, hogy a tervezett architektúra ezt teljesíti-e.

A feltételes módosítók óvatos használata

Ha a ReaktorVéd-forgatókönyvben a kezelő jelenlétét (0,25) és a halálos kimenetelt (0,9) is beszámítanánk, a csökkentett gyakoriság $1{,}1 \cdot 10^{-3}$/év lenne, és a biztonsági funkcióra csak PFD $\leq 8{,}9 \cdot 10^{-3}$ követelmény jutna. A SIL ugyan most is 2 maradna, de a követelmény négy és félszer enyhébb. A tartózkodási valószínűségek nehezen ellenőrizhetők és könnyen változnak (például egy indítás vagy egy karbantartás idején sokkal többen vannak a csarnokban), ezért sok vállalat nem engedi a feltételes módosítók használatát, vagy csak szigorú feltételekkel (Center for Chemical Process Safety 2015). Ugyanígy tilos ugyanazt a hatást kétszer beszámítani: ha a tolerálható gyakoriság már egy halálos kimenetelre vonatkozik, a halálos kimenetel valószínűségét nem szabad ismét szorzóként alkalmazni, ha pedig a forgatókönyvre, akkor éppen fordítva.

Buncefield LOPA-szemmel

Az 1. fejezetben bemutatott 2005-ös Buncefield-balesetben egy benzintartály túltöltését két, egymástól függetlennek tekintett réteg sem akadályozta meg. A tartály automatikus szintmérője az éjszaka folyamán beragadt (a vizsgálat szerint korábban is többször előfordult ilyen), a túltöltés elleni független szintkapcsoló pedig azért nem működött, mert a próbakarját működő helyzetben rögzítő lakatot nem szerelték fel; ennek szükségességét a gyártó nem közölte egyértelműen sem a telepítővel és a karbantartóval, sem az üzemeltetővel (Buncefield Major Incident Investigation Board 2008).
LOPA-szemmel mindkét hiba *rejtett* volt, és mindkettő az auditálhatóság követelményét sértette. A szintmérő ismétlődő beragadása nem vezetett a hiba okának feltárásához, a szintkapcsoló működésképtelenségét pedig éppen a tesztelési szerkezet hiányos ismerete okozta. A tanulság: egy réteg csak akkor ér annyit, amennyit beszámítunk, ha a tesztelése valóban feltárja a rejtett hibáit, és a karbantartás nem teremt közös hibaforrást. A vizsgálóbizottság ajánlásai nyomán a tárolótartályok túltöltés elleni védelmét a funkcionális biztonsági szabványok szerint kell tervezni és igazolni (Buncefield Major Incident Investigation Board 2008).

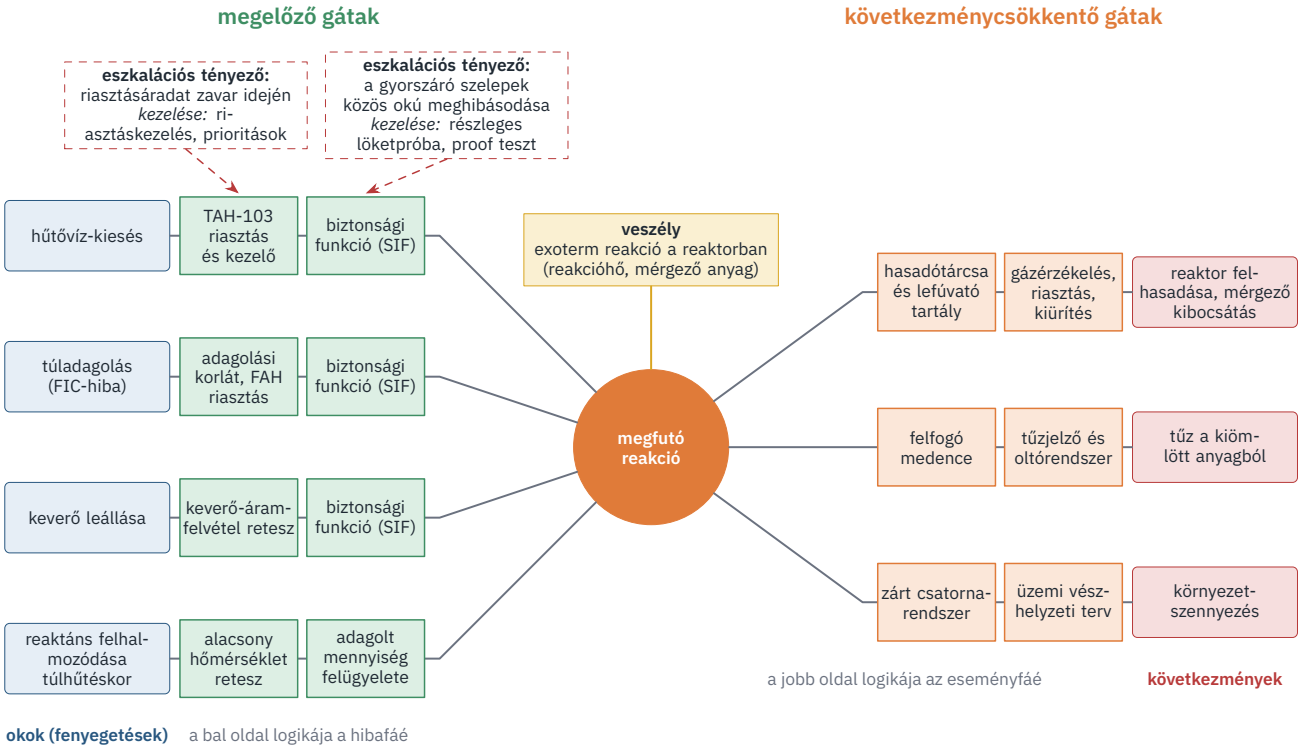
11.2.2. A LOPA erősségei és korlátai

A LOPA egyszerű, gyors, és a HAZOP eredményeire közvetlenül építhető; következetesebb, mint a kockázati gráf vagy a kalibrálatlan kockázati mátrix; a független védelmi rétegek kritériumai pedig

rákényszerítik a csapatot, hogy minden beszámított védelmet kritikusan vizsgáljon meg. Korlátai abból fakadnak, hogy egyszerűsítés: forgatókönyvenként egyetlen kezdő eseményt és egyetlen következményt vizsgál; a táblázatos értékek nagyságrendi pontosságúak; a rétegek közötti finomabb függőségeket és a közös okú meghibásodásokat nem modellezi; és ha ugyanaz a védelem sok forgatókönyvben szerepel, a teljes kockázat összegzése külön feladat. Ha a LOPA határesetet ad, vagy a forgatókönyv bonyolult, részletes eseményfa- és hibafa-elemzés szükséges (Center for Chemical Process Safety 2001, 2015).

11.3. A csokornyakkendő-modell

A **csokornyakkendő-diagram** (*bow-tie*) egyetlen ábrán mutatja meg egy nemkívánatos esemény okait, következményeit és az ezek közé épített gátakat (11.5. ábra). A módszer a vegyiparban jelent meg az 1970-es évek végén, és az 1990-es évektől az olaj- és gázipar, majd a légi közlekedés és az egészségügy is átvette (Center for Chemical Process Safety és Energy Institute 2018). A diagram középpontjában a **csúcsesemény** áll: az a pillanat, amikor a veszély feletti uralom elvész (például „megfutó reakció”, „a jármű elhagyja a sávját”). Fölötte a **veszély**, a kár forrása (2. fejezet). Bal oldalon az **okok** (fenyegetések), jobb oldalon a **következmények** láthatók. Az okok és a csúcsesemény közé a **megelőző gátak**, a csúcsesemény és a következmények közé a **következménycsökkentő gátak** kerülnek. Az **eszkalációs tényezők** (*escalation factors*) azok a körülmények, amelyek egy gátat gyengítenek vagy hatástalanná tesznek; ezek ellen további intézkedések védenek.



11.5. ábra. A ReaktorVéd megfutó reakciójának csokornyakkendő-diagramja. A biztonsági funkció három okkal szemben is gát, ezért közös okú meghibásodása különösen súlyos; az eszkalációs tényezők a gátak gyengülésének lehetséges okait mutatják.

A diagram bal oldala logikailag egy hibafa, amelynek csúcsa alatt VAGY-kapu áll (bármelyik ok elég), és minden ok alatt ÉS-kapu (az oknak és az összes gátjának kudarca együtt szükséges). A jobb oldal

logikailag egy eseményfa, amelynek fejlécei a következménycsökkentő gátak. A csokornyakkendő tehát nem új matematikai modell, hanem a két meglévő szemléletes, *kommunikációs* összekapcsolása. Ebben rejlik az ereje és a korlátja is: a vezetők, a kezelők és a karbantartók számára érthető, megmutatja, melyik gát melyik veszély ellen véd, és mire kell a karbantartásnak, a képzésnek és az ellenőrzésnek figyelnie. A számszerű kockázatot viszont csak akkor adja meg, ha a két oldalt valódi hibafává és eseményfává alakítjuk.

A jó csokornyakkendő-diagram néhány szabálya (Center for Chemical Process Safety és Energy Institute 2018):

- A **gát** teljes, önállóan hatásos védelem. Egy aktív gát három elemből áll: érzékel, dönt és beavatkozik (például távadó, logika és szelep); ha ezek bármelyike hiányzik, nem gát, hanem egy gát része.
- A **tevékenységek** (képzés, karbantartás, eljárások) nem gátak; ezek a gátak fenntartását szolgálják, és az eszkalációs tényezők kezeléseként jelennek meg.
- Minden oknak legyen legalább egy, de lehetőleg több független gátja, és a gátak legyenek függetlenek az októl és egymástól.
- Ha ugyanaz a gát több ok ellen is véd (a ReaktorVéd biztonsági funkciója három ok ellen), annak kudarca több utat is megnyit; az ilyen gátak különös figyelmet érdemelnek.

A csokornyakkendő-diagram a **gátak állapotának nyomon követésére** is alkalmas. Ha minden gáthoz teljesítménymutatót rendelünk (tesztelési lefedettség, elmaradt karbantartások, kiiktatások ideje), a diagram a működő üzem kockázatának élő térképévé válik. Ezt a gondolatot a biztonsági érvelés (17. fejezet) általánosítja.

11.4. A modellek kapcsolata és választása

Az eddigi fejezetek öt logikai modellt mutattak be. Ezek nem versenytársai, hanem különböző nézőpontjai ugyanannak a rendszernek (11.3. táblázat).

11.3. táblázat. Az elemzési modellek összevetése

Modell	Irány	Fő kérdés	Erőssége	Korlátja
megbízhatósági blokkdiagram	a működés útjai	mikor működik a funkció?	szemléletes, gyors	függetlenség, statikus
hibafa	deduktív, okok felé	mi okozza a csúcseseményt?	kombinációk, közös okok	egy csúcsesemény, statikus
eseményfa	induktív, következmények felé	mi lesz a kezdő esemény után?	időrend, kimenetek	ágak közti függőségek
csokornyakkendő	okok és következmények	mely gátak védenek?	kommunikáció, gátkezelés	önmagában nem kvantitatív
Markov-modell	állapotok és átmenetek	hogyan változik a rendszer időben?	javítás, sorrend, tesztelés	állapotrobbanás

A modellek egymásba alakíthatók. A blokkdiagram soros kapcsolata a hibafában VAGY-kapu, a párhuzamos kapcsolata ÉS-kapu (a meghibásodások szintjén), a hibafa minimális útjai pedig a blokkdiagram útjai (10.8. szakasz). Az eseményfa minden eseménysorozata egy hibafa csúcseseményeként írható fel: a kezdő esemény és a kudarcot vallott fejlécek hibafáinak ÉS-kapcsolata. A csokornyakkendő

bal oldala hibafa, jobb oldala eseményfa. A sorrendfüggő, javítható vagy időben változó viselkedés ott kezdődik, ahol a statikus logikai modellek véget érnek; ezt a 12. fejezet Markov-modelljei írják le.

A gyakorlati választás a kérdéstől függ. A tervezés korai szakaszában, a koncepciók összevetésére a blokkdiagram a leggyorsabb. Egy súlyos csúcsesemény okainak mélyreható feltárására és a biztonsági funkciók igazolására a hibafa a megfelelő. A forgatókönyvek kimeneteinek és gyakoriságainak meghatározására az eseményfa, a SIL-kiosztásra a LOPA, a vezetőkkel és a kezelőkkel való kommunikációra és a gátak üzemeltetés közbeni követésére a csokornyakkendő. A teljes kép sokszor csak a modellek együttes használatával áll össze.

Gyakorlat

A gyakorlati útmutató 9. laborjában egy böngészős eseményfa- és LOPA-alkalmazással elemezheted a ReaktorVéd forgatókönyveit: a kezdő esemény, a feltételek és a független védelmi rétegek változtatásával azonnal látod a kimenetek gyakoriságát és a biztonsági funkció szükséges SIL-jét. Az alkalmazás figyelmeztet, ha egy védelmet a függetlenségi kritériumok sérülése miatt nem lehet beszámítani.

Összefoglalás

- Az eseményfa induktív módszer: egy kezdő eseménytől, a védelmek sikerén és kudarcán elágazva, időrendben követi az eseménysorozatokat; egy sorozat gyakorisága a kezdő esemény gyakoriságának és a feltételes valószínűségeknek a szorzata.
- Az ágak közötti függőségeket (közös támogató rendszerek) a fejlécekhez kapcsolt hibafákkal kell kezelni; a függetlenségi feltevés nagyságrendi alábecsléshez vezethet.
- A LOPA egyetlen ágra szűkített, nagyságrendi eseményfa; a független védelmi rétegek (hatásos, független, megbízható, auditálható) kockázatcsökkentését veti össze a tolerálható gyakorisággal, és ebből adja a biztonsági funkció szükséges SIL-jét.
- A feltételes módosítókat óvatosan kell használni, és egyetlen hatás sem számítható be kétszer.
- A csokornyakkendő-diagram egy nemkívánatos esemény okait, következményeit, megelőző és következménycsökkentő gátait, valamint az eszkalációs tényezőket mutatja; bal oldala hibafa, jobb oldala eseményfa.
- A blokkdiagram, a hibafa, az eseményfa, a csokornyakkendő és a Markov-modell ugyanannak a rendszernek különböző nézetei, és egymásba alakíthatók.

Önellenőrzés

1. Egy éghető folyadék szivárgásának gyakorisága évi 10^{-2}. Azonnali gyújtás valószínűsége 0,1; késleltetett gyújtásé (ha nincs azonnali) 0,3. Rajzold fel az eseményfát, és számold ki a jet tűz (azonnali gyújtás), a gőzfelhő-robbanás (késleltetett gyújtás) és a gyújtás nélküli kibocsátás gyakoriságát!
2. Miért nem számítható be független védelmi rétegként a BPCS szabályozókör, ha a kezdő esemény éppen ennek a körnek a meghibásodása?
3. Tegyük fel, hogy a ReaktorVéd hasadótárcsájának kétfázisú méretezését igazolták, és a tárcsa PFD $= 10^{-2}$ független védelmi rétegként beszámítható. Mekkora PFD-t kell ekkor teljesítenie a biztonsági funkciónak? Szükség van-e egyáltalán SIL-besorolású funkcióra?
4. Mi a különbség a gát és a tevékenység között a csokornyakkendő-diagramon? Hova kerül a „kezelők képzése”?
5. Két fejléc közös tápellátásra épül. A tápellátás kiesésének valószínűsége 10^{-3}, a fejlécek egyéb okú meghibásodásáé 10^{-2} és 10^{-2}. Mekkora a két fejléc együttes kudarcának valószínűsége? Mekkora lenne a függetlenségi feltevéssel?

Megoldás.

1. Jet tűz: $10^{-2} \cdot 0{,}1 = 10^{-3}$/év. Késleltetett gyújtás: $10^{-2} \cdot 0{,}9 \cdot 0{,}3 = 2{,}7 \cdot 10^{-3}$/év. Gyújtás nélküli kibocsátás: $10^{-2} \cdot 0{,}9 \cdot 0{,}7 = 6{,}3 \cdot 10^{-3}$/év. Ellenőrzés: a három összege 10^{-2}.

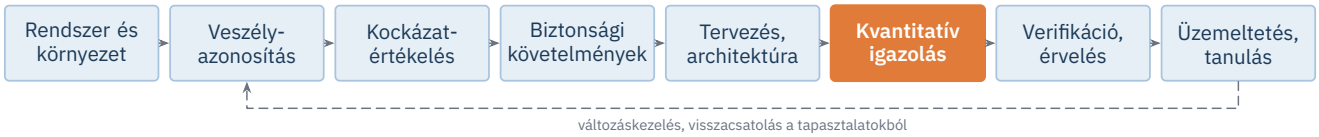
2. Mert nem független a kezdő eseménytől: ha a szabályozókör meghibásodott, ugyanaz a kör nem védhet a meghibásodása következményei ellen. Csak egy másik, független érzékelővel, logikával és beavatkozóval rendelkező kör lehet IPL.
3. A csökkentett gyakoriság SIF nélkül $5 \cdot 10^{-3} \cdot 10^{-2} = 5 \cdot 10^{-5}$/év, a szükséges PFD $\leq 10^{-5}/(5 \cdot 10^{-5}) = 0{,}2$, RRF ≥ 5. Ez a SIL 1 tartományánál (PFD $< 10^{-1}$) is enyhébb, így SIL-besorolású funkció nem feltétlenül szükséges; a kockázatcsökkentés azonban ekkor a hasadótárcsán és a lefúvató rendszeren múlik, amelynek méretezését, karbantartását és tesztelését szigorúan kell kezelni. A döntésnél azt is mérlegelni kell, hogy a hasadótárcsa a mérgező anyagot a lefúvató rendszerbe engedi, tehát a következmény más, de nem feltétlenül elhanyagolható.
4. A gát önállóan hatásos védelem (érzékel, dönt, beavatkozik), a tevékenység a gát fenntartását szolgálja, de önmagában nem állítja meg az eseménysort. A kezelők képzése tevékenység: az eszkalációs tényezők kezelései között jelenik meg (például a „riasztásáradat” eszkalációs tényező kezeléseként a kezelői gát mellett).
5. Közös táppal: $P \approx 10^{-3} + 10^{-2} \cdot 10^{-2} = 1{,}1 \cdot 10^{-3}$. Függetlenséggel: $(1{,}1 \cdot 10^{-2})^2 \approx 1{,}2 \cdot 10^{-4}$, ami majdnem egy nagyságrenddel kisebb.

Továbbolvasás

- Center for Chemical Process Safety (2001): *Layer of Protection Analysis: Simplified Process Risk Assessment.* AIChE – a LOPA alapműve (Center for Chemical Process Safety 2001).
- Center for Chemical Process Safety (2015): *Guidelines for Initiating Events and Independent Protection Layers in Layer of Protection Analysis.* AIChE–Wiley (Center for Chemical Process Safety 2015).
- CCPS – Energy Institute (2018): *Bow Ties in Risk Management: A Concept Book for Process Safety.* AIChE–Wiley (Center for Chemical Process Safety és Energy Institute 2018).
- IEC 62502:2010 – az eseményfa-elemzés szabványa (IEC 2010b); IEC 61511-3:2016 – a SIL-meghatározás módszerei, köztük a LOPA (IEC 2016b).
- Rausand, M. (2020): *System Reliability Theory*, 4. fejezet – eseményfák és a modellek kapcsolata (Rausand és mtsai. 2020).

12.

Markov-modellek és rendelkezésre állás



A blokkdiagram és a hibafa statikus modellek: azt írják le, *mely* meghibásodás-kombinációk okoznak bajt, de azt nem, hogy *mikor* és *milyen sorrendben*. Sok fontos kérdésre így nem adnak választ. Mekkora egy javítható rendszer rendelkezésre állása? Mennyit ér, ha a redundáns egység meghibásodása után gyorsan javítunk? Mennyire rontja a hideg tartalék előnyét az indulási hiba? Hogyan alakul egy rejtett hibákra hajlamos védelem meghibásodási valószínűsége a próbák között, és mi történik, ha a próba nem fedi fel az összes hibát? Ezekre a kérdésekre a Markov-modellek felelnek. Ebben a fejezetben megismerjük a folytonos idejű Markov-láncok alapjait, a rendelkezésre állás fogalmait, a javítható és a tartalékolt rendszerek modelljeit, végül a biztonsági rendszerek periodikus tesztelésének leírására szolgáló többfázisú Markov-modelleket.

Tanulási eredmények

A fejezet végére képes leszel

- egy rendszer viselkedését állapotokkal és átmeneti rátákkal leírni, és felírni az intenzitásmátrixot;
- kiszámolni egy javítható rendszer pillanatnyi és tartós rendelkezésre állását, valamint az átlagos meghibásodásig eltelt időt;
- Markov-modellel értékelni a javítás, a közös okú meghibásodás, az indulási hiba és a rejtett tartalékhiba hatását;
- periodikusan tesztelt biztonsági rendszerek PFD-jét többfázisú Markov-modellel számolni, és értelmezni a nem tökéletes proof teszt hatását;
- felismerni a Markov-modellek korlátait, és tudni, mikor van szükség szimulációra vagy más modellre.

12.1. Miért van szükség Markov-modellekre?

A statikus logikai modellek három feltevésre épülnek: az elemek két állapotúak, egymástól függetlenek, és a meghibásodások sorrendje nem számít (9. fejezet). A valódi rendszerekben ezek a feltevések gyakran sérülnek:

- **Javítás:** a meghibásodott elemet javítják, és a rendszer visszatér a működő állapotba. A rendszer állapota ilyenkor időben ide-oda vált.
- **Sorrend:** a hideg tartalék csak a fő egység kiesése után indul; egy átkapcsoló hibája csak akkor számít, ha a fő egység előtt következik be.

- **Állapotfüggő igénybevétel:** ha egy kétcsatornás rendszer egyik csatornája kiesik, a másikra nagyobb terhelés jut, és nagyobb rátával hibásodhat meg.
- **Tesztelés:** a rejtett hibák csak a próbák időpontjaiban derülnek ki, így a rendszer állapota a próbák ütemében változik.

A Markov-modell ezeket közvetlenül ábrázolja. A rendszer lehetséges állapotait csomópontok, az állapotok közötti átmeneteket pedig rátákkal címkézett nyilak jelölik. A Markov-modelleket az IEC 61165 szabványosítja (IEC 2006b), és az IEC 61508-6 a biztonsági funkciók PFD- és PFH-értékének számítására is elfogadott módszerként ismerteti (IEC 2010a).

12.2. A folytonos idejű Markov-lánc

A rendszer állapotát egy $X(t)$ valószínűségi változó írja le, amely véges sok érték valamelyikét veszi fel. A **Markov-tulajdonság** szerint a jövőbeli viselkedés csak a jelenlegi állapottól függ, attól nem, hogyan jutott a rendszer ide:

$$P\big(X(t+s)=j \mid X(t)=i,\ X(u),\ u<t\big) = P\big(X(t+s)=j \mid X(t)=i\big). \tag{12.1}$$

Ha az átmeneti valószínűségek az időben nem változnak (**homogén** lánc), minden állapotban az ott töltött idő exponenciális eloszlású, és az i állapotból a j állapotba vezető átmenetet egy állandó q_{ij} ráta jellemzi. Ez az exponenciális élettartam-modell (8. fejezet) természetes kiterjesztése: az emlékezetnélküliség az élettartamok, a Markov-tulajdonság a rendszerállapotok szintjén ugyanazt jelenti. A rátákból alkotott mátrix az **intenzitásmátrix** (generátormátrix) $\mathbf{Q}$: főátlón kívüli elemei az átmeneti ráták, főátlójában pedig $q_{ii} = -\sum_{j \neq i} q_{ij}$ áll, így minden sor összege nulla.

Az állapotvalószínűségek $\mathbf{p}(t) = [p_0(t), p_1(t), \dots]$ sorvektorára a **Kolmogorov-egyenletek** érvényesek:

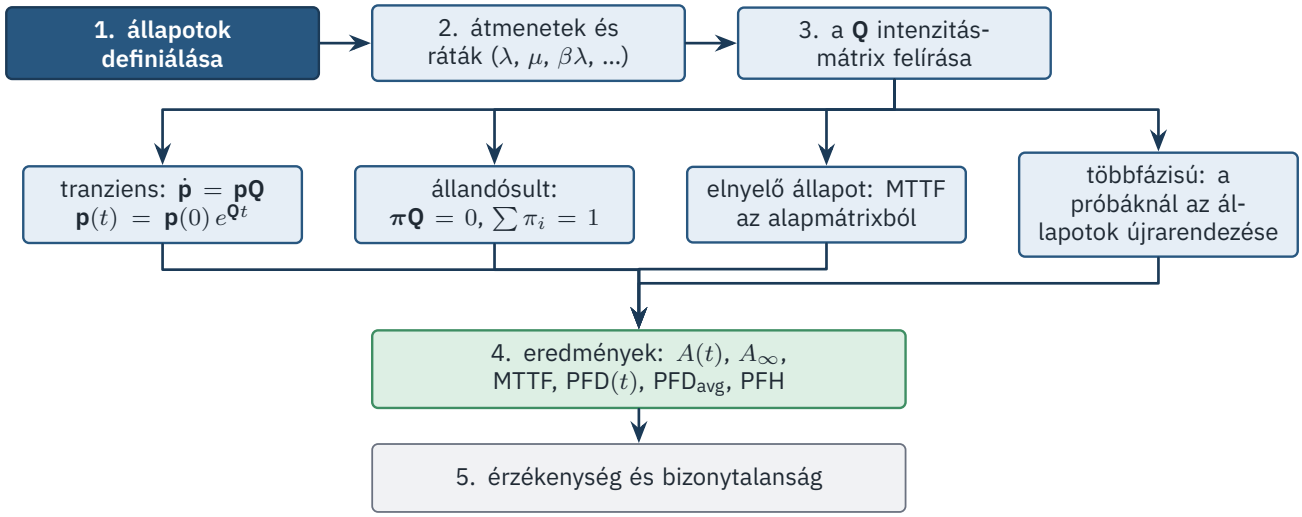
$$\frac{d\mathbf{p}(t)}{dt} = \mathbf{p}(t)\,\mathbf{Q}, \qquad \mathbf{p}(t) = \mathbf{p}(0)\,e^{\mathbf{Q}t}. \tag{12.2}$$

Három kérdéstípusra szokás választ keresni (12.1. ábra):

- **tranziens viselkedés:** az állapotvalószínűségek időfüggése, például a rendelkezésre állás $A(t)$ vagy a meghibásodási valószínűség $\mathrm{PFD}(t)$;
- **tartós (állandósult) viselkedés:** hosszú idő után az állapotvalószínűségek egy π határeloszláshoz tartanak, amely a $\pi\mathbf{Q} = \mathbf{0}$, $\sum_i \pi_i = 1$ lineáris egyenletrendszer megoldása;
- **első elérési idő:** ha a rendszerhiba állapotából nincs visszaút (**elnyelő állapot**), az odáig eltelt idő várható értéke a rendszer MTTF-je. Ha $\mathbf{Q}_T$ a nem elnyelő állapotokra szűkített mátrix, az $\mathbf{N} = -\mathbf{Q}_T^{-1}$ **alapmátrix** (i,j) eleme a j állapotban töltött várható idő az i állapotból indulva, így az MTTF az $\mathbf{N}$ kezdőállapothoz tartozó sorának összege.

12.3. A rendelkezésre állás

A javítható rendszerek legfontosabb jellemzője a **rendelkezésre állás** (*availability*): annak képessége, hogy a rendszer adott időpontban vagy időtartam alatt el tudja látni a funkcióját, feltéve, hogy a szükséges külső erőforrások rendelkezésre állnak (IEC 2015). Három változatát különböztetjük meg:



12.1. ábra. A Markov-modellezés lépései. A kérdéstől függően a tranziens, a tartós, az elnyelő állapotos vagy a többfázisú megoldásra van szükség.

- **pillanatnyi rendelkezésre állás** $A(t)$: annak valószínűsége, hogy a rendszer a t időpontban működőképes;
- **átlagos rendelkezésre állás** egy $[0, T]$ időszakra: $\bar{A}(T) = \frac{1}{T}\int_0^T A(t)\,dt$, a működőképes idő várható aránya;
- **tartós rendelkezésre állás** $A_\infty = \lim_{t\to\infty} A(t)$.

A rendelkezésre állás nem azonos a megbízhatósággal. A megbízhatóság $R(t)$ annak valószínűsége, hogy a rendszer $[0, t]$ alatt *egyszer sem* hibásodik meg; a rendelkezésre állás megengedi, hogy közben meghibásodjon és megjavítsák. Ezért mindig $A(t) \geq R(t)$. A rendelkezésre nem állás $U = 1 - A$ gyakran szemléletesebb: évi állásidőként (órában) is kifejezhető.

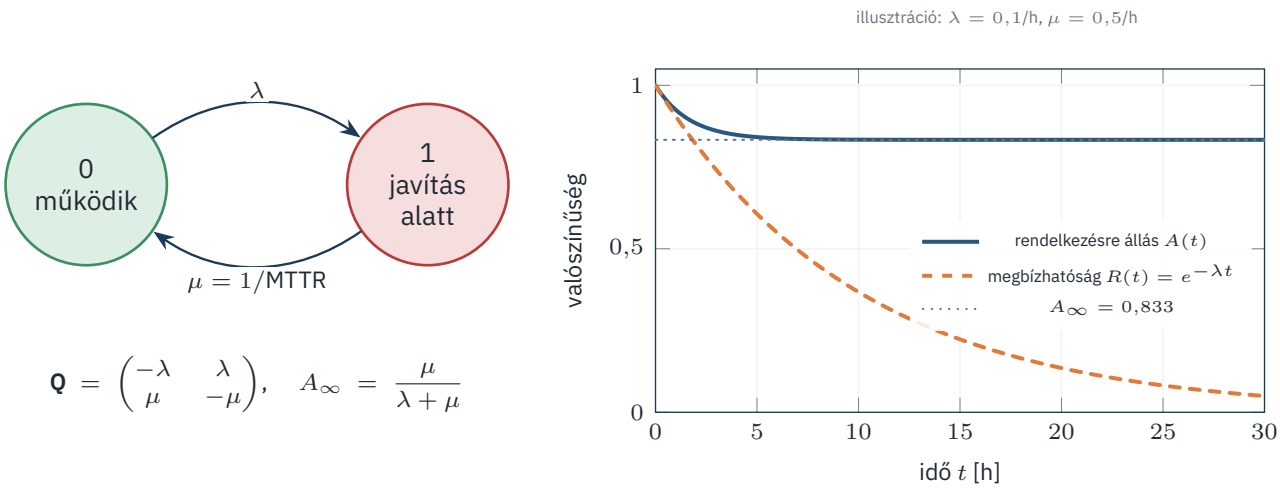
A gyakorlatban a javítási idő összetevői is számítanak. A **belső** (*inherent*) rendelkezésre állás csak a tényleges javítási időt veszi figyelembe; az **üzemi** (*operational*) rendelkezésre állás a hiba felismerésének, a szakember kiérkezésének, az alkatrész beszerzésének és az adminisztrációnak az idejét is (Rausand és mtsai. 2020). Egy biztonsági rendszer PFD-számításánál az MTTR-be mindezt bele kell érteni, ameddig a biztonsági funkció nem áll rendelkezésre.

12.3.1. Egyetlen javítható elem

A legegyszerűbb Markov-modellnek két állapota van: a 0 állapotban az elem működik, az 1 állapotban javítják (12.2. ábra). A meghibásodási ráta λ, a javítási ráta $\mu = 1/\mathrm{MTTR}$. A Kolmogorov-egyenletek megoldása $p_0(0) = 1$ kezdőfeltétellel

$$A(t) = \frac{\mu}{\lambda+\mu} + \frac{\lambda}{\lambda+\mu}\,e^{-(\lambda+\mu)t}, \qquad A_\infty = \frac{\mu}{\lambda+\mu} = \frac{\mathrm{MTTF}}{\mathrm{MTTF}+\mathrm{MTTR}}. \tag{12.3}$$

A rendelkezésre állás exponenciálisan, $1/(\lambda+\mu)$ időállandóval áll be a tartós értékére. Mivel a javítás általában sokkal gyorsabb a meghibásodásnál ($\mu \gg \lambda$), ez az időállandó közel az MTTR, vagyis a tartós állapot gyorsan beáll. Kis $\lambda\,\mathrm{MTTR}$ szorzatnál a rendelkezésre nem állás $U_\infty \approx \lambda\,\mathrm{MTTR}$. A $\mathrm{MTBF} = \mathrm{MTTF} + \mathrm{MTTR}$ a két egymást követő meghibásodás közötti átlagos idő.



12.2. ábra. Egyetlen javítható elem Markov-modellje és rendelkezésre állása. A rendelkezésre állás a tartós értékéhez tart, a megbízhatóság nullához; a görbék szemléltető rátákkal készültek.

> **Egy szivattyú rendelkezésre állása**
>
> Egy hűtővízszivattyú meghibásodási rátája $\lambda = 10^{-4}\,\text{h}^{-1}$ (MTTF = 10 000 óra), az átlagos javítási idő 8 óra. A tartós rendelkezésre állás $A_\infty = 0{,}125/(0{,}125+0{,}0001) = 0{,}99920$, a rendelkezésre nem állás $U_\infty = 7{,}99\cdot10^{-4}$, alig kevesebb, mint a $\lambda\,\text{MTTR} = 8\cdot10^{-4}$ közelítés. Az évi átlagos állásidő $7{,}99\cdot10^{-4}\cdot8760 \approx 7$ óra. Az időállandó $1/(\lambda+\mu) \approx 8$ óra, tehát a tartós érték egy-két nap alatt beáll.

12.4. Redundáns, javítható rendszerek

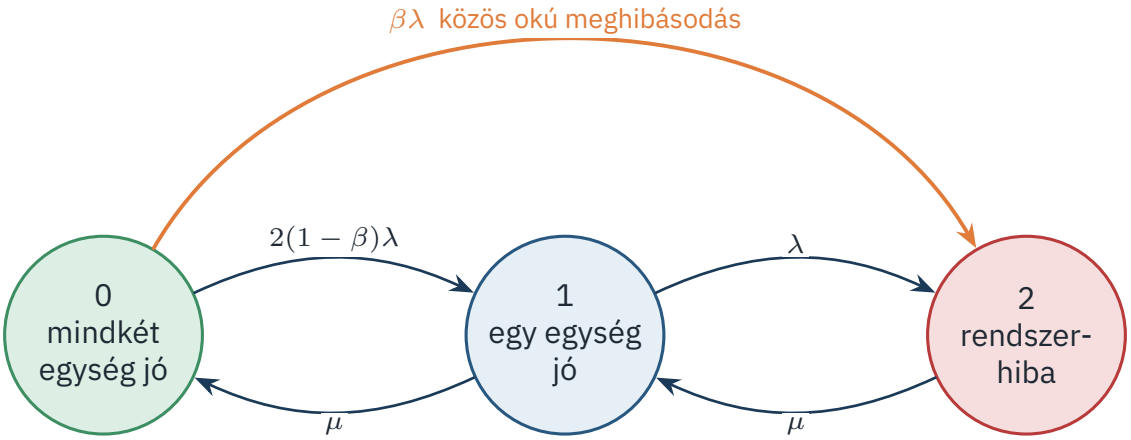
Két azonos, párhuzamosan működő egység (1oo2) Markov-modelljének három állapota van: mindkét egység jó, egy egység jó, a rendszer meghibásodott (12.3. ábra). Egy javítócsapat esetén a 0 állapotból 2λ rátával jutunk az 1 állapotba, onnan λ rátával a rendszerhibába, a javítás pedig mindkét lépést μ rátával fordítja vissza. A közös okú meghibásodást a β-tényezős modell (10.6. szakasz) szerint egy közvetlen, $\beta\lambda$ rátájú átmenet írja le a 0 állapotból a rendszerhibába; ilyenkor a független meghibásodás rátája $2(1-\beta)\lambda$.

Ha a 2 állapotot elnyelőnek tekintjük, az alapmátrixból a rendszer MTTF-je $\beta = 0$ esetén

$$\text{MTTF}_{1\text{oo}2} = \frac{3\lambda+\mu}{2\lambda^2}. \tag{12.4}$$

Javítás nélkül ($\mu = 0$) ez a 9. fejezetből ismert $1{,}5/\lambda$, javítással viszont nagyságrendekkel nagyobb lehet. A szivattyú adataival ($\lambda = 10^{-4}\,\text{h}^{-1}$, MTTR $= 8$ óra) a két szivattyúból álló rendszer MTTF-je javítás nélkül 15 000 óra, javítással $6{,}3\cdot10^6$ óra, azaz több mint hétszáz év. A redundancia igazi ereje tehát a *gyors javításban* rejlik: a rendszer csak akkor hibásodik meg, ha a második egység éppen a javítás néhány órája alatt esik ki.

A közös okú meghibásodás ezt az előnyt drámaian csökkenti. Már $\beta = 5\%$ mellett is az MTTF $1{,}9\cdot10^5$ órára (22 év) esik, harmincketted részére, a tartós rendelkezésre nem állás pedig $1{,}3\cdot10^{-6}$-ról $4{,}1\cdot10^{-5}$-re nő. A magyarázat egyszerű: a független kettős meghibásodás rátája körülbelül



egy javítócsapat, $\lambda = 10^{-4}$/h, MTTR $= 8$ h

	javítás nélkül	javítással, $\beta = 0$	javítással, $\beta = 5\%$
rendszer MTTF	$1{,}5 \cdot 10^{4}$ h	$6{,}3 \cdot 10^{6}$ h	$1{,}9 \cdot 10^{5}$ h
tartós rendelkezésre nem állás	–	$1{,}3 \cdot 10^{-6}$	$4{,}1 \cdot 10^{-5}$

12.3. ábra. Javítható 1oo2 rendszer Markov-modellje közös okú meghibásodással. A táblázat a javítás és a közös okú meghibásodás hatását mutatja.

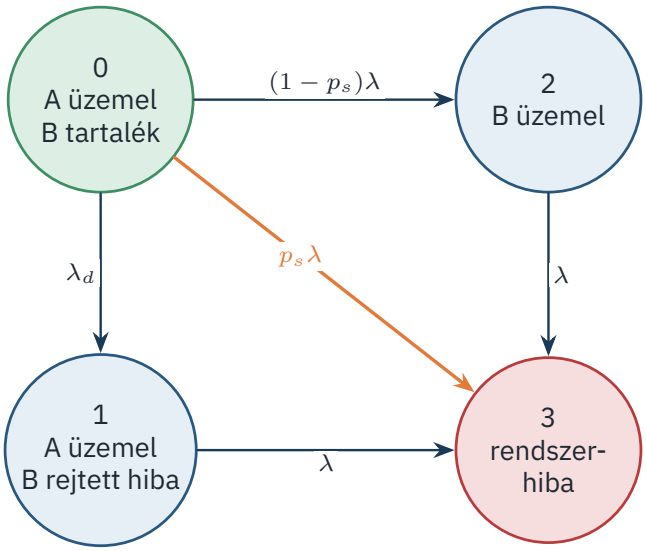
$2\lambda^2\,\mathrm{MTTR} = 1{,}6 \cdot 10^{-7}\,\mathrm{h}^{-1}$, a közös okúé $\beta\lambda = 5 \cdot 10^{-6}\,\mathrm{h}^{-1}$, harmincszor nagyobb. A javítható redundáns rendszerekben a közös okú meghibásodás szinte mindig uralja a rendszerhibát.

> **Egy vagy két javítócsapat?**
>
> A modell azt is megmutatja, mennyit ér egy második javítócsapat. Ha a rendszerhiba állapotában két csapat egyszerre javít, a 2 állapotból az 1 állapotba 2μ rátával lehet visszajutni. Ez a tartós rendelkezésre nem állást felére csökkenti, az MTTF-et viszont nem változtatja, mert az MTTF az első rendszerhibáig tart. A rendelkezésre állás és a megbízhatóság tehát különböző intézkedésekre érzékeny, és a kérdés megfogalmazásától függ, melyik a mérvadó.

12.4.1. A hideg tartalék valósághű modellje

A 9.5. szakasz szerint az ideális hideg tartalék MTTF-je $2/\lambda$, jobb, mint a melegé $(1{,}5/\lambda)$. A valóságban azonban a tartalék a bekapcsoláskor nem mindig indul el, és a várakozás alatt is meghibásodhat úgy, hogy ez csak az igénykor derül ki. A 12.4. ábra Markov-modellje mindkét hatást tartalmazza: a p_s indulási hiba közvetlenül a rendszerhibába vezet, a λ_d alvó (rejtett) meghibásodás pedig egy olyan állapotba, amelyben a fő egység még működik, de már nincs mögötte tartalék.

Két százalékos indulási hiba alig csökkenti az MTTF-et $(1{,}98/\lambda)$, de ha a tartalék a várakozás alatt a fő egység rátájának tizedével hibásodik meg, az MTTF már $1{,}89/\lambda$, öt százalék indulási hibával és ötödnyi alvó rátával pedig $1{,}79/\lambda$. A hideg tartalék előnye a melegével szemben így a felére olvad. Ha a tartalékot rendszeresen kipróbálják, a rejtett hibák időben kiderülnek; ennek modellezésére szolgálnak a következő szakasz többfázisú modelljei.



p_s: indulási hiba valószínűsége igénykor
λ_d: a tartalék rejtett (alvó) meghibásodási rátája

változat	MTTF
ideális hideg tartalék	$2{,}00/\lambda$
$p_s = 0{,}02$	$1{,}98/\lambda$
$p_s = 0{,}02$, $\lambda_d = 0{,}1\lambda$	$1{,}89/\lambda$
$p_s = 0{,}05$, $\lambda_d = 0{,}2\lambda$	$1{,}79/\lambda$
meleg tartalék (aktív 1oo2)	$1{,}50/\lambda$

12.4. ábra. A hideg tartalék Markov-modellje indulási hibával és rejtett tartalékhibával (javítás nélkül). Néhány százalékos indulási hiba és kis alvó meghibásodási ráta is jelentősen csökkenti a hideg tartalék előnyét.

12.5. Biztonsági rendszerek: rejtett hibák és periodikus tesztelés

A biztonsági rendszerek nagy része nyugalmi állapotban van, és csak igénykor kell működnie. Veszélyes meghibásodásaik egy része a diagnosztika révén azonnal kiderül (DD), más része rejtve marad (DU), amíg egy **proof teszt** vagy egy valódi igény fel nem fedi (6.5. szakasz). A rendszer állapota így két különböző időskálán változik: a meghibásodások folytonos időben, véletlenszerűen következnek be, a próbák viszont rögzített időpontokban, determinisztikusan történnek.

Ezt a **többfázisú Markov-modell** írja le. A próbák közötti szakaszban a rendszer egy közönséges Markov-láncként fejlődik, amelyben a DU-állapotokból nincs javítás. A próba pillanatában egy átrendezési mátrix a rejtett hibás állapotokat a javítás alatti vagy a hibátlan állapotba képezi le, és a következő szakasz ebből az eloszlásból indul. A $\mathrm{PFD}(t)$ a veszélyes állapotok valószínűségeinek összege; jellegzetes **fűrészfog** alakú görbe, amely minden próba után nullára (vagy közel nullára) esik vissza, és a következő próbáig nő. A biztonsági funkció jellemzője ennek időátlaga, a $\mathrm{PFD}_{\mathrm{avg}}$ (Rausand 2014; IEC 2010a).

ReaktorVéd: a gyorszáró szelepek többfázisú Markov-modellje

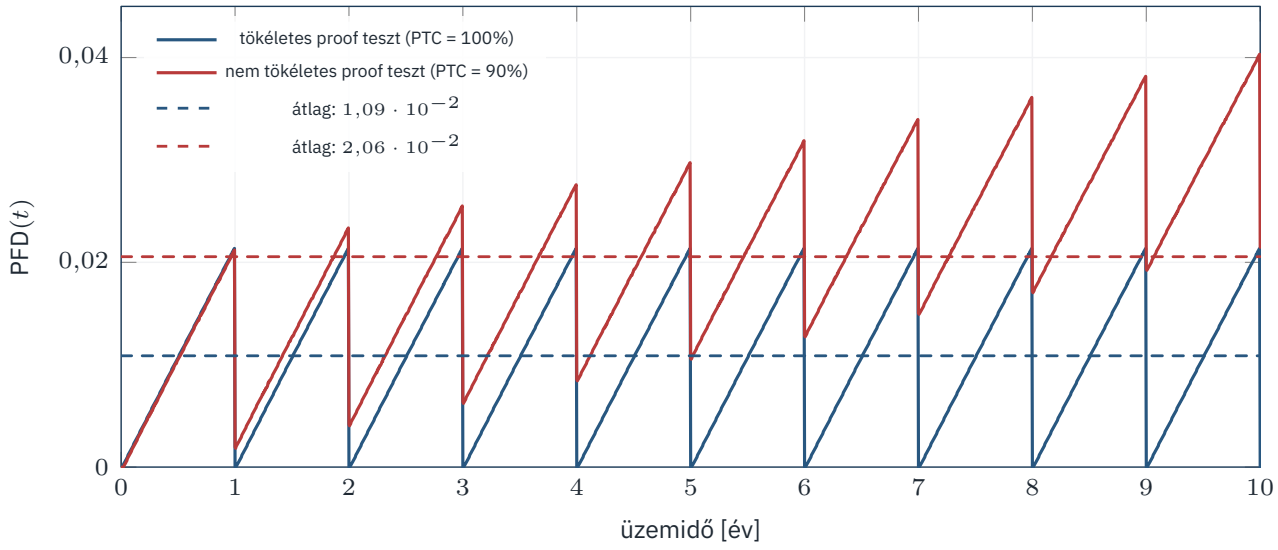
A két gyorszáró szelep (1oo2, $\lambda_{DU} = 2{,}5 \cdot 10^{-6}\,\mathrm{h}^{-1}$, $\beta = 10\%$, évenkénti proof teszt) három állapottal írható le: mindkét szelep jó; egy szelep rejtett hibás; mindkettő hibás (veszélyes). A 0 állapotból $2(1-\beta)\lambda_{DU}$ rátával jutunk az 1-be és $\beta\lambda_{DU}$ rátával a 2-be, az 1-ből λ_{DU} rátával a 2-be. A próba a rendszert visszaállítja a 0 állapotba.
A Markov-modellből a $\mathrm{PFD}(t)$ a próba előtti pillanatban $2{,}57 \cdot 10^{-3}$-ig nő, az időátlaga $1{,}22 \cdot 10^{-3}$. Az IEC 61508-6 egyszerűsített képletei (13. fejezet) $1{,}23 \cdot 10^{-3}$ értéket adnak; a különbség fél százalék alatti, és a képlet a biztonságos oldalon van. A Markov-modell előnye nem ebben az egyszerű esetben mutatkozik meg, hanem ott, ahol a képletek feltevései sérülnek: eltolt idejű próbák, nem tökéletes próba, a próba alatti kiiktatás, állapotfüggő javítás vagy több, egymástól függő alrendszer esetén.

12.5.1. A nem tökéletes proof teszt

A proof teszt ritkán fed fel minden rejtett hibát. Egy gyorszáró szelep éves próbája például ellenőrizheti, hogy a szelep zár-e, de azt nem feltétlenül, hogy a zárás a valódi üzemi nyomáskülönbség mellett is tömör-e. A próba **lefedettsége** (*proof test coverage*, PTC) a rejtett veszélyes meghibásodások azon hányada, amelyet a próba felfed. A fel nem fedett hibák csak egy alaposabb vizsgálatnál, például a T_M időközű nagyjavításnál vagy cserénél derülnek ki. Az átlagos PFD egycsatornás elemre ekkor közelítőleg

$$\mathrm{PFD}_{\mathrm{avg}} \approx \mathrm{PTC} \cdot \lambda_{DU} \left(\frac{T_1}{2} + \mathrm{MTTR} \right) + (1 - \mathrm{PTC}) \cdot \lambda_{DU} \frac{T_M}{2}. \tag{12.5}$$

A második tag a nem tesztelt hibák „lassan halmozódó” hozzájárulása, amely a hosszú T_M miatt már kis $1 - \mathrm{PTC}$ mellett is jelentős lehet (Rausand 2014; IEC 2010a).



12.5. ábra. Egy egycsatornás gyorszáró szelep meghibásodási valószínűsége tökéletes és 90%-os lefedettségű évenkénti proof teszt mellett, tízévenkénti nagyjavítással. A nem felfedett hibák miatt a fűrészfog alapvonala folyamatosan emelkedik, és az átlagos PFD közel kétszeresére nő.

A ReaktorVéd egycsatornás szelepére ($\lambda_{DU} = 2{,}5 \cdot 10^{-6}\,\mathrm{h}^{-1}$, $T_1 = 1$ év, $T_M = 10$ év) a tökéletes próba mellett $\mathrm{PFD}_{\mathrm{avg}} = 1{,}10 \cdot 10^{-2}$, 90%-os lefedettség mellett $2{,}08 \cdot 10^{-2}$, 70%-os mellett $4{,}05 \cdot 10^{-2}$ (12.5. ábra). Egy látszólag apró részlet, a próba módja, tehát akár négyszeresére is növelheti a PFD-t. Ezért az IEC 61511 megköveteli, hogy a proof teszt eljárásai írásban rögzítsék, mely meghibásodásokat fedik fel, és a PFD-számítás a tényleges lefedettséggel készüljön (IEC 2016b).

12.6. Megoldási módszerek és korlátok

Kis modellek (két-három állapot) zárt alakban is megoldhatók, mint az 12.3. és az 12.4. egyenlet. Nagyobb modelleknél numerikus módszereket használunk: a mátrixexponenciális függvény közvetlen számítását, az egyenletes randomizálást (*uniformization*) vagy a differenciálegyenlet-rendszer numerikus integrálását; a tartós eloszláshoz pedig egy lineáris egyenletrendszert oldunk meg (Trivedi és Bobbio 2017).

A Markov-modellek fő korlátja az **állapotrobbanás**: n kétállapotú elem 2^n állapotot ad, tíz elem már 1024-et, húsz elem több mint egymilliót. Ezt a szimmetriák kihasználásával (azonos elemek összevonása), a modell modulokra bontásával (a statikus részek hibafával, a dinamikus részek Markov-modellel), illetve a kis valószínűségű állapotok elhagyásával lehet kezelni. A második korlát az exponenciális tartózkodási idő feltevése. A determinisztikus időtartamok (például egy rögzített javítási idő vagy a periodikus próba) a többfázisú modellekkel, az általános eloszlások fázistípusú közelítéssel vagy szemi-Markov-modellekkel kezelhetők. A legbonyolultabb viselkedésekre (állapotfüggő stratégiák, korlátozott javítási erőforrások, öregedés) a sztochasztikus Petri-hálók és a Monte Carlo-szimuláció a rugalmasabb eszközök (Trivedi és Bobbio 2017; Rausand 2014).

KoboCella: biztonság kontra rendelkezésre állás

A kollaboratív cellában a biztonsági funkciók nemcsak a sérüléseket előzik meg, hanem a termelést is befolyásolják. Ha a lézerszkenner minden belépéskor *biztonsági megállást* vált ki, és a szerelő óránként hatszor lép be, a robot pedig minden alkalommal 20 másodpercre áll meg, egy egyszerű kétállapotú modell szerint a rendelkezésre állás $1/(1 + 6 \cdot 20/3600) = 0{,}968$, vagyis a műszakidő több mint 3%-a elvész. A sebesség- és távolságfelügyelet (*speed and separation monitoring*) ehelyett csak lassítja a robotot, amíg a szerelő a közelben van, így a veszteség kisebb (ISO 2016). A biztonsági funkciók téves vagy túl gyakori beavatkozása ráadásul arra ösztönözheti a dolgozókat, hogy megkerüljék a védelmet. A biztonság és a rendelkezésre állás együttes modellezése ezért a jó biztonsági tervezés része, nem pusztán gazdasági kérdés.

Gyakorlat

A gyakorlati útmutató 10. laborjában egy böngészős Markov-szerkesztővel állapotgráfot rajzolhatsz, és az alkalmazás kiszámolja a pillanatnyi és a tartós rendelkezésre állást, valamint az MTTF-et. A hozzá tartozó Colab-notebook a javítható 1oo2 rendszert, a hideg tartalékot és a gyorszáró szelepek többfázisú modelljét valósítja meg, és megmutatja a nem tökéletes proof teszt hatását a PFD fűrészfog-görbéjére.

Összefoglalás

- A Markov-modell állapotokkal és átmeneti rátákkal írja le a rendszert; a homogén lánc az exponenciális élettartam-modell kiterjesztése a rendszer szintjére.
- Az állapotvalószínűségek a Kolmogorov-egyenletekből számolhatók; a tartós eloszlás a $\pi\mathbf{Q} = \mathbf{0}$ egyenletrendszerből, az MTTF az elnyelő állapotos modell alapmátrixából adódik.
- A rendelkezésre állás megengedi a meghibásodást és a javítást, ezért mindig legalább akkora, mint a megbízhatóság; egyetlen elemre $A_\infty = \mathrm{MTTF}/(\mathrm{MTTF} + \mathrm{MTTR})$, és $U_\infty \approx \lambda\,\mathrm{MTTR}$.
- A javítható redundáns rendszerek MTTF-je nagyságrendekkel nagyobb a javítás nélkülinél, de a közös okú meghibásodás ezt az előnyt jelentősen csökkenti.
- A hideg tartalék előnyét az indulási hiba és a rejtett tartalékhiba nagyrészt eltüntetheti.
- A periodikusan tesztelt biztonsági rendszerek PFD-je fűrészfog alakú; a többfázisú Markov-modell pontosan leírja, és a nem tökéletes proof teszt hatását is megmutatja.
- A fő korlát az állapotrobbanás és az exponenciális tartózkodási idő feltevése; bonyolult esetekben Petri-hálók és Monte Carlo-szimuláció segít.

Önellenőrzés

1. Egy berendezés meghibásodási rátája $2 \cdot 10^{-4}\,\mathrm{h}^{-1}$, az átlagos javítási idő 12 óra. Mekkora a tartós rendelkezésre állás és az évi állásidő?
2. Két azonos, javítható egység párhuzamosan működik, $\lambda = 10^{-3}\,\mathrm{h}^{-1}$, $\mu = 0{,}1\,\mathrm{h}^{-1}$, egy javítócsapat. Mekkora a rendszer MTTF-je javítással és javítás nélkül?
3. Miért nem változtat a második javítócsapat az 1oo2 rendszer MTTF-jén, ha a rendelkezésre állását javítja?

4. Egy szelep proof tesztjének lefedettsége 80%, $\lambda_{DU} = 2{,}5 \cdot 10^{-6}\,\mathrm{h}^{-1}$, a próba évenkénti, a nagyjavítás tízévenkénti, az MTTR 8 óra. Mekkora az átlagos PFD? Melyik tag a nagyobb?
5. Hány állapota van egy olyan rendszer Markov-modelljének, amely hat, egymástól különböző, kétállapotú elemből áll? Miért kevesebb, ha az elemek azonosak?

Megoldás.

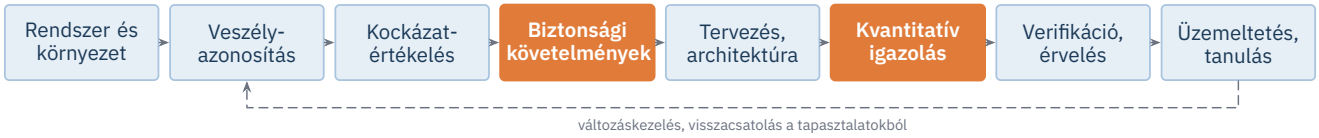
1. $\lambda\,\mathrm{MTTR} = 2{,}4 \cdot 10^{-3}$, $U_\infty = 2{,}4 \cdot 10^{-3}/1{,}0024 = 2{,}39 \cdot 10^{-3}$, $A_\infty = 0{,}99761$. Az évi állásidő $2{,}39 \cdot 10^{-3} \cdot 8760 \approx 21$ óra.
2. Javítással $\mathrm{MTTF} = (3 \cdot 10^{-3} + 0{,}1)/(2 \cdot 10^{-6}) = 51\,500$ óra; javítás nélkül $1{,}5/\lambda = 1500$ óra.
3. Mert az MTTF az első rendszerhibáig eltelt idő; a második csapat csak akkor dolgozik, ha a rendszer már meghibásodott (mindkét egység rossz), tehát az első rendszerhiba időpontját nem befolyásolja. A rendelkezésre állásra viszont hat, mert rövidíti a rendszerhiba állapotában töltött időt.
4. $0{,}8 \cdot 2{,}5 \cdot 10^{-6} \cdot (4380 + 8) + 0{,}2 \cdot 2{,}5 \cdot 10^{-6} \cdot 43\,800 = 8{,}78 \cdot 10^{-3} + 2{,}19 \cdot 10^{-2} = 3{,}07 \cdot 10^{-2}$. A nem tesztelt hibák tagja a nagyobb, pedig csak a hibák ötödét adják: a tízéves nagyjavítási időköz miatt ezek tízszer tovább maradnak rejtve.
5. $2^6 = 64$ állapot. Azonos elemeknél nem kell tudni, *melyik* elem hibás, csak azt, *hány*; ez $n + 1 = 7$ állapotra csökkenti a modellt.

Továbbolvasás

- Rausand, M. (2014): *Reliability of Safety-Critical Systems: Theory and Applications.* Wiley – a biztonsági rendszerek Markov-modelljei, a többfázisú modellek és a nem tökéletes proof teszt (Rausand 2014).
- Trivedi, K. S., Bobbio, A. (2017): *Reliability and Availability Engineering.* Cambridge University Press – a Markov-láncok, a szemi-Markov-modellek és a Petri-hálók részletes tárgyalása (Trivedi és Bobbio 2017).
- IEC 61165:2006 – a Markov-technikák alkalmazása a megbízhatósági elemzésben (IEC 2006b).
- Rausand, M., Barros, A., Høyland, A. (2020): *System Reliability Theory*, 11–12. fejezet (Rausand és mtsai. 2020).

13.

A biztonsági integritás igazolása: PFD, PFH és SIL



A kockázatelemzés és a LOPA megmondja, mekkora kockázatcsökkentést kell egy biztonsági funkciónak biztosítania (3. és 11. fejezet). A tervezés ezután egy konkrét architektúrát javasol: érzékelőket, logikát, beavatkozókat, redundanciát és tesztelési stratégiát. Ez a fejezet arról szól, hogyan igazolható, hogy a tervezett funkció valóban teljesíti a követelményt. Az igazolásnak három pillére van (4.3.2. szakasz): a véletlenszerű hardvermeghibásodások számszerű célértéke, az architekturális korlátok és a szisztematikus képesség. A fejezet a megbízhatósági, a hibafa- és a Markov-modellek eredményeit egyetlen gyakorlati eljárásba foglalja, és a ReaktorVéd példáján végigvezet egy teljes SIL-igazoláson, a téves leállásokat is beleértve.

Tanulási eredmények

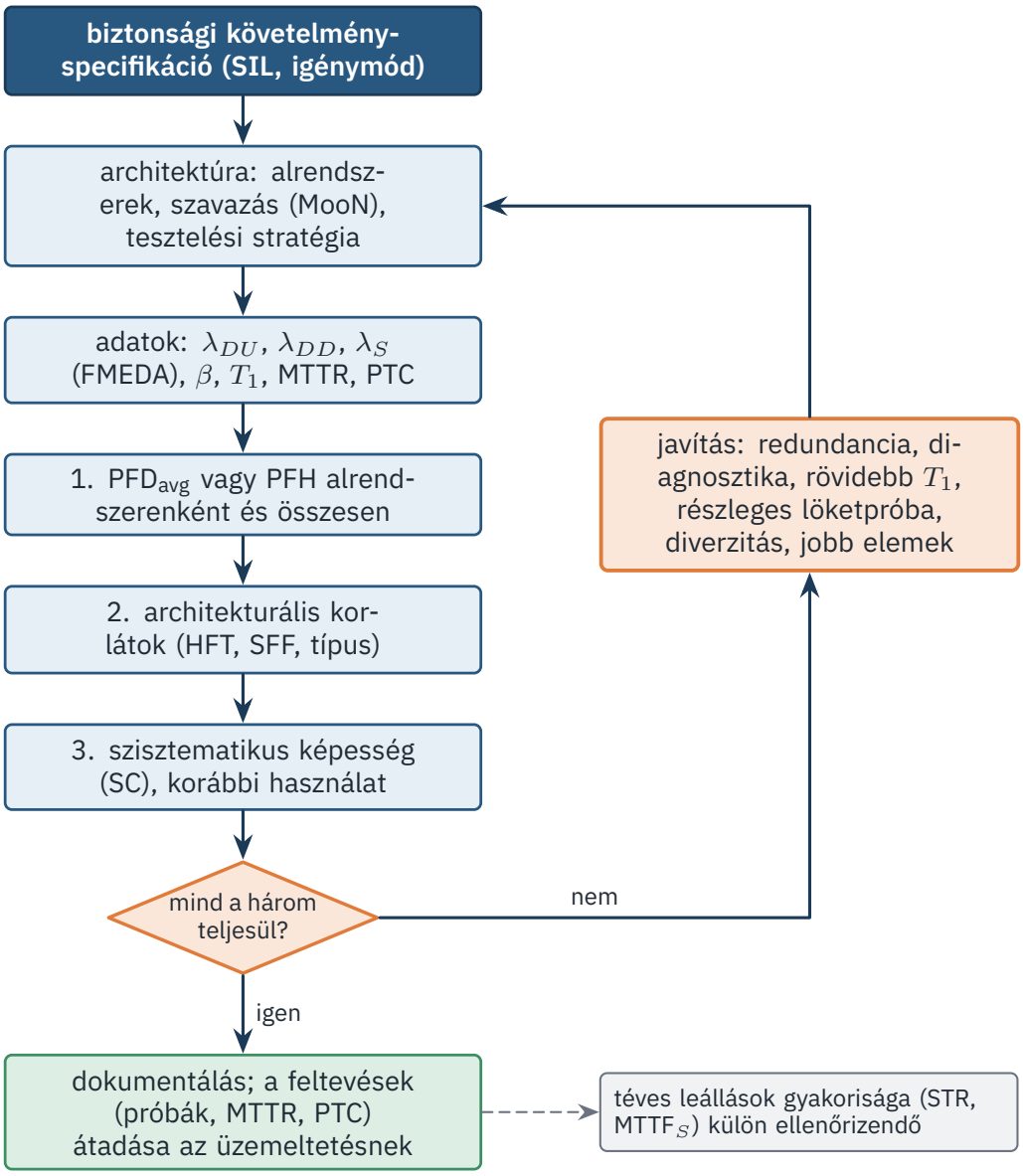
A fejezet végére képes leszel

- a biztonsági funkció elemeinek meghibásodásait biztonságos, felfedett veszélyes és rejtett veszélyes kategóriákba sorolni, és kiszámolni a diagnosztikai lefedettséget és a biztonságos meghibásodási hányadot;
- az IEC 61508-6 egyszerűsített képleteivel kiszámolni az 1oo1, 1oo2 és 2oo3 alrendszerek PFD_{avg}-értékét, valamint a nagy igénybevételű funkciók PFH-értékét;
- ellenőrizni az architekturális korlátokat az IEC 61508 és az IEC 61511 szerint, és értelmezni a szisztematikus képesség követelményét;
- érzékenységvizsgálattal azonosítani a PFD-t leginkább befolyásoló tényezőket, és értékelni a részleges löketpróba és a nem tökéletes proof teszt hatását;
- a biztonsági funkció téves leállási gyakoriságát is becsülni, és a biztonság és a rendelkezésre állás közötti egyensúlyt mérlegelni.

13.1. Az igazolás menete

A SIL-igazolás nem egyetlen számítás, hanem egy ciklikus tervezési folyamat része (13.1. ábra). A kiindulópont a **biztonsági követelményspecifikáció** (SRS), amely rögzíti a funkciót, a SIL-t, az igénymódot, a válaszidőt, a biztonságos állapotot és a megengedett téves leállási gyakoriságot (IEC 2016b). A tervezők egy architektúrát javasolnak, összegyűjtik az elemek meghibásodási adatait, majd a három pillér szerint ellenőrzik a tervet. Ha bármelyik nem teljesül, módosítanak (redundancia,

jobb diagnosztika, rövidebb próbaidőköz, részleges löketpróba, diverzitás, más elemek), és újra ellenőriznek. A végén a számításban használt feltevések (a próbák gyakorisága és lefedettsége, a javítási idő, a kiiktatások szabályai) az üzemeltetési és karbantartási követelmények részévé válnak, hiszen az igazolás csak addig érvényes, amíg ezek teljesülnek.



13.1. ábra. A SIL-igazolás folyamata. A három pillér együtt szükséges; a számítás feltevései az üzemeltetés követelményeivé válnak.

13.1.1. Meghibásodási kategóriák

A biztonsági funkció elemeinek meghibásodásait a funkcióra gyakorolt hatásuk szerint soroljuk be (2.6.2. szakasz, 6.5. szakasz). A meghibásodási ráta

$$\lambda = \lambda_S + \lambda_D, \qquad \lambda_D = \lambda_{DD} + \lambda_{DU}, \tag{13.1}$$

ahol λ_S a **biztonságos** (a funkciót nem akadályozó, legfeljebb téves működést okozó), λ_{DD} a **felfedett veszélyes** (a diagnosztika által rövid időn belül észlelt) és λ_{DU} a **rejtett veszélyes** meghibásodások rátája. Két származtatott jellemző:

$$\mathrm{DC} = \frac{\lambda_{DD}}{\lambda_D}, \qquad \mathrm{SFF} = \frac{\lambda_S + \lambda_{DD}}{\lambda_S + \lambda_D}. \tag{13.2}$$

A **diagnosztikai lefedettség** (DC) a veszélyes meghibásodások felfedett hányada, a **biztonságos meghibásodási hányad** (SFF) pedig azoknak a meghibásodásoknak az aránya, amelyek nem vezetnek rejtett veszélyes állapothoz (IEC 2010a). A PFD szempontjából a λ_{DU} a döntő: ezek a hibák csak a proof teszten vagy egy valódi igénynél derülnek ki.

A meghibásodási adatok forrásai a gyártók FMEDA-alapú tanúsítványai, az iparági adatbázisok és kézikönyvek (például a SINTEF PDS-módszerének adatai (Hauge és mtsai. 2013)), valamint az üzem saját tapasztalati adatai. Mindegyiknek megvannak a maga feltevései: a tanúsítvány rendszerint ideális telepítést és környezetet feltételez, az adatbázisok sokféle alkalmazás átlagát adják, a saját adatok pedig kevés meghibásodásra épülnek. A 8. fejezetben láttuk, hogy kevés megfigyelésből a ráta felső konfidenciakorlátja akár másfél-kétszerese is lehet a pontbecslésnek. Az igazolásban mindig rögzíteni kell, melyik adatot honnan vettük, és miért tekintjük az adott alkalmazásban érvényesnek.

13.2. A PFD$_{\text{avg}}$ számítása

Alacsony igénybevételű funkcióknál (az igények gyakorisága legfeljebb évi egy, 4.3. szakasz) a mérőszám az igénykori meghibásodás átlagos valószínűsége, a PFD$_{\text{avg}}$. A teljes funkció PFD$_{\text{avg}}$-értéke az alrendszerek (érzékelők, logika, beavatkozók) értékének összege, hiszen bármelyik alrendszer kudarca a funkció kudarcát jelenti:

$$\mathrm{PFD}_{\mathrm{SIF}} \approx \mathrm{PFD}_{\text{érzékelők}} + \mathrm{PFD}_{\text{logika}} + \mathrm{PFD}_{\text{beavatkozók}}. \tag{13.3}$$

Az alrendszerek PFD$_{\text{avg}}$-értékére az IEC 61508-6 B melléklete egyszerűsített képleteket ad (IEC 2010a). Ezek a 12. fejezet többfázisú Markov-modelljeinek közelítései, amelyek akkor pontosak, ha $\lambda T_1 \ll 1$, a csatornákat egyszerre tesztelik, és a próba tökéletes. Két **ekvivalens leállási időt** definiálnak: a csatornáét (t_{CE}) és a csoportét (t_{GE}),

$$t_{CE} = \frac{\lambda_{DU}}{\lambda_D}\left(\frac{T_1}{2} + \mathrm{MRT}\right) + \frac{\lambda_{DD}}{\lambda_D}\,\mathrm{MTTR}, \qquad t_{GE} = \frac{\lambda_{DU}}{\lambda_D}\left(\frac{T_1}{3} + \mathrm{MRT}\right) + \frac{\lambda_{DD}}{\lambda_D}\,\mathrm{MTTR}, \tag{13.4}$$

ahol T_1 a proof teszt időköze, MRT az átlagos javítási idő a rejtett hiba felfedése után, MTTR pedig a felfedett hiba helyreállításának átlagos ideje. (A következő példákban egyszerűségből mindkettőt MTTR-rel jelöljük.) A $T_1/3$ a 10. fejezetben látott 4/3-os időátlagolási tényező forrása. A három legfontosabb architektúra képlete:

$$\begin{aligned}
\mathrm{PFD}_{1\mathrm{oo}1} &= \lambda_D\, t_{CE},\\
\mathrm{PFD}_{1\mathrm{oo}2} &= 2\big[(1-\beta_D)\lambda_{DD} + (1-\beta)\lambda_{DU}\big]^2 t_{CE}\, t_{GE} + \beta_D\lambda_{DD}\,\mathrm{MTTR} + \beta\lambda_{DU}\left(\frac{T_1}{2} + \mathrm{MRT}\right),\\
\mathrm{PFD}_{2\mathrm{oo}3} &= 6\big[(1-\beta_D)\lambda_{DD} + (1-\beta)\lambda_{DU}\big]^2 t_{CE}\, t_{GE} + \beta_D\lambda_{DD}\,\mathrm{MTTR} + \beta\lambda_{DU}\left(\frac{T_1}{2} + \mathrm{MRT}\right).
\end{aligned} \tag{13.5}$$

A redundáns képletek első tagja a független kettős meghibásodások, a második és a harmadik a felfedett, illetve a rejtett hibák közös okú hozzájárulása; β és β_D a rejtett, illetve a felfedett hibákra vonatkozó közös okú tényező (10.6. szakasz). A 2oo3 független tagjának 6-os szorzója abból adódik, hogy három csatornapárból bármelyik kettős meghibásodása elég a kudarchoz. Egy 2oo2 alrendszer egyszerűen két 1oo1 csatorna soros kapcsolata: $\mathrm{PFD}_{2oo2} = 2\lambda_D t_{CE}$.

ReaktorVéd: a PFD~avg~ számítása

A ReaktorVéd biztonsági funkciójának adatait a 13.1. táblázat foglalja össze. A funkció hibabiztos kivitelű: a szelepek energiamentesítésre zárnak, a távadók élő nullás áramhurokkal csatlakoznak (10.6. szakasz).

13.1. táblázat. A ReaktorVéd biztonsági funkciójának adatai ($T_1 = 8760$ h, MTTR = 8 h)

Alrendszer	Architektúra	λ_{DU} [h^{-1}]	λ_{DD} [h^{-1}]	λ_S [h^{-1}]	β / β_D	Típus
hőmérséklet-távadók (TT-101/102)	1oo2	$4 \cdot 10^{-7}$	$6 \cdot 10^{-7}$	$5 \cdot 10^{-7}$	5% / 2,5%	B
biztonsági PLC	1oo1 (tanúsított)	–	–	–	–	SIL 3
gyorszáró szelepek (XV-101/102)	1oo2	$2{,}5 \cdot 10^{-6}$	0	–	10% / –	A

Távadók. $\lambda_D = 10^{-6}\,\mathrm{h}^{-1}$, DC $= 60\%$. Az ekvivalens leállási idők $t_{CE} = 0{,}4 \cdot 4388 + 0{,}6 \cdot 8 = 1760$ óra és $t_{GE} = 0{,}4 \cdot 2928 + 0{,}6 \cdot 8 = 1176$ óra. A független tag $2 \cdot (5{,}85 \cdot 10^{-7} + 3{,}8 \cdot 10^{-7})^2 \cdot 1760 \cdot 1176 = 3{,}86 \cdot 10^{-6}$, a közös okú tagok $0{,}025 \cdot 6 \cdot 10^{-7} \cdot 8 + 0{,}05 \cdot 4 \cdot 10^{-7} \cdot 4388 = 8{,}79 \cdot 10^{-5}$. Összesen $\mathrm{PFD}_{\mathrm{TT}} = 9{,}17 \cdot 10^{-5}$.
Logika. A tanúsított biztonsági PLC gyártói értéke $\mathrm{PFD}_{\mathrm{L}} = 5 \cdot 10^{-5}$ az adott próbaidőközre.
Szelepek. $\lambda_{DD} = 0$, így $t_{CE} = 4388$ óra, $t_{GE} = 2928$ óra. A független tag $2 \cdot (2{,}25 \cdot 10^{-6})^2 \cdot 4388 \cdot 2928 = 1{,}30 \cdot 10^{-4}$, a közös okú tag $0{,}1 \cdot 2{,}5 \cdot 10^{-6} \cdot 4388 = 1{,}097 \cdot 10^{-3}$. Összesen $\mathrm{PFD}_{\mathrm{V}} = 1{,}227 \cdot 10^{-3}$.
A funkció. $\mathrm{PFD}_{\mathrm{SIF}} = 9{,}17 \cdot 10^{-5} + 5 \cdot 10^{-5} + 1{,}227 \cdot 10^{-3} = 1{,}369 \cdot 10^{-3}$, azaz RRF $= 731$. A LOPA követelménye (PFD $\leq 2 \cdot 10^{-3}$, RRF ≥ 500) teljesül. A PFD 90%-át a szelepek adják, és ezen belül is a közös okú meghibásodás; a távadók és a logika együtt csak 10%-ot.
Az eredmény jól egyezik a hibafa ($1{,}335 \cdot 10^{-3}$, 10.6. szakasz) és a többfázisú Markov-modell ($1{,}22 \cdot 10^{-3}$ a szelepekre, 12.5. szakasz) eredményével, és jóval nagyobb a blokkdiagram naiv becslésénél ($1{,}7 \cdot 10^{-4}$, 9. fejezet). A különbség oka a közös okú meghibásodás és a helyes időátlagolás; a blokkdiagram egyiket sem veszi figyelembe.

Gyakori hibák a PFD-számításban

- **λ_D helyett λ_{DU}, vagy fordítva:** a felfedett hibák csak a javítási idő alatt rontják a PFD-t; ha a teljes λ_D-vel számolunk $T_1/2$-vel, nagyon pesszimista, ha a felfedett hibákat egyáltalán nem vesszük figyelembe, enyhén optimista az eredmény.
- **A közös okú meghibásodás elhagyása:** a redundáns alrendszerekben gyakran ez a domináns tag, ahogy a ReaktorVéd szelepeinél is.
- **Tökéletes proof teszt feltételezése:** a valóságos próbák ritkán fednek fel minden hibát (13.5. szakasz).
- **A gyártói PFD-érték átvétele a feltevések ellenőrzése nélkül:** a tanúsítvány egy adott próbaidőközre, javítási időre és környezetre érvényes.
- **A redundáns csatornák PFD-jének szorzása:** az átlagok szorzata nem a szorzat átlaga, és a közös okú tagot is elhagyja (10. fejezet).

- **„SIL 2-es eszköz”:** a SIL a biztonsági *funkció* tulajdonsága. Egy eszköz legfeljebb *alkalmas* egy adott SIL-ű funkcióban való használatra, a szisztematikus képessége és a hardveres jellemzői alapján.

13.3. Architekturális korlátok

A számított PFD önmagában nem elég, mert a meghibásodási adatok bizonytalanok, és a modell feltevései sérülhetnek. A szabványok ezért minimális **hardveres hibatűrést** (*hardware fault tolerance*, HFT) írnak elő: HFT = N azt jelenti, hogy N+1 hiba okozhatja a funkció elvesztését (egy 1oo2 alrendszer HFT-je 1, egy 2oo3-é szintén 1, egy 1oo3-é 2).

Az IEC 61508-2 két utat kínál (IEC 2010a). Az **1H út** a biztonságos meghibásodási hányadhoz és az elem típusához köti a megengedett SIL-t (13.2. táblázat). Az **A típusú** elemek meghibásodási módjai jól ismertek, és viselkedésük hibás állapotban is teljesen meghatározható (például szelepek, relék, egyszerű analóg érzékelők); a **B típusú** elemek összetettek, jellemzően mikroprocesszorosak, és nem minden meghibásodási módjuk ismert (például okos távadók, PLC-k).

13.2. táblázat. Az elérhető legnagyobb SIL az IEC 61508-2 1H útja szerint

SFF	A típus, HFT = 0	HFT = 1	HFT = 2	B típus, HFT = 0	HFT = 1	HFT = 2
< 60%	SIL 1	SIL 2	SIL 3	nem megengedett	SIL 1	SIL 2
60% – < 90%	SIL 2	SIL 3	SIL 4	SIL 1	SIL 2	SIL 3
90% – < 99%	SIL 3	SIL 4	SIL 4	SIL 2	SIL 3	SIL 4
≥ 99%	SIL 3	SIL 4	SIL 4	SIL 3	SIL 4	SIL 4

A **2H út** a tényleges üzemi tapasztalaton alapuló meghibásodási adatokra épít, és egyszerűbb HFT-követelményt támaszt: SIL 4-hez HFT = 2, SIL 3-hoz HFT = 1, SIL 2-höz nagy igénybevételnél HFT = 1, alacsony igénybevételnél HFT = 0, SIL 1-hez HFT = 0. Cserébe a meghibásodási rátákat legalább 70%-os konfidenciaszinten kell becsülni, és a célérték teljesülését is a bizonytalanság figyelembevételével kell igazolni (IEC 2010a). A folyamatipari IEC 61511 2016-os kiadása ehhez hasonló, egyszerű táblázatot használ (13.3. táblázat), és az SFF-et elhagyta a követelmények közül (IEC 2016b).

13.3. táblázat. A minimális hardveres hibatűrés az IEC 61511-1:2016 szerint

SIL	Igénymód	Minimális HFT
1	bármely	0
2	alacsony igénybevétel	0
2	nagy igénybevétel vagy folyamatos	1
3	bármely	1
4	bármely	2

Az architekturális korlátok szerepe vitatott. Lundteigen és Rausand elemzése szerint a korlátok és a PFD-számítás nem mindig ugyanabba az irányba terelik a tervezőt, és az SFF ösztönözheti a biztonságos meghibásodások „felhizlalását” is, ami a valódi biztonságot nem javítja, a téves leállásokat viszont növeli (Lundteigen és Rausand 2009). Ez is szerepet játszott abban, hogy az IEC 61511 új kiadása egyszerűbb, HFT-alapú követelményre tért át.

ReaktorVéd: az architekturális korlátok ellenőrzése

A távadók SFF-je $(5+6)\cdot 10^{-7}/(1{,}5\cdot 10^{-6}) = 73\%$; B típusúak, HFT = 1, így az 1H út szerint SIL 2-ig alkalmazhatók. A szelepek A típusúak, diagnosztika nélkül az SFF-jük 60% alatt van, de HFT = 1 mellett SIL 2-ig alkalmazhatók. A biztonsági PLC tanúsítványa szerint SIL 3-ig alkalmas. A funkció tehát az IEC 61508 1H útja szerint teljesíti a SIL 2 architekturális követelményeit. Az IEC 61511 szerint SIL 2-höz alacsony igénybevételnél HFT = 0 is elég volna; a redundanciára itt a PFD-célérték miatt van szükség, nem az architekturális korlát miatt. Egycsatornás szelepekkel a funkció PFD-je $1{,}1\cdot 10^{-2}$ fölött lenne, SIL 1 tartományban.

13.4. Szisztematikus képesség

A számszerű célérték és az architekturális korlátok a véletlenszerű hardvermeghibásodások ellen védenek. A szisztematikus hibák (specifikációs, tervezési, szoftver- és gyártási hibák) azonban nem írhatók le meghibásodási rátával (2. fejezet). Ellenük a fejlesztési folyamat szigorúsága véd: az IEC 61508 minden SIL-hez meghatározott módszereket és technikákat ír elő a specifikációtól a validálásig. Egy elem **szisztematikus képessége** (*systematic capability*, SC 1–4) azt fejezi ki, hogy milyen SIL-ű funkcióban használható a szisztematikus integritás szempontjából (IEC 2010a).

A szisztematikus képesség háromféleképpen igazolható: az IEC 61508 szerinti fejlesztéssel (1S út), az üzemi tapasztalattal igazolt megbízhatósággal (*proven in use*, 2S út), illetve a korábban kifejlesztett szoftverek esetén a 3S úttal. A folyamatiparban az IEC 61511 a **korábbi használat** (*prior use*) fogalmát vezeti be: egy eszköz akkor választható, ha az üzemeltető hasonló alkalmazásban és környezetben szerzett, dokumentált tapasztalata igazolja az alkalmasságát (IEC 2016b). Az alkalmazási szoftverre (a biztonsági PLC-ben futó logikára) az IEC 61511 saját életciklus-követelményeket támaszt; a szoftverek biztonságát a 15. fejezet tárgyalja.

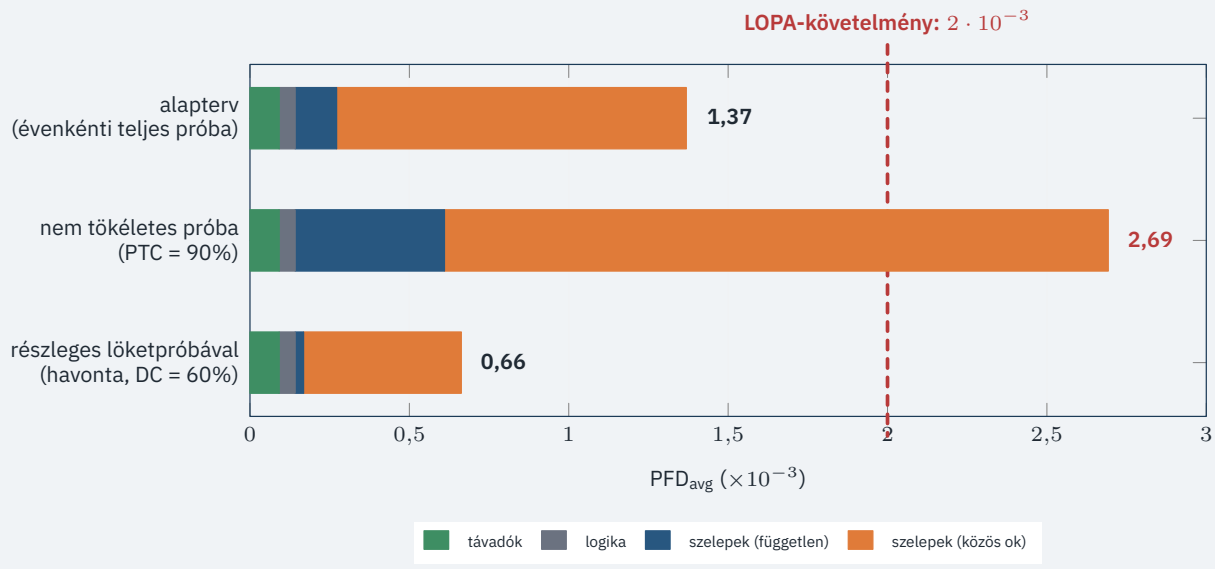
13.5. Részleges löketpróba és nem tökéletes proof teszt

A gyorszáró szelepek éves proof tesztje a folyamat leállítását igényli, ezért a próbaidőköz nem rövidíthető tetszés szerint. A **részleges löketpróba** (*partial stroke testing*, PST) üzem közben, a szelepet csak részben (például 10–20%-ban) megmozdítva ellenőrzi, hogy a szelep nem ragadt-e be, és a működtető, a mágnesszelep és a rugó működik-e. A PST nem helyettesíti a teljes próbát, mert a zárás tömörségét és a teljes löketet nem ellenőrzi, de a rejtett hibák egy részét sokkal gyakrabban fedi fel (Rausand 2014).

A PST hatása a 12.5.1. szakasz gondolatmenetével számolható: a szelep rejtett hibáinak $\mathrm{DC}_{\mathrm{PST}}$ hányada a PST időközében, a többi a teljes próba időközében derül ki. Ennek fordítottja a **nem tökéletes proof teszt**: ha a teljes próba lefedettsége PTC $< 100\%$, a fel nem fedett hibák csak a nagyjavításkor derülnek ki.

ReaktorVéd: mit tesz a próba minősége?

Ha a szelepek éves próbája csak a hibák 90%-át fedi fel (például mert a próba nem ellenőrzi a zárás tömörségét az üzemi nyomáskülönbség mellett), és a szelepeket tízévenként újítják fel, a funkció PFD-je $2{,}69\cdot 10^{-3}$-ra nő (13.2. ábra). Ez *nem teljesíti* a LOPA követelményét. Egy apró eljárási részlet tehát érvényteleníthet a SIL-igazolást.
Ha viszont havonta részleges löketpróbát végeznek, amely a szelephibák 60%-át felfedi, a szelepek közös okú tagja $1{,}097\cdot 10^{-3}$-ról $4{,}95\cdot 10^{-4}$-re, a független tag $1{,}30\cdot 10^{-4}$-ről $2{,}6\cdot 10^{-5}$-re csökken. A funkció PFD-je $6{,}63\cdot 10^{-4}$, RRF $= 1508$: kétszeres tartalék a követelményhez képest. A számításokat numerikus időátlagolással is ellenőriztük; az eltérés a képletektől 1–3%.



13.2. ábra. A ReaktorVéd biztonsági funkciójának PFD-összetevői három tesztelési stratégia mellett. A nem tökéletes próba a követelmény fölé viszi a PFD-t, a részleges löketpróba kétszeres tartalékot ad.

A PST lefedettségét azonban igazolni kell: függ a szelep és a működtető típusától, a meghibásodási módoktól és a próba kivitelezésétől. A 60% egy gondos FMEDA-val alátámasztandó feltevés, nem általános érték.

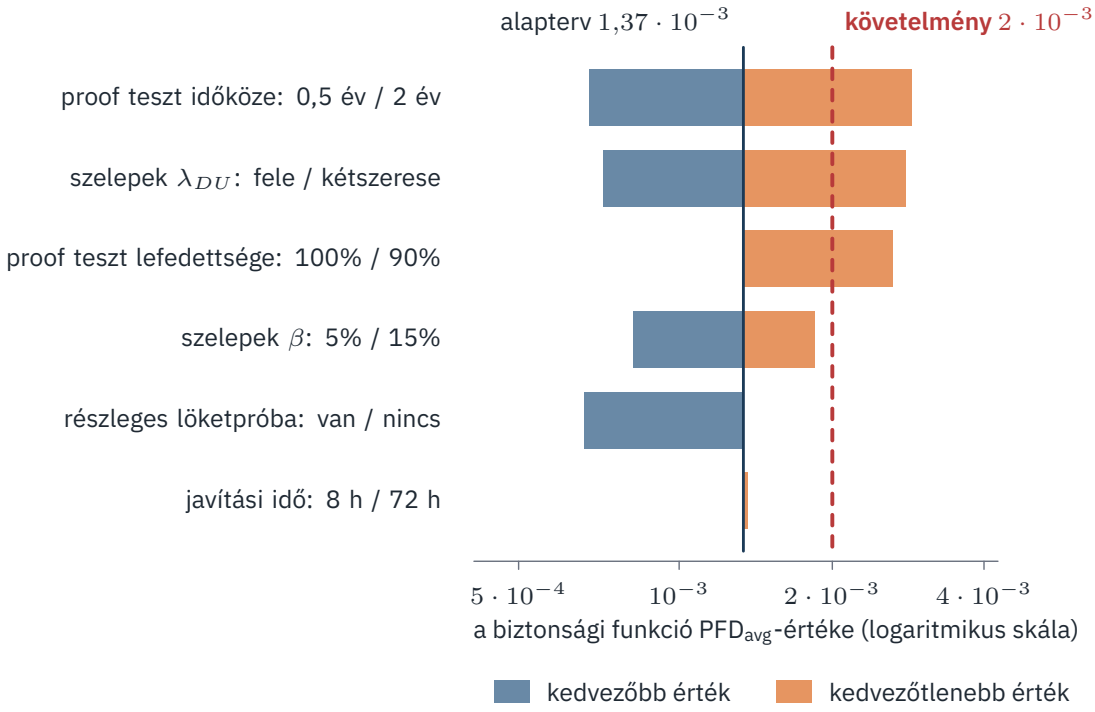
13.6. Érzékenységvizsgálat

A PFD-számítás bemenő adatai bizonytalanok, és a feltevések üzem közben változhatnak. Az érzékenységvizsgálat megmutatja, mely tényezők befolyásolják leginkább az eredményt, és ezzel azt is, mire kell az üzemeltetésnek a legjobban figyelnie. A ReaktorVéd tornádódiagramja (13.3. ábra) szerint a legerősebb hatása a proof teszt időközének (fél és két év között $6{,}8 \cdot 10^{-4}$ és $2{,}95 \cdot 10^{-3}$ között változik a PFD), a szelepek rejtett meghibásodási rátájának, a próba lefedettségének és a szelepek közös okú tényezőjének van. A javítási idő hatása elhanyagolható: még 72 óra is csak 1,6%-kal növeli a PFD-t, mert a redundáns szelepeknél egyetlen hiba javítása alatt a másik szelep még véd.

A tanulság gyakorlati: a ReaktorVéd biztonsága a szelepeken múlik. A kétéves próbaidőköz, a kétszer rosszabb szelepadat vagy a 90%-os próbalefedettség egyaránt elég ahhoz, hogy a funkció ne teljesítse a követelményt. Az igazolásnak ezért a szelepek kiválasztására (üzemi tapasztalat, megfelelő típus), a közös okú hibák csökkentésére (különálló mágnesszelepek és tápellátás, eltérő gyártójú vagy típusú szelepek, eltolt karbantartás) és a próbaeljárás minőségére kell összpontosítania. A távadók és a logika ennél jóval kevésbé kritikusak.

13.7. Téves leállások

A biztonsági funkció kétféleképpen hibázhat: veszélyesen és biztonságosan (9.3.1. szakasz). A biztonságos meghibásodások a folyamat **téves leállását** (*spurious trip*) okozzák. Ez nemcsak termelési veszteség: a leállások és az újraindítások maguk is veszélyes átmeneti állapotok, a gyakori téves leállás pedig arra ösztönözheti a kezelőket, hogy kiiktassák a védelmet. Az IEC 61511 ezért előírja, hogy a biztonsági követelményspecifikáció rögzítse a megengedett legnagyobb téves leállási gyakoriságot



13.3. ábra. A ReaktorVéd biztonsági funkciójának érzékenysége a fő paraméterekre (tornádódiagram). A szaggatott vonal a LOPA-követelmény; a proof teszt időköze, a szelepadatok és a próba lefedettsége mind a követelmény fölé vihetik a PFD-t.

(IEC 2016b). A jellemzők a **téves leállási gyakoriság** (*spurious trip rate*, STR) és annak reciproka, a téves leállásig eltelt átlagos idő (MTTF_S).

Az 1oo2 alrendszer bármelyik csatornájának téves jelzésére leállít, ezért téves leállási gyakorisága $\mathrm{STR}_{1oo2} \approx 2(\lambda_S + \lambda_{DD})$, ha a felfedett veszélyes hibákra is leállás a reakció. A 2oo3 alrendszer csak két csatorna egyidejű téves jelzésére állít le, ezért $\mathrm{STR}_{2oo3} \approx 6(\lambda_S + \lambda_{DD})^2\,\mathrm{MTTR}$, ami nagyságrendekkel kisebb.

ReaktorVéd: a hibabiztos kialakítás ára

A hibabiztos átalakítás (10.6. szakasz) a tápegységek kiesését a veszélyes hibafából a téves leállások hibafájába helyezte át. A 13.4. táblázat a következő feltevésekkel számol: a távadók $\lambda_S + \lambda_{DD} = 1{,}1 \cdot 10^{-6}\,\mathrm{h}^{-1}$, egy szelep téves zárása $10^{-6}\,\mathrm{h}^{-1}$, a logika téves kimenete $10^{-6}\,\mathrm{h}^{-1}$, egy tápegység kiesése $5 \cdot 10^{-6}\,\mathrm{h}^{-1}$.

13.4. táblázat. A ReaktorVéd téves leállási gyakorisága különböző kialakítások mellett

Változat	STR [1/év]	MTTF_S [év]	Domináns tag
1oo2 távadók, két egyszeres tápegység	0,133	7,5	tápegységek (66%)
1oo2 távadók, redundáns tápellátás	0,046	21,9	szelepek, távadók, logika
2oo3 távadók, redundáns tápellátás	0,026	38,0	szelepek, logika

Az egyszeres tápegységekkel átlagosan hétévente egyszer állna le tévesen a reaktor; ennek kétharmadát a tápegységek okoznák. A redundáns (diódás VAGY-kapcsolású) tápellátás ezt a hozzájárulást gyakorlatilag megszünteti. A 2oo3 távadókkal a távadók téves leállásai is eltűnnek, a PFD pedig szinte változatlan ($9{,}94 \cdot 10^{-5}$ az 1oo2 $9{,}17 \cdot 10^{-5}$ értékéhez képest), mert mindkét esetben a közös okú tag dominál. Ugyanaz a probléma, a közös tápellátás, a 10. fejezetben a biztonságot, itt a rendelkezésre állást veszélyeztette; a megoldás mindkét esetben a függőség felismerése és megszüntetése.

13.8. Nagy igénybevétel: a PFH számítása

Ha a biztonsági funkció gyakran, évente többször működésbe lép, vagy folyamatosan működik (például egy gép biztonsági sebességfelügyelete vagy egy jármű kormányrendszerének felügyelete), a mérőszám a veszélyes meghibásodás átlagos gyakorisága, a PFH (4.2. táblázat). Ilyenkor a rejtett hiba gyakorlatilag azonnal veszélyes eseményhez vezet, hiszen az igény hamarosan bekövetkezik. Egycsatornás alrendszerre $\mathrm{PFH}_{1oo1} \approx \lambda_{DU}$, ha a felfedett hibákra a rendszer időben biztonságos állapotba kerül. Az 1oo2 alrendszer képlete az IEC 61508-6 szerint (IEC 2010a)

$$\mathrm{PFH}_{1oo2} = 2\big[(1-\beta_D)\lambda_{DD} + (1-\beta)\lambda_{DU}\big](1-\beta)\lambda_{DU}\,t_{CE} + \beta_D\lambda_{DD} + \beta\lambda_{DU}. \tag{13.6}$$

A gépiparban az ISO 13849-1 egyszerűsített eljárást is kínál, amely a kategóriából, a csatornánkénti $\mathrm{MTTF_D}$-ből és az átlagos diagnosztikai lefedettségből táblázatosan adja a PFH-t és a PL-t; a közös okú meghibásodás elleni intézkedéseket pedig egy pontozásos lista értékeli, amelyen a 2-es kategóriától felfelé legalább 65 pontot kell elérni (ISO 2023).

KoboCella: a biztonságos megállítás PFH-ja

A KoboCella biztonságos megállítási funkciója két független csatornán valósul meg (1oo2). Csatornánként $\lambda_D = 10^{-6}\,\mathrm{h}^{-1}$ ($\mathrm{MTTF_D} \approx 114$ év), a diagnosztikai lefedettség 99%, így $\lambda_{DD} = 9{,}9 \cdot 10^{-7}$ és $\lambda_{DU} = 10^{-8}\,\mathrm{h}^{-1}$; a közös okú tényezők $\beta = 2\%$, $\beta_D = 1\%$; a rejtett hibák csak a húszéves használati idő végén derülnének ki. Az 13.6. egyenletből $\mathrm{PFH} = 1{,}01 \cdot 10^{-8}\,\mathrm{h}^{-1}$, amely teljesíti a $\mathrm{PL_r}$ = e követelményét ($\mathrm{PFH} < 10^{-7}$, 4.6. táblázat).
A számításból egy meglepő tanulság is kiderül: a PFH szinte teljes egészében a $\beta_D\lambda_{DD}$ tagból adódik, vagyis a *felfedett* hibák közös okú hányadából. Nagy igénybevételnél a diagnosztika csak akkor véd, ha a hiba felismerése és a biztonságos állapot elérése gyorsabb, mint a következő igény; két csatorna egyidejű, közös okú meghibásodása pedig a diagnosztikai reakció előtt is veszélyes helyzetet teremthet. A gyors diagnosztikai reakció és a közös okú hibák elleni intézkedések (szétválasztás, diverzitás) ezért a nagy igénybevételű funkciókban különösen fontosak.

VészFék: hardvermetrikák az ISO 26262 szerint

A közúti járműveknél a véletlenszerű hardvermeghibásodásokat az ISO 26262-5 három metrikával értékeli (4.5. táblázat) (ISO 2018a): az egypontos hibák metrikájával (SPFM), a látens hibák metrikájával (LFM) és a véletlenszerű hardvermeghibásodások miatti biztonsági célsértés valószínűségének metrikájával (PMHF), amely a PFH megfelelője. Tegyük fel, hogy a vészfékező vezérlőegység biztonságrelevánsnak minősülő elemeinek teljes meghibásodási rátája 500 FIT ($5{\cdot}10^{-7}\,\mathrm{h}^{-1}$), ebből az egypontos és a diagnosztika által nem fedett maradó hibák rátája 3 FIT, a többpontos hibák közül a látensek (a diagnosztika és a vezető által sem észlelt hibák) rátája 20 FIT. Ekkor $\mathrm{SPFM} = 1 - 3/500 = 99{,}4\%$ és $\mathrm{LFM} = 1 - 20/(500-3) = 96{,}0\%$, ami teljesíti az ASIL D célértékeit (legalább 99%, illetve 90%). A PMHF-et az egypontos és maradó hibák 3 FIT-es rátája uralja, mert a látens kettős hibák járuléka (egy látens hiba és egy második hiba egyidejű fennállása) nagyságrendekkel kisebb; így a 10 FIT-es ASIL D célérték is teljesül. A gondolatmenet ugyanaz, mint az IEC 61508-ban: a felfedett és a rejtett hibák szétválasztása, és a rejtett hibák felfedési idejének korlátozása.

Gyakorlat

A gyakorlati útmutató 11. laborjában egy böngészős PFD/SIL-kalkulátorral állíthatod össze a ReaktorVéd biztonsági funkcióját: az alrendszerek architektúrájának, meghibásodási adatainak, a próbaidőköznek, a közös okú tényezőknek, a részleges löketpróbának és a proof teszt lefedettségének változtatásával azonnal látod a $\mathrm{PFD}_{\mathrm{avg}}$-t, az elérhető SIL-t, az architekturális korlátokat és a téves leállási gyakoriságot. A hozzá tartozó Colab-notebook a képleteket a többfázisú Markov-modellel és Monte Carlo-szimulációval is összeveti, és elkészíti a tornádódiagramot.

Összefoglalás

- A SIL-igazolás három pillére a számszerű célérték ($\mathrm{PFD}_{\mathrm{avg}}$ vagy PFH), az architekturális korlátok és a szisztematikus képesség; a számítás feltevései az üzemeltetés követelményeivé válnak.
- A meghibásodások biztonságos, felfedett veszélyes és rejtett veszélyes kategóriákba sorolhatók; a PFD-t elsősorban a rejtett veszélyes hibák határozzák meg.
- Az IEC 61508-6 képletei a csatorna és a csoport ekvivalens leállási idején alapulnak; a redundáns alrendszerekben a közös okú tag gyakran domináns.
- A ReaktorVéd funkciója $\mathrm{PFD} = 1{,}37 \cdot 10^{-3}$ értékkel teljesíti a $2 \cdot 10^{-3}$-os követelményt; a PFD 90%-át a szelepek adják.
- Az architekturális korlátok az IEC 61508 1H útja szerint az SFF-től, a típustól és a HFT-től, a 2H út és az IEC 61511 szerint csak a HFT-től függenek.
- A próba minősége döntő: a 90%-os próbalefedettség érvényteleníthető, a részleges löketpróba megerősítheti az igazolást.
- A hibabiztos kialakítás a téves leállások gyakoriságát növeli; a redundáns tápellátás és a 2oo3 szavazás egyszerre őrzi meg a biztonságot és a rendelkezésre állást.
- Nagy igénybevételnél a PFH a mérőszám; a felfedett hibák közös okú hányada és a diagnosztikai reakció ideje ilyenkor különösen fontos.

Önellenőrzés

1. Egy távadó $\lambda_S = 3 \cdot 10^{-7}$, $\lambda_{DD} = 9 \cdot 10^{-7}$, $\lambda_{DU} = 3 \cdot 10^{-7}\ \mathrm{h}^{-1}$. Mekkora a DC és az SFF? B típusú elemként HFT = 0 mellett milyen SIL-ig alkalmazható az 1H út szerint?
2. Számold ki egy 1oo1 szelep $\mathrm{PFD}_{\mathrm{avg}}$-értékét $\lambda_{DU} = 2{,}5 \cdot 10^{-6}\ \mathrm{h}^{-1}$, $T_1 = 2$ év és MTTR = 8 óra mellett! Milyen SIL-tartományba esik?
3. Miért csökkenti a PFD-t csak kis mértékben, ha a ReaktorVéd távadóit 2oo3 architektúrára cseréljük? Akkor miért érdemes mégis megfontolni?
4. A ReaktorVéd szelepeinek közös okú tényezője 10%. Milyen műszaki intézkedésekkel lehetne csökkenteni?
5. Miért érvényteleníthető a SIL-igazolást, ha az üzemeltető a próbaidőközt egy évről kettőre növeli?

Megoldás.

1. $\lambda_D = 1{,}2 \cdot 10^{-6}$, $\mathrm{DC} = 9/12 = 75\%$; $\mathrm{SFF} = (3+9)/15 = 80\%$. B típusú elemként, HFT = 0 mellett, 60–90% közötti SFF-fel az 1H út szerint SIL 1-ig alkalmazható.
2. $\mathrm{PFD} = 2{,}5 \cdot 10^{-6} \cdot (8760 + 8) = 2{,}19 \cdot 10^{-2}$; SIL 1 tartomány ($10^{-2} \le \mathrm{PFD} < 10^{-1}$).
3. Mert a távadók PFD-jét már 1oo2 esetén is a közös okú tag uralja ($8{,}8 \cdot 10^{-5}$), amelyen a 2oo3 nem változtat, és a funkció PFD-jének csak 7%-át adják a távadók. A 2oo3 előnye a téves leállások jelentős csökkentése, és hogy egy távadó karbantartása alatt is megmarad a hibatűrés (1oo2 üzemmódban).
4. Különálló mágnesszelepek és működtető levegőellátás, eltérő típusú vagy gyártójú szelepek (diverzitás), fizikai szétválasztás, eltolt időpontú karbantartás és tesztelés különböző személyekkel, a folyamatközeghez jobban illeszkedő anyagválasztás, valamint a részleges löketpróba, amely a közös okú hibák egy részét is korán felfedi.
5. Mert a PFD közel arányos a próbaidőközzel: két év mellett a funkció PFD-je $2{,}95 \cdot 10^{-3}$, ami meghaladja a $2 \cdot 10^{-3}$-os követelményt. A próbaidőköz az igazolás feltevése, tehát a változtatás előtt az igazolást meg kell ismételni (változáskezelés, 4. fejezet).

Továbbolvasás

- IEC 61508-2 és -6:2010 – architekturális korlátok, a PFD- és PFH-képletek (IEC 2010a).
- IEC 61511-1:2016 – a folyamatipari biztonsági műszeres rendszerek követelményei, a HFT-táblázat és a téves leállások specifikálása (IEC 2016b).
- Rausand, M. (2014): *Reliability of Safety-Critical Systems: Theory and Applications.* Wiley – a PFD/PFH számítása, a részleges löketpróba és a nem tökéletes proof teszt (Rausand 2014).
- Hauge, S. et al. (2013): *Reliability Prediction Method for Safety Instrumented Systems – PDS Method Handbook.* SINTEF (Hauge és mtsai. 2013).
- Lundteigen, M. A., Rausand, M. (2009): Architectural constraints in IEC 61508: Do they have the intended effect? *Reliability Engineering & System Safety* (Lundteigen és Rausand 2009).
- Smith, D. J., Simpson, K. G. L. (2020): *The Safety Critical Systems Handbook*, 5. kiadás – gyakorlati útmutató a funkcionális biztonsági számításokhoz (Smith és Simpson 2020).

IV. rész – Tervezés biztonságra



Pannon Egyetem
University of Pannonia

14.

Biztonságos architektúrák és tervezési elvek



Az eddigi fejezetek azt mutatták meg, hogyan azonosítjuk a veszélyeket, és hogyan értékeljük egy adott terv kockázatát és megbízhatóságát. Ez a fejezet a tervezőé: milyen elvek és architektúrák vezetnek eleve biztonságos rendszerekhez? A sorrend nem mellékes. A leghatékonyabb a veszélyt magát megszüntetni vagy csökkenteni (inherens biztonság), ezt követik a működés nélkül védő passzív, majd az aktív védelmek. Az aktív védelmeknél azt kell eldönteni, mi történjen hiba esetén: álljon le a rendszer biztonságosan, kapcsoljon ki egy funkció, működjön tovább csökkentett teljesítménnyel, vagy maradjon teljesen működőképes. Ezek a döntések határozzák meg a redundancia, a diagnosztika és a közös okú hibák elleni védekezés szükséges mértékét. A fejezet a folyamatipari, a gépipari, a közúti és a repülési gyakorlat példáin, köztük a 2014-es jegyzetben bemutatott haszonjármű-fékrendszer korszerűsített elemzésén vezet végig.

Tanulási eredmények

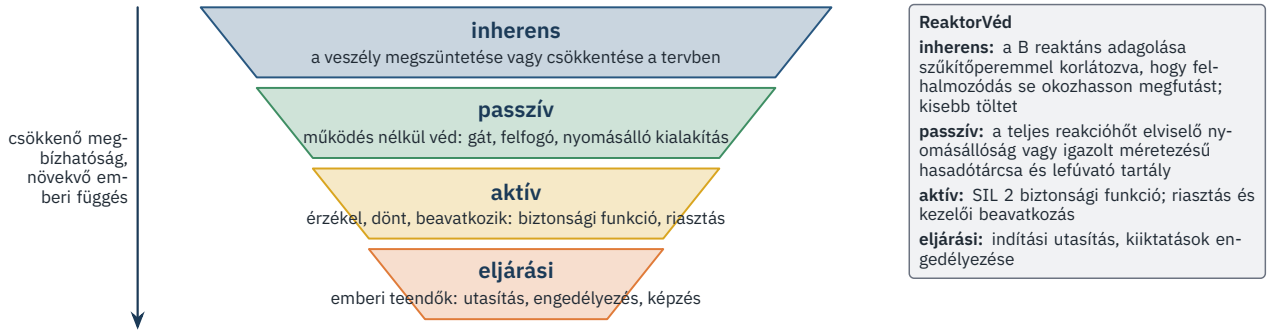
A fejezet végére képes leszel

- alkalmazni a kockázatcsökkentő intézkedések hierarchiáját és az inherens biztonság elveit;
- megkülönböztetni a hibabiztos, a hibára elnémuló, a csökkentett működésű és a hibatűrő rendszereket, és egy adott funkcióhoz kiválasztani a szükséges hibaviselkedést;
- jellemezni a hardveres, információs, időbeli és szoftveres redundancia formáit, és értékelni a homogén és a diverz redundanciát;
- diagnosztikai megoldásokat választani, és összehangolni a diagnosztikai időket a hibatűrési időkerettel;
- felismerni a közös okú hibák forrásait, és tervezési intézkedéseket javasolni ellenük;
- elemezni a többszintű felügyeleti koncepciókat és a redundáns fékrendszer-architektúrákat.

14.1. A kockázatcsökkentés hierarchiája és az inherens biztonság

A 3. fejezetben láttuk, hogy az ISO 12100 háromlépcsős módszere a természeténél fogva biztonságos tervezéssel kezdi a kockázatcsökkentést, és csak ezután alkalmaz védőberendezéseket és tájékoztatást (ISO 2010). A folyamatipar ugyanezt a gondolatot négy szinttel fejezi ki (14.1. ábra): **inherens** (a veszély megszüntetése vagy csökkentése), **passzív** (működés nélkül védő eszközök, például gátak és felfogó medencék), **aktív** (érzékelő, döntő és beavatkozó védelmek) és **eljárási** (emberi teendők) intézkedések

(Kletz és Amyotte 2010). Lefelé haladva a védelem egyre több elem helyes működésén és egyre inkább az emberen múlik, ezért egyre kevésbé megbízható.



14.1. ábra. A kockázatcsökkentő stratégiák hierarchiája a ReaktorVéd példáival. A felső szintek kevesebb elem helyes működésén múlnak, ezért megbízhatóbbak.

Az **inherens biztonság** fogalmát Trevor Kletz vezette be az 1970-es években, azzal a ma is idézett gondolattal, hogy ami nincs, az nem is szivároghat el. Fő elvei (Kletz és Amyotte 2010):

- **minimalizálás (intenzifikálás):** kevesebb veszélyes anyag, kisebb berendezések, kisebb tárolt mennyiség;
- **helyettesítés:** kevésbé veszélyes anyag vagy eljárás (például vizes oldószer a gyúlékony helyett);
- **mérséklés:** enyhébb körülmények (alacsonyabb hőmérséklet és nyomás, hígítás), vagy olyan kialakítás, amely korlátozza a lehetséges energiát;
- **a hatások korlátozása:** olyan tervezés, amely a meghibásodás következményeit szűk körre korlátozza (például a térköz, a falak, a betáplálás fizikai korlátja);
- **egyszerűsítés:** kevesebb lehetőség a hibára, könnyen érthető, ellenőrizhető állapotok, a hibás összeszerelés lehetetlenné tétele.

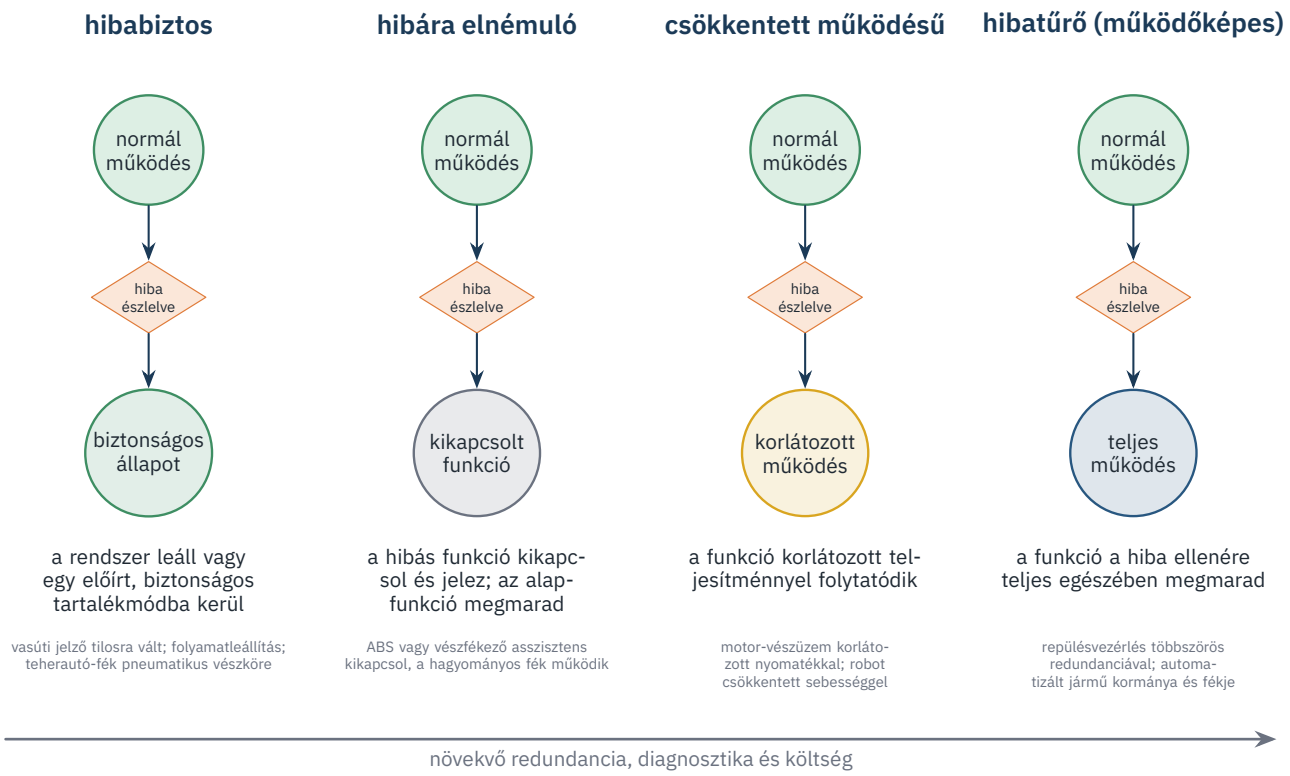
Az inherens biztonság a legnagyobb hatású, de csak a tervezés korai szakaszában érvényesíthető olcsón (4. fejezet). Később a változtatás egyre drágább, és a tervezők kénytelenek aktív védelmekkel kompenzálni azt, amit a koncepcióban elmulasztottak. Az energia–gát gondolat (1.4.2. szakasz) szerint az inherens biztonság az energiaforrás csökkentését jelenti, minden további intézkedés pedig gátakat épít a forrás és a védendő érték közé.

ReaktorVéd: inherens megoldások vizsgálata

A ReaktorVéd félszakaszos (*semi-batch*) reaktor: a B reaktánst folyamatosan adagolják. A tervezőcsapat a HAZOP után inherens megoldásokat is megvizsgált. A legfontosabb: az adagolás maximális áramlását egy szűkítőperem fizikailag korlátozza, így a reaktorban felhalmozódó, el nem reagált B mennyisége akkor sem elég a megfutáshoz, ha a hűtés kiesik (a hatások korlátozása). Egy másik lehetőség olyan oldószer, amelynek forráspontja a bomlási hőmérséklet alatt van, így a forralás elvonja a hőt (mérséklés); ez azonban a termék minőségét rontaná. Ha az inherens megoldások a forgatókönyvet megszüntetik vagy a következmény súlyosságát csökkentik, a LOPA kockázatcsökkentési követelménye (11.2. szakasz) és vele a biztonsági funkció SIL-je is csökken. A biztonsági funkcióra így is szükség van, de a nagyobb biztonsági tartalék és az egyszerűbb igazolás a tervezés elején tett döntés eredménye.

14.2. Hibaviselkedés: mi történjen hiba esetén?

Minden aktív biztonsági rendszer tervezésének kulcskérdése, hogy a rendszer hogyan viselkedjen, ha egy elemében hiba keletkezik. Négy alapvető stratégiát különböztetünk meg (14.2. ábra). Ezeket a 2014-es jegyzet a haszonjárművek elektronikus fékrendszerein mutatta be.



14.2. ábra. A hibaviselkedés négy alapvető stratégiája. Balról jobbra nő a szükséges redundancia, diagnosztika és költség.

Hibabiztos (*fail-safe*) rendszer hiba esetén egy előre meghatározott biztonságos állapotba kerül. A klasszikus példa a vasúti jelző, amely áramkimaradáskor a gravitáció vagy egy rugó hatására tilosat mutat, és a nyugalmi áramú elv, amely szerint a védelmet áram tartja üzemkész állapotban, és bármilyen szakadás leállást okoz. A ReaktorVéd energiamentesítésre záró szelepei is hibabiztosak (10.6. szakasz). A hibabiztos stratégia feltétele, hogy *létezzen* biztonságos állapot, amely gyorsan és megbízhatóan elérhető. Egy vegyi reaktorban ez a betáplálás leállítása, egy repülőgépen viszont nincs ilyen: a repülés közben „leállított" gép nem biztonságos.

Hibára elnémuló (*fail-silent*) rendszer hiba esetén kikapcsolja a hibás funkciót, és ezt jelzi; az alapfunkció egy másik, egyszerűbb rendszer révén megmarad. Az ABS hibája esetén a fék blokkolásgátló nélkül, de működik; a vészfékező asszisztens hibája esetén a vezető figyelmeztetést kap, és a hagyományos fék változatlanul rendelkezésére áll. A hibára elnémuló stratégia lényege, hogy a hibás rendszer ne avatkozzon be hibásan: a „hallgatás" biztonságosabb, mint a téves beavatkozás.

Csökkentett működésű (*fail-degraded*, *fail-soft*) rendszer hiba esetén korlátozott teljesítménnyel folytatja a működést. A motorvezérlés vészüzemben korlátozott nyomatékkal engedi a járművet a szervizig (*limp home*), a kollaboratív robot érzékelőhiba esetén csökkentett sebességgel dolgozik, vagy egy kétcsatornás folyamatirányítás az egyik csatorna kiesése után egycsatornásként, de szigorúbb felügyelettel működik.

Hibatűrő vagy **működőképes** (*fail-operational*) rendszer a hiba ellenére teljes egészében ellátja a funkcióját. Erre ott van szükség, ahol nincs biztonságos állapot, vagy az csak a funkció további működésével érhető el: a repülésvezérlésben, a magasan automatizált járművek kormány- és fékrendszerében, ahol a rendszernek a hiba után is biztonságos helyre kell vinnie a járművet (*minimal risk manoeuvre*), vagy a vasúti jelzőrendszerek rendelkezésre állást igénylő részein (UNECE 2021b).

A választás a funkciótól és a biztonságos állapot létezésétől függ, nem a műszaki ízléstől. Az SAE J3016 automatizálási szintjei (16. fejezet) ebből a szempontból is határvonalat jelentenek: amíg a vezető a tartalék (2. szintig), az automatizált funkció lehet hibára elnémuló; ha a rendszernek kell átvennie a tartalék szerepét is (3. szinttől), legalább egy átmeneti időre hibatűrőnek kell lennie (SAE International 2021).

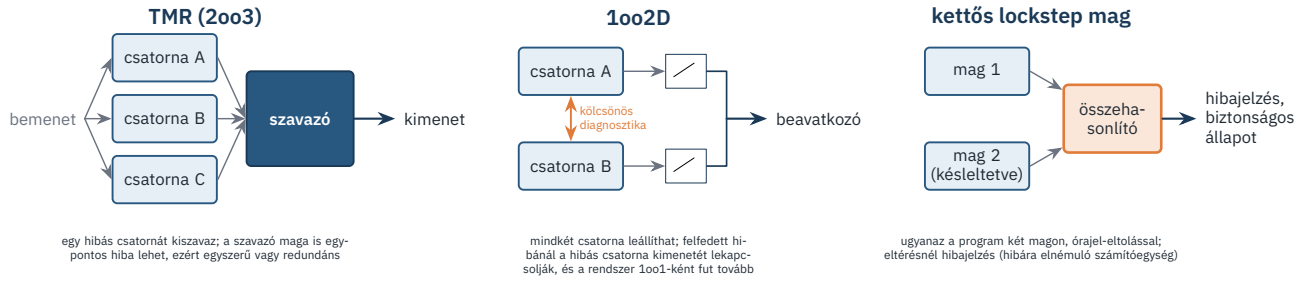
14.3. Redundancia

A redundancia az a többlet, amely a funkcióhoz önmagában nem szükséges, de hiba esetén lehetővé teszi a funkció fenntartását vagy a hiba felismerését (Avižienis és mtsai. 2004). Négy formája van:

- **hardveres redundancia:** több fizikai csatorna (érzékelő, számítóegység, beavatkozó);
- **információs redundancia:** többletinformáció az adatokban (paritás, ellenőrző összeg, hibajavító kód, biztonsági kommunikációs protokoll sorszámmal és időbélyeggel);
- **időbeli redundancia:** a művelet megismétlése (például egy számítás kétszeri elvégzése, egy üzenet ismételt küldése), amely az átmeneti hibák ellen véd;
- **szoftveres redundancia:** ugyanazon funkció eltérő megvalósításai (N-verziós programozás, helyreállító blokkok).

A hardveres redundancia lehet **statikus** (a csatornák egyidejűleg működnek, és szavazás dönti el a kimenetet) vagy **dinamikus** (a hibát észlelik, és átkapcsolnak a tartalékra). A legismertebb statikus architektúra a **háromszoros moduláris redundancia** (*triple modular redundancy*, TMR): három csatorna és egy többségi szavazó, amely egy hibás csatornát „kiszavaz" (14.3. ábra). A TMR a 2oo3 struktúra számítástechnikai megfelelője (9.3. szakasz). Gyenge pontja a szavazó, amely egypontos hiba lehet; ezért a szavazót vagy nagyon egyszerűre és ellenőrizhetőre tervezik, vagy maga is redundáns. A folyamatiparban elterjedt **1oo2D** architektúrában két csatorna kölcsönösen diagnosztizálja egymást; felfedett hiba esetén a hibás csatorna kimenetét lekapcsolják, és a rendszer egycsatornásként, de szigorúbb felügyelettel működik tovább. A gépjárművekben gyakori a **kettős lockstep** mikrokontroller,

amelyben két processzormag ugyanazt a programot futtatja kis időeltolással, és egy összehasonlító áramkör az első eltérésnél hibát jelez: ez egy hibára elnémuló számítóegység.



14.3. ábra. Három jellegzetes redundáns architektúra: a többségi szavazású TMR, a kölcsönösen diagnosztizált 1oo2D és a kettős lockstep processzor.

A többcsatornás rendszerekben sajátos veszély a **bizánci hiba**: egy hibás csatorna a többi csatornának *különböző* értékeket közöl, így azok nem tudnak egyetérteni. Lamport, Shostak és Pease klasszikus eredménye szerint f ilyen hiba tolerálásához legalább $3f + 1$ csatorna és megfelelő kommunikációs protokoll szükséges, ha az üzenetek nincsenek hitelesítve (Lamport és mtsai. 1982). Ezért van a nagy integritású repülésvezérlő és űrtechnikai rendszerekben négy vagy több csatorna, illetve gondosan tervezett, hitelesített adatcsere.

14.3.1. Homogén és diverz redundancia

A **homogén** redundancia azonos csatornákat használ. Véletlenszerű hardverhibák ellen hatékony, de a közös okú hibák és különösen a szisztematikus hibák (tervezési, szoftverhibák) ellen nem: ha a hiba a tervben van, minden csatornában ott van. A **diverz** redundancia eltérő technológiájú, eltérő tervezésű vagy eltérő elven működő csatornákat használ, így egy tervezési hiba kisebb valószínűséggel érinti mindegyiket.

A diverzitás a repülésben a legkidolgozottabb. Az Airbus repülésvezérlő rendszerében két különböző típusú számítógép működik, eltérő processzorokkal, eltérő csapatok által, eltérő eszközökkel fejlesztett szoftverrel (Traverse és mtsai. 2004). A Boeing 777 elsődleges repülésvezérlő számítógépe „háromszor hármas” redundanciájú: három csatorna, mindegyikben három, különböző processzorcsaládra épülő számítási út (Yeh 1996). A szoftveres diverzitás hatékonysága azonban korlátozott. Knight és Leveson kísérletében egymástól függetlenül fejlesztett programváltozatok sokkal gyakrabban hibáztak ugyanazon bemeneteknél, mint ahogy függetlenségük esetén várható lett volna (Knight és Leveson 1986); a programozók hajlamosak ugyanazokat a nehéz eseteket elrontani. Az N-verziós programozás ezért nem helyettesíti a gondos fejlesztést (Avižienis 1985), és a diverzitás akkor a leghatékonyabb, ha különböző *elveken* alapul (például egy összetett szabályozó mellett egy egyszerű, fizikai határértéket figyelő felügyelet).

14.4. Diagnosztika

A diagnosztika a rejtett hibákból felfedett hibákat csinál: csökkenti a λ_{DU}-t, és ezzel a PFD-t vagy a PFH-t (13. fejezet). A gyakori diagnosztikai megoldások:

- **plauzibilitás-ellenőrzés:** a mért érték fizikailag lehetséges-e (tartomány, változási sebesség), összhangban van-e más mérésekkel;

- **csatornák összehasonlítása:** két érzékelő eltérése egy határértéknél nagyobb (kölcsönös diagnosztika);
- **élő nulla és vonalfelügyelet:** a 4–20 mA-es hurokban a 3,6 mA alatti áram szakadást, a 21 mA feletti zárlatot jelez;
- **tesztimpulzusok:** a kimeneteket rövid impulzusokkal ellenőrzik, hogy le tudnak-e kapcsolni;
- **memória- és processzortesztek:** a programmemória ellenőrző összege, RAM-tesztek, utasításkészlet-tesztek;
- **programfutás-felügyelet:** az egyszerű *watchdog* csak azt ellenőrzi, hogy a program időben jelentkezik-e, az ablakos watchdog azt is, hogy nem túl korán, a logikai programfutás-felügyelet pedig azt, hogy a program a helyes sorrendben járja-e be az ellenőrzőpontokat.

Az IEC 61508-2 A melléklete minden megoldáshoz megadja az elérhető diagnosztikai lefedettséget (alacsony: 60%, közepes: 90%, nagy: 99%) (IEC 2010a). Az egyszerű watchdog például csak alacsony lefedettséget ad, a logikai programfutás-felügyelet közepeset vagy nagyot.

14.4.1. A diagnosztika időzítése

A diagnosztika csak akkor ér valamit, ha a hibát *időben* felismeri, és a rendszer időben eljut a biztonságos állapotba. Az ISO 26262 ezt a **hibatűrési időkerettel** (*fault tolerant time interval*, FTTI) fejezi ki: ennyi idő telhet el egy hiba keletkezésétől a veszélyes esemény bekövetkezéséig, ha a biztonsági mechanizmusok nem lépnek közbe. A hiba felismerésének ideje és a hibareakció ideje együtt nem haladhatja meg az FTTI-t (ISO 2018a). A folyamatiparban ugyanezt a **folyamatbiztonsági idő** (*process safety time*) fejezi ki, amelyen belül a biztonsági funkciónak le kell állítania a folyamatot; a ReaktorVéd 10 másodperces válaszideje (4. fejezet) ebből következik. Ha egy diagnosztikai teszt csak óránként fut, de a hiba néhány másodperc alatt veszélyes helyzetet teremthet, a teszt a PFD-számításban nem tekinthető diagnosztikának.

14.5. Védekezés a közös okú hibák ellen

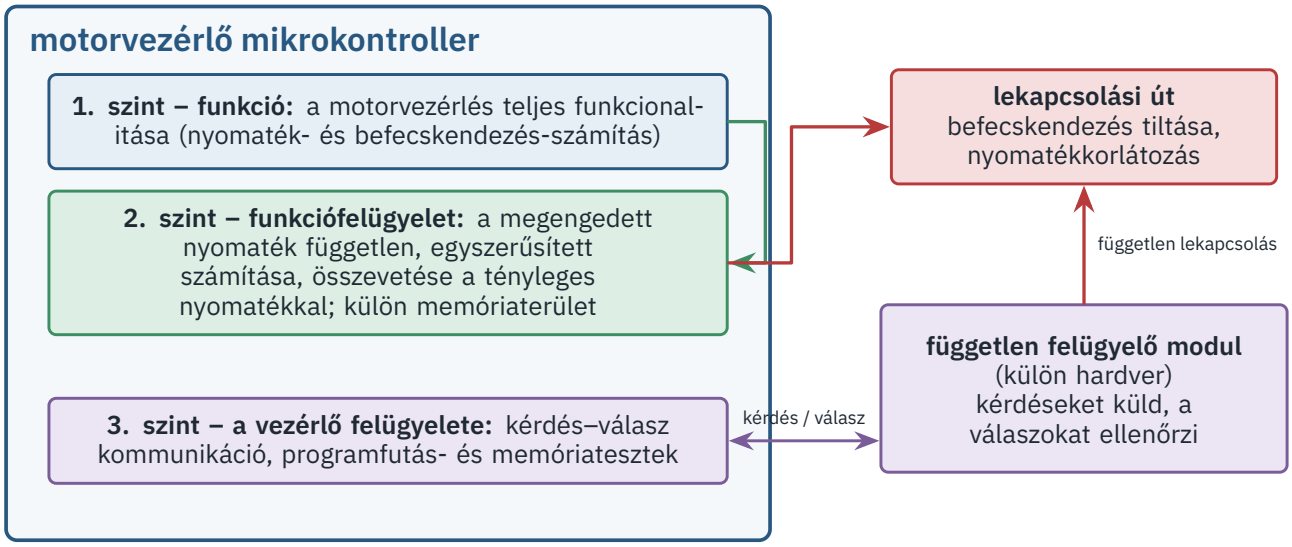
A 10–13. fejezet számításai egyértelművé tették, hogy a redundáns rendszerekben a közös okú meghibásodás gyakran uralja a kockázatot. A közös okok főbb forrásai (Mosleh és mtsai. 1988; IEC 2010a):

- **környezet:** hőmérséklet, rezgés, nedvesség, elektromágneses zavarok, tűz, elárasztás, amely egyszerre több csatornát ér;
- **közös támogató rendszerek:** tápellátás, műszerlevegő, hűtés, kommunikációs hálózat;
- **tervezés:** azonos tervezési vagy szoftverhiba, azonos alkatrész-gyengeség;
- **gyártás és telepítés:** azonos gyártási sorozat, azonos telepítési hiba;
- **üzemeltetés és karbantartás:** ugyanaz a hibás kalibrálás vagy beállítás minden csatornán, ugyanaz a személy, ugyanaz a hibás eljárás.

Az ellenük alkalmazható intézkedések három csoportba sorolhatók: **szétválasztás** (térbeli elkülönítés, külön kábelút, külön tápellátás, külön szekrény), **diverzitás** (eltérő technológia, gyártó, elv vagy fejlesztőcsapat) és **függetlenség a folyamatokban** (eltolt időpontú karbantartás különböző személyekkel, független ellenőrzés, a változtatások gondos kezelése). Az IEC 61508-6 D mellékletének pontozásos eljárása éppen ezeket az intézkedéseket értékeli, és a pontszámból adja a β-tényezőt (10.6. szakasz). A gépiparban az ISO 13849-1 hasonló listát használ, amelyen legalább 65 pontot kell elérni (ISO 2023). A közös okú hibák elleni védekezés tehát nem csak számítási kérdés: minden pont egy konkrét tervezési vagy szervezeti döntést jelent.

14.6. Többszintű felügyeleti koncepciók: az E-Gas példája

Az elektronikus gázpedál (*drive-by-wire*) bevezetésével a motor nyomatékát már egy szoftver számolja ki, és egy szoftverhiba nem kívánt gyorsítást okozhat. A német autóipar ezért egy szabványosított, háromszintű felügyeleti koncepciót dolgozott ki, amely az ISO 26262 előtt keletkezett, de máig a biztonságos járműelektronika egyik mintapéldája (14.4. ábra) (EGAS Workgroup 2015).



14.4. ábra. Az E-Gas háromszintű felügyeleti koncepciója. A funkciót egy diverz funkciófelügyelet és egy független hardveres felügyelet ellenőrzi, és mindkettő a funkciótól független úton állíthatja biztonságos állapotba a motort.

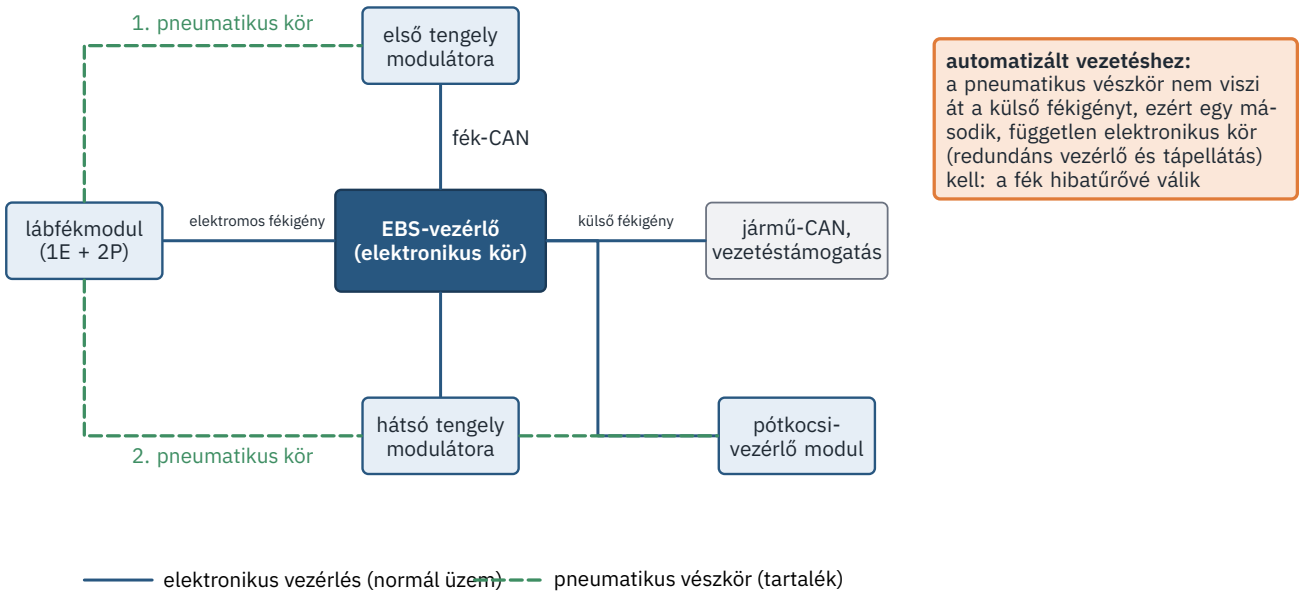
- **1. szint (funkció):** a motorvezérlés teljes funkcionalitása, a vezető kívánságából és a sok egyéb bemenetből a nyomaték és a befecskendezés kiszámítása. Ez a bonyolult, gyakran változó szoftver.
- **2. szint (funkciófelügyelet):** egy egyszerűsített, a funkciótól eltérő módon számított *megengedett* nyomaték, amelyet összevetnek a tényleges (az 1. szint által kért) nyomatékkal. Eltérés esetén nyomatékkorlátozás vagy a befecskendezés tiltása következik. A 2. szint ugyanazon a processzoron fut, de külön memóriaterületen, védett adatokkal.
- **3. szint (a vezérlő felügyelete):** egy független felügyelő modul (külön hardver) kérdés–válasz kommunikációval ellenőrzi, hogy a fő processzor helyesen működik-e: helyesen hajtja-e végre az utasításokat, és lefut-e a 2. szint programja. Hiba esetén a felügyelő modul a fő processzortól függetlenül kapcsolja le a beavatkozókat.

A koncepció három általános tervezési elvet testesít meg: a bonyolult funkciót egy egyszerűbb, *diverz* felügyelet ellenőrzi; a felügyeletet egy *független* hardver felügyeli; és a biztonságos állapotba vezető út *független* a felügyelt funkciótól. Ugyanezt a mintát követi a ReaktorVéd is, ahol a bonyolult BPCS-t egy egyszerű, független biztonsági funkció felügyeli, és a gépek biztonsági vezérlései, ahol a funkcionális vezérlő mellett egy biztonsági vezérlő gondoskodik a biztonságos leállásról.

14.7. Esettanulmány: haszonjármű-fékrendszerek

A 2014-es jegyzet Fülep Tímea által írt fejezetei a haszonjárművek elektronikus fékrendszerein mutatták be a hibaviselkedési stratégiákat. Ez a példa ma is tanulságos, sőt az automatizált vezetés miatt aktuálisabb, mint valaha.

Az Európában az 1990-es évek második fele óta elterjedt **elektronikus fékrendszer** (EBS) fő eleme a központi vezérlőegység, amely saját fék-CAN-hálózaton irányítja a tengelyek fékmodulátorait és a pótkocsi-vezérlő modult (14.5. ábra). Normál üzemben a vezető fékigényét a lábfékmodul elektromos jelként adja át, és az elektronika számolja ki a fékerő-elosztást, az ABS- és a stabilitási beavatkozásokat. A redundancia szempontjából a rendszer **1E+2P** felépítésű: egy elektronikus kör és két pneumatikus vészkör. Ha az elektronikus körben hiba keletkezik, a rendszer a hiba súlyosságától függően részleges vagy teljes pneumatikus vészüzembe kapcsol, amelyben az alapfékezés teljes értékű, de az elektronikus funkciók (ABS, fékerő-elosztás, stabilitásvezérlés) nem érhetők el. Ez a fékrendszer **hibabiztos**: egyszeres hiba esetén is teljesíti a jogszabályban előírt fékhatást (UNECE 2016). Az ABS és a stabilitásvezérlés önmagában **hibára elnémuló**: hiba esetén kikapcsol, és az alapfék megmarad.



14.5. ábra. A haszonjárművek 1E+2P felépítésű elektronikus fékrendszerének vázlata. A pneumatikus vészkörök a vezető fékigényét átviszik, de a külső (automatizált) fékigényt nem.

A 2014-es jegyzet már rámutatott, hogy költségokokból vita folyik az egyik pneumatikus kör elhagyásáról (1E+1P), amely a jogszabályi előírásokat még kielégíti, és megtartja a hibabiztos tulajdonságot. Az automatizált vezetés szempontjából azonban mindkét változatnak ugyanaz a gyengesége: a pneumatikus vészkör csak a vezető lábának parancsát viszi át, a külső, elektronikus fékigényt nem. Ha a jármű vezetője nem figyel (mert a rendszer vezet), az elektronikus kör hibája után nincs, aki fékezzen. Egy magasan automatizált járműben a fékrendszernek ezért **hibatűrőnek** kell lennie: egy második, független elektronikus vezérlési körre van szükség, saját tápellátással és kommunikációval, amely az első kör kiesése után is végrehajtja a külső fékigényt, legalább addig, amíg a jármű egy minimális kockázatú manőverrel biztonságosan megáll (UNECE 2021b). A hibaviselkedési stratégia tehát nem a rendszer, hanem a *funkció és a felelősségmegosztás* tulajdonsága: ugyanaz a fék a vezetővel hibabiztos lehet, a vezető nélkül hibatűrőnek kell lennie.

VészFék: hibára elnémuló asszisztens, hibatűrő fék

A vészfékező asszisztens a vezető kiegészítője (2. szintig), ezért hibára elnémuló: ha a radar vagy a kamera hibás, vagy az érzékelés megbízhatatlan (például erős esőben), a funkció kikapcsol, a műszerfalon jelzés jelenik meg, és a vezető a hagyományos fékkel fékez. A téves fékezés elleni védelem ugyanakkor kritikus (10. fejezet): a rendszernek akkor is el kell némulnia, ha nem biztos a dolgában. A fékrendszer maga (a fékerő-erősítő, a stabilitásvezérlő hidraulikája) a vezetővel együtt hibabiztos. Ha ugyanezt a funkciót egy magasan automatizált jármű látja el, ahol a vezető nem tartalék, a fékrendszer és a környezetérzékelés is hibatűrő kell legyen: redundáns érzékelés, redundáns számítóegység és redundáns fékbeavatkozás.

KoboCella: inherens biztonság és felügyelt mozgás

A kollaboratív robot biztonsága elsősorban inherens: a könnyű, lekerekített élű kar, a korlátozott tömeg és sebesség miatt egy ütközés ereje eleve kisebb (4. fejezet). Erre épülnek az aktív funkciók: az erő- és teljesítménykorlátozás, a sebesség- és távolságfelügyelet és a biztonságosan felügyelt megállás, amelyeket a robotvezérlő biztonsági része kétcsatornásan, kölcsönös diagnosztikával valósít meg (ISO 2025a, 2016). Hiba esetén a robot hibabiztosan megáll (a hajtások biztonságos nyomatékmentesítése, STO). A cella tervezésének fő kérdése, hogy a biztonsági funkciók ne állítsák le feleslegesen a munkát (12.6. szakasz), mert a gyakori leállás a védelem megkerülésére ösztönöz.

Összefoglalás

- A kockázatcsökkentés hierarchiája: inherens, passzív, aktív, eljárási; a felső szintek kevesebb elem működésén múlnak, ezért megbízhatóbbak. Az inherens biztonság a tervezés elején a legolcsóbb.
- A hibaviselkedés négy stratégiája: hibabiztos (biztonságos állapot), hibára elnémuló (a hibás funkció kikapcsol), csökkentett működésű és hibatűrő (a funkció megmarad). A választás attól függ, létezik-e biztonságos állapot, és ki a tartalék.
- A redundancia lehet hardveres, információs, időbeli és szoftveres, statikus vagy dinamikus; a homogén redundancia a szisztematikus hibák ellen nem véd, a diverz redundancia igen, de a szoftveres diverzitás hatékonysága korlátozott.
- A diagnosztika a rejtett hibákat felfedett hibákká alakítja; a felismerési és a reakcióidőnek a hibatűrési időkereten vagy a folyamatbiztonsági időn belül kell maradnia.
- A közös okú hibák ellen szétválasztás, diverzitás és a folyamatok függetlensége véd.
- A többszintű felügyelet (E-Gas) mintája: diverz felügyelet, független hardveres felügyelő, független lekapcsolási út.
- Az automatizált vezetéshez a hibabiztos 1E+2P fékrendszer helyett hibatűrő, kétszeres elektronikus vezérlésű fék kell.

Önellenőrzés

1. Mondj egy-egy példát hibabiztos, hibára elnémuló, csökkentett működésű és hibatűrő rendszerre a saját szakterületedről! Miért éppen az a stratégia a megfelelő?
2. Miért nem lehet egy repülőgép repülésvezérlő rendszere hibabiztos?
3. Egy TMR-rendszer csatornái azonos szoftvert futtatnak. Milyen hibák ellen véd, és milyenek ellen nem?
4. Egy hőmérséklet-távadó hibáját óránként egyszer futó teszt ismeri fel. A folyamatbiztonsági idő 2 perc. Tekinthető-e ez a teszt diagnosztikának a PFD-számításban?
5. Sorolj fel öt konkrét intézkedést a ReaktorVéd két gyorszáró szelepének közös okú meghibásodása ellen, és sorold be őket a szétválasztás, a diverzitás és a folyamatok függetlensége csoportokba!

Megoldás.

1. Például: hibabiztos a felvonó fékje (áramkimaradáskor rugó fékez); hibára elnémuló egy kerékpár elektromos rásegítése (hiba esetén kikapcsol, pedálozni lehet); csökkentett működésű egy szerver RAID-tömbje (egy lemez kiesése után lassabban, de működik); hibatűrő egy kórházi szünetmentes tápellátás. A választás mindig attól függ, mi történik a funkció elvesztésekor, és van-e biztonságos állapot.

2. Mert repülés közben nincs biztonságos állapot, amelybe a rendszer egyszerűen „leállhatna”; a gép irányíthatóságát a hiba után is fenn kell tartani a leszállásig. Ezért a repülésvezérlés hibatűrő, többszörös és diverz redundanciával.
3. Véd a független véletlenszerű hardverhibák ellen (egy csatorna hibáját kiszavazza), és a nem közös átmeneti hibák ellen. Nem véd a szoftver tervezési hibái ellen (mindhárom csatorna ugyanúgy hibázik), a közös okú hardverhibák ellen (például közös táp vagy túlmelegedés), és a szavazó hibája ellen.
4. Nem. A hiba felismerésének ideje (legfeljebb egy óra) sokkal hosszabb a folyamatbiztonsági időnél, ezért a hiba felismerése előtt bekövetkező igény veszélyes eseményhez vezethet. A számításban ezt a hibát rejtettnek kell tekinteni, vagy sokkal gyakoribb diagnosztika kell.
5. Szétválasztás: külön mágnesszelep és külön levegőellátás a két működtetőnek; külön kábelút és tápellátás. Diverzitás: eltérő típusú vagy gyártójú szelepek (például egy golyóscsap és egy pillangószelep). A folyamatok függetlensége: eltolt időpontú karbantartás és proof teszt, különböző szerelőkkel; a karbantartás utáni működéspróba független ellenőrzéssel.

Továbbolvasás

- Kletz, T., Amyotte, P. (2010): *Process Plants: A Handbook for Inherently Safer Design*, 2. kiadás. CRC Press (Kletz és Amyotte 2010).
- Avižienis, A. et al. (2004): Basic concepts and taxonomy of dependable and secure computing. *IEEE TDSC* – a hibatűrés fogalomrendszere (Avižienis és mtsai. 2004).
- EGAS Workgroup (2015): *Standardized E-Gas Monitoring Concept for Gasoline and Diesel Engine Control Units*, 6.0 változat (EGAS Workgroup 2015).
- Traverse, P. et al. (2004): Airbus fly-by-wire: A total approach to dependability (Traverse és mtsai. 2004); Yeh, Y. C. (1996): Triple-triple redundant 777 primary flight computer (Yeh 1996).
- Knight, J. C., Leveson, N. G. (1986): An experimental evaluation of the assumption of independence in multiversion programming. *IEEE TSE* (Knight és Leveson 1986).

15.

A szoftver biztonsága



A mai biztonságkritikus rendszerekben a funkciók egyre nagyobb részét szoftver valósítja meg: a vegyi reaktor biztonsági PLC-je, a jármű fékvezérlője, a robot biztonsági felügyelete, a repülőgép repülésvezérlése. A szoftver nem kopik, nem fárad, és nem hibásodik meg véletlenszerűen; amikor mégis rosszul működik, annak oka mindig a specifikációban, a tervben vagy a kódban rejlő **szisztematikus hiba**, amely meghatározott körülmények között determinisztikusan előjön (2. fejezet). Ezért a szoftver biztonsága nem meghibásodási rátákkal, hanem a fejlesztési folyamat szigorával, az architektúrával és a verifikációval érhető el. Ebben a fejezetben áttekintjük, miben különbözik a szoftver a hardvertől, mit tanultunk a nagy szoftveres balesetekből, mit írnak elő az ágazati szabványok, és milyen technikák (kódolási szabványok, particionálás, statikus elemzés, strukturális lefedettség, formális módszerek) támogatják a biztonságos szoftver fejlesztését.

Tanulási eredmények

A fejezet végére képes leszel

- megmagyarázni, miért nem jellemezhető a szoftver megbízhatósága a hardveréhez hasonló meghibásodási rátával, és miért nem igazolható nagyon kis hibagyakoriság pusztán teszteléssel;
- a nagy szoftveres balesetek (Ariane 5, Therac-25, Boeing 737 MAX) tanulságait a szoftver-életciklus lépéseihez kötni;
- összevetni az IEC 61508-3, az ISO 26262-6, a DO-178C és a rokon szabványok fő követelményeit;
- kiválasztani a megfelelő kódolási, architekturális és verifikációs technikákat egy adott integritási szinthez;
- kiszámolni egy döntés MC/DC-lefedettségéhez szükséges teszteket;
- értékelni a formális módszerek, a meglévő szoftverek és a fejlesztőeszközök szerepét a biztonsági igazolásban.

15.1. Miben más a szoftver?

A hardver meghibásodását fizikai folyamatok okozzák (kopás, öregedés, igénybevétel), ezért statisztikailag leírható, és a redundáns elemek nagyrészt egymástól függetlenül hibásodnak meg. A szoftver hibája ezzel szemben a gyártás pillanatától jelen van minden példányban, és ugyanazon bemenetre mindig ugyanúgy jelentkezik. Néhány következmény:

- **A homogén redundancia nem véd.** Két azonos szoftvert futtató csatorna ugyanott hibázik, ahogy az Ariane 5 két inerciális referenciarendszerénél történt (15.2.1. szakasz).
- **Nincs folytonosság.** Egy mechanikus szerkezet kis terhelésváltozásra kis deformációval felel; egy programban egyetlen bit eltérése teljesen más viselkedést okozhat. Ezért a szoftvernél nem lehet néhány pontban tesztelni, és a többit „interpolálni”.
- **A bonyolultság nem látszik.** A szoftver állapottere a változók számával exponenciálisan nő, és a fizikai korlátok, amelyek a hardver bonyolultságát természetes módon fékezik, itt hiányoznak.
- **A tesztelés korlátos.** Ahogy Dijkstra megfogalmazta: a tesztelés a hibák jelenlétét mutathatja ki, a hiányukat nem.

Az utolsó pontot kvantitatívan is meg lehet fogalmazni. Butler és Finelli kimutatta, hogy egy repülésvezérlő szoftver 10^{-9} óránkénti hibagyakoriságát teszteléssel igazolni több mint százezer évnyi, reprezentatív üzemi körülmények közötti hibamentes futtatást igényelne (Butler és Finelli 1993). Littlewood és Strigini bayesi elemzése szerint a tesztelésből és az üzemi tapasztalatból reálisan csak mérsékelt megbízhatóság igazolható; a nagyon nagy megbízhatóság állítása mindig a fejlesztési folyamatra, az architektúrára és a formális érvelésre is épül (Littlewood és Strigini 1993). Ugyanez a gondolatmenet ismétlődik ma az automatizált járműveknél: Kalra és Paddock szerint a balesetek gyakoriságának statisztikailag megalapozott összevetéséhez több száz millió mérföld vezetés kellene (18. fejezet) (Kalra és Paddock 2016).

Ezért a szabványok a szoftvernél nem PFD-t vagy PFH-t, hanem **folyamat- és termékkövetelményeket** írnak elő, amelyek szigora az integritási szinttel nő. A SIL, az ASIL vagy a DAL a szoftvernél azt határozza meg, milyen módszereket kell alkalmazni, milyen részletességű dokumentációt kell készíteni, és milyen függetlenséggel kell ellenőrizni a munkát.

15.2. Tanulságok a nagy szoftveres balesetekből

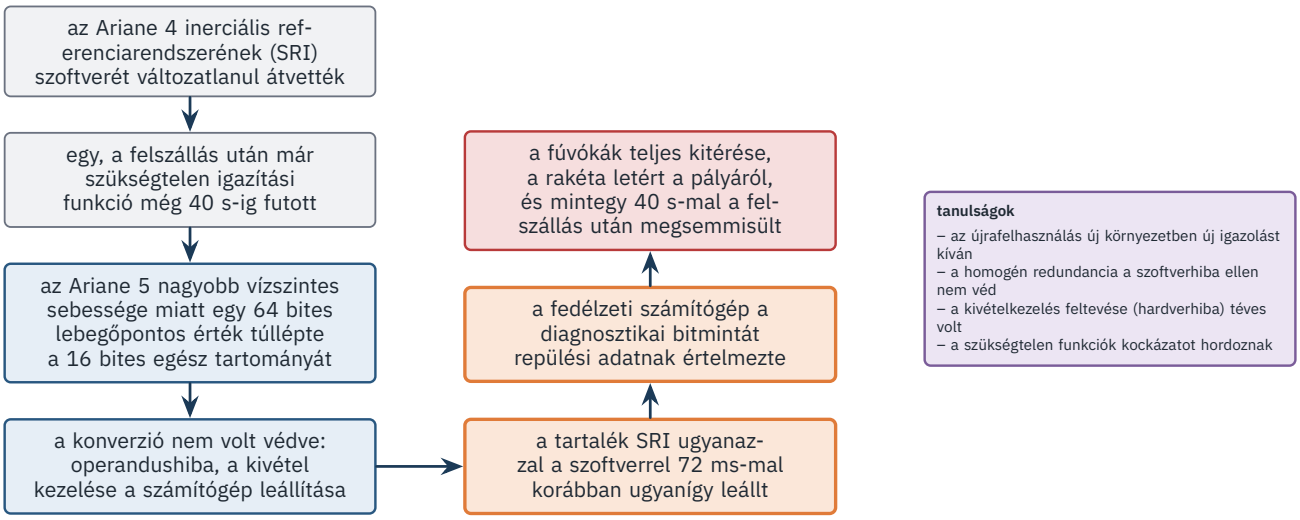
15.2.1. Ariane 5, 501-es repülés (1996)

Az 1. fejezetben bemutatott balesetek közül három a szoftver szerepe szempontjából különösen tanulságos; itt ezeket a szoftver-életciklus szemszögéből elemezzük. Az Ariane 5 1996-os első repülése mintegy negyven másodperccel a felszállás után ért véget. A vizsgálóbizottság jelentése a hibaláncot pontosan rekonstruálta (15.1. ábra) (Lions 1996). Az inerciális referenciarendszer szoftverét az Ariane 4-ből vették át. Egy igazítási funkció, amelyre a felszállás után már nem volt szükség, még negyven másodpercig futott. Az Ariane 5 nagyobb vízszintes sebessége miatt egy 64 bites lebegőpontos érték 16 bites egész számmá alakítása túlcsordult. A konverziót nem védték, mert az Ariane 4 pályáján ez nem fordulhatott elő; a kivételkezelés pedig, amely hardverhibát feltételezett, leállította a számítógépet. A tartalék rendszer ugyanazzal a szoftverrel néhány tized másodperccel korábban ugyanígy leállt. A fedélzeti számítógép a leállt rendszer diagnosztikai adatait repülési adatnak értelmezte, és teljesen kitérítette a fúvókákat.

A tanulságok általánosak: az újrafelhasználás nem mentesít az igazolás alól, mert a követelmények az új környezetben mások lehetnek; a homogén redundancia nem véd a szoftverhiba ellen; a kivételkezelés feltevései (itt: minden hiba hardverhiba) maguk is hibásak lehetnek; és a szükségtelen funkciók szükségtelen kockázatot hordoznak.

15.2.2. Therac-25 (1985–1987)

A Therac-25 sugárterápiás gyorsítóval (1. fejezet) 1985 és 1987 között hat ismert baleset történt, amelyekben a betegek a tervezett dózis sokszorosát kapták, súlyos sérüléseket és haláleseteket okozva (Leveson és Turner 1993). Az elődmodellekben hardveres reteszek akadályozták meg a veszélyes



15.1. ábra. Az Ariane 5 501-es repülésének hibalánca a vizsgálóbizottság jelentése alapján. Egyetlen szoftverhiba azért vezetett katasztrófához, mert több tervezési feltevés egyszerre bizonyult tévesnek.

üzemállapotokat; a Therac-25-ben ezeket elhagyták, és a biztonságot teljesen a szoftverre bízták. A szoftverben versenyhelyzet (*race condition*) volt: ha a kezelő gyakorlott gyorsasággal módosította a beállításokat, a gép a módosítás előtti és utáni beállítások veszélyes keverékével sugárzott. Egy másik hiba egy egybájtos számláló túlcsordulásából eredt. A hibaüzenetek érthetetlenek voltak, a kezelők megszokták, hogy figyelmen kívül hagyják őket, és a gyártó az első jelzések után sokáig nem hitte el, hogy a szoftver hibás lehet.

A Therac-25 a szoftverbiztonság tankönyvi esete lett: a szoftver nem helyettesítheti a független védelmeket, a régi kód új környezetben új hibákat hozhat elő, az ember–gép felület a biztonsági rendszer része (16. fejezet), és a balesetek kivizsgálása a szervezet kultúráján is múlik.

15.2.3. Boeing 737 MAX (2018–2019)

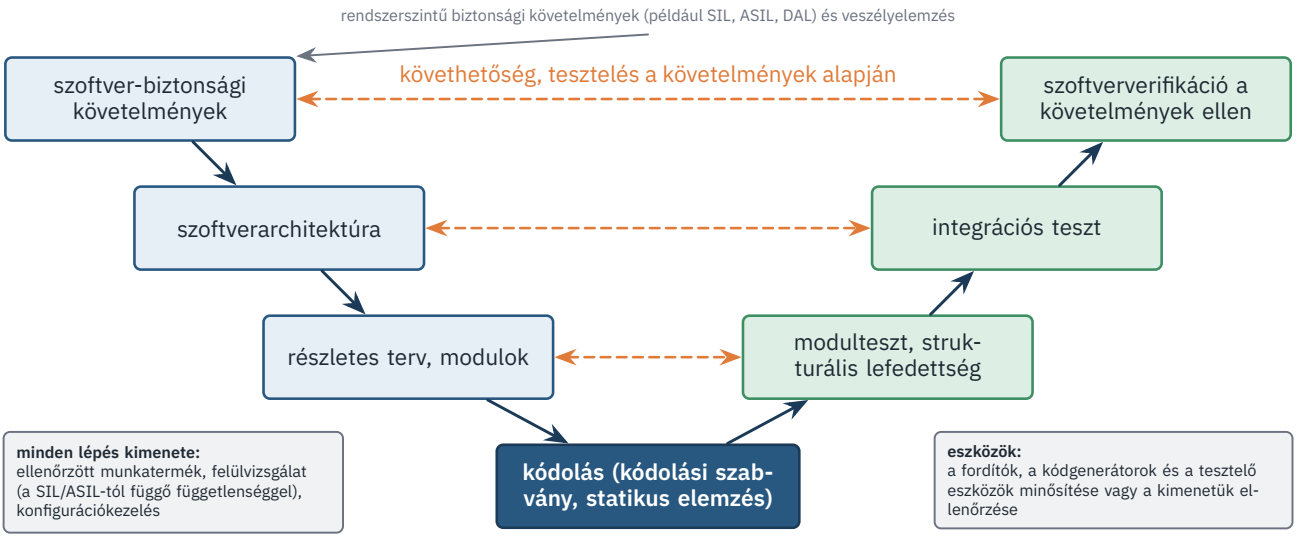
A Lion Air 610-es (2018) és az Ethiopian Airlines 302-es járatának (2019) katasztrófájában (1. fejezet) összesen 346 ember halt meg. Mindkét esetben a manőverezési jellemzőket javító rendszer (MCAS) egyetlen, hibás állásszög-érzékelő jele alapján ismételten orrnehéz irányba trimmelte a gépet. A független vizsgálatok szerint az MCAS jogosultságát a fejlesztés késői szakaszában jelentősen megnövelték, de a biztonsági értékelést nem frissítették ennek megfelelően; a rendszer egyetlen érzékelőre támaszkodott, és a biztonsági értékelés olyan pilótareakciókat feltételezett, amelyek több egyidejű figyelmeztetés mellett nem voltak reálisak (Joint Authorities Technical Review 2019).

Ez a baleset már nem programozási hibáról szól: a szoftver azt tette, amit a követelmények szerint tennie kellett. A hiba a követelményekben, a rendszerszintű biztonsági elemzésben, a változáskezelésben és az emberi teljesítményre vonatkozó feltevésekben volt. Leveson régóta hangsúlyozza, hogy a szoftverrel kapcsolatos balesetek döntő többsége a hibás vagy hiányos követelményekre vezethető vissza, nem a kódolási hibákra (Leveson 2011). Lutz a NASA űrszondáinak szoftverhibáit elemezve hasonlóra jutott: a biztonsági szempontból lényeges hibák többnyire a követelmények félreértéséből és az interfészekből eredtek (Lutz 1993). A szoftverbiztonság ezért a rendszerszintű veszélyelemzéssel kezdődik (5. és 7. fejezet).

15.3. A szabványok követelményei

Az ágazati szoftverszabványok közös szerkezetűek: egy életciklus-modellt írnak elő, amelynek minden lépéséhez követelményeket, módszereket és ellenőrzési tevékenységeket rendelnek, és ezek szigorát az integritási szinthez kötik (15.2. ábra). A legfontosabbak:

- **IEC 61508-3:** az általános szabvány szoftverrésze. Részletes táblázatokban minden SIL-hez ajánlott (R) vagy erősen ajánlott (HR) technikákat ad a követelményektől a verifikációig; ha egy erősen ajánlott technikát nem alkalmaznak, azt indokolni kell (IEC 2010a).
- **IEC 61511:** a folyamatipari *alkalmazási szoftvert* szabályozza, és megkülönbözteti a rögzített programú nyelveket (csak paraméterezés), a korlátozott változékonyságú nyelveket (például funkcióblokk-diagram a biztonsági PLC-ben) és a teljes változékonyságú nyelveket (általános programozási nyelvek); az utóbbiakra az IEC 61508-3 vonatkozik (IEC 2016b).
- **ISO 26262-6:** a közúti járművek szoftverfejlesztése ASIL-ekhez kötött módszertáblázatokkal; a 9. rész a különböző ASIL-ű szoftverelemek együttélésének (*freedom from interference*) követelményeit is rögzíti (ISO 2018a).
- **DO-178C:** a repülés szoftverszabványa öt fejlesztési biztosítottsági szinttel (DAL A–E). Nem módszereket, hanem *célkitűzéseket* ír elő: az A szinten 71, a B szinten 69, a C szinten 62, a D szinten 26 célkitűzést kell teljesíteni, egyre több közülük függetlenséggel (RTCA 2011a). Kiegészítései a formális módszerekre (DO-333) és a modellalapú fejlesztésre vonatkoznak (RTCA 2011b).
- **EN 50716:** a vasúti szoftverfejlesztés szabványa, amely 2023-ban váltotta fel az EN 50128-at és az EN 50657-et (CENELEC 2023).
- **IEC 62304:** az orvostechnikai szoftverek életciklusa, a lehetséges sérülés súlyossága szerinti A, B és C szoftverbiztonsági osztállyal, az ISO 14971 kockázatkezelési folyamatához kapcsolódva (IEC 2006c).



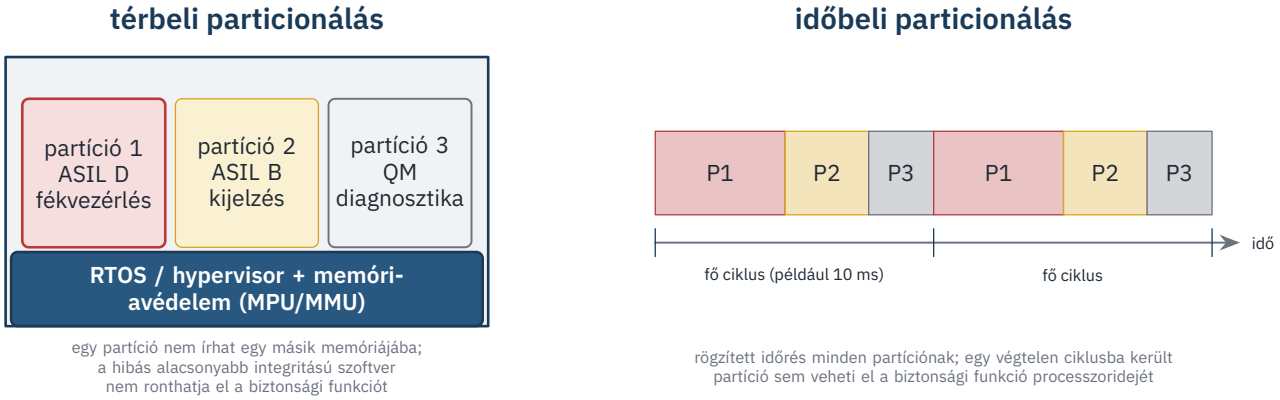
15.2. ábra. A szoftver V-modellje. A bal ág minden szintjének a jobb ágon egy verifikációs szint felel meg; a követelmények és a tesztek közötti kétirányú követhetőség az igazolás gerince.

A szabványokban közös a **kétirányú követhetőség**: minden rendszerszintű biztonsági követelményhez követhetően tartozik szoftverkövetelmény, minden szoftverkövetelményhez terv- és kódelem, valamint teszteset, és fordítva, minden kódrészletnek valamelyik követelményből kell erednie. Ez egyszerre biztosítja, hogy semmi ne maradjon ki, és hogy ne kerüljön a szoftverbe nem szándékolt funkció (ami az Ariane 5 igazítási funkciójának esete volt).

15.4. Architektúra: particionálás és vegyes kritikusság

A modern beágyazott rendszerekben egyetlen processzoron különböző integritású szoftverek futnak: egy járművezérlő egységen a fékvezérlés (ASIL D) mellett diagnosztika és kijelzés (QM vagy alacsony ASIL). Ha a kevésbé gondosan fejlesztett szoftver felülírhatja a biztonsági funkció memóriáját vagy elveheti a processzoridejét, az egész egység a legalacsonyabb integritású elem szintjére süllyed. A megoldás a **particionálás** (15.3. ábra):

- **térbeli particionálás:** a memóriavédelmi egység (MPU vagy MMU) és a valós idejű operációs rendszer vagy hypervisor biztosítja, hogy egy partíció ne írhasson a másik memóriájába;
- **időbeli particionálás:** minden partíció rögzített időrést kap, így egy végtelen ciklusba került partíció sem veheti el a biztonsági funkció idejét.



15.3. ábra. A térbeli és az időbeli particionálás. A különböző integritású szoftverek egy processzoron akkor élhetnek együtt, ha egyik sem zavarhatja a másik memóriáját és futási idejét.

A repülésben az integrált moduláris avionika és az ARINC 653 szabvány, a járműiparban az AUTOSAR és az ISO 26262-6 és -9 követelményei szabályozzák ezt (ISO 2018a). A particionálás mellett a jó biztonsági architektúra egyéb elvei: modularitás és egyszerű interfészek; a biztonsági funkciók elkülönítése a nem biztonságiaktól; a hibák terjedésének korlátozása; a defenzív programozás (a bemenetek és a köztes eredmények ellenőrzése); és a 14.6. szakaszban látott diverz felügyelet.

15.5. Kódolás és statikus elemzés

A biztonságkritikus szoftvereket olyan nyelvi részhalmazban írják, amely kizárja a nem definiált vagy nehezen elemezhető konstrukciókat. A C nyelvre a **MISRA C** irányelvei a legelterjedtebbek, legújabb kiadása a MISRA C:2025 (MISRA 2025). Ezek többek között tiltják vagy korlátozzák a dinamikus memóriafoglalást, a rekurziót, a nem definiált viselkedést okozó műveleteket és az implicit típuskonverziókat; az Ariane 5 hibája éppen egy nem védett típuskonverzió volt. Hasonló szerepet töltenek be a MISRA C++ és a CERT C irányelvei, a repülésben pedig az Ada és annak formálisan ellenőrizhető részhalmaza, a SPARK.

A **statikus elemzés** a kód futtatása nélkül vizsgálja a programot. Az egyszerű eszközök a kódolási szabályok megsértését keresik, a fejlettebbek, például az **absztrakt interpretáción** alapuló elemzők, matematikai bizonyosságot adnak bizonyos hibaosztályok (nullával osztás, túlcsordulás, tömbtúlindexelés, inicializálatlan változó) hiányáról. Az Astrée elemzővel így igazolták, hogy az Airbus repülésvezérlő szoftvereinek kódjában nincsenek futásidejű hibák (Delmas és Souyris 2007). Az ilyen elemzés olyan bizonyosságot ad, amelyet a tesztelés sosem adhatna: minden lehetséges bemenetre érvényes.

15.6. Tesztelés és strukturális lefedettség

A biztonságkritikus szoftver tesztelése **követelményalapú**: minden teszteset egy követelmény ellenőrzésére szolgál, és a követelmények normál és határesetei, valamint az érvénytelen bemenetek is lefedendők. A **strukturális lefedettség** mérése ezután megmutatja, hogy a követelményalapú tesztek a kód mely részeit futtatták meg. A le nem fedett kód vagy hiányzó követelményre (és tesztre), vagy nem szándékolt funkcióra, vagy halott kódra utal; mindhárom eset vizsgálatot igényel.

A lefedettségi mértékek egyre szigorúbbak (Hayhurst és mtsai. 2001):

- **utasításlefedettség:** minden utasítás legalább egyszer lefut;
- **döntési lefedettség:** minden döntés (elágazás) igaz és hamis kimenete is előfordul;
- **módosított feltétel/döntés lefedettség** (*modified condition/decision coverage*, MC/DC): ezen felül minden döntés minden feltételéről ki kell mutatni, hogy önállóan, a többi feltétel rögzítése mellett befolyásolja a döntés kimenetét.

Az MC/DC a repülésben a DAL A szoftver követelménye, a döntési lefedettség a DAL B-é, az utasításlefedettség a DAL C-é (RTCA 2011a). Az ISO 26262-6 az ASIL D egységtesztjénél erősen ajánlja az MC/DC-t, az ASIL B és C szinten a döntési lefedettséget (ISO 2018a). Az MC/DC előnye, hogy n feltételhez már $n+1$ teszttel elérhető, szemben a 2^n kombináció kimerítő tesztelésével, mégis kimutatja, ha egy feltétel hatását a kód rosszul valósítja meg (15.4. ábra).

teszt	A	B	C	$D = A \wedge (B \vee C)$	szerep
1	**1**	1	0	1	A párja: 4
2	1	**0**	0	0	B párja: 1, C párja: 3
3	1	0	**1**	1	
4	**0**	1	0	0	

MC/DC-feltétel: minden feltételnek egyszer igaz és egyszer hamis értéket kell felvennie, és minden feltételre kell egy olyan tesztpár, amelyben *csak az a feltétel* változik, és ez megváltoztatja a döntés kimenetét.

A: 1 és 4 B: 1 és 2 C: 2 és 3

négy teszt ($n+1$) elég a nyolc lehetséges kombináció helyett

Összehasonlítás: az utasításlefedettséghez egyetlen igaz kimenetű teszt, a döntési lefedettséghez egy igaz és egy hamis kimenetű teszt elég; egyik sem bizonyítja, hogy minden feltétel valóban hat a döntésre.

15.4. ábra. MC/DC-lefedettség az $A \wedge (B \vee C)$ döntésre. Négy teszt elég ahhoz, hogy mindhárom feltételre legyen egy olyan tesztpár, amelyben csak az adott feltétel változik, és ez megváltoztatja a döntést.

MC/DC-tesztek tervezése

Egy reteszfeltétel: a robot csak akkor indulhat, ha az ajtó zárva van (A), *és* vagy a kezelő a biztonsági zónán kívül van (B), vagy a robot csökkentett sebességű üzemmódban van (C). A döntés $D = A \wedge (B \vee C)$. Az MC/DC-tesztek (15.4. ábra): (1) $A=1, B=1, C=0 \Rightarrow D=1$; (2) $A=1, B=0, C=0 \Rightarrow D=0$; (3) $A=1, B=0, C=1 \Rightarrow D=1$; (4) $A=0, B=1, C=0 \Rightarrow D=0$. Az 1–4 pár mutatja A, az 1–2 pár B, a 2–3 pár C önálló hatását. Ha a programozó elfelejtette a C feltételt ($D = A \wedge B$), a (3)-as teszt hibát mutat (az elvárt 1 helyett 0-t ad). Döntési lefedettséghez viszont az (1) és a (2) teszt is elég volna, és egyik sem venné észre a hiányzó feltételt. Az MC/DC sem tévedhetetlen: ha a programozó a műveleti sorrendet tévesztette el, és $(A \wedge B) \vee C$-t írt, mind a négy teszt átmegy, mert a hiba csak az $A=0, B=0, C=1$ bemenetnél jelentkezik. A strukturális lefedettség tehát a tesztkészlet *elégtelenségét* mutatja ki, a kód helyességét nem bizonyítja; a felülvizsgálat és a statikus elemzés ezért elengedhetetlen kiegészítése.

A lefedettség szükséges, de nem elégséges feltétel: a 100%-os MC/DC sem bizonyítja, hogy a követelmények helyesek, vagy hogy a program időzítése és erőforrás-használata megfelelő. Ezért a biztonságkritikus szoftver verifikációja a tesztelés mellett a legrosszabb futási idő elemzését, a veremhasználat elemzését, a felülvizsgálatokat és a statikus elemzést is magában foglalja.

15.7. Formális módszerek

A formális módszerek a követelményeket és a tervet matematikai nyelven írják le, és a tulajdonságokat (például „a két gyorszáró szelep sosem kap egyszerre nyitási parancsot, ha a hőmérséklet a határ fölött van”) matematikai bizonyítással vagy kimerítő állapottér-bejárással ellenőrzik. Két fő águk a **modellellenőrzés** (*model checking*), amely egy véges állapotú modell minden lehetséges viselkedését automatikusan bejárja, és ellenpéldát ad, ha egy tulajdonság sérül; és a **tételbizonyítás**, amely interaktívan, gépi ellenőrzés mellett igazolja, hogy a program megfelel a specifikációnak.

A formális módszerek ipari alkalmazása az elmúlt évtizedekben kinőtte a kutatási kuriózum státuszt. A párizsi 14-es, vezető nélküli metróvonal biztonságkritikus szoftverét a B-módszerrel fejlesztették, és a bizonyított részekben a későbbi validálás és üzemeltetés során sem találtak hibát (Behm és mtsai. 1999). A seL4 mikrokernel az első általános célú operációsrendszer-mag, amelynek funkcionális helyességét teljes körű gépi bizonyítás igazolja (Klein és mtsai. 2009). Az Amazon Web Services a TLA+ specifikációs nyelvvel olyan tervezési hibákat talált elosztott rendszereiben, amelyeket a tesztelés és a felülvizsgálat nem tárt fel (Newcombe és mtsai. 2015). A DO-333 kiegészítés a repülésben lehetővé teszi, hogy a formális bizonyítás bizonyos tesztelési célkitűzéseket kiváltson (RTCA 2011b).

A formális módszereknek is vannak korlátai: a bizonyítás csak a specifikációhoz képest igazolja a helyességet, és ha a specifikáció hibás (mint a 737 MAX esetében a követelmények), a bizonyítás sem segít. A modellellenőrzés pedig az állapotrobbanás miatt nagy rendszereknél csak absztrakciókkal alkalmazható. Leginkább a kritikus magok (protokollok, reteszlogika, ütemezés, felügyeleti funkciók) igazolására érdemes őket használni.

15.8. Meglévő szoftverek, eszközök és konfigurációk

A biztonságkritikus termékek ritkán épülnek teljesen saját fejlesztésű szoftverből. Operációs rendszerek, kommunikációs könyvtárak, nyílt forrású komponensek és korábbi termékekből átvett modulok egyaránt előfordulnak. Ezek felhasználhatósága a szabványok szerint igazolható a megfelelő fejlesztési folyamattal (ha a dokumentáció rendelkezésre áll), az üzemi tapasztalattal (ha a korábbi használat körülményei, a változatok és a meghibásodások dokumentáltak) vagy az elem körüli védelmekkel (például a kimenet független felügyeletével) (IEC 2010a). Az orvostechnikában az ismeretlen eredetű szoftver (*software of unknown provenance*, SOUP) kockázatait külön kell értékelni (IEC 2006c). Az Ariane 5 tanulsága szerint az üzemi tapasztalat csak akkor érvényes, ha az új alkalmazás bemeneti tartománya és környezete a korábbiéval azonos.

A **fejlesztőeszközök** (fordítók, kódgenerátorok, statikus elemzők, tesztelő eszközök) maguk is hibát vihetnek a termékbe, vagy elrejthetik a hibákat. Az ISO 26262-8 az eszköz hatása és a hibái felderíthetősége alapján eszközbizalmi szintet rendel hozzájuk, és ennek megfelelően kell minősíteni őket vagy a kimenetüket ellenőrizni (ISO 2018a); a repülésben a DO-178C kiegészítése szabályozza az eszközminősítést. Végül a **konfiguráció és a paraméterek** (határértékek, időzítések, mértékegységek) a szoftver részei: egy rosszul beállított kioldási határérték ugyanolyan veszélyes, mint egy programhiba, ezért ezek is a változáskezelés és a verifikáció hatálya alá tartoznak.

> **ReaktorVéd: az alkalmazási program a biztonsági PLC-ben**
>
> A ReaktorVéd biztonsági logikáját a tanúsított PLC korlátozott változékonyságú nyelvén, tanúsított funkcióblokkokból építik fel (1oo2 szavazás a távadókra, időkésleltetés, kiiktatáskezelés, a szelepek kimenetei). Az IEC 61511 szerinti fő lépések: az alkalmazási program követelményei a biztonsági követelményspecifikációból erednek; a programot felülvizsgálják és szimulációval tesztelik; a gyári átvételi teszt minden ok–okozati kapcsolatot ellenőriz; a változtatásokat a változáskezelési eljárás szerint engedélyezik és újratesztelik (IEC 2016b). A tipikus hibák nem a funkcióblokkokban, hanem az alkalmazásban vannak: rossz mértékegység (°C helyett K), felcserélt kimenetek, olyan kiiktatás,

amely karbantartás után bekapcsolva marad, vagy egy időkésleltetés, amely a folyamatbiztonsági időnél hosszabb. A gyári és a helyszíni átvételi teszt ezért a teljes jellistát és minden kiiktatást lefed.

VészFék: szoftver ASIL-lal és teljesítménykorlátokkal

A vészfékező asszisztens szoftvere két, eltérő jellegű részből áll. A döntési és beavatkozási lánc (a fékigény kiszámítása, a téves fékezés elleni felügyelet, a kommunikáció a fékrendszerrel) hagyományos, ASIL szerint fejlesztett szoftver: követelményalapú tesztelés, MC/DC-lefedettség a legmagasabb ASIL-ű részeken, particionálás a többi funkciótól. Az érzékelési lánc (a tárgyak felismerése a kamera képéből) viszont gyakran gépi tanulásra épül, amelynek viselkedése nem vezethető vissza egyenként megírt követelményekre, és a hagyományos lefedettségi mértékek sem értelmezhetők rá. Ennek biztonsága a rendeltetésszerű funkció biztonságának (ISO 21448) és a mesterséges intelligencia biztonsági szabványainak tárgya (18. fejezet). Az architektúra ezért gyakran egy egyszerű, hagyományos módon igazolható felügyeletet tesz a gépi tanulási komponens mellé, amely a fizikailag nem plauzibilis döntéseket kiszűri.

Összefoglalás

- A szoftver hibái szisztematikusak; a szoftver nem jellemezhető a hardveréhez hasonló meghibásodási rátával, és a nagyon kis hibagyakoriság teszteléssel nem igazolható.
- A nagy szoftveres balesetek (Ariane 5, Therac-25, 737 MAX) többnyire a követelményekből, az újrafelhasználás feltevéseiből, a független védelmek hiányából és a rendszerszintű elemzés hiányosságaiból fakadtak, nem a kódolásból.
- Az ágazati szabványok az integritási szinttől függő szigorú életciklust, módszereket és független ellenőrzést írnak elő; közös elvük a kétirányú követhetőség.
- A particionálás teszi lehetővé, hogy különböző integritású szoftverek egy processzoron fussanak.
- A kódolási szabványok (MISRA C:2025) és a statikus elemzés egész hibaosztályokat zárnak ki; az absztrakt interpretáció a futásidejű hibák hiányát is bizonyíthatja.
- A követelményalapú tesztelést strukturális lefedettség egészíti ki; az MC/DC n feltételhez $n+1$ teszttel ellenőrzi minden feltétel önálló hatását.
- A formális módszerek a kritikus magok helyességét matematikai bizonyossággal igazolhatják, de csak a specifikációhoz képest.
- A meglévő szoftverek, a fejlesztőeszközök és a konfigurációs adatok is az igazolás tárgyai.

Önellenőrzés

1. Miért nem javítja a szoftverhibák elleni védelmet, ha ugyanazt a programot két processzoron futtatjuk? Milyen hibák ellen véd mégis ez az elrendezés?
2. Hány teszt kell az $A \wedge B \wedge C$ döntés MC/DC-lefedettségéhez? Add meg őket!
3. Miért nem elég a 100%-os utasításlefedettség egy biztonsági funkció szoftverének igazolásához?
4. Egy beszállító azt állítja, hogy a szoftvere tíz év alatt több ezer berendezésben hibátlanul működött, ezért SIL 3-as alkalmazásban is használható. Milyen kérdéseket tennél fel?
5. Mi a közös az Ariane 5 és a Boeing 737 MAX balesetében a szoftver szempontjából, és mi a különbség?

Megoldás.

1. Mert a szoftverhiba mindkét példányban ott van, és ugyanazon bemenetre mindkettő ugyanúgy hibázik (közös okú hiba). Az elrendezés a véletlenszerű hardverhibák és az átmeneti hibák (például sugárzás okozta bitváltás) ellen véd, ha a két csatorna eredményét összehasonlítják.
2. Négy teszt: $(1,1,1) \Rightarrow 1$, valamint $(0,1,1)$, $(1,0,1)$ és $(1,1,0)$, mindegyik $\Rightarrow 0$. Mindegyik feltételre az első teszt és a hozzá tartozó, csak abban a feltételben eltérő teszt alkot párt.
3. Mert csak azt mutatja, hogy minden utasítás lefutott, azt nem, hogy minden döntés mindkét kimenete és minden feltétel hatása ellenőrzött; és semmit sem mond a követelmények helyességéről, az időzítésről, a hiányzó funkciókról vagy a nem szándékolt viselkedésről.
4. Ugyanabban a változatban futott-e a szoftver? Hasonló volt-e az alkalmazás, a bemeneti tartomány, a környezet és a használati mód? Hogyan gyűjtötték és elemezték a meghibásodásokat, és ki ellenőrizte ezt? Mennyi az összes üzemóra, és ebből milyen konfidenciával mekkora hibagyakoriság állítható? Ismertek-e a

szoftver fejlesztési folyamata és dokumentációja? Az IEC 61508 üzemi tapasztalatra vonatkozó követelményei mindezek dokumentálását kívánják.

5. Mindkettőben a szoftver a specifikációja szerint működött, és a hiba a rendszerszintű feltevésekben volt. Az Ariane 5-nél a hiba egy újrafelhasznált modul feltevéseiben (az Ariane 4 pályája) és a nem védett konverzióban rejlett, a homogén redundancia pedig nem védett ellene. A 737 MAX-nál a követelmények, a változáskezelés, az egyetlen érzékelőre épülő kialakítás és a pilótareakciókra vonatkozó feltevések voltak hibásak.

Továbbolvasás

- Leveson, N. G. (2011): *Engineering a Safer World.* MIT Press – a szoftver szerepe a balesetekben és a követelmények elsődlegessége (Leveson 2011).
- Lions, J.-L. (1996): *ARIANE 5 Flight 501 Failure: Report by the Inquiry Board.* ESA (Lions 1996); Leveson, N. G., Turner, C. S. (1993): An investigation of the Therac-25 accidents. *IEEE Computer* (Leveson és Turner 1993).
- Butler, R. W., Finelli, G. B. (1993): The infeasibility of quantifying the reliability of life-critical real-time software. *IEEE TSE* (Butler és Finelli 1993).
- Hayhurst, K. J. et al. (2001): *A Practical Tutorial on Modified Condition/Decision Coverage.* NASA (Hayhurst és mtsai. 2001).
- IEC 61508-3, ISO 26262-6, DO-178C, EN 50716, IEC 62304 – az ágazati szoftverszabványok (IEC 2010a, 2006c; ISO 2018a; RTCA 2011a; CENELEC 2023).

16.

Az ember a biztonságkritikus rendszerben



A balesetvizsgálati jelentések visszatérő kifejezése az „emberi hiba". A 1. fejezetben azonban láttuk, hogy ez a megállapítás ritkán magyaráz bármit: a frontvonalban dolgozók tévedései többnyire a rendszer tervezéséből, a munkakörülményekből és a szervezet döntéseiből fakadnak, és ugyanezek az emberek a legtöbb nap éppen azzal tartják biztonságban a rendszert, hogy alkalmazkodnak a váratlan helyzetekhez. Ez a fejezet az embert a biztonságkritikus rendszer részeként vizsgálja: hogyan és miért tévedünk, mitől függ, hogy a kezelő érti-e a helyzetet, milyen csapdákat rejt az automatizálás, hogyan tervezhetők jó riasztási rendszerek, és hogyan számszerűsíthető az emberi teljesítmény a kockázatelemzésben.

Tanulási eredmények

A fejezet végére képes leszel

- az emberi tévedéseket Rasmussen viselkedési szintjei és Reason hibatípusai szerint osztályozni, és mindegyikhez megfelelő megelőző intézkedést rendelni;
- elemezni a helyzetismeret szintjeit, és felismerni a kijelzők, a riasztások és az automatizálás hatását a kezelő teljesítményére;
- megmagyarázni az automatizálás ironiáit és a járműautomatizálási szintek biztonsági következményeit;
- a riasztáskezelés fő elveit és teljesítménymutatóit alkalmazni;
- emberi hibavalószínűséget becsülni a HEART és a SPAR-H módszerrel, és értelmezni a függőségek hatását;
- megítélni, mikor számítható be a kezelői beavatkozás független védelmi rétegként.

16.1. Emberi hiba: tünet vagy ok?

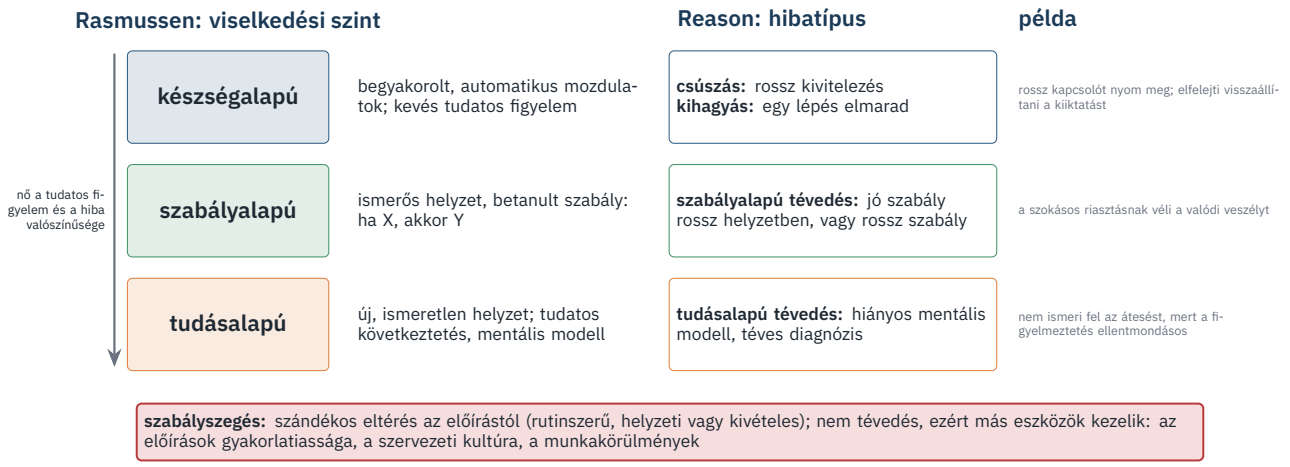
Az emberi hibának két szemlélete van. A **hagyományos szemlélet** szerint a rendszer alapvetően biztonságos, és az emberek megbízhatatlansága veszélyezteti; a megoldás a szigorúbb szabályozás, a fegyelmezés és az ember kiváltása automatizálással. Az **újabb szemlélet** szerint az emberi hiba a rendszer mélyebb problémáinak *tünete*: azt jelzi, hol nem illeszkednek egymáshoz a feladatok, az eszközök, a környezet és az emberi képességek (Reason 1997; Leveson 2011). Hollnagel Safety-II szemlélete ennél is tovább megy: a biztonság nem a hibák hiánya, hanem a rendszer képessége arra, hogy változó körülmények között is sikeresen működjön, és ennek legfontosabb forrása éppen az emberi alkalmazkodás (Hollnagel 2014).

A két szemlélet nem zárja ki egymást, de a gyakorlati következményük eltér. Ha egy kezelő rossz szelepet zár el, a hagyományos szemlélet a kezelőt oktatja ki, az újabb azt kérdezi meg, miért lehetett a két szelepet összetéveszteni (azonos kinézet, rossz címkézés, rossz megvilágítás, időnyomás), és a rendszert alakítja át úgy, hogy a tévedés nehezebb legyen, vagy következmények nélkül maradjon.

16.1.1. Viselkedési szintek és hibatípusok

Rasmussen három viselkedési szintet különböztetett meg (16.1. ábra) (Rasmussen 1983). A **készségalapú** viselkedés begyakorolt, automatikus mozdulatsorokból áll, kevés tudatos figyelemmel (egy rutinszerű kapcsolási művelet). A **szabályalapú** viselkedés ismerős helyzetben betanult szabályokat alkalmaz („ha a nyomás magas, nyisd a lefúvatást"). A **tudásalapú** viselkedés új, ismeretlen helyzetben tudatos következtetésre és a rendszer mentális modelljére épül.

Reason erre építve osztályozta a nem szándékos hibákat (Reason 1990). A készségalapú szinten **csúszások** (a szándék jó, a kivitelezés rossz: rossz gombot nyom meg) és **kihagyások** (egy lépés elmarad: elfelejti visszaállítani a kiiktatást) fordulnak elő. A szabály- és a tudásalapú szinten **tévedések**: jó szabály rossz helyzetben, rossz szabály, illetve hiányos mentális modellen alapuló téves diagnózis. A **szabályszegés** ezektől elkülönül: szándékos eltérés az előírástól, amely lehet rutinszerű (mindenki így csinálja), helyzeti (az előírás az adott helyzetben nem követhető) vagy kivételes.



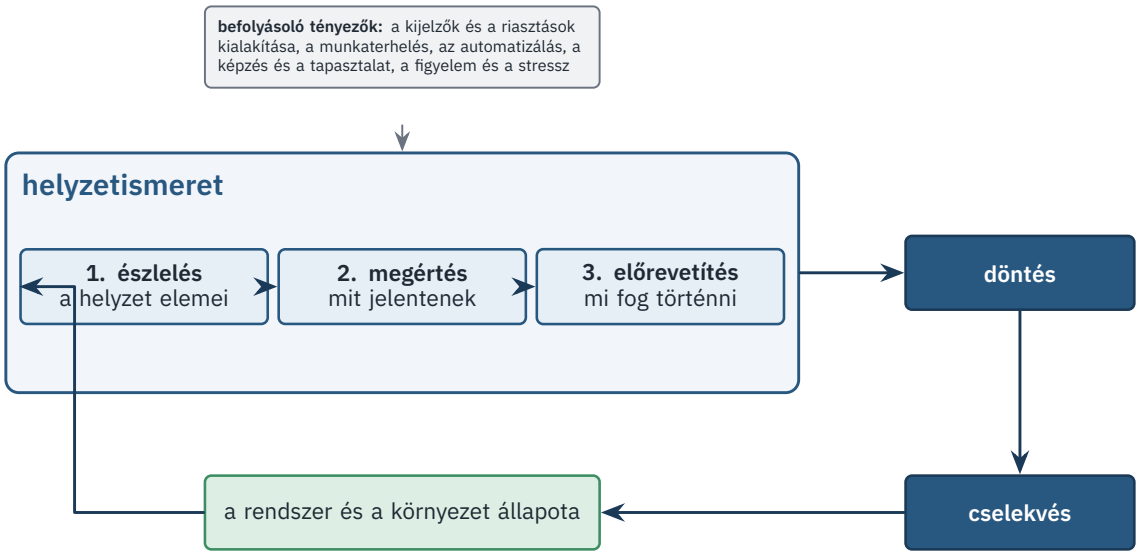
16.1. ábra. Rasmussen viselkedési szintjei és Reason hibatípusai. A szinttől függ, milyen intézkedés hatékony; a szabályszegés nem tévedés, ezért más eszközökkel kezelendő.

Az osztályozásnak gyakorlati haszna van, mert minden hibatípusra más intézkedés hat. A csúszások és kihagyások ellen a jó ergonómia, a megkülönböztethető kezelőszervek, az ellenőrző listák és a visszaigazolást kérő kialakítás véd; a képzés itt keveset segít, hiszen a kezelő tudja, mit kellene tennie. A szabályalapú tévedések ellen a helyzetek egyértelmű jelzése és a jó eljárások védenek. A tudásalapú tévedések ellen a rendszer működésének mély megértése, a szimulátoros képzés, a jó kijelzők és a döntéstámogatás. A szabályszegések ellen pedig a gyakorlatias, végrehajtható előírások, a munkaterhelés és az időnyomás csökkentése, és egy olyan szervezeti kultúra, amely a jelentést és a tanulást jutalmazza, nem a hibáztatást (Reason 1997).

16.2. Helyzetismeret és ember–gép felület

A kezelő akkor dönt jól, ha érti, mi történik. Endsley modellje szerint a **helyzetismeret** (*situation awareness*) három szintű (16.2. ábra) (Endsley 1995): a helyzet lényeges elemeinek **észlelése** (milyen

értékeket mutatnak a műszerek), azok **megértése** (mit jelentenek együtt, milyen állapotban van a rendszer), és a közeli jövő **előrevetítése** (mi fog történni, ha nem avatkozom be). A helyzetismeret elvesztése a vizsgált légi balesetek jelentős részében szerepet játszott, és a folyamatipari és a közúti balesetekben is gyakori.



16.2. ábra. Endsley helyzetismereti modellje. A döntés és a cselekvés minősége a helyzetismeret három szintjén múlik, amelyet a kijelzők, a riasztások, az automatizálás és a munkaterhelés erősen befolyásol.

Az ember–gép felület tervezésének célja a helyzetismeret támogatása. Néhány jól bevált elv: a kijelző a *feladathoz* és ne a rendszer belső szerkezetéhez igazodjon; az eltérést a normál állapottól mutassa, ne csak a nyers értéket; a trendeket is jelenítse meg, mert az előrevetítéshez a változás iránya kell; az üzemmódot (kézi, automatikus, kiiktatott) mindig egyértelműen mutassa; és a ritkán használt, de kritikus funkciók is legyenek könnyen elérhetők. A Therac-25 érthetetlen hibaüzenetei (15.2.2. szakasz) és a texasi finomító hibás szintkijelzése (1.2. szakasz) azt mutatják, hogy a rossz felület a balesetek egyik leggyakoribb látens feltétele.

16.3. Az automatizálás iróniái

Az automatizálás a rutinfeladatokat megbízhatóbban és fáradhatatlanabbul végzi, mint az ember. Bainbridge klasszikus tanulmánya azonban rámutatott az automatizálás **iróniáira** (Bainbridge 1983). A tervezők azt automatizálják, amit tudnak, és az embernek a nehezen automatizálható, váratlan helyzetek maradnak. A kezelő feladata a felügyelet lesz, amire az ember kifejezetten rosszul alkalmas: a ritkán változó kijelzők figyelése fárasztó és unalmas. Közben a kézi készségei és a rendszer működésének megértése elsorvad, mert nem gyakorolja őket. Amikor pedig az automatika a képességei határán kiszáll, éppen akkor kell átvennie az irányítást, amikor a helyzet a legnehezebb, és ő a legkevésbé felkészült.

Parasuraman, Sheridan és Wickens modellje szerint az automatizálás négy funkcióra terjedhet ki: az információ gyűjtésére, elemzésére, a döntés kiválasztására és a végrehajtásra, és mindegyiknél különböző fokú lehet (Parasuraman és mtsai. 2000). A biztonság szempontjából a legkritikusabb a döntés automatizálása, mert ott a legnagyobb a veszélye a **túlzott bizalomnak** (*complacency*) és az **automatizálási torzításnak**, amikor a kezelő akkor is elfogadja a rendszer javaslatát, ha a rendelkezésére álló más információ ellentmond neki. A másik gyakori probléma az **üzemmód-**

tévesztés: a kezelő azt hiszi, a rendszer egy adott üzemmódban van, miközben egy másikban működik.

> **Air France 447, 2009: amikor az automatika kiszáll**
>
> 2009. június 1-jén az Air France Rio de Janeiróból Párizsba tartó Airbus A330-asa éjszaka, viharfelhők között repült az Atlanti-óceán felett. A sebességmérő Pitot-csöveket jégkristályok tömték el, a mért sebességek ellentmondásossá váltak, és a robotpilóta kikapcsolt. A repülésvezérlés egy kevesebb védelmet nyújtó üzemmódba váltott. A gépet irányító pilóta a váratlan helyzetben tartósan felfelé húzta az orrot, a gép átesett, és a személyzet a hátralévő négy percben nem ismerte fel az átesést. A gép az óceánba csapódott, mind a 228 ember meghalt a fedélzeten (Bureau d'Enquêtes et d'Analyses 2012).
> A francia vizsgálóhatóság jelentése az emberi teljesítményt a rendszer egészének összefüggésében elemezte. A pilóták a kézi magaslégköri repülést és az átesés felismerését ritkán gyakorolták; a váratlan helyzet meglepetés- és ijedtségreakciót váltott ki; a feladatmegosztás a pilóták között felbomlott; és az átesésre figyelmeztető jelzés viselkedése maga is zavaró volt: nagyon kis mért sebességnél (az adatot érvénytelennek tekintve) elhallgatott, és amikor a pilóta lenyomta az orrot, és a sebesség nőni kezdett, újra megszólalt, mintha a helyes beavatkozás okozná a veszélyt (Bureau d'Enquêtes et d'Analyses 2012). Az eset Bainbridge iróniáinak tankönyvi példája: az automatika éppen a legnehezebb helyzetben adta vissza az irányítást, olyan személyzetnek, amelynek kézi készségei és helyzetismerete erre nem voltak felkészítve.

16.3.1. Járműautomatizálás

A közúti járművekben az automatizálás fokozatait az SAE J3016 hat szintje írja le (16.3. ábra) (SAE International 2021). A 0–2. szinten a vezető folyamatosan figyeli a környezetet, és ő a tartalék; a rendszer csak támogatja. A 3. szinttől a rendszer figyeli a környezetet a működési tartományán belül, a 3. szinten azonban a vezetőnek átvételi kérésre még ésszerű időn belül át kell vennie az irányítást. A 4. és az 5. szinten a rendszer maga a tartalék.

szint	ki kormányoz és gyorsít/fékez?	ki figyeli a környezetet?	ki a tartalék hiba esetén?	működési tartomány
0: nincs automatizálás	vezető	vezető	vezető	–
1: vezetéstámogatás	vezető és rendszer	vezető	vezető	korlátozott
2: részleges automatizálás	rendszer	vezető	vezető	korlátozott
3: feltételes automatizálás	rendszer	rendszer	vezető (átvételi kérésre)	korlátozott
4: magas fokú automatizálás	rendszer	rendszer	rendszer	korlátozott
5: teljes automatizálás	rendszer	rendszer	rendszer	korlátlan

itt kerül a környezet figyelése a vezetőtől a rendszerhez

16.3. ábra. Az SAE J3016 automatizálási szintjei. A 2. és a 3. szint között a környezet figyelésének felelőssége a vezetőről a rendszerre kerül; a biztonsági követelmények itt változnak meg alapvetően.

A legnehezebb a 2. és a 3. szint. A 2. szinten a rendszer annyira jól működik, hogy a vezető hajlamos elveszíteni a figyelmét, holott neki kell azonnal beavatkoznia, ha a rendszer hibázik. A 1. fejezetben tárgyalt 2018-as tempei baleset ennek drámai példája: a fejlesztési célú automatizált jármű biztonsági vezetője a vizsgálat szerint az út helyett a mobiltelefonját nézte, a vállalat pedig nem gondoskodott megfelelő felügyeletről és figyelemellenőrzésről (National Transportation Safety Board 2019). A

3. szinten az átvételi idő a kérdés: egy hosszú ideje nem figyelő vezetőnek másodpercek, sőt több másodperc kell, mire újra kialakítja a helyzetismeretét. Ezért írják elő a szabályozások a vezető figyelmének folyamatos ellenőrzését, és azt, hogy a rendszer az átvétel elmaradása esetén maga hajtson végre minimális kockázatú manővert (UNECE 2021b).

VészFék: az ember mint tartalék és mint zavarforrás

A vészfékező asszisztens 1. vagy 2. szintű funkció: a vezető felelőssége, hogy figyeljen, a rendszer csak akkor avatkozik be, ha a vezető nem reagál időben. Az emberi tényezők itt két irányból hatnak. Egyrészt a figyelmeztetésnek olyannak kell lennie, hogy a vezető gyorsan és helyesen reagáljon (hangjelzés, vizuális jelzés, rövid fékrándítás), és ne ijedjen meg annyira, hogy rosszul cselekedjen. Másrészt a túl sok téves figyelmeztetés a vezető bizalmát rontja: kikapcsolja a rendszert, vagy figyelmen kívül hagyja a jelzéseit. A téves fékezés és a téves figyelmeztetés ezért nemcsak műszaki, hanem emberi tényezői kockázat is, amelyet a rendeltetésszerű funkció biztonságának elemzése (18. fejezet) is figyelembe vesz.

16.4. Riasztáskezelés

A riasztás a kezelő figyelmét egy beavatkozást igénylő helyzetre hívja fel. A modern irányítórendszerekben egy riasztás beállítása szinte semmibe sem kerül, ezért a rendszerek gyakran túl sok riasztással rendelkeznek. Zavar idején percenként tucatnyi riasztás szólalhat meg, és a kezelő nem tudja kiválasztani a lényegeset. A **riasztásáradat** számos súlyos ipari baleset hátterében szerepelt, és a 11.3. szakasz csokornyakkendő-diagramján a kezelői gát eszkalációs tényezője volt.

A riasztáskezelés szakmai irányelveit az EEMUA 191 útmutató és az ANSI/ISA-18.2 szabvány foglalja össze (EEMUA 2013; ISA 2016). Fő elveik: minden riasztásnak egyértelmű beavatkozást kell igényelnie, és a kezelőnek legyen ideje erre; a riasztásokat racionalizálni kell (felesleges, ismétlődő, állandóan aktív riasztások megszüntetése); a prioritásokat a következmény súlyossága és a rendelkezésre álló idő alapján kell kiosztani; a riasztásokat üzem közben folyamatosan mérni és javítani kell. A szokásos teljesítménymutatók: egy kezelőre jutó átlagos riasztásszám (az ajánlás tízpercenként nagyjából egy), a riasztásáradatok aránya (áradatnak a tíz percen belüli tíznél több riasztás számít), a leggyakoribb riasztások részaránya, az állandóan aktív és a tartósan elnémított riasztások száma, és a prioritások megoszlása (a riasztások nagy része alacsony, kis része magas prioritású legyen) (ISA 2016).

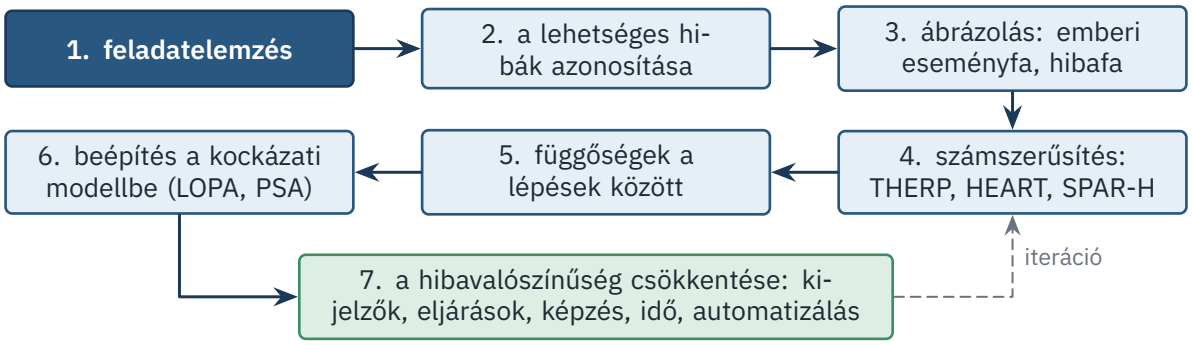
A biztonsági szempontból kritikus riasztások külön figyelmet érdemelnek: ha egy riasztást és a hozzá tartozó kezelői beavatkozást a LOPA független védelmi rétegként számít be (11.2.1. szakasz), akkor a riasztásnak a többi riasztástól megkülönböztethetőnek, a kezelőnek pedig az előírt idő alatt képesnek kell lennie felismerni és végrehajtani a teendőt.

16.5. Az emberi megbízhatóság elemzése

Ha az emberi beavatkozás egy védelmi réteg vagy egy hibafa alapeseménye, a kockázatelemzéshez számszerű **emberi hibavalószínűség** (*human error probability*, HEP) kell. Ezt az emberi megbízhatóság elemzése (*human reliability analysis*, HRA) adja (16.4. ábra) (Kirwan 1994). A folyamat a feladat elemzésével kezdődik (milyen lépésekből áll, milyen információra és időre van szükség), majd azonosítja a lehetséges hibákat, emberi eseményfában vagy hibafában ábrázolja őket, számszerűsíti a valószínűségüket, figyelembe veszi a lépések közötti függőségeket, és az eredményt beépíti a kockázati modellbe. A legértékesebb kimenet sokszor nem a szám, hanem annak felismerése, mi csökkentené a hibavalószínűséget.

A három legelterjedtebb számszerűsítő módszer:

- **THERP** (*technique for human error rate prediction*): a nukleáris ipar számára kidolgozott, részletes módszer, amely a feladatot elemi lépésekre bontja, mindegyikhez táblázatos alapvalószínűséget



16.4. ábra. Az emberi megbízhatóság elemzésének folyamata. A számszerűsítés csak egy lépés; a feladatelemzés és a hibavalószínűség csökkentésére tett javaslatok legalább ilyen fontosak.

rendel, és ezeket a teljesítményt befolyásoló tényezőkkel módosítja (Swain és Guttmann 1983). A lépések közötti **függőséget** öt szinttel kezeli: ha a független valószínűség N, a feltételes valószínűség kis függőségnél $(1 + 19N)/20$, közepesnél $(1 + 6N)/7$, nagynál $(1 + N)/2$, teljes függőségnél 1. Ez fontos: ha ugyanaz a kezelő ellenőrzi a saját munkáját, a második lépés hibavalószínűsége közel sem független az elsőétől.

- **HEART** (*human error assessment and reduction technique*): egyszerűbb, gyorsan alkalmazható módszer. A feladatot kilenc általános feladattípus egyikébe sorolja, mindegyikhez névleges hibavalószínűség tartozik, majd a hibát előidéző körülmények (*error producing conditions*, EPC) szorzóival módosítja, a hatásuk becsült arányával súlyozva (Williams 1988).
- **SPAR-H**: az amerikai nukleáris hatóság egyszerűsített módszere, amely a feladatot diagnózisra (névleges HEP = 0,01) és végrehajtásra (névleges HEP = 0,001) bontja, és nyolc teljesítményt befolyásoló tényező (rendelkezésre álló idő, stressz, bonyolultság, tapasztalat és képzés, eljárások, ergonómia és ember–gép felület, munkaképesség, munkafolyamatok) szorzóival módosítja (Gertman és mtsai. 2005).

Emberi hibavalószínűség becslése a HEART módszerrel

Egy kezelőnek egy rutinszerű, jól begyakorolt, gyors és viszonylag kis szakértelmet igénylő műveletet kell végrehajtania (egy kézi szelep elzárása egy ismert helyen). A HEART szerint ez az E feladattípus, névleges hibavalószínűsége 0,02 (Williams 1988). Két hibát előidéző körülmény áll fenn: kevés idő van a hiba észlelésére és kijavítására (szorzója 11), és a kezelő tapasztalatlan (szorzója 3). Az elemző a két körülmény hatásának arányát 0,4-re, illetve 0,5-re becsüli. A súlyozott szorzók: $(11 - 1) \cdot 0{,}4 + 1 = 5$ és $(3 - 1) \cdot 0{,}5 + 1 = 2$, így

$$\text{HEP} = 0{,}02 \cdot 5 \cdot 2 = 0{,}2.$$

Ez tízszerese a névleges értéknek. A számítás egyben azt is megmutatja, hol érdemes beavatkozni: ha a kezelőnek több idő jut (például mert egy automatikus előriasztás korábban jelez), az első szorzó 1-re csökken, és a HEP 0,04 lesz; ha tapasztalt kezelő végzi a műveletet, további felére.

A HRA-módszerek eredményei bizonytalanok: ugyanarra a feladatra különböző elemzők és módszerek akár egy nagyságrenddel eltérő értéket is kaphatnak (Kirwan 1994). A HEP-értékeket ezért nagyságrendi becslésként, a feltevések gondos dokumentálásával és érzékenységvizsgálattal érdemes használni, és a módszer ereje inkább a hibát előidéző körülmények szisztematikus azonosításában rejlik.

ReaktorVéd: beszámítható-e a kezelő?

A LOPA a TAH-103 riasztást és a kezelői beavatkozást PFD $= 0{,}1$ értékű független védelmi rétegként számította be (11.2. táblázat). Ezt a SPAR-H módszerrel ellenőrizzük. A *diagnózis* (a riasztás felismerése és a helyes teendő kiválasztása) névleges HEP-je 0,01. A rendelkezésre álló idő (legalább 20 perc a felfutásig) a szükségeshez képest elegendő, szorzója 1. A stressz magas (egy megfutó reakció veszélye), szorzója 2. A diagnózis közepesen bonyolult (a riasztás oka lehet a hűtővízszivattyú, a szabályozószelep vagy egy mérési hiba), szorzója 2. A többi tényező névleges. Így a diagnózis HEP-je $0{,}01 \cdot 2 \cdot 2 = 0{,}04$. A *végrehajtás* (az adagolás leállítása és a tartalék szivattyú indítása a kezelőpultról) névleges HEP-je 0,001, magas stresszel 0,002. Az együttes HEP ≈ 0,042, ami alátámasztja a 0,1-es beszámítást, és mutatja, hogy a tartalék nem túl nagy.

A feltételek azonban kritikusak: a riasztásnak magas prioritásúnak és a riasztásáradatban is megkülönböztethetőnek kell lennie; a teendőt írásos eljárás rögzíti, amelyet a kezelők szimulátoron gyakorolnak; és éjszakai műszakban is van képzett kezelő a pultnál. Ha a riasztási rendszer túlterhelt, vagy az eljárás hiányos, a SPAR-H szorzói (ergonómia, eljárások) tízszeresre vagy ötvenszeresre is nőhetnek (Gertman és mtsai. 2005), és a réteg nem számítható be.

KoboCella: együttműködés, bizalom és megkerülés

A kollaboratív robotcellában az ember és a gép szoros együttműködése új emberi tényezői kérdéseket vet fel. A szerelőnek meg kell bíznia a robotban ahhoz, hogy hatékonyan dolgozzon mellette, de nem szabad túlzottan megbíznia benne. A robot mozgásának kiszámíthatónak kell lennie (azonos pályák, jelzőfények a mozgás irányáról és a sebességről), mert a meglepő mozdulatok ijedtséget és hirtelen, veszélyes reakciókat váltanak ki. A biztonsági funkciók túl gyakori beavatkozása (12.6. szakasz) arra ösztönözheti a dolgozókat, hogy megkerüljék a védelmet. Az ISO 12100 ezért a kockázatbecslés részeként kifejezetten vizsgáltatja a védőintézkedések megkerülésének lehetőségét és ösztönzőit, és az előrelátható helytelen használatot is (ISO 2010). A tervezésnek ezért a szerelők bevonásával, a valós munkafolyamatok megfigyelésével kell történnie.

Összefoglalás

- Az emberi hiba többnyire a rendszer mélyebb problémáinak tünete; a Safety-II szemlélet szerint az emberi alkalmazkodás a biztonság forrása is.
- Rasmussen készség-, szabály- és tudásalapú szintjei és Reason hibatípusai (csúszás, kihagyás, tévedés, szabályszegés) más-más intézkedést kívánnak.
- A helyzetismeret (észlelés, megértés, előrevetítés) a jó döntés feltétele; a kijelzők, a riasztások és az automatizálás erősen befolyásolja.
- Az automatizálás iróniái: az embernek a legnehezebb helyzetek maradnak, készségei elsorvadnak, és a legrosszabb pillanatban kell átvennie az irányítást.
- Az SAE J3016 szerint a 2. és a 3. szint között kerül a környezet figyelése a vezetőtől a rendszerhez; a vezető figyelmének ellenőrzése és a minimális kockázatú manőver ezért alapkövetelmény.
- A riasztáskezelés elvei: minden riasztás beavatkozást igényel, racionalizálás, következmény- és időalapú prioritás, folyamatos mérés.
- Az emberi hibavalószínűség THERP-, HEART- vagy SPAR-H-módszerrel becsülhető; az eredmény nagyságrendi, és a lépések közötti függőség döntő lehet.

Önellenőrzés

1. Egy karbantartó egy szivattyú cseréje után elfelejti kinyitni a nyomóoldali szelepet. Melyik hibatípus ez? Milyen intézkedés hatékony ellene?
2. Miért nem elég egy tudásalapú tévedés után a kezelőt újra kiképezni?
3. Egy feladat diagnózisát SPAR-H szerint értékeljük: a rendelkezésre álló idő épphogy elegendő (szorzó 10), a stressz magas (2), a többi tényező névleges. Mekkora a diagnózis hibavalószínűsége?

4. Egy kezelő elvégez egy kritikus lépést ($N = 0{,}01$), majd ugyanő ellenőrzi. Mekkora annak a valószínűsége, hogy az ellenőrzés is elmarad, ha a THERP szerint nagy a függőség? Mennyi lenne függetlenséggel?
5. Miért veszélyesebb lehet egy 2. szintű vezetéstámogató rendszer, mint egy 1. szintű, pedig többet segít?

Megoldás.

1. Kihagyás (készségalapú szint). Hatékony ellene az ellenőrző lista, a munkaengedélyben rögzített visszaállítási lépés, a független ellenőrzés, és olyan műszaki megoldás, amely a szivattyú indítását megakadályozza, ha a szelep zárva van (retesz).
2. Mert a tudásalapú tévedés oka gyakran a rendszer viselkedésének nehezen érthetősége, a hiányos vagy félrevezető információ, vagy egy olyan helyzet, amelyet a képzés nem fedett le. A képzés mellett a kijelzőket, a döntéstámogatást és a rendszer átláthatóságát is javítani kell.
3. $0{,}01 \cdot 10 \cdot 2 = 0{,}2$.
4. Nagy függőségnél $(1 + N)/2 = (1 + 0{,}01)/2 \approx 0{,}505$, tehát az ellenőrzés az esetek felében nem fogja meg a hibát; az együttes valószínűség $0{,}01 \cdot 0{,}505 \approx 5 \cdot 10^{-3}$. Függetlenséggel $0{,}01^2 = 10^{-4}$. A független ellenőrzés (más személy, más időpont) ezért sokkal értékesebb.
5. Mert a 2. szintű rendszer egyszerre kormányoz és gyorsít, így a vezető aktív szerepe megszűnik, de felelőssége megmarad: folyamatosan figyelnie kell, és azonnal be kell avatkoznia. A jól működő rendszer túlzott bizalmat és figyelemvesztést okoz, ezért a ritka hibákra a vezető lassabban vagy egyáltalán nem reagál.

Továbbolvasás

- Reason, J. (1990): *Human Error.* Cambridge University Press (Reason 1990); Reason, J. (1997): *Managing the Risks of Organizational Accidents.* Ashgate (Reason 1997).
- Endsley, M. R. (1995): Toward a theory of situation awareness in dynamic systems. *Human Factors* (Endsley 1995).
- Bainbridge, L. (1983): Ironies of automation. *Automatica* (Bainbridge 1983); Parasuraman, R., Sheridan, T. B., Wickens, C. D. (2000): A model for types and levels of human interaction with automation (Parasuraman és mtsai. 2000).
- Kirwan, B. (1994): *A Guide to Practical Human Reliability Assessment.* Taylor & Francis (Kirwan 1994); Gertman, D. et al. (2005): *The SPAR-H Human Reliability Analysis Method* (NUREG/CR-6883) (Gertman és mtsai. 2005).
- BEA (2012): *Final Report on the Accident on 1st June 2009 to the Airbus A330-203, Flight AF 447* (Bureau d'Enquêtes et d'Analyses 2012).
- EEMUA 191 és ANSI/ISA-18.2 – a riasztáskezelés irányelvei (EEMUA 2013; ISA 2016).

V. rész – Bizonyítás és új kihívások



Pannon Egyetem
University of Pannonia

17.

Biztonsági érvelés és igazolás



Az előző fejezetek sokféle elemzést, számítást és tervezési döntést mutattak be. Egy biztonságkritikus rendszer üzembe helyezésekor azonban nem elég, hogy ezek egyenként elkészültek: valakinek (a hatóságnak, egy független értékelőnek, a vállalat vezetésének, végső soron a társadalomnak) meg kell győződnie arról, hogy *együtt* elegendők a rendszer biztonságának alátámasztására. Erre szolgál a **biztonsági érvelés** (*safety case*): egy strukturált érvelés, amely bizonyítékokra támaszkodva meggyőzően és ellenőrizhetően mutatja be, hogy a rendszer egy adott alkalmazásban és környezetben elfogadhatóan biztonságos. Ebben a fejezetben megismerjük a biztonsági érvelés eredetét és elemeit, a célstruktúra-jelölést (GSN), a ReaktorVéd biztonsági érvelésének vázát, a független értékelés formáit, a biztonsági érvelések tipikus gyengeségeit, végül az üzemeltetés során folyamatosan karbantartott, „élő” biztonsági érvelés gondolatát.

Tanulási eredmények

A fejezet végére képes leszel

- megmagyarázni a biztonsági érvelés célját, elemeit és szerepét a különböző ágazatok szabályozásában;
- GSN-jelöléssel felépíteni egy biztonsági érvelés vázát, és azonosítani benne az állításokat, a stratégiákat, a bizonyítékokat, a kontextust és a feltevéseket;
- az eddigi elemzéseket (veszélyelemzés, LOPA, hibafa, SIL-igazolás, szoftverigazolás) egy koherens érvelésbe rendezni;
- felismerni a biztonsági érvelések tipikus gyengeségeit, és a független értékelés szerepét;
- megtervezni, hogyan marad érvényes egy biztonsági érvelés az üzemeltetés és a változások során.

17.1. A biztonsági érvelés eredete és célja

A biztonsági érvelés mint szabályozási eszköz súlyos balesetek tanulságaiból született. Az Északi-tengeren 1988. július 6-án a Piper Alpha olajfúró platformon bekövetkezett robbanás és tűz 167 ember életét követelte. A Cullen-vizsgálat megállapította, hogy az előíró jellegű szabályozás (amely részletes műszaki előírások betartását ellenőrzi) nem volt képes kezelni egy bonyolult létesítmény kockázatait, és azt javasolta, hogy az üzemeltetőnek kelljen bizonyítania: azonosította a súlyos baleseti

veszélyeket, és a kockázatokat a lehető legalacsonyabbra csökkentette (Cullen 1990). Ez a **célkitűzés-alapú szabályozás** (*goal-based regulation*) a brit offshore iparban 1992-ben lépett életbe, és azóta sok ágazatban elterjedt.

A biztonsági érvelés ma számos szabályozásban és szabványban kötelező vagy ajánlott munkatermék: az Egyesült Királyságban az offshore és a nukleáris iparban, a vasúti biztonsági jelzőrendszereknél az EN 50129 szerint (CENELEC 2018), a közúti járművekben az ISO 26262 szerint (ISO 2018a), az autonóm termékekre pedig az UL 4600 egész szabványa a biztonsági érvelésre épül (UL Standards & Engagement 2023). A folyamatiparban az IEC 61511 nem használja ezt a kifejezést, de a funkcionális biztonsági értékelés és a dokumentációs követelmények lényegében ugyanezt szolgálják (IEC 2016b).

A biztonsági érvelés három dolgot kapcsol össze: az **állításokat** (mit állítunk a rendszer biztonságáról), a **bizonyítékokat** (elemzések, tesztek, tanúsítványok, üzemi tapasztalat) és az **érvelést**, amely megmutatja, *miért* támasztják alá a bizonyítékok az állításokat (Kelly 1998). A harmadik elem a legfontosabb és a leggyakrabban hiányzó: egy halom dokumentum önmagában nem érv. Ahhoz, hogy egy értékelő meggyőződhessen a rendszer biztonságáról, látnia kell, hogyan függenek össze az egyes bizonyítékok, mely feltevésekre épülnek, és mi hiányzik.

17.2. A célstruktúra-jelölés (GSN)

A biztonsági érvelések leírására a legelterjedtebb grafikus nyelv a **célstruktúra-jelölés** (*Goal Structuring Notation*, GSN), amelyet Tim Kelly dolgozott ki a York-i Egyetemen (Kelly 1998; Kelly és Weaver 2004), és amelynek ma a 3. változatú közösségi szabványa érvényes (Assurance Case Working Group 2021). A GSN elemei (17.1. ábra):

- **cél (állítás):** egy állítás a rendszerről, amelyet igazolni kell („a funkció PFD-je legfeljebb $2 \cdot 10^{-3}$”);
- **stratégia:** az érvelés módja, ahogyan egy állítást részállításokra bontunk („érvelés a SIL három pillére szerint”);
- **megoldás (bizonyíték):** egy elemzés, teszt, tanúsítvány vagy üzemi adat, amely egy állítást közvetlenül alátámaszt;
- **kontextus:** a hatókör, a definíciók, a működési környezet, amelyre az állítás vonatkozik;
- **feltevés és indoklás:** elfogadott, de nem igazolt állítás, illetve egy stratégia vagy döntés indoka;
- **kifejtetlen állítás:** amelynek alátámasztása még hiányzik (a fejlesztés közben hasznos jelölés).



17.1. ábra. A GSN fő elemei és kapcsolatai. Az „alátámasztja” kapcsolat az állításokat, a stratégiákat és a bizonyítékokat köti össze, az „összefüggésben van” kapcsolat a kontextust és a feltevéseket.

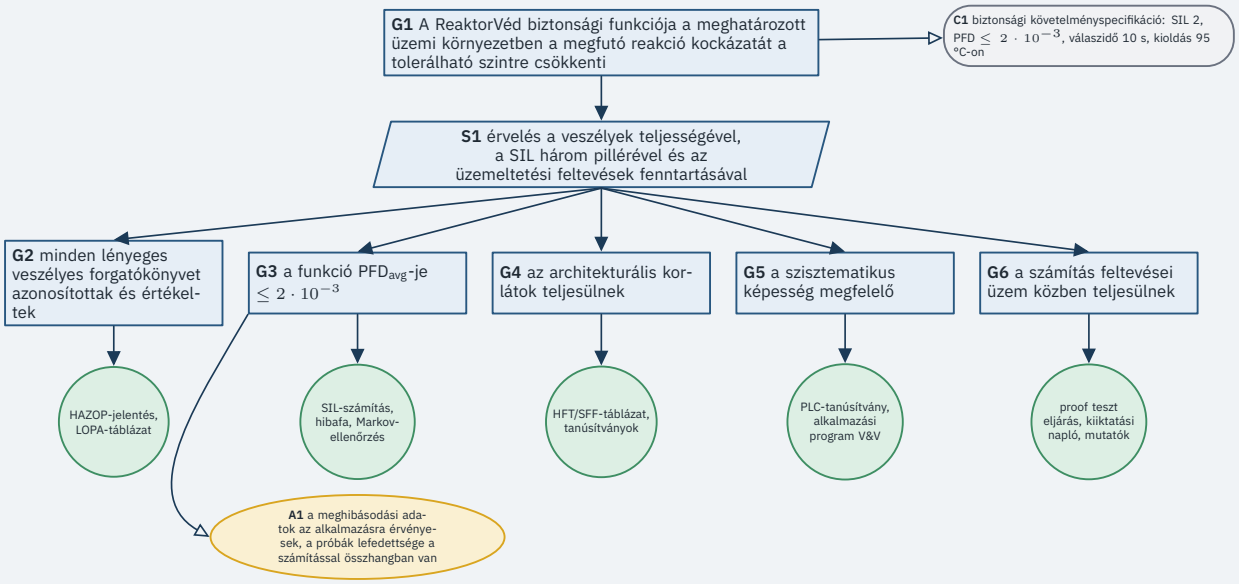
A GSN-érvelés fentről lefelé épül: a legfelső állítás (rendszerint „a rendszer elfogadhatóan biztonságos az adott környezetben”) stratégiákon keresztül részállításokra bomlik, és minden ág egy vagy több bizonyítékkal zárul. A jól felépített érvelésben minden állítás egyértelmű, ellenőrizhető mondat, minden stratégia megmondja, miért elegendő a részállítások összessége, és a kontextus és a feltevések pontosan rögzítik az érvelés érvényességi körét. A GSN szabvány ismétlődő érvelési mintákat

(*patterns*) és moduláris érveléseket is támogat, amelyek a nagy, több beszállítós rendszereknél nélkülözhetetlenek (Assurance Case Working Group 2021).

ReaktorVéd: a biztonsági érvelés váza

A ReaktorVéd biztonsági funkciójának érvelése (17.2. ábra) a G1 állításból indul: a funkció a meghatározott üzemi környezetben a megfutó reakció kockázatát a tolerálható szintre csökkenti. A kontextus (C1) a biztonsági követelményspecifikáció. Az S1 stratégia öt részállításra bontja az érvelést:

- **G2:** minden lényeges veszélyes forgatókönyvet azonosítottak és értékeltek – bizonyíték a HAZOP-jelentés és a LOPA-táblázat (5. és 11. fejezet);
- **G3:** a funkció $\mathrm{PFD}_{\mathrm{avg}}$-je legfeljebb $2 \cdot 10^{-3}$ – bizonyíték a SIL-számítás, a hibafa és a Markov-modell összevetése (10–13. fejezet); a számítás az A1 feltevésre épül (a meghibásodási adatok érvényesek, a próbák lefedettsége a számítással összhangban van);
- **G4:** az architekturális korlátok teljesülnek – bizonyíték a HFT/SFF-táblázat és a tanúsítványok;
- **G5:** a szisztematikus képesség megfelelő – bizonyíték a PLC tanúsítványa és az alkalmazási program verifikációja és validálása (15. fejezet);
- **G6:** a számítás feltevései üzem közben teljesülnek – bizonyíték a proof teszt eljárása, a kiiktatások naplója és a teljesítménymutatók.



17.2. ábra. A ReaktorVéd biztonsági funkciója biztonsági érvelésének felső szintje GSN-jelöléssel. A G6 ág teszi az érvelést „élővé”: a számítás feltevéseit az üzemeltetés során is igazolni kell.

Az érvelés felépítése közben két hiányosság derült ki, ami a módszer egyik fő hasznát mutatja. Az A1 feltevés első változata egyszerűen „a meghibásodási adatok érvényesek” volt; a 13.5. szakasz elemzése után a csapat felismerte, hogy a próbák lefedettsége a feltevés kritikus része, és ezt explicitté tette. A G6 állítás eredetileg hiányzott; a csapat úgy gondolta, hogy az üzemeltetés „már nem a projekt része”. A biztonsági érvelés azonban arról szól, hogy a rendszer az *életciklusa során* biztonságos, és ez nem igazolható az üzemeltetési feltételek nélkül.

17.3. Független értékelés

A biztonsági érvelés értéke azon múlik, mennyire alapos és pártatlan a felülvizsgálata. A szabványok ezért az integritási szinttől függő **függetlenséget** írnak elő a verifikációban és az értékelésben. Az IEC 61511 a **funkcionális biztonsági értékelést** (*functional safety assessment*, FSA) öt ponton írja elő vagy

ajánlja: a veszély- és kockázatelemzés és a követelményspecifikáció után, a tervezés után, a telepítés, az üzembe helyezés és a validálás után (a veszélyes anyagok bevezetése előtt), az üzemeltetési tapasztalatok megszerzése után, és a módosítások után (IEC 2016b). Az ISO 26262 megerősítő intézkedéseket (*confirmation measures*) ír elő: a munkatermékek megerősítő felülvizsgálatát, a funkcionális biztonsági auditot és a funkcionális biztonsági értékelést, az ASIL-tól függő függetlenségi szinttel (ISO 2018a).

A független értékelő nem a hibák vadászata miatt fontos, hanem mert a fejlesztőcsapat óhatatlanul a saját feltevéseinek foglya. Egy kívülálló könnyebben észreveszi a hiányzó veszélyeket, az indokolatlan feltevéseket és az érvelés gyenge pontjait.

17.4. A biztonsági érvelések gyengeségei

A biztonsági érvelés hatékony eszköz, de visszaélhető vagy kiüresíthető. A legtanulságosabb példa a brit királyi légierő Nimrod XV230 felderítő repülőgépének 2006-os afganisztáni katasztrófája, amelyben tizennégy katona halt meg. A gép biztonsági érvelését néhány évvel korábban készítették el, de a független vizsgálat szerint az érvelés súlyos hibákat tartalmazott, és egy olyan veszélyt, amely a balesethez vezetett, nem értékelt megfelelően. Haddon-Cave jelentése szerint a biztonsági érvelés papírmunkává, a megfelelés kipipálásává vált, amely a dokumentáció mennyiségét tévesztette össze a biztonság igazolásával, és a szervezetben mindenki azt feltételezte, hogy valaki más gondoskodik a valódi kockázatokról (Haddon-Cave 2009).

A tipikus gyengeségek:

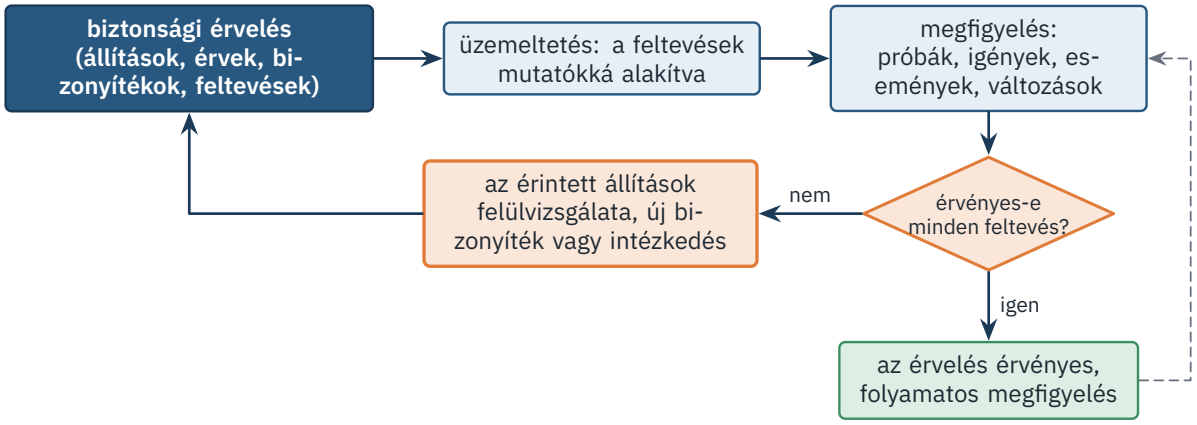
- **megerősítési torzítás:** az érvelés természeténél fogva arra ösztönöz, hogy a biztonságot alátámasztó bizonyítékokat keressük, ne a cáfolókat; a jó érvelés ezért kifejezetten foglalkozik az ellenérvekkel és a kétségekkel;
- **a bizonyíték és az állítás közti rés:** a bizonyíték csak egy szűkebb állítást támaszt alá, mint amire használják (például egy teszt csak normál körülményeket fedett le);
- **rejtett feltevések:** az érvelés egy olyan feltevésen múlik, amelyet sehol nem mondtak ki (például hogy a próbák minden hibát felfednek);
- **elavulás:** a rendszer, a környezet vagy a használat megváltozott, az érvelés nem;
- **a mennyiség mint minőség:** több ezer oldal dokumentáció, amelyből senki sem látja, mi az érv lényege.

A GSN 3. változata ezért lehetővé teszi, hogy az érvelés kapcsolataihoz bizonyossági pontokat (*assurance claim points*) rendeljünk, ahol külön érvelés mutatja be, mennyire bízhatunk az adott lépésben (Assurance Case Working Group 2021). Az UL 4600 pedig minden fejezetéhez jellegzetes hibalistákat ad, amelyeket az értékelőnek ellenőriznie kell (UL Standards & Engagement 2023).

17.5. Az élő biztonsági érvelés

A biztonsági érvelést hagyományosan az üzembe helyezés előtt készítik el, és utána legfeljebb a nagyobb módosítások idején veszik elő. Ez a gyakorlat két okból elégtelen. Egyrészt az érvelés feltevései (meghibásodási ráták, próbák lefedettsége, a kezelők reakcióideje, az igények gyakorisága) üzem közben változnak, és csak a működő rendszer adataiból ellenőrizhetők. Másrészt a tanuló, adaptív rendszerek (például a gépi tanulási komponensek) viselkedése az üzemeltetés során is változhat (18. fejezet). Denney, Pai és Habli ezért **dinamikus biztonsági érvelést** javasolnak, amelynek érvényességét a rendszer működése során folyamatosan ellenőrzik, és amely a megfigyelt adatok alapján frissül (Denney és mtsai. 2015). Az UL 4600 hasonló elven **biztonsági teljesítménymutatókat** (*safety performance indicators*) követel meg, amelyek az érvelés feltevéseinek érvényességét mérik az üzemeltetés során (UL Standards & Engagement 2023).

Az élő érvelés ciklusa (17.3. ábra) a feltevéseket mérhető mutatókká alakítja; a próbák, az igények, az események és a változások adatait folyamatosan összeveti velük; és ha egy feltevés sérül, az érintett állításokat felülvizsgálja, új bizonyítékot gyűjt vagy intézkedést hoz. Ez a gondolat a 11.3. szakaszban látott gátállapot-követést általánosítja, és összekapcsolja a biztonsági érvelést a változáskezeléssel (4. fejezet).



17.3. ábra. Az élő biztonsági érvelés ciklusa. A feltevések mutatókká alakulnak, és ha egy mutató a feltevés sérülését jelzi, az érintett állításokat felül kell vizsgálni.

ReaktorVéd: a feltevések mutatói

A G6 állítás alatti mutatók: a gyorszáró szelepek proof tesztjeinek és részleges löketpróbáinak teljesítése az ütemterv szerint; a próbák során talált hibák száma és jellege (a meghibásodási ráta feltevésének ellenőrzésére); a valódi igények gyakorisága (a LOPA kezdő esemény gyakoriságának ellenőrzésére); a kiiktatások száma és időtartama; a biztonsági funkció téves leállásainak gyakorisága; a TAH-103 riasztás megszólalása és a kezelői reakcióidő; a riasztási rendszer teljesítménymutatói. Ha például a próbákon a vártnál több szelephiba derül ki, vagy a kiiktatások összideje meghaladja a feltételezett értéket, az érvelést felül kell vizsgálni, és szükség esetén módosítani a karbantartási stratégiát vagy a tervet. A biztonsági érvelés így a karbantartási és üzemeltetési döntések mindennapi eszközévé válik.

VészFék: érvelés hiba nélküli veszélyekre is

A vészfékező asszisztens biztonsági érvelése két nagy ágból áll. Az egyik a funkcionális biztonságé (ISO 26262): a hardver- és a szoftverhibák elleni védelem, az ASIL-követelmények teljesülése. A másik a rendeltetésszerű funkció biztonságáé (ISO 21448): annak igazolása, hogy a teljesítőképesség korlátai (például az érzékelés ködben vagy a szokatlan alakú akadályok felismerése) nem okoznak elfogadhatatlan kockázatot, és a fennmaradó ismeretlen veszélyes helyzetek valószínűsége kellően kicsi (18. fejezet). A második ág bizonyítékai nagyrészt statisztikai és szimulációs jellegűek, és az érvelés üzem közbeni frissítése (a terepen gyűjtött adatok alapján) itt különösen fontos.

KoboCella: a műszaki dokumentáció mint érvelés

A gépekre vonatkozó (EU) 2023/1230 rendelet szerint a gyártónak műszaki dokumentációt kell összeállítania, amely tartalmazza a kockázatértékelést, az alkalmazott szabványokat, a biztonsági funkciók igazolását és a vizsgálati eredményeket, és ennek alapján állítja ki az EU-megfelelőségi nyilatkozatot (Európai Parlament és Tanács 2023). Bár a rendelet nem használja a biztonsági érvelés kifejezést, egy jól felépített műszaki dokumentáció valójában az: állítások (a cella megfelel az alapvető egészségvédelmi és biztonsági követelményeknek), bizonyítékok (PL-számítások, mérések, ütközési erőmérések) és érvelés (miért elegendők ezek). A GSN szerkezet itt is segít abban, hogy a dokumentáció ne csak egy iratgyűjtő, hanem egy ellenőrizhető érvelés legyen.

Összefoglalás

- A biztonsági érvelés strukturált érvelés, amely bizonyítékokra támaszkodva mutatja be, hogy a rendszer egy adott alkalmazásban és környezetben elfogadhatóan biztonságos; a célkitűzés-alapú szabályozás eszköze, a Piper Alpha-katasztrófa tanulságaiból született.
- Elemei az állítások, az érvelés (stratégiák), a bizonyítékok, a kontextus és a feltevések; az érvelés a leggyakrabban hiányzó elem.
- A GSN szabványos grafikus jelölés, amely fentről lefelé bontja az állításokat, és mintákat és modularitást is támogat.
- A ReaktorVéd érvelése a veszélyek teljességére, a SIL három pillérére és az üzemeltetési feltevések fenntartására épül; felépítése során rejtett feltevések kerültek felszínre.
- A független értékelés az integritási szinttől függő függetlenséggel szükséges.
- A tipikus gyengeségek a megerősítési torzítás, a bizonyíték és az állítás közti rés, a rejtett feltevések, az elavulás és a dokumentációmennyiség; a Nimrod-katasztrófa mutatta meg a kiüresedett érvelés veszélyét.
- Az élő biztonsági érvelés a feltevéseket mutatókká alakítja, és az üzemeltetés során folyamatosan ellenőrzi.

Önellenőrzés

1. Miben különbözik a célkitűzés-alapú szabályozás az előíró jellegű szabályozástól? Milyen előnyei és kockázatai vannak?
2. Egy biztonsági érvelésben ez az állítás szerepel: „A szoftvert alaposan tesztelték.” Mi a baj vele? Fogalmazd meg jobban!
3. Melyik GSN-elemmel ábrázolnád: (a) a ReaktorVéd üzemi hőmérséklet-tartományát; (b) azt, hogy a távadók meghibásodási rátája a gyártói adatnak felel meg; (c) a HAZOP-jelentést?
4. Miért nem elég a biztonsági érvelést az üzembe helyezés előtt elkészíteni?
5. Milyen tanulságokat fogalmazott meg a Nimrod-vizsgálat a biztonsági érvelésekről?

Megoldás.

1. Az előíró szabályozás részletes műszaki követelmények betartását ellenőrzi; a célkitűzés-alapú szabályozás azt követeli meg, hogy az üzemeltető bizonyítsa: azonosította a veszélyeket, és elfogadható szintre csökkentette a kockázatokat. Előnye, hogy az egyedi és új rendszerekhez is alkalmazható, és a kockázatokra összpontosít; kockázata, hogy a minősége a felkészültségen és az őszinteségen múlik, és könnyen formális papírmunkává válhat.
2. Nem ellenőrizhető: nem mondja meg, mit, mi ellen és milyen kritérium szerint teszteltek. Jobban: „A szoftver minden biztonsági követelményét követelményalapú teszt ellenőrzi, a tesztek a kód MC/DC-lefedettségét 100%-ban elérik, és minden teszt sikeres.” Ez állítás, amelyet a tesztjelentés és a lefedettségi jelentés közvetlenül alátámaszt.
3. (a) kontextus; (b) feltevés (ha nem igazoljuk) vagy egy részállítás, amelyet üzemi adatokkal támasztunk alá; (c) megoldás (bizonyíték).
4. Mert az érvelés feltevései (meghibásodási ráták, próbák, igénygyakoriság, kezelői teljesítmény) üzem közben változhatnak, és csak a működő rendszer adataiból ellenőrizhetők; a módosítások és a környezet változásai pedig az érvelés egyes részeit érvényteleníthetik.
5. Hogy a biztonsági érvelés könnyen papírmunkává válik, ha a megfelelés igazolására és nem a valódi veszélyek feltárására készül; hogy a dokumentáció mennyisége nem pótolja az érvelés minőségét; hogy a felelősség szétaprózódása miatt mindenki másra számít; és hogy az érvelést a rendszer egész életciklusa során karban kell tartani.

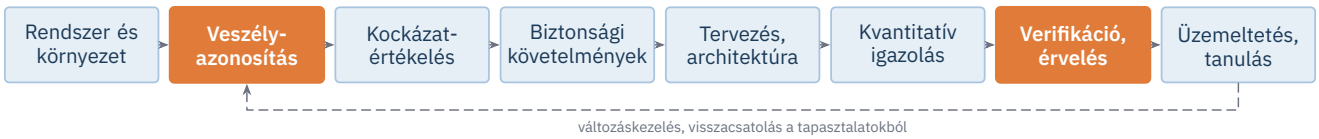
Továbbolvasás

- Kelly, T. P. (1998): *Arguing Safety – A Systematic Approach to Managing Safety Cases.* PhD-értekezés, University of York (Kelly 1998).
- Assurance Case Working Group (2021): *Goal Structuring Notation Community Standard*, 3. változat. SCSC (Assurance Case Working Group 2021).
- Haddon-Cave, C. (2009): *The Nimrod Review.* The Stationery Office (Haddon-Cave 2009).

- UL 4600:2023 – az autonóm termékek biztonsági érvelésen alapuló értékelése (UL Standards & Engagement 2023).
- Denney, E., Pai, G., Habli, I. (2015): Dynamic safety cases for through-life safety assurance. ICSE (Denney és mtsai. 2015).

18.

Új kihívások: rendeltetésszerű funkció, mesterséges intelligencia és kiberbiztonság



A könyv eddigi módszerei egy közös feltevésre épültek: a veszélyes helyzetek forrása valaminek a *meghibásodása* (egy alkatrész, egy szoftver, egy emberi művelet), vagy legalábbis az elemek közötti nem szándékolt kölcsönhatás. Három fejlemény ezt a képet alapjaiban bővíti. Az automatizált járművek és robotok veszélyt okozhatnak úgy is, hogy semmi sem hibásodik meg, egyszerűen azért, mert a funkciójuk teljesítőképessége egy adott helyzetben nem elegendő. A gépi tanulással készült komponensek viselkedését nem egyenként megírt követelmények, hanem adatok határozzák meg, és a hagyományos verifikációs módszerek nagy része nem alkalmazható rájuk. A hálózatba kapcsolt rendszerekben pedig egy szándékos támadás is veszélyes helyzetet teremthet, amely ellen a valószínűségi modellek nem védenek. Ez a fejezet ezt a három kihívást, a rájuk adott szabványosítási és szabályozási válaszokat, végül a mesterséges intelligencia szerepét a biztonsági mérnök munkájában tekinti át.

Tanulási eredmények

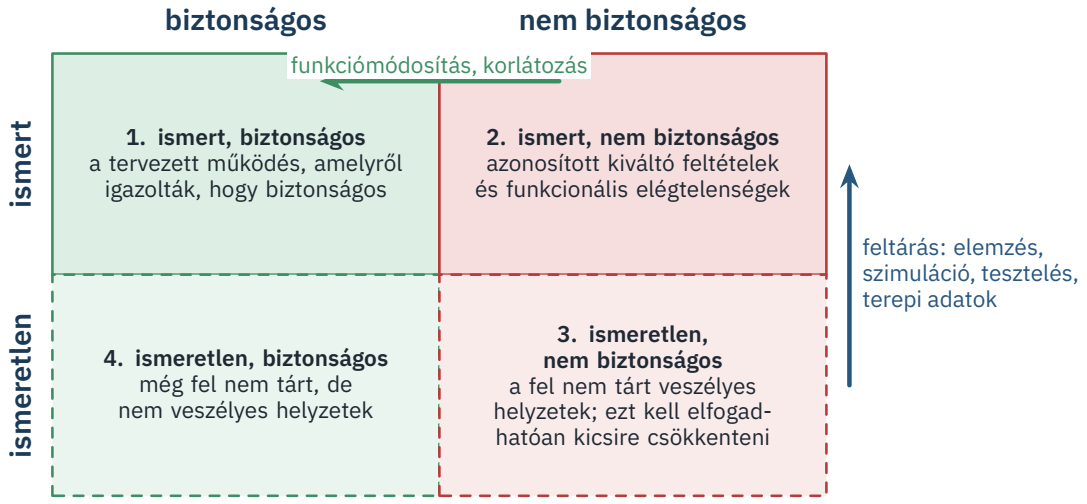
A fejezet végére képes leszel

- megkülönböztetni a funkcionális biztonság és a rendeltetésszerű funkció biztonságának (SOTIF) tárgykörét, és alkalmazni a SOTIF négy területének gondolatmenetét;
- megbecsülni, mekkora üzemi tapasztalat kell egy automatizált rendszer biztonságának statisztikai igazolásához;
- összefoglalni a gépi tanulási komponensek biztonságának fő kihívásait és az AMLAS, az ISO/PAS 8800 és az ISO/IEC TR 5469 válaszait;
- biztonsági architektúrát javasolni gépi tanulási komponenst tartalmazó rendszerre;
- elemezni a kiberbiztonság és a funkcionális biztonság kapcsolatát, és alkalmazni az IEC 62443 zóna- és csatornamodelljét;
- áttekinteni az MI-rendelet, a gépekre vonatkozó rendelet és a kiberrezilienciáról szóló jogszabály biztonsági vonatkozásait és ütemezését;
- felelősen használni a nagy nyelvi modelleket a biztonsági elemzések támogatására.

18.1. A rendeltetésszerű funkció biztonsága

A 2018-as tempei baleset (1. fejezet) jól mutatja a problémát: az automatizált jármű érzékelői működtek, a szoftver a specifikációja szerint futott, mégis halálos baleset történt, mert a rendszer nem tudta időben helyesen osztályozni a kerékpárját toló gyalogost (National Transportation Safety Board 2019). Az ISO 26262 a hibák és a meghibásodások okozta veszélyekkel foglalkozik, ezért ilyen esetekre nem ad választ. Ezt a rést tölti ki az ISO 21448, a **rendeltetésszerű funkció biztonsága** (*safety of the intended functionality*, SOTIF) (ISO 2022).

A SOTIF szerint veszélyt okozhatnak a **funkcionális elégtelenségek**: a specifikáció hiányosságai (a tervezők nem gondoltak egy helyzetre) és a teljesítőképesség korlátai (az érzékelő vagy az algoritmus egy helyzetben nem elég pontos). Ezeket **kiváltó feltételek** (*triggering conditions*) hozzák felszínre: alacsonyan álló nap, köd, erős eső, tükröződő felület, szokatlan alakú tárgy, váratlan emberi viselkedés. A SOTIF célja, hogy a lehetséges forgatókönyveket négy területre osztva (18.1. ábra) a **nem biztonságos ismert** (2.) és különösen a **nem biztonságos ismeretlen** (3.) területet elfogadhatóan kicsire csökkentse. Az ismert veszélyes forgatókönyveket funkciómódosítással kezeljük (jobb érzékelés, érzékelőfúzió, a működési tartomány korlátozása, a vezető figyelmeztetése); az ismeretlen területet pedig elemzéssel, szimulációval, célzott teszteléssel és a terepi adatok gyűjtésével „tárjuk fel”, azaz ismertté tesszük.



18.1. ábra. A SOTIF négy területe. A cél a nem biztonságos forgatókönyvek, különösen a fel nem tártak arányának elfogadhatóan kicsire csökkentése.

A SOTIF kulcsfogalma a **működési tervezési tartomány** (*operational design domain*, ODD): azoknak a körülményeknek a köre (úttípus, sebesség, időjárás, napszak, forgalmi helyzetek), amelyekre a rendszert tervezték. A biztonsági igazolás csak az ODD-n belül érvényes, és a rendszernek fel kell ismernie, ha elhagyja azt. Az automatizált vezetési rendszerek tervezésére, verifikációjára és validálására az ISO/TS 5083 ad átfogó keretet, amely a funkcionális biztonságot, a SOTIF-ot és a kiberbiztonságot egyetlen biztonsági érvelésbe foglalja (ISO 2025b); az ENSZ 157. számú előírása pedig a sávtartó automatizált rendszerek jóváhagyási követelményeit rögzíti (UNECE 2021b).

VészFék: kiváltó feltételek és fantomfékezés

A vészfékező asszisztens SOTIF-elemzése két irányban keres kiváltó feltételeket. A **téves fékezés** (fantomfékezés) jellegzetes kiváltói: hidak és felüljárók, amelyeket a radar akadálynak érzékel; az úttestbe épített fémfedlapok és -lemezek erős radarvisszaverődése; a kamerát elvakító alacsony nap; árnyékok és útburkolati jelek, amelyeket a képfeldolgozás tárgynak vél. A **fékezés elmaradásának** kiváltói: sötét ruhás gyalogos éjszaka, szokatlan alakú tárgy (például kerékpárt toló ember vagy egy felborult jármű), köd és

erős eső. A funkciómódosítások: a radar és a kamera eredményének összevetése (a téves fékezés ellen mindkettő megerősítése, a fékezés elmaradása ellen a figyelmeztetés már egyetlen érzékelő jelzésére), a működés korlátozása rossz látási viszonyok között, és a vezető tájékoztatása. Az ismeretlen terület csökkentésére a gyártó nagy mennyiségű terepi adatot és szimulált forgatókönyvet elemez, és a kiadott járművek adataiból folyamatosan keresi az új kiváltó feltételeket.

18.1.1. Mennyi tesztelés kell?

Az automatizált járművek biztonságának egyik kézenfekvő igazolása az lenne, hogy elég sokat vezetünk velük baleset nélkül. A 8. fejezet megbízhatósági becslései megmutatják, miért nem járható ez az út. Ha egy rendszer t ideig (vagy távolságon) hibamentesen működött, akkor C konfidenciaszinten a hibagyakoriság felső korlátja $\lambda_C = -\ln(1-C)/t$. Kalra és Paddock ezt az amerikai közúti halálozási gyakoriságra alkalmazta, amely 2013-ban körülbelül 1,09 haláleset volt százmillió mérföldenként (Kalra és Paddock 2016). Ahhoz, hogy 95%-os konfidenciával kimutassuk, hogy egy automatizált járműflotta *legalább ilyen jó*, halálos baleset nélkül

$$t = \frac{-\ln(1-0{,}95)}{1{,}09 \cdot 10^{-8}\ \text{mérföld}^{-1}} = \frac{3{,}0}{1{,}09 \cdot 10^{-8}} \approx 2{,}75 \cdot 10^{8}\ \text{mérföld}$$

vezetés kellene, és annak kimutatásához, hogy *lényegesen jobb* is, nagyságrendekkel több. Ez a gondolatmenet ugyanaz, amelyet Butler és Finelli a szoftverekre alkalmazott (15.1. szakasz): a nagyon ritka eseményekre vonatkozó biztonsági állítás nem igazolható pusztán üzemi tapasztalattal. A megoldás a **forgatókönyv-alapú validálás**, amely a releváns és a kritikus helyzeteket szisztematikusan azonosítja, szimulációban és tesztpályán vizsgálja, és a közúti tapasztalatot ennek kiegészítésére használja (Koopman és Wagner 2016); valamint a biztonsági érvelés, amely ezeket a bizonyítékokat összefogja, és az üzemeltetés során folyamatosan frissül (17.5. szakasz) (UL Standards & Engagement 2023).

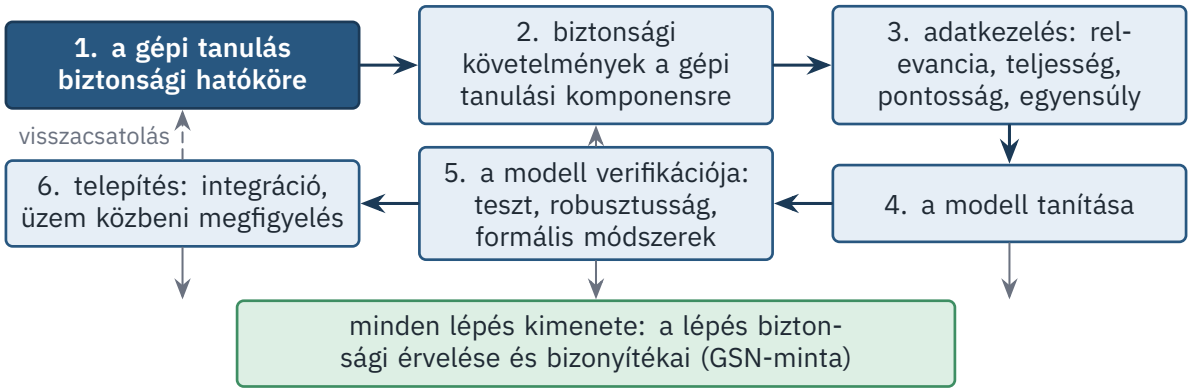
18.2. Gépi tanulás a biztonságkritikus rendszerekben

A gépi tanulási (ML) komponensek, különösen a mély neurális hálók, olyan feladatokban érnek el kiemelkedő teljesítményt, amelyekre hagyományos algoritmust nem lehet írni: képek és jelenetek értelmezésében, összetett mintázatok felismerésében, az emberi viselkedés előrejelzésében. A biztonsági igazolás szempontjából azonban alapvetően különböznek a hagyományos szoftvertől:

- **Az adat a specifikáció.** A komponens viselkedését nem egyenként megírt követelmények, hanem a tanítóadatok határozzák meg. Ha egy helyzet nem szerepel megfelelő arányban az adatokban, a modell viselkedése ott kiszámíthatatlan.
- **Eloszláseltolódás.** A modell a tanítási adatok eloszlásán működik jól; ha a valós környezet ettől eltér (új járműtípusok, más éghajlat, kopott útburkolati jelek), a teljesítmény észrevétlenül romolhat.
- **Robusztusság.** Szegedy és munkatársai kimutatták, hogy szemmel észrevehetetlen, gondosan megválasztott változtatások a bemeneten egy neurális hálót teljesen téves osztályozásra késztethetnek (Szegedy és mtsai. 2014); ez egyszerre biztonsági és kiberbiztonsági kockázat.
- **Átláthatatlanság.** A modell belső működése nem vezethető vissza ember által érthető szabályokra, ezért a hagyományos felülvizsgálat és a strukturális lefedettség nem alkalmazható.
- **Nem szándékolt optimalizálás.** Amodei és munkatársai számos módját írták le annak, ahogyan egy tanuló rendszer a célfüggvényét a tervezők szándékával ellentétesen, nem kívánt mellékhatásokkal teljesítheti (Amodei és mtsai. 2016).

Salay, Queiroz és Czarnecki elemzése szerint az ISO 26262 szoftverfejlesztési módszereinek jelentős része csak részben vagy egyáltalán nem alkalmazható a gépi tanulási komponensekre (Salay és mtsai. 2017). Erre a kihívásra az elmúlt években több útmutató és szabvány született.

Az **AMLAS** (*Assurance of Machine Learning for use in Autonomous Systems*) a York-i Egyetem módszertana, amely a gépi tanulási komponens fejlesztését hat lépésre bontja, és mindegyikhez egy biztonsági érvelési mintát és a szükséges bizonyítékokat rendeli (18.2. ábra) (Hawkins és mtsai. 2021): a gépi tanulás biztonsági hatókörének meghatározása, a komponens biztonsági követelményei (például a teljesítmény, a robusztusság és az értelmezhetőség követelményei), az adatkezelés (a tanító-, a validáló- és a tesztadatok relevanciája, teljessége, pontossága és kiegyensúlyozottsága), a modell tanítása, a modell verifikációja (független tesztadatokon, robusztussági vizsgálatokkal, ahol lehet, formális módszerekkel), végül a telepítés és az üzem közbeni megfigyelés.



18.2. ábra. Az AMLAS hat lépése. Minden lépés egy biztonsági érvelési mintát és a hozzá tartozó bizonyítékokat állít elő; a verifikáció és a telepítés tapasztalatai visszahatnak a követelményekre.

Az **ISO/PAS 8800** a közúti járművek mesterséges intelligenciát tartalmazó rendszereinek biztonságára ad útmutatást: az ISO 26262 és az ISO 21448 kiegészítéseként az MI-alapú rendszerelemek biztonsági életciklusát, a biztonsági követelmények levezetését, az adatok életciklusát és az üzem közbeni megfigyelést szabályozza (ISO 2024). Az **ISO/IEC TR 5469** ágazatfüggetlen technikai jelentés a mesterséges intelligencia funkcionális biztonsági alkalmazásairól: áttekinti, milyen szerepben (a biztonsági funkción belül vagy azon kívül) és milyen feltételekkel használható MI, milyen tulajdonságai befolyásolják a biztonságot, és milyen intézkedések segítenek a kockázatok kezelésében (ISO/IEC 2024).

18.2.1. Architekturális válasz: egyszerűség a bonyolultság felügyeletére

A gépi tanulási komponensek biztonságának talán legerősebb eszköze az architektúra. A **Simplex-architektúra** gondolata szerint egy nagy teljesítményű, de nehezen igazolható vezérlő mellé egy egyszerű, hagyományos módszerekkel igazolható biztonsági vezérlőt és egy döntési logikát teszünk, amely átkapcsol az egyszerű vezérlőre, ha a rendszer állapota a biztonságos tartomány határához közelít (Sha 2001). Ugyanez a minta jelenik meg a 14.6. szakasz E-Gas felügyeletében és a ReaktorVéd független biztonsági funkciójában: a bonyolult funkciót egy egyszerű, diverz, független felügyelet ellenőrzi. Gépi tanulási komponensekre alkalmazva ez lehet egy fizikai plauzibilitás-ellenőrzés (a jármű nem fékezhet erősen, ha egyik független érzékelő sem lát akadályt), egy hagyományos érzékelőre épülő biztonsági burok (a robot sebességét a tanúsított lézerszkenner korlátozza, akármit is javasol a kamerás MI), vagy a bemenet eloszláson kívüliségének (*out-of-distribution*) felismerése, amely ilyenkor egy óvatosabb üzemmódba kapcsol.

KoboCella: kamerás személyfelismerés gépi tanulással

A KoboCella fejlesztői szeretnék a lézerszkennert egy kamerás, gépi tanulásos személyfelismerővel kiegészíteni, amely a szerelő testtartását és mozgásának irányát is felismeri, és így finomabban szabályozhatná a robot sebességét. A gépekre vonatkozó (EU) 2023/1230 rendelet szerint azonban a teljesen vagy részben önfejlesztő viselkedésű, gépi tanulást alkalmazó biztonsági alkatrészek, amelyek biztonsági funkciót látnak el, a fokozott kockázatú termékek közé tartoznak, és bejelentett szervezet bevonásával kell értékelni a megfelelőségüket (Európai Parlament és Tanács 2023). A csapat ezért olyan architektúrát választ, amelyben a biztonsági funkciót továbbra is a tanúsított lézerszkenner és a robotvezérlő kétcsatornás biztonsági része látja el (PL_r szerint igazolva); a kamerás MI csak a *termelékenységet* javítja, a biztonsági határokon belül (például a robot pályáját tervezi át, hogy elkerülje a szerelőt). Az MI hibája így legfeljebb felesleges lassítást okozhat, veszélyes helyzetet nem. Ha a jövőben az MI-t a biztonsági funkcióba is be akarják építeni, az AMLAS-hoz és az ISO/IEC TR 5469-hez hasonló teljes igazolásra lesz szükség.

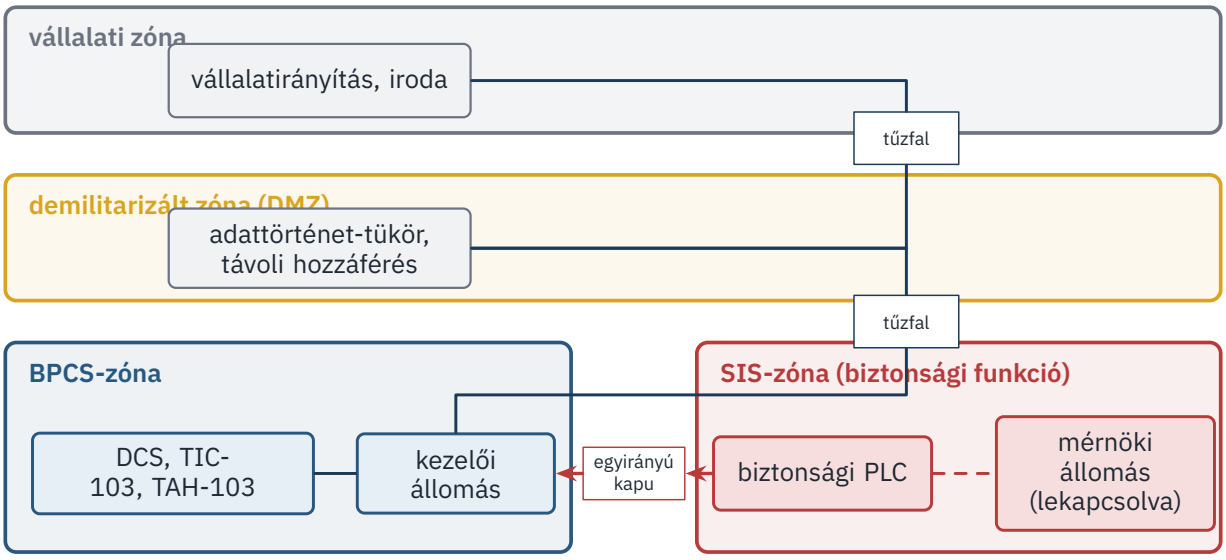
18.3. Kiberbiztonság és funkcionális biztonság

A biztonsági elemzések hagyományosan véletlen meghibásodásokkal és nem szándékos tévedésekkel számolnak. A támadó azonban nem véletlenszerűen viselkedik: a leggyengébb pontot keresi, több „meghibásodást” idézhet elő egyszerre, és éppen a ritka, nem tesztelt állapotokat használja ki. A valószínűségi modellek függetlenségi feltevései vele szemben értelmetlenek. Ezért a kiberbiztonság (*security*) nem illeszthető be egyszerűen a megbízhatósági számításokba, hanem külön elemzést és külön intézkedéseket igényel, amelyeket a funkcionális biztonsággal (*safety*) össze kell hangolni (IEC 2019b; Young és Leveson 2014).

Az ipari irányítórendszerek elleni támadások közül a biztonsági mérnök számára a legtanulságosabb a 2017-ben egy közel-keleti petrolkémiai üzemben felfedezett kártevő, amelyet az amerikai hatóságok HatMan, a kutatók Triton vagy Trisis néven írtak le: a támadók a biztonsági műszeres rendszer vezérlőit vették célba, és megpróbálták átprogramozni őket. A támadás azért derült ki, mert a vezérlők hibát észleltek, és biztonságos állapotba, leállásba vitték az üzemet (Cybersecurity and Infrastructure Security Agency 2019). A tanulság súlyos: a biztonsági funkció, az utolsó műszaki védelmi réteg maga is célpont lehet, és egy sikeres támadás egyszerre iktathatja ki a védelmet és idézheti elő a veszélyes helyzetet.

A folyamatiparban az IEC 61511 2016-os kiadása előírja, hogy a biztonsági műszeres rendszerre kiberbiztonsági kockázatértékelést kell végezni (IEC 2016b), az ipari automatizálás kiberbiztonságának szabványsorozata pedig az IEC 62443. Ennek alapgondolata a **zónák és csatornák** modellje (18.3. ábra): a rendszert azonos biztonsági követelményű zónákra osztjuk, a zónák közötti kommunikációt ellenőrzött csatornákra korlátozzuk, és minden zónához célzott biztonsági szintet rendelünk (IEC 2009--2024). A közúti járművekben az ISO/SAE 21434 a kiberbiztonsági mérnöki folyamatot, benne a fenyegetés- és kockázatelemzést (TARA) szabályozza (ISO/SAE 2021), az ENSZ 155. számú előírása pedig 2024 júliusától minden új jármű forgalomba helyezésének feltétele egy tanúsított kiberbiztonsági irányítási rendszer (UNECE 2021a).

A biztonság és a kiberbiztonság nemcsak kiegészítik, hanem befolyásolhatják is egymást. Egy kiberbiztonsági intézkedés veszélyeztetheti a biztonságot: ha a kezelői állomás néhány hibás jelszó után zárol, egy vészhelyzetben a kezelő kizárhatja magát; ha egy biztonsági frissítés a biztonsági PLC szoftverét módosítja, az újra validálást igényel. Fordítva, egy biztonsági funkció kiberbiztonsági kockázatot teremthet: egy távoli kioldás-visszaállítási lehetőség támadási útvonal lehet. Ezért a két terület elemzését közös keretben érdemes végezni; az STPA rendszerelméleti megközelítése (7. fejezet) például kiterjeszthető a nem biztonságos irányítási műveletek szándékos előidézésére is (STPA-Sec) (Young és Leveson 2014).



A zónák közötti kapcsolatok (csatornák) ellenőrzöttek; a SIS-zónából a BPCS felé csak állapotadat megy ki, befelé nincs írási út; a mérnöki állomás csak engedélyezett módosításkor csatlakozik.

18.3. ábra. A ReaktorVéd irányítórendszerének zónái és csatornái az IEC 62443 szerint. A biztonsági funkció külön zónában van, ahonnan csak állapotadat megy ki, befelé nincs írási út.

ReaktorVéd: a biztonsági funkció kiberbiztonsága

A ReaktorVéd kiberbiztonsági kockázatértékelése megállapította, hogy a biztonsági PLC diagnosztikai adatait a karbantartók a vállalati hálózatról is szeretnék látni, és a mérnöki állomás az eredeti tervben állandóan csatlakozott a biztonsági hálózathoz. A módosított architektúrában a biztonsági funkció önálló zóna; a BPCS felé csak egy egyirányú adatkapun keresztül megy ki állapotadat; a mérnöki állomás csak engedélyezett módosításkor, fizikai kulccsal csatlakoztatható; a biztonsági PLC-n a program módosítása csak a helyszínen, kulcskapcsolóval engedélyezhető. A biztonsági érvelés (17. fejezet) G5 ága kiegészül egy állítással: a biztonsági funkció integritását kiberbiztonsági intézkedések védik, és a feltevések (például hogy a mérnöki állomás csak engedéllyel csatlakozik) teljesülését a változáskezelés és az auditok ellenőrzik.

18.4. A szabályozás új rétegei

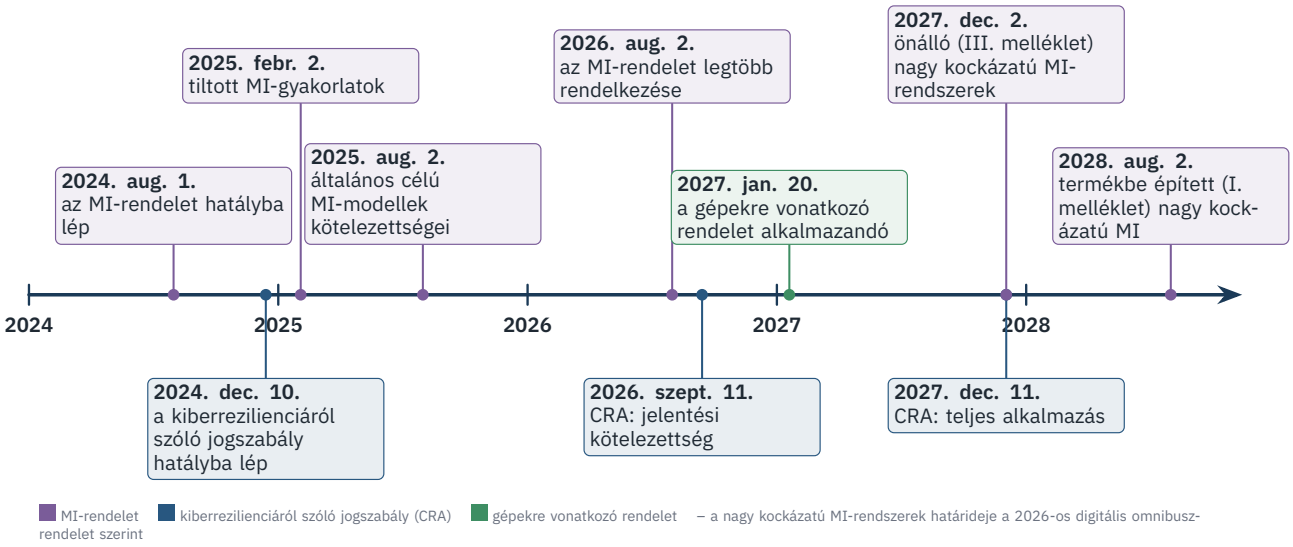
Az Európai Unióban három új rendelet alakítja át a biztonságkritikus termékek szabályozását (18.4. ábra).

A **gépekre vonatkozó (EU) 2023/1230 rendelet** 2027. január 20-tól váltja fel a gépi irányelvet. Az új rendelet kifejezetten kezeli a mesterséges intelligenciát és a kiberbiztonságot: a gépeknek a kiszámíthatóan változó, önfejlesztő viselkedés mellett is biztonságosnak kell maradniuk, és a biztonsági funkciókat védeni kell a rosszindulatú befolyásolással szemben (Európai Parlament és Tanács 2023).

A **mesterséges intelligenciáról szóló (EU) 2024/1689 rendelet** (MI-rendelet) kockázatalapú szabályozást vezet be. A nagy kockázatú MI-rendszerek közé tartoznak azok, amelyek az uniós harmonizációs jogszabályok (például a gépekre, a játékokra, az orvostechnikai eszközökre vonatkozó szabályok) hatálya alá tartozó termékek biztonsági alkatrészei. Rájuk kockázatkezelési rendszer, adatkezelési követelmények, műszaki dokumentáció, naplózás, átláthatóság, emberi felügyelet, valamint pontossági, robusztussági és kiberbiztonsági követelmények vonatkoznak (Európai

Parlament és Tanács 2024a). A nagy kockázatú rendszerekre vonatkozó kötelezettségek alkalmazását a 2026 júliusában kihirdetett digitális omnibusz-rendelet elhalasztotta: az önálló (III. mellékletbeli) rendszerekre 2027. december 2-tól, a termékekbe biztonsági alkatrészként beépített (I. mellékletbeli) rendszerekre 2028. augusztus 2-tól kell alkalmazni őket (Európai Parlament és Tanács 2026).

A **kiberrezilienciáról szóló (EU) 2024/2847 rendelet** a digitális elemeket tartalmazó termékekre ír elő kiberbiztonsági követelményeket a teljes életciklusra, a sérülékenységek kezelésével és a biztonsági frissítésekkel együtt. A súlyos incidensek és az aktívan kihasznált sérülékenységek jelentési kötelezettsége 2026. szeptember 11-től, a rendelet többi része 2027. december 11-től alkalmazandó (Európai Parlament és Tanács 2024b).



18.4. ábra. Az EU biztonsági szempontból lényeges új jogszabályainak ütemezése. A nagy kockázatú MI-rendszerekre vonatkozó határidők a 2026-os digitális omnibusz-rendelet szerint módosultak.

A három rendelet egymásra épül: egy MI-alapú biztonsági alkatrészt tartalmazó, hálózatba kapcsolt gépre egyszerre vonatkozhat mindhárom. A harmonizált szabványok kidolgozása folyamatban van, és a következő években várhatóan a funkcionális biztonsági szabványok (köztük az IEC 61508 harmadik, 2027-re várt kiadása) is beépítik az MI-re és a kiberbiztonságra vonatkozó követelményeket. A mérnöki gyakorlat szempontjából a legfontosabb üzenet: a biztonság, a kiberbiztonság és az MI megbízhatósága nem kezelhető külön silókban; közös kockázatelemzés, közös architektúra és közös, élő biztonsági érvelés kell hozzájuk.

18.5. A mesterséges intelligencia mint a biztonsági mérnök eszköze

A nagy nyelvi modellek a biztonsági mérnök munkáját is megváltoztatják. Segíthetnek a veszélyazonosítás ötletgyűjtésében (vezérszavas eltérések, meghibásodási módok, nem biztonságos irányítási műveletek felvetésében), a szabványok és a szakirodalom áttekintésében, a dokumentáció előkészítésében, a számítások ellenőrzésében és a gyakorlati útmutató notebookjaihoz hasonló elemző kódok megírásában. Diemert és Weber kísérleti vizsgálata szerint egy nagy nyelvi modell hasznos javaslatokkal segítheti a veszélyelemzést, de a javaslatok egy része téves vagy irreleváns, és a modell nem helyettesíti a szakértői ítéletet (Diemert és Weber 2023).

A felelős használat elvei a 16. fejezet automatizálási tanulságaiból következnek: a modell ötletgyűjtő partner, nem döntéshozó; minden állítását ellenőrizni kell, különösen a számokat, a szabványhivatkozásokat

és a jogi állításokat; vigyázni kell az automatizálási torzításra, amely a gépi javaslatot hajlamos elfogadni akkor is, ha a saját tudásunk ellentmond neki; a bizalmas tervadatokat csak megfelelő adatvédelmi feltételekkel szabad megosztani; és a biztonsági érvelésben dokumentálni kell, hogy egy elemzésben milyen szerepet játszott a modell, és ki ellenőrizte a kimenetét.

Nagy nyelvi modell mint STPA-segítő

„Egy kollaboratív robotcella STPA-elemzését végezzük. Az irányító a robotvezérlő biztonsági része; irányítási műveletei: sebességcsökkentés, biztonságos megállás, újraindítás. A visszacsatolás a lézerszkenner jele (személy a 0,5 m-es, illetve az 1,2 m-es zónában) és a robot sebességmérése. A veszteség: a szerelő sérülése; a veszély: a robot nagy sebességgel mozog, miközben a szerelő a közös munkatérben van. Mind a négy nem biztonságos irányítási művelet típusára (nem adja ki; kiadja, amikor nem kellene; rossz időben vagy sorrendben adja ki; túl korán leállítja vagy túl sokáig alkalmazza) sorolj fel lehetséges eseteket, és mindegyikhez a kontextust. Utána javasolj okozati forgatókönyveket a visszacsatolás hibáira is. Jelöld, ha egy javaslatod bizonytalan, és ne hivatkozz olyan szabványpontra, amelyben nem vagy biztos.”
A modell válaszát a csapat a saját elemzésével veti össze (7. fejezet); a kimenet csak akkor kerülhet a jegyzőkönyvbe, ha egy szakértő ellenőrizte és elfogadta.

18.6. Kitekintés

A biztonságkritikus rendszerek tudománya az első hibafáktól a tanuló, hálózatba kapcsolt, ember és gép közötti munkamegosztásra épülő rendszerekig jutott. A könyv módszerei ehhez a fejlődéshez három irányban adnak kapaszkodót. A **valószínűségi modellek** (megbízhatóság, hibafa, eseményfa, Markov-modellek, SIL-igazolás) ott maradnak nélkülözhetetlenek, ahol a meghibásodások véletlenszerűek és az adatok megbízhatók. A **rendszerelméleti szemlélet** (STAMP, STPA, a biztonsági kultúra, a Safety-II) ott, ahol a balesetek a kölcsönhatásokból, a szervezetből és a változásból fakadnak. A **biztonsági érvelés** pedig az a keret, amely a kettőt, a tervezési döntéseket és az üzemeltetés tapasztalatait egyetlen, ellenőrizhető és élő egésszé fogja össze. Az új technológiák nem teszik feleslegessé ezeket az eszközöket; éppen ellenkezőleg, a gépi tanulás, az autonómia és a hálózatosodás korában a biztonsági mérnök legfontosabb képessége az lesz, hogy tudja, melyik eszköz mire alkalmas, és mire nem.

Gyakorlat

A gyakorlati útmutató 12. laborja egy projektfeladat: egy választott rendszerre (például egy automatizált raktári járműre vagy egy orvostechnikai infúziós pumpára) a könyv módszereit végigvezetve kell elkészíteni a veszélyelemzést, a kockázatértékelést, a biztonsági követelményeket, a kvantitatív igazolást és a biztonsági érvelés vázát. Az útmutató mintapromptokat is ad a nagy nyelvi modellek felelős használatához az egyes lépésekben.

Összefoglalás

- A SOTIF (ISO 21448) a meghibásodás nélkül, a funkcionális elégtelenségekből és a kiváltó feltételekből eredő veszélyekkel foglalkozik; célja a nem biztonságos ismert és ismeretlen forgatókönyvek elfogadhatóan kicsire csökkentése.
- A ritka eseményekre vonatkozó biztonsági állítás nem igazolható pusztán üzemi tapasztalattal: az emberi vezetővel egyenértékű halálozási gyakoriság 95%-os konfidenciájú kimutatásához mintegy 275 millió mérföld kellene; a forgatókönyv-alapú validálás és a biztonsági érvelés a megoldás.
- A gépi tanulási komponensek biztonságának kihívásai: az adat mint specifikáció, az eloszláseltolódás, a robusztusság, az átláthatatlanság és a nem szándékolt optimalizálás; az AMLAS, az ISO/PAS 8800 és az ISO/IEC TR 5469 ad rájuk választ.
- Az architektúra a legerősebb eszköz: egy egyszerű, igazolható felügyelet (Simplex-minta) korlátozza a bonyolult komponens hatását.

- A támadó nem véletlenszerű; a kiberbiztonságot külön, de a funkcionális biztonsággal összehangoltan kell kezelni (IEC 62443, ISO/SAE 21434, IEC 61511 kiberbiztonsági értékelés). A Triton-támadás a biztonsági funkciót vette célba.
- A gépekre vonatkozó rendelet, az MI-rendelet és a kiberrezilienciáról szóló jogszabály 2026 és 2028 között fokozatosan alkalmazandó; a nagy kockázatú MI-rendszerek határidejét a 2026-os digitális omnibusz-rendelet 2027 decemberére, illetve 2028 augusztusára tolta ki.
- A nagy nyelvi modellek hasznos segítők a biztonsági elemzésben, de minden kimenetüket szakértőnek kell ellenőriznie.

Önellenőrzés

1. Miben különbözik a fantomfékezés oka egy hibás radarérzékelő okozta téves fékezéstől? Melyik szabvány melyiket kezeli?
2. Egy automatizált rendszer hibagyakoriságára 10^{-7} óránkénti felső korlátot szeretnénk igazolni 90%-os konfidenciával, hibamentes üzemi tapasztalatból. Hány üzemóra kell?
3. Miért nem elég egy gépi tanulási modell tesztadatokon mért 99,9%-os pontossága a biztonsági igazoláshoz?
4. Javasolj Simplex-jellegű architektúrát egy MI-alapú gyalogosfelismerőt tartalmazó vészfékező rendszerre!
5. Mondj egy példát arra, amikor egy kiberbiztonsági intézkedés a funkcionális biztonságot rontja, és egyet a fordítottjára!

Megoldás.

1. A fantomfékezés során semmi sem hibásodik meg: az érzékelés teljesítőképessége egy kiváltó feltétel (például egy híd vagy fémlemez) mellett nem elegendő, ez a SOTIF (ISO 21448) tárgya. A hibás radar okozta téves fékezés meghibásodás következménye, ez az ISO 26262 tárgya.
2. $t = -\ln(1-0{,}9)/10^{-7} = 2{,}30 \cdot 10^{7}$ óra, azaz több mint 2600 év egyetlen rendszerre, vagy például ezer rendszer 2,6 évnyi hibamentes működése.
3. Mert a pontosság egy átlag a tesztadatok eloszlásán: nem mond semmit a ritka, de veszélyes helyzetekről, az eloszláson kívüli bemenetekről, a robusztusságról és arról, hogy a tévedések *milyen* hibák (veszélyes vagy ártalmatlan). A biztonsági követelményeket a veszélyes hibamódokra kell megfogalmazni és igazolni, a működési tartomány teljes lefedésével.
4. Például: az MI-alapú felismerő javaslatot tesz a fékezésre, de az erős fékezéshez egy független, hagyományos módszerrel igazolt csatorna (például a radar távolság- és sebességmérése alapján számolt ütközési idő) megerősítése kell; a téves fékezés ellen az MI egyedül csak figyelmeztetést vagy enyhe fékezést kezdeményezhet. Ha a felismerő bemenete eloszláson kívülinek tűnik (például szokatlan időjárás), a rendszer óvatosabb üzemmódba vált, és a vezetőt figyelmezteti.
5. Kiberbiztonság rontja a biztonságot: a kezelői állomás zárolása hibás jelszavak után vészhelyzetben kizárhatja a kezelőt; egy kötelező frissítés újravalidálás nélkül megváltoztathatja a biztonsági funkció viselkedését. Biztonság rontja a kiberbiztonságot: egy távoli kioldás-visszaállítás vagy egy állandóan csatlakozó karbantartói hozzáférés, amelyet a gyors helyreállítás érdekében építettek be, támadási útvonalat nyit.

Továbbolvasás

- ISO 21448:2022 – a rendeltetésszerű funkció biztonsága (ISO 2022); ISO/TS 5083:2025 – az automatizált vezetési rendszerek biztonsága (ISO 2025b).
- Kalra, N., Paddock, S. M. (2016): Driving to safety. *Transportation Research Part A* (Kalra és Paddock 2016).
- Hawkins, R. et al. (2021): *Guidance on the Assurance of Machine Learning in Autonomous Systems (AMLAS)* (Hawkins és mtsai. 2021); ISO/PAS 8800:2024 (ISO 2024); ISO/IEC TR 5469:2024 (ISO/IEC 2024).
- Sha, L. (2001): Using simplicity to control complexity. *IEEE Software* (Sha 2001).
- IEC 62443 és IEC TR 63069 – az ipari kiberbiztonság és a funkcionális biztonság összehangolása (IEC 2009--2024, 2019b).
- (EU) 2023/1230, (EU) 2024/1689 és (EU) 2024/2847 rendelet, valamint a 2026-os digitális omnibusz-rendelet (Európai Parlament és Tanács 2023, 2024a, 2024b, 2026).

Irodalom

Abernethy, Robert B. 2006. *The New Weibull Handbook*. 5. kiad. R. B. Abernethy.

Abonyi, János, és Tímea Fülep. 2014. *Biztonságkritikus rendszerek*. Pannon Egyetem.

AIAG, és VDA. 2019. *Failure Mode and Effects Analysis – FMEA Handbook*. Automotive Industry Action Group; Verband der Automobilindustrie.

Amodei, Dario, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, és Dan Mané. 2016. „Concrete Problems in AI Safety". *arXiv preprint arXiv:1606.06565*.

Assurance Case Working Group. 2021. *Goal Structuring Notation Community Standard, Version 3*. SCSC-141C. Safety-Critical Systems Club.

Avižienis, Algirdas. 1985. „The N-Version Approach to Fault-Tolerant Software". *IEEE Transactions on Software Engineering* SE-11 (12): 1491–501.

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.

Bainbridge, Lisanne. 1983. „Ironies of Automation". *Automatica* 19 (6): 775–79. https://doi.org/10.1016/0005-1098(83)90046-8.

Behm, Patrick, Paul Benoit, Alain Faivre, és Jean-Marc Meynadier. 1999. „Météor: A Successful Application of B in a Large Project". *FM'99 – Formal Methods*, Lecture Notes in Computer Science, köt. 1708: 369–87.

Birnbaum, Zygmunt W. 1969. „On the Importance of Different Components in a Multicomponent System". In *Multivariate Analysis II*, szerkesztette P. R. Krishnaiah. Academic Press.

Borst, M. van der, és H. Schoonakker. 2001. „An Overview of PSA Importance Measures". *Reliability Engineering & System Safety* 72 (3): 241–45.

Bryant, Randal E. 1986. „Graph-Based Algorithms for Boolean Function Manipulation". *IEEE Transactions on Computers* C-35 (8): 677–91. https://doi.org/10.1109/TC.1986.1676819.

Buncefield Major Incident Investigation Board. 2008. *The Buncefield Incident 11 December 2005: The Final Report of the Major Incident Investigation Board*. HSE Books.

Bureau d'Enquêtes et d'Analyses. 2012. *Final Report on the Accident on 1st June 2009 to the Airbus A330-203 Registered F-GZCP Operated by Air France Flight AF 447 Rio de Janeiro–Paris*. BEA.

Butler, Ricky W., és George B. Finelli. 1993. „The Infeasibility of Quantifying the Reliability of Life-Critical Real-Time Software”. *IEEE Transactions on Software Engineering* 19 (1): 3–12. https://doi.org/10.1109/32.210303.

CENELEC. 2017. *EN 50126-1: Railway Applications – The Specification and Demonstration of Reliability, Availability, Maintainability and Safety (RAMS) – Part 1: Generic RAMS Process*.

CENELEC. 2018. *EN 50129: Railway Applications – Communication, Signalling and Processing Systems – Safety Related Electronic Systems for Signalling*.

CENELEC. 2023. *EN 50716: Railway Applications – Requirements for Software Development*.

Center for Chemical Process Safety. 2001. *Layer of Protection Analysis: Simplified Process Risk Assessment*. American Institute of Chemical Engineers.

Center for Chemical Process Safety. 2015. *Guidelines for Initiating Events and Independent Protection Layers in Layer of Protection Analysis*. American Institute of Chemical Engineers – Wiley.

Center for Chemical Process Safety, és Energy Institute. 2018. *Bow Ties in Risk Management: A Concept Book for Process Safety*. American Institute of Chemical Engineers – Wiley.

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.

Cullen, W. Douglas. 1990. *The Public Inquiry into the Piper Alpha Disaster*. Cm 1310. HMSO.

Cybersecurity and Infrastructure Security Agency. 2019. *MAR-17-352-01 HatMan – Safety System Targeted Malware (Update B)*. CISA.

Delmas, David, és Jean Souyris. 2007. „Astrée: From Research to Industry”. *Static Analysis (SAS 2007)*, Lecture Notes in Computer Science, köt. 4634: 437–51.

Denney, Ewen, Ganesh Pai, és Ibrahim Habli. 2015. „Dynamic Safety Cases for Through-Life Safety Assurance”. *Proceedings of the 37th IEEE/ACM International Conference on Software Engineering (ICSE)* 2: 587–90. https://doi.org/10.1109/ICSE.2015.199.

Diemert, Simon, és Jens H. Weber. 2023. „Can Large Language Models Assist in Hazard Analysis?” *Computer Safety, Reliability, and Security. SAFECOMP 2023 Workshops*, Lecture Notes in Computer Science, köt. 14182: 410–22.

Dugan, Joanne Bechta, Salvatore J. Bavuso, és Mark A. Boyd. 1992. „Dynamic Fault-Tree Models for Fault-Tolerant Computer Systems”. *IEEE Transactions on Reliability* 41 (3): 363–77. https://doi.org/10.1109/24.159800.

EEMUA. 2013. *EEMUA Publication 191: Alarm Systems – A Guide to Design, Management and Procurement*. 3. kiad.

EGAS Workgroup. 2015. *Standardized E-Gas Monitoring Concept for Gasoline and Diesel Engine Control Units*. Version 6.0. Verband der Automobilindustrie (VDA) – Arbeitskreis EGAS.

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.

Ericson, Clifton A. 1999. „Fault Tree Analysis – A History”. *Proceedings of the 17th International System Safety Conference* (Orlando, FL).

Európai Parlament és Tanács. 2012. *2012/18/EU irányelv a veszélyes anyagokkal kapcsolatos súlyos balesetek veszélyének kezeléséről (Seveso III.)*.

Európai Parlament és Tanács. 2023. *(EU) 2023/1230 rendelet a gépekről*.

Európai Parlament és Tanács. 2024a. *(EU) 2024/1689 rendelet a mesterséges intelligenciára vonatkozó harmonizált szabályok megállapításáról (mesterséges intelligenciáról szóló rendelet)*.

Európai Parlament és Tanács. 2024b. *(EU) 2024/2847 rendelet a digitális elemeket tartalmazó termékekre vonatkozó horizontális kiberbiztonsági követelményekről (kiberrezilienciáról szóló jogszabály)*.

Európai Parlament és Tanács. 2026. *Rendelet az (EU) 2024/1689 rendeletnek a mesterséges intelligenciára vonatkozó harmonizált szabályok végrehajtásának egyszerűsítése tekintetében történő módosításáról (digitális omnibusz a mesterséges intelligenciáról)*.

Fussell, Jerry B. 1975. „How to Hand-Calculate System Reliability and Safety Characteristics”. *IEEE Transactions on Reliability* R-24 (3): 169–74.

Fussell, Jerry B., és William E. Vesely. 1972. „A New Methodology for Obtaining Cut Sets for Fault Trees”. *Transactions of the American Nuclear Society* 15: 262–63.

Gertman, David, Harold Blackman, Julie Marble, James Byers, és Curtis Smith. 2005. *The SPAR-H Human Reliability Analysis Method*. NUREG/CR-6883. U.S. Nuclear Regulatory Commission.

Goldberg, Bruce E., Kay Everhart, Rachel Stevens, Nelson Babbitt, Pat Clemens, és Leslie Stout. 1994. *System Engineering „Toolbox” for Design-Oriented Engineers*. NASA Reference Publication No. 1358. NASA Marshall Space Flight Center.

Haddon, William. 1973. „Energy Damage and the Ten Countermeasure Strategies”. *Journal of Trauma* 13 (4): 321–31.

Haddon-Cave, Charles. 2009. *The Nimrod Review: An Independent Review into the Broader Issues Surrounding the Loss of the RAF Nimrod MR2 Aircraft XV230 in Afghanistan in 2006*. HC 1025. The Stationery Office.

Hauge, Stein, Mary Ann Lundteigen, Per Hokstad, és Solfrid Håbrekke. 2013. *Reliability Prediction Method for Safety Instrumented Systems – PDS Method Handbook*. SINTEF A24442. SINTEF.

Hawkins, Richard, Colin Paterson, Chiara Picardi, Yan Jia, Radu Calinescu, és Ibrahim Habli. 2021. *Guidance on the Assurance of Machine Learning in Autonomous Systems (AMLAS)*. Assuring Autonomy International Programme, University of York.

Hayhurst, Kelly J., Dan S. Veerhusen, John J. Chilenski, és Leanna K. Rierson. 2001. *A Practical Tutorial on Modified Condition/Decision Coverage*. NASA/TM-2001-210876. NASA Langley Research Center.

Health and Safety Executive. 2001. *Reducing Risks, Protecting People: HSE’s Decision-Making Process*. HSE Books.

Health and Safety Executive. 2003. *Out of Control: Why Control Systems Go Wrong and How to Prevent Failure*. 2. kiad. HSE Books.

Heinrich, Herbert W. 1931. *Industrial Accident Prevention: A Scientific Approach*. McGraw-Hill.

Hollnagel, Erik. 2014. *Safety-I and Safety-II: The Past and Future of Safety Management*. Ashgate.

IEC. 2009--2024. *IEC 62443: Security for Industrial Automation and Control Systems (szabványsorozat)*.

IEC. 2006a. *IEC 61025: Fault Tree Analysis (FTA)*. 2. kiad.

IEC. 2006b. *IEC 61165: Application of Markov Techniques*. 2. kiad.

IEC. 2006c. *IEC 62304: Medical Device Software – Software Life Cycle Processes*.

IEC. 2010a. *IEC 61508-1...7: Functional Safety of Electrical/Electronic/Programmable Electronic Safety-Related Systems*. 2. kiad. International Electrotechnical Commission.

IEC. 2010b. *IEC 62502: Analysis Techniques for Dependability – Event Tree Analysis (ETA)*.

IEC. 2015. *IEC 60050-192: International Electrotechnical Vocabulary – Part 192: Dependability*.

IEC. 2016a. *IEC 61078: Reliability Block Diagrams*. 3. kiad.

IEC. 2016b. *IEC 61511-1: Functional Safety – Safety Instrumented Systems for the Process Industry Sector – Part 1: Framework, Definitions, System, Hardware and Application Programming Requirements*. 2. kiad.

IEC. 2016c. *IEC 61882: Hazard and Operability Studies (HAZOP Studies) – Application Guide*. 2. kiad.

IEC. 2018. *IEC 60812: Failure Modes and Effects Analysis (FMEA and FMECA)*. 3. kiad.

IEC. 2019a. *IEC 31010: Risk Management – Risk Assessment Techniques*. 2. kiad.

IEC. 2019b. *IEC TR 63069: Industrial-Process Measurement, Control and Automation – Framework for Functional Safety and Security*.

IEC. 2021. *IEC 62061: Safety of Machinery – Functional Safety of Safety-Related Control Systems*. 2. kiad.

ISA. 2016. *ANSI/ISA-18.2: Management of Alarm Systems for the Process Industries*.

ISO. 2010. *ISO 12100: Safety of Machinery – General Principles for Design – Risk Assessment and Risk Reduction*.

ISO. 2016. *ISO/TS 15066: Robots and Robotic Devices – Collaborative Robots*.

ISO. 2018a. *ISO 26262-1...12: Road Vehicles – Functional Safety*. 2. kiad.

ISO. 2018b. *ISO 31000: Risk Management – Guidelines*. 2. kiad.

ISO. 2019. *ISO 14971: Medical Devices – Application of Risk Management to Medical Devices*. 3. kiad.

ISO. 2022. *ISO 21448: Road Vehicles – Safety of the Intended Functionality*.

ISO. 2023. *ISO 13849-1: Safety of Machinery – Safety-Related Parts of Control Systems – Part 1: General Principles for Design*. 4. kiad.

ISO. 2024. *ISO/PAS 8800: Road Vehicles – Safety and Artificial Intelligence*.

ISO. 2025a. *ISO 10218-1/-2: Robotics – Safety Requirements*.

ISO. 2025b. *ISO/TS 5083: Road Vehicles – Safety for Automated Driving Systems – Design, Verification and Validation*.

ISO/IEC. 2014. *ISO/IEC Guide 51: Safety Aspects – Guidelines for Their Inclusion in Standards*. 3. kiad.

ISO/IEC. 2024. *ISO/IEC TR 5469: Artificial Intelligence – Functional Safety and AI Systems*.

ISO/SAE. 2021. *ISO/SAE 21434: Road Vehicles – Cybersecurity Engineering*.

Joint Authorities Technical Review. 2019. *Boeing 737 MAX Flight Control System: Observations, Findings, and Recommendations*. Federal Aviation Administration.

Kalra, Nidhi, és Susan M. Paddock. 2016. „Driving to Safety: How Many Miles of Driving Would It Take to Demonstrate Autonomous Vehicle Reliability?" *Transportation Research Part A: Policy and Practice* 94: 182–93. https://doi.org/10.1016/j.tra.2016.09.010.

Kaplan, Edward L., és Paul Meier. 1958. „Nonparametric Estimation from Incomplete Observations". *Journal of the American Statistical Association* 53 (282): 457–81. https://doi.org/10.1080/01621459.1958.10501452.

Kelly, Timothy P. 1998. „Arguing Safety – A Systematic Approach to Managing Safety Cases". Phdthesis, University of York.

Kelly, Tim, és Rob Weaver. 2004. „The Goal Structuring Notation – A Safety Argument Notation". *Proceedings of the Dependable Systems and Networks 2004 Workshop on Assurance Cases*.

Kirwan, Barry. 1994. *A Guide to Practical Human Reliability Assessment*. Taylor & Francis.

Klein, Gerwin, Kevin Elphinstone, Gernot Heiser, és mtsai. 2009. „seL4: Formal Verification of an OS Kernel". *Proceedings of the 22nd ACM Symposium on Operating Systems Principles (SOSP)*, 207–20. https://doi.org/10.1145/1629575.1629596.

Kletz, Trevor, és Paul Amyotte. 2010. *Process Plants: A Handbook for Inherently Safer Design*. 2. kiad. CRC Press.

Knight, John C., és Nancy G. Leveson. 1986. „An Experimental Evaluation of the Assumption of Independence in Multiversion Programming". *IEEE Transactions on Software Engineering* SE-12 (1): 96–109.

Koopman, Philip, és Michael Wagner. 2016. „Challenges in Autonomous Vehicle Testing and Validation". *SAE International Journal of Transportation Safety* 4 (1): 15–24. https://doi.org/10.4271/2016-01-0128.

Lamport, Leslie, Robert Shostak, és Marshall Pease. 1982. „The Byzantine Generals Problem". *ACM Transactions on Programming Languages and Systems* 4 (3): 382–401. https://doi.org/10.1145/357172.357176.

Leveson, Nancy G. 2011. *Engineering a Safer World: Systems Thinking Applied to Safety*. MIT Press. https://doi.org/10.7551/mitpress/8179.001.0001.

Leveson, Nancy G., és John P. Thomas. 2018. *STPA Handbook*. MIT Partnership for Systems Approaches to Safety; Security.

Leveson, Nancy G., és Clark S. Turner. 1993. „An Investigation of the Therac-25 Accidents". *Computer* 26 (7): 18–41. https://doi.org/10.1109/MC.1993.274940.

Lions, Jacques-Louis. 1996. *ARIANE 5 Flight 501 Failure: Report by the Inquiry Board*. European Space Agency.

Littlewood, Bev, és Lorenzo Strigini. 1993. „Validation of Ultrahigh Dependability for Software-Based Systems". *Communications of the ACM* 36 (11): 69–80. https://doi.org/10.1145/163359.163373.

Lundteigen, Mary Ann, és Marvin Rausand. 2009. „Architectural Constraints in IEC 61508: Do They Have the Intended Effect?" *Reliability Engineering & System Safety* 94 (2): 520–25. https://doi.org/10.1016/j.ress.2008.06.003.

Lutz, Robyn R. 1993. „Analyzing Software Requirements Errors in Safety-Critical, Embedded Systems". *Proceedings of the IEEE International Symposium on Requirements Engineering*, 126–33.

Meeker, William Q., és Luis A. Escobar. 1998. *Statistical Methods for Reliability Data*. Wiley.

MISRA. 2025. *MISRA C:2025 – Guidelines for the Use of the C Language in Critical Systems*. The MISRA Consortium.

Modarres, Mohammad, Mark P. Kaminskiy, és Vasiliy Krivtsov. 2016. *Reliability Engineering and Risk Analysis: A Practical Guide*. 3. kiad. CRC Press.

Mosleh, Ali, Karl N. Fleming, Gareth W. Parry, Henrique M. Paula, David H. Worledge, és Dale M. Rasmuson. 1988. *Procedures for Treating Common Cause Failures in Safety and Reliability Studies*. NUREG/CR-4780. U.S. Nuclear Regulatory Commission.

National Transportation Safety Board. 2019. *Collision Between Vehicle Controlled by Developmental Automated Driving System and Pedestrian, Tempe, Arizona, March 18, 2018*. NTSB/HAR-19/03. NTSB.

Newcombe, Chris, Tim Rath, Fan Zhang, Bogdan Munteanu, Marc Brooker, és Michael Deardeuff. 2015. „How Amazon Web Services Uses Formal Methods". *Communications of the ACM* 58 (4): 66–73. https://doi.org/10.1145/2699417.

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* 30 (3): 286–97. https://doi.org/10.1109/3468.844354.

Perrow, Charles. 1984. *Normal Accidents: Living with High-Risk Technologies*. Basic Books.

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.

Rasmussen, Jens. 1997. „Risk Management in a Dynamic Society: A Modelling Problem”. *Safety Science* 27 (2–3): 183–213. https://doi.org/10.1016/S0925-7535(97)00052-0.

Rausand, Marvin. 2014. *Reliability of Safety-Critical Systems: Theory and Applications*. Wiley. https://doi.org/10.1002/9781118776353.

Rausand, Marvin, Anne Barros, és Arnljot Høyland. 2020. *System Reliability Theory: Models, Statistical Methods, and Applications*. 3. kiad. Wiley. https://doi.org/10.1002/9781119373940.

Rauzy, Antoine. 1993. „New Algorithms for Fault Trees Analysis”. *Reliability Engineering & System Safety* 40 (3): 203–11. https://doi.org/10.1016/0951-8320(93)90060-C.

Reason, James. 1990. *Human Error*. Cambridge University Press.

Reason, James. 1997. *Managing the Risks of Organizational Accidents*. Ashgate.

RTCA. 2011a. *DO-178C: Software Considerations in Airborne Systems and Equipment Certification*.

RTCA. 2011b. *DO-333: Formal Methods Supplement to DO-178C and DO-278A*.

Ruijters, Enno, és Mariëlle Stoelinga. 2015. „Fault Tree Analysis: A Survey of the State-of-the-Art in Modeling, Analysis and Tools”. *Computer Science Review* 15–16: 29–62. https://doi.org/10.1016/j.cosrev.2015.03.001.

SAE International. 2021. *J3016: Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles*.

SAE International. 2023a. *ARP4754B: Guidelines for Development of Civil Aircraft and Systems*. https://doi.org/10.4271/ARP4754B.

SAE International. 2023b. *ARP4761A: Guidelines for Conducting the Safety Assessment Process on Civil Aircraft, Systems, and Equipment*.

Salay, Rick, Rodrigo Queiroz, és Krzysztof Czarnecki. 2017. *An Analysis of ISO 26262: Using Machine Learning Safely in Automotive Software*. arXiv:1709.02435.

Sha, Lui. 2001. „Using Simplicity to Control Complexity”. *IEEE Software* 18 (4): 20–28. https://doi.org/10.1109/MS.2001.936213.

Smith, David J., és Kenneth G. L. Simpson. 2020. *The Safety Critical Systems Handbook: A Straightforward Guide to Functional Safety*. 5. kiad. Butterworth-Heinemann.

Stamatelatos, Michael, William Vesely, Joanne Dugan, Joseph Fragola, Joseph Minarick, és Jan Railsback. 2002. *Fault Tree Handbook with Aerospace Applications*. Version 1.1. NASA Office of Safety; Mission Assurance.

Swain, Alan D., és Henry E. Guttmann. 1983. *Handbook of Human Reliability Analysis with Emphasis on Nuclear Power Plant Applications*. NUREG/CR-1278. U.S. Nuclear Regulatory Commission.

Szegedy, Christian, Wojciech Zaremba, Ilya Sutskever, és mtsai. 2014. „Intriguing Properties of Neural Networks”. *International Conference on Learning Representations (ICLR)*.

Traverse, Pascal, Isabelle Lacaze, és Jean Souyris. 2004. „Airbus Fly-by-Wire: A Total Approach to Dependability”. In *Building the Information Society (IFIP 18th World Computer Congress)*. Springer.

Trivedi, Kishor S., és Andrea Bobbio. 2017. *Reliability and Availability Engineering: Modeling, Analysis, and Applications*. Cambridge University Press. https://doi.org/10.1017/9781316163047.

UL Standards & Engagement. 2023. *UL 4600: Standard for Evaluation of Autonomous Products*. 3. kiad.

UNECE. 2016. *UN Regulation No. 13: Uniform Provisions Concerning the Approval of Vehicles of Categories M, N and O with Regard to Braking*.

UNECE. 2021a. *UN Regulation No. 155: Cyber Security and Cyber Security Management System*.

UNECE. 2021b. *UN Regulation No. 157: Automated Lane Keeping Systems (ALKS)*.

U.S. Chemical Safety and Hazard Investigation Board. 2007. *Investigation Report: Refinery Explosion and Fire, BP Texas City, Texas, March 23, 2005*. 2005-04-I-TX. CSB.

U.S. Department of Defense. 2012. *MIL-STD-882E: Standard Practice – System Safety*.

U.S. Nuclear Regulatory Commission. 1975. *Reactor Safety Study: An Assessment of Accident Risks in U.S. Commercial Nuclear Power Plants*. WASH-1400 (NUREG-75/014). U.S. Nuclear Regulatory Commission.

Vesely, William E., Francine F. Goldberg, Norman H. Roberts, és David F. Haasl. 1981. *Fault Tree Handbook*. NUREG-0492. U.S. Nuclear Regulatory Commission.

Weibull, Waloddi. 1951. „A Statistical Distribution Function of Wide Applicability”. *Journal of Applied Mechanics* 18: 293–97.

Williams, Jeremy C. 1988. „A Data-Based Method for Assessing and Reducing Human Error to Improve Operational Performance”. *Proceedings of the IEEE Fourth Conference on Human Factors and Power Plants* (Monterey, CA), 436–50.

Yeh, Y. C. 1996. „Triple-Triple Redundant 777 Primary Flight Computer”. *Proceedings of the IEEE Aerospace Applications Conference* 1: 293–307.

Young, William, és Nancy G. Leveson. 2014. „An Integrated Approach to Safety and Security Based on Systems Theory”. *Communications of the ACM* 57 (2): 31–35. https://doi.org/10.1145/2556938.

A.

Valószínűségszámítási összefoglaló

Ez a függelék a könyvben használt valószínűségszámítási fogalmakat és összefüggéseket foglalja össze. Nem helyettesít egy tankönyvet, hanem a fejezetekben felhasznált eszközök gyors felidézését szolgálja; a részletes tárgyalás például Rausand, Barros és Høyland (Rausand és mtsai. 2020), valamint Modarres, Kaminskiy és Krivtsov (Modarres és mtsai. 2016) könyvében található.

A.1. Események és valószínűség

Egy véletlen kísérlet lehetséges kimeneteinek halmaza az **eseménytér** Ω, ennek részhalmazai az **események**. A P valószínűség Kolmogorov axiómái szerint: $P(A) \geq 0$; $P(\Omega) = 1$; egymást kizáró eseményekre $P(A \cup B) = P(A) + P(B)$. Ebből következik a komplementer szabálya, $P(\bar{A}) = 1 - P(A)$, és két tetszőleges eseményre

$$P(A \cup B) = P(A) + P(B) - P(A \cap B).$$

Több eseményre a **szita-formula** (inklúzió–exklúzió) általánosítja ezt; a hibafa csúcseseményének pontos valószínűsége ezzel számolható a minimális vágatokból (10.4.2. szakasz). Ha az események valószínűsége kicsi, a metszetek elhanyagolhatók, és $P(\bigcup A_i) \approx \sum P(A_i)$: ez a **ritkaesemény-közelítés**, amely koherens rendszerekre mindig felső korlát.

A **feltételes valószínűség** $P(A \mid B) = P(A \cap B)/P(B)$, ha $P(B) > 0$. Két esemény **független**, ha $P(A \cap B) = P(A)\,P(B)$, azaz $P(A \mid B) = P(A)$. A függetlenség és a kizárólagosság két különböző fogalom: két egymást kizáró, pozitív valószínűségű esemény sosem független. A redundáns rendszerek számításában a függetlenség feltevése a legkritikusabb; a közös okú meghibásodások éppen ennek sérülését írják le (10.6. szakasz).

Ha $B_1, \dots, B_n$ az eseménytér egy felosztása, a **teljes valószínűség tétele** szerint $P(A) = \sum_i P(A \mid B_i)\,P(B_i)$, és a **Bayes-tétel** szerint

$$P(B_j \mid A) = \frac{P(A \mid B_j)\,P(B_j)}{\sum_i P(A \mid B_i)\,P(B_i)}.$$

Mit jelent egy riasztás?

Egy berendezésnél a veszélyes hiba előfordulási valószínűsége egy adott ellenőrzéskor 1%. A diagnosztika a valódi hibát 99%-os valószínűséggel jelzi, de hibátlan állapotban is 5%-os valószínűséggel ad téves riasztást. Ha riasztás van, mekkora a valódi hiba valószínűsége? A Bayes-tétel szerint $0{,}01 \cdot 0{,}99 / (0{,}01 \cdot 0{,}99 + 0{,}99 \cdot$

$0{,}05) = 0{,}167$. A riasztások hatoda valódi: egy ritka hibánál még egy jó diagnosztika is többségében téves riasztást ad. Ez a jelenség a riasztáskezelés és a kezelői bizalom egyik alapproblémája (16.4. szakasz).

A.2. Valószínűségi változók és eloszlások

Egy X valószínűségi változó **eloszlásfüggvénye** $F(x) = P(X \leq x)$; folytonos esetben a **sűrűségfüggvény** $f(x) = F'(x)$. A **várható érték** $E[X] = \int x f(x)\,dx$, a **szórásnégyzet** $\mathrm{Var}(X) = E[(X - E[X])^2]$. Ha X egy elem élettartama, akkor $F(t)$ a meghibásodási valószínűség, $R(t) = 1 - F(t)$ a megbízhatóság, és $\mathrm{MTTF} = E[X] = \int_0^\infty R(t)\,dt$ (8. fejezet). A könyvben használt fő eloszlásokat a A.1. táblázat foglalja össze.

A.1. táblázat. A könyvben használt fő eloszlások

Eloszlás	Sűrűség- vagy valószínűségfüggvény	Várható érték	Alkalmazás a könyvben
binomiális (n, p)	$\binom{n}{k} p^k (1-p)^{n-k}$	np	k-ból-n rendszerek
Poisson (μ)	$\mu^k e^{-\mu}/k!$	μ	meghibásodások száma adott idő alatt
exponenciális (λ)	$\lambda e^{-\lambda t}$	$1/\lambda$	állandó meghibásodási ráta, Markov-modellek
Weibull (β, η)	$\frac{\beta}{\eta}\left(\frac{t}{\eta}\right)^{\beta-1} e^{-(t/\eta)^\beta}$	$\eta\,\Gamma(1 + 1/\beta)$	korai, véletlen és elhasználódásos meghibásodások
normális (μ, σ)	$\frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/(2\sigma^2)}$	μ	mérési hibák, közelítések
lognormális (μ, σ)	$\frac{1}{x\sigma\sqrt{2\pi}} e^{-(\ln x - \mu)^2/(2\sigma^2)}$	$e^{\mu + \sigma^2/2}$	bizonytalan paraméterek, javítási idők
gamma (a, b)	$\frac{b^a}{\Gamma(a)} x^{a-1} e^{-bx}$	a/b	a meghibásodási ráta bayesi becslése
$\chi^2\,(\nu)$	a gamma $(\nu/2, 1/2)$ speciális esete	ν	a ráta konfidenciakorlátai

A **lognormális eloszlás** a valószínűségi biztonsági elemzések szokásos modellje a bizonytalan paraméterekre. Ha $\ln X$ normális eloszlású μ várható értékkel és σ szórással, akkor X mediánja e^{μ}, és a **hibatényező** $\mathrm{EF} = e^{1{,}645\sigma}$ a 95%-os és az 50%-os kvantilis hányadosa (Modarres és mtsai. 2016). Például $\mathrm{EF} = 3$ esetén $\sigma = 0{,}668$, és a várható érték a medián 1,25-szöröse; ezért a 10.7. szakasz Monte Carlo-szimulációjában a csúcsesemény várható értéke nagyobb volt a pontbecslésnél.

A.3. A Poisson-folyamat

Ha a meghibásodások egymástól függetlenül, állandó λ intenzitással következnek be (és a meghibásodott elemet azonnal kicseréljük), a $(0, t]$ időszakban bekövetkező meghibásodások száma λt paraméterű Poisson-eloszlású:

$$P(N(t)=k)=\frac{(\lambda t)^k}{k!}\,e^{-\lambda t},\qquad k=0,1,2,\dots$$

Az első meghibásodásig eltelt idő exponenciális eloszlású, és a folyamat **emlékezet nélküli**: $P(T>t+s\mid T>t)=P(T>s)$. Egy már t ideje működő elem további élettartama tehát ugyanolyan eloszlású, mint egy újé. Ez a tulajdonság teszi lehetővé a Markov-modellek egyszerű felírását (12. fejezet), de egyben a korlátja is: öregedő elemekre nem érvényes.

Kis λt esetén $P(N(t)\ge 1)=1-e^{-\lambda t}\approx\lambda t$, és két vagy több meghibásodás valószínűsége elhanyagolható. Például $\lambda t=0{,}1$ mellett $P(N\ge1)=0{,}095$, $P(N=2)=0{,}0045$.

A.4. A meghibásodási ráta becslése

Ha T összes üzemidő alatt r meghibásodást figyeltünk meg (időcenzúrázott adat), a ráta maximum likelihood becslése $\hat\lambda=r/T$, és az egyoldali $1-\alpha$ szintű felső konfidenciakorlát

$$\lambda_{1-\alpha}=\frac{\chi^2_{1-\alpha}(2r+2)}{2T},$$

ahol $\chi^2_{1-\alpha}(\nu)$ a ν szabadsági fokú χ^2-eloszlás $1-\alpha$ kvantilise (8. fejezet). Hibamentes tapasztalat ($r=0$) esetén ebből $\lambda_{1-\alpha}=-\ln\alpha/T$; ez a 18.1.1. szakasz becslésének alapja.

A **bayesi becslés** az előzetes tudást (például iparági adatbázisok értékét) kombinálja a saját megfigyelésekkel. Ha a ráta előzetes eloszlása gamma (a,b), és T idő alatt r meghibásodás történt, az utólagos eloszlás gamma $(a+r,b+T)$, várható értéke $(a+r)/(b+T)$.

> **Bayesi frissítés**
>
> A 8. fejezet távadóiról 250 darab négy éven át gyűjtött adata szerint $T=8{,}76\cdot10^6$ óra alatt $r=3$ meghibásodás történt ($\hat\lambda=3{,}42\cdot10^{-7}\,\mathrm{h}^{-1}$). Tegyük fel, hogy egy iparági adatbázis alapján az előzetes várakozás $5\cdot10^{-7}\,\mathrm{h}^{-1}$, amelyet egy gyengén informatív, $a=2$, $b=4\cdot10^6$ óra paraméterű gamma-eloszlással írunk le. Az utólagos eloszlás gamma $(5;\ 1{,}276\cdot10^7\,\mathrm{h})$, várható értéke $3{,}92\cdot10^{-7}\,\mathrm{h}^{-1}$, 90%-os felső korlátja $6{,}26\cdot10^{-7}\,\mathrm{h}^{-1}$. A saját adatok a becslést az iparági értéknél kedvezőbb irányba mozdították, de nem annyira, mint a puszta maximum likelihood becslés, mert az előzetes tudás is számít.

A.5. Hasznos közelítések

A biztonsági számításokban gyakran használt közelítések, amelyek kis valószínűségeknél ($x\ll1$) pontosak:

- $1-e^{-x}\approx x$, így $F(t)\approx\lambda t$, ha $\lambda t\ll1$;
- $1-\prod_i(1-q_i)\approx\sum_i q_i$ (a VAGY-kapu és a soros rendszer ritkaesemény-közelítése);
- egy periodikusan, T_1 időközönként tesztelt elem átlagos meghibásodási valószínűsége $\frac{1}{T_1}\int_0^{T_1}\lambda t\,dt=\lambda T_1/2$;
- két egyszerre tesztelt elem együttes meghibásodási valószínűségének időátlaga $\frac{1}{T_1}\int_0^{T_1}(\lambda t)^2\,dt=\lambda^2T_1^2/3$, ami az egyedi átlagok szorzatának 4/3-szorosa (10. fejezet).

Mindegyik közelítésnél érdemes ellenőrizni, hogy a feltétel ($\lambda t \ll 1$, kis q_i) valóban teljesül-e; például egy évtizedes nagyjavítási időközzel számolt, rejtett meghibásodásoknál ($\lambda T_M \approx 0{,}2$) a lineáris közelítés már néhány százalékos hibát okoz (12.5.1. szakasz).

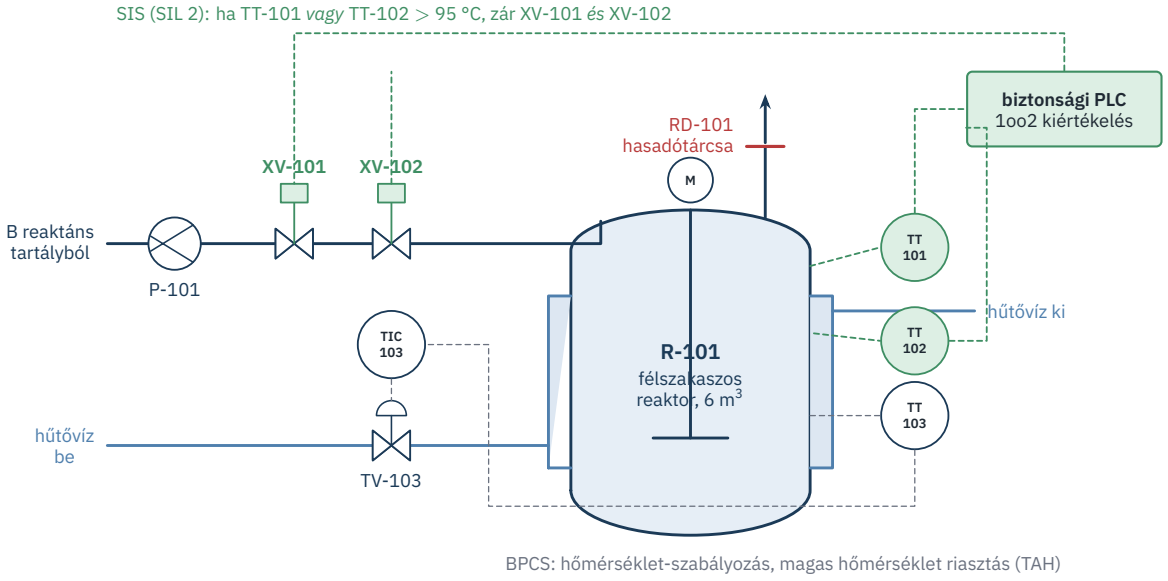
B.

A három esettanulmány

A könyv módszereit három kitalált, de valós mérnöki problémákat modellező esettanulmány köti össze. Ez a függelék összefoglalja a leírásukat, a számításokban használt adatokat és a fejezetekben kapott eredményeket. A számértékek a szakirodalomban szokásos nagyságrendekben mozognak; a ReaktorVéd minden számát egyetlen Python-szkript állítja elő, amely a könyv forrásában (`_shared/esetek/`) és a gyakorlati útmutató notebookjaiban is megtalálható, így minden eredmény reprodukálható.

B.1. A – ReaktorVéd: exoterm reaktor védelmi rendszere

A folyamat. Félszakaszos üzemű keverős reaktor, amelyben exoterm reakció zajlik. A B reaktánst a P-101 szivattyú adagolja két, sorba kötött gyorszáró szelepen (XV-101, XV-102) át. A reakcióhőt a köpenyben áramló hűtővíz vonja el; a hőmérsékletet a TIC-103 szabályozó tartja az alapjelen. A reaktoron hasadótárcsa (RD-101) is van, amelynek kétfázisú áramlásra való méretezése az elemzés idején nem igazolt (5. fejezet).



B.1. ábra. A ReaktorVéd egyszerűsített folyamatábrája.

A veszélyes forgatókönyv. A hűtővíz kiesése (évi 0,1 gyakorisággal) exoterm fázisban (az esetek felében) megfutó reakcióhoz, a reaktor felhasadásához, mérgező kibocsátáshoz és halálos sérüléshez vezethet. A vállalati kritérium szerint egy halálos kimenetelű forgatókönyv tolerálható gyakorisága 10^{-5}/év.

A biztonsági funkció. Két hőmérséklet-távadó (TT-101/102, 1oo2) méri a reaktor hőmérsékletét; 95 °C fölött a tanúsított biztonsági PLC mindkét gyorszáró szelepet zárja (1oo2 a „zárás” funkcióra), 10 másodpercen belül. A végleges kialakítás hibabiztos: a szelepek energiamentesítésre zárnak, a távadók élő nullás áramhurokkal csatlakoznak, a tápellátás redundáns.

B.1. táblázat. A ReaktorVéd fő adatai

Paraméter	Érték	Fejezet
kezdő esemény (hűtővíz-kiesés)	$0{,}1$/év	3., 11.
feltételt teremtő tényező (exoterm fázis)	$0{,}5$	11.
riasztás és kezelői beavatkozás (IPL)	PFD $= 0{,}1$ (SPAR-H szerint $\approx 0{,}04$)	11., 16.
tolerálható gyakoriság	10^{-5}/év	3.
a SIF szükséges PFD-je	$\leq 2 \cdot 10^{-3}$ (SIL 2, RRF ≥ 500)	11.
távadók	$\lambda_{DU} = 4 \cdot 10^{-7}$, $\lambda_{DD} = 6 \cdot 10^{-7}$, $\lambda_S = 5 \cdot 10^{-7}\,\mathrm{h}^{-1}$; $\beta = 5\%$, $\beta_D = 2{,}5\%$	13.
biztonsági PLC	PFD $= 5 \cdot 10^{-5}$, SIL 3-ra tanúsított	13.
gyorszáró szelepek	$\lambda_{DU} = 2{,}5 \cdot 10^{-6}\,\mathrm{h}^{-1}$, $\beta = 10\%$	13.
proof teszt időköze, javítási idő	$T_1 = 8760$ h, MTTR $= 8$ h	13.
tápegység (energiával működtetett változat)	igénykori kiesés 10^{-3}	10.

Az eredmények áttekintése. A B.2. táblázat megmutatja, hogyan finomodott a biztonsági funkció értékelése a könyv során. A sorozat maga is tanulság: a naiv blokkdiagram nagyságrenddel optimistább a helyes eredménynél, az első tervváltozat rejtett egypontos hibákat tartalmazott, és a próbák minősége döntően befolyásolja az eredményt.

B.2. táblázat. A ReaktorVéd biztonsági funkciójának értékelése a könyv fejezeteiben

Modell, változat	Eredmény	Fejezet
megbízhatósági blokkdiagram (független elemek)	PFD $\approx 1{,}7 \cdot 10^{-4}$	9.
hibafa, energiával működtetett szelepek, közös tápegységek	PFD $= 2{,}17 \cdot 10^{-3}$	10.
hibafa, hibabiztos kialakítás, közös okok nélkül	PFD $= 1{,}73 \cdot 10^{-4}$	10.
hibafa, hibabiztos kialakítás, β-tényezős közös okokkal	PFD $= 1{,}335 \cdot 10^{-3}$	10.

Modell, változat	Eredmény	Fejezet
többfázisú Markov-modell (csak a szelepek)	PFD $= 1{,}22 \cdot 10^{-3}$	12.
IEC 61508-6 képletek (a végleges igazolás)	PFD $= 1{,}369 \cdot 10^{-3}$, RRF $= 731$	13.
90%-os proof teszt lefedettséggel	PFD $= 2{,}69 \cdot 10^{-3}$ (nem teljesül)	13.
havi részleges löketpróbával	PFD $= 6{,}63 \cdot 10^{-4}$, RRF $= 1508$	13.
reaktorfelhasadás gyakorisága (eseményfa)	$6{,}8 \cdot 10^{-6}$/év	11.
egy kezelő egyéni kockázata	$1{,}5 \cdot 10^{-6}$/év	3.
ebből téves leállások (egyszeres → redundáns táp → 2oo3 távadók)	$0{,}133 \rightarrow 0{,}046 \rightarrow 0{,}026$/év	13.

B.2. B – VészFék: automatikus vészfékező asszisztens

A funkció. Személygépkocsi vészfékező asszisztense (AEB), amely a radar és a kamera adatai alapján felismeri a jármű előtti akadályokat (járművek, gyalogosok, kerékpárosok), figyelmezteti a vezetőt, és ha az nem reagál, önállóan fékez. A rendszer elemei: frontradar (77 GHz), frontkamera, érzékelőfúziós és döntési vezérlőegység, valamint az elektrohidraulikus fékaktuátor és a stabilitásvezérlő, amely a fékezést végrehajtja. A funkció a vezető kiegészítője (SAE J3016 szerint legfeljebb 2. szint), ezért hibára elnémuló kialakítású: hiba esetén kikapcsol és jelez (14. fejezet).

A két veszély. A funkció kétféleképpen okozhat kárt: indokolatlan erős fékezéssel (ráfutásos baleset) és a fékezés elmaradásával (ütközés). A HARA szerint az indokolatlan teljes erejű fékezés autópályán, követő jármű jelenlétében S3, E4, C3, azaz **ASIL D** (4. fejezet); a biztonsági cél az, hogy a funkció ne kezdeményezhessen indokolatlan, egy határértéknél nagyobb lassulású fékezést 60 km/h fölött.

A fejezetekben. A VészFék a balesetek tanulságainál (a tempei baleset, 1. fejezet), a funkcionális veszélyelemzésnél (5. fejezet), a radarérzékelő FMEA-jánál (6. fejezet), az STPA nem biztonságos irányítási műveleteinél (7. fejezet), a két csúcseseményű hibafánál (10. fejezet), a hardvermetrikáknál (13. fejezet), a hibaviselkedési stratégiáknál (14. fejezet), a szoftver és a gépi tanulás határán (15. fejezet), az ember mint tartalék kérdésénél (16. fejezet), a kétágú biztonsági érvelésnél (17. fejezet) és a SOTIF kiváltó feltételeinél (18. fejezet) jelenik meg.

B.3. C – KoboCella: kollaboratív robotcella

A cella. Egy szerelőcella, amelyben egy könnyű, kollaboratív robotkar és egy szerelő közös munkatérben dolgozik. A fő veszélyek a robotkar mozgási energiája (ütközés) és a megfogó szorítóereje (becsípődés).

A biztonsági funkciók. Sebesség- és távolságfelügyelet: ha a lézerszkenner a robot 1,2 m-es körzetében személyt érzékel, a robot sebessége legfeljebb 250 mm/s; 0,5 m-en belül a robot biztonságosan megáll. Erő- és teljesítménykorlátozás az ISO/TS 15066 biomechanikai határértékei szerint. Hiba esetén a hajtások biztonságos nyomatékmentesítése (STO).

B.3. táblázat. A KoboCella fő adatai

Jellemző	Érték	Fejezet
szükséges teljesítményszint a sebességfelügyeletre	$\mathrm{PL_r}$ = e (S2, F2, P2); inherens tervezéssel (S1) $\mathrm{PL_r}$ = c lehetne	4.
architektúra	4-es kategória, kétcsatornás (1oo2), kölcsönös diagnosztika	9., 14.
a biztonságos megállítás PFH-ja	$1{,}01 \cdot 10^{-8}\,\mathrm{h}^{-1}$ (PL e)	13.
a robothajtás csapágyainak Weibull-paraméterei	$\beta = 2{,}73$, $\eta = 10\,170$ h, MTTF $\approx$ 9050 h	8.
rendelkezésre állás gyakori biztonsági megállásokkal	0,968 (óránként 6 belépés, 20 s megállás)	12.

A fejezetekben. A KoboCella a háromlépcsős kockázatcsökkentésnél (3. fejezet), az előzetes veszélyelemzésnél (5. fejezet), a $\mathrm{PL_r}$ meghatározásánál (4. fejezet), az STPA részletes példájában (7. fejezet), a Weibull-elemzésnél (8. fejezet), a nagy igénybevételű eseményfánál (11. fejezet), a biztonság és a rendelkezésre állás kapcsolatánál (12. fejezet), a PFH-számításnál (13. fejezet), az ember–robot együttműködés emberi tényezőinél (16. fejezet), a műszaki dokumentációnál (17. fejezet) és a gépi tanulásos személyfelismerés szabályozási kérdéseinél (18. fejezet) szerepel.

C.

Magyar–angol szószedet

A könyv a magyar szakszavakat használja, és első előfordulásukkor megadja az angol megfelelőjüket. Ez a szószedet a fontosabb fogalmakat gyűjti össze ábécérendben. Ahol a szakirodalomban az angol rövidítés terjedt el, azt is feltüntetjük.

C.1. táblázat. Magyar–angol szószedet

Magyar	Angol	Rövidítés
alacsony igénybevételű üzemmód	low demand mode	
alapesemény	basic event	
alapfolyamat-irányító rendszer	basic process control system	BPCS
átlagos javítási (helyreállítási) idő	mean time to restoration	MTTR
átlagos meghibásodásig eltelt idő	mean time to failure	MTTF
automatizálási torzítás	automation bias	
bináris döntési diagram	binary decision diagram	BDD
biztonsági érvelés	safety case	
biztonsági funkció	safety function	
biztonsági integritási szint	safety integrity level	SIL
biztonsági követelményspecifikáció	safety requirements specification	SRS
biztonsági műszeres funkció	safety instrumented function	SIF
biztonsági műszeres rendszer	safety instrumented system	SIS
biztonsági teljesítménymutató	safety performance indicator	SPI
biztonságos meghibásodási hányad	safe failure fraction	SFF
célstruktúra-jelölés	goal structuring notation	GSN
csokornyakkendő-diagram	bow-tie diagram	
csúcsesemény	top event	
diagnosztikai lefedettség	diagnostic coverage	DC
eloszláseltolódás	distributional shift	
emberi hibavalószínűség	human error probability	HEP
emberi megbízhatóság elemzése	human reliability analysis	HRA
energia–gát modell	energy–barrier model	
eseményfa-elemzés	event tree analysis	ETA
eszkalációs tényező	escalation factor	

Magyar	Angol	Rövidítés
fejlesztési biztosítottsági szint	development assurance level	DAL
felfedett veszélyes meghibásodás	dangerous detected failure	DD
feltételes módosító	conditional modifier	
feltételt teremtő tényező	enabling condition	
folyamatbiztonsági idő	process safety time	
funkcionális biztonság	functional safety	
funkcionális biztonsági értékelés	functional safety assessment	FSA
funkcionális veszélyelemzés	functional hazard assessment	FHA
független védelmi réteg	independent protection layer	IPL
gépi tanulás	machine learning	ML
hardveres hibatűrés	hardware fault tolerance	HFT
helyzetismeret	situation awareness	
hibabiztos	fail-safe	
hibafa-elemzés	fault tree analysis	FTA
hibamód-, hatás- és diagnosztikai elemzés	failure modes, effects and diagnostic analysis	FMEDA
hibamód- és hatáselemzés	failure mode and effects analysis	FMEA
hibára elnémuló	fail-silent	
hibatényező	error factor	EF
hibatűrési időkeret	fault tolerant time interval	FTTI
hibatűrő (működőképes)	fail-operational	
igénykori meghibásodás átlagos valószínűsége	average probability of dangerous failure on demand	$\mathrm{PFD}_{\mathrm{avg}}$
inherens biztonság	inherent safety	
intézkedési prioritás	action priority	AP
irányítási struktúra	control structure	
k-ból-n rendszer	k-out-of-n system	koon
kezdő esemény	initiating event	IE
kiváltó feltétel	triggering condition	
kockázat	risk	
kockázatcsökkentési érték	risk reduction worth	RRW
kockázatcsökkentési tényező	risk reduction factor	RRF
kockázati gráf	risk graph	
kockázati mátrix	risk matrix	
kockázatnövelési érték	risk achievement worth	RAW
korábbi használat	prior use	
közös okú meghibásodás	common cause failure	CCF
látens feltétel	latent condition	
megbízhatóság	reliability	
megbízhatósági blokkdiagram	reliability block diagram	RBD
megerősítő intézkedések	confirmation measures	
meghibásodási ráta	failure rate, hazard rate	λ, $h(t)$
minimális vágat	minimal cut set	MCS
módosított feltétel/döntés lefedettség	modified condition/decision coverage	MC/DC
működési tervezési tartomány	operational design domain	ODD
nagy igénybevételű üzemmód	high demand mode	

Magyar	Angol	Rövidítés
nem biztonságos irányítási művelet	unsafe control action	UCA
particionálás	partitioning	
proof teszt	proof test	
proof teszt lefedettsége	proof test coverage	PTC
rejtett veszélyes meghibásodás	dangerous undetected failure	DU
rendelkezésre állás	availability	
rendeltetésszerű funkció biztonsága	safety of the intended functionality	SOTIF
rendszerelméleti baleseti modell	systems-theoretic accident model and processes	STAMP
rendszerelméleti folyamatelemzés	system-theoretic process analysis	STPA
részleges löketpróba	partial stroke testing	PST
rétegzett védelem elemzése	layer of protection analysis	LOPA
riasztásáradat	alarm flood	
ritkaesemény-közelítés	rare event approximation	
sajtszelet-modell	Swiss cheese model	
sikerfa	success tree	
struktúrafüggvény	structure function	
szabályszegés	violation	
szisztematikus képesség	systematic capability	SC
szisztematikus meghibásodás	systematic failure	
teljesítményszint	performance level	PL
téves leállás	spurious trip	
téves leállási gyakoriság	spurious trip rate	STR
tolerálható kockázat	tolerable risk	
üzemi tapasztalattal igazolt	proven in use	
vágat	cut set	
véletlenszerű hardvermeghibásodás	random hardware failure	
veszély	hazard	
veszély- és működőképesség-vizsgálat	hazard and operability study	HAZOP
veszélyelemzés és kockázatbecslés	hazard analysis and risk assessment	HARA
veszélyes esemény	hazardous event	
veszélyes helyzet	hazardous situation	
veszélyes meghibásodás átlagos gyakorisága	average frequency of dangerous failure per hour	PFH
zóna és csatorna	zone and conduit	

D.

Szabványok és jogszabályok

Ez a függelék a könyvben hivatkozott szabványokat, irányelveket és jogszabályokat foglalja össze területenként, a könyv lezárásakor (2026 novemberében) érvényes kiadással. A szabványokat rendszeresen felülvizsgálják; alkalmazásuk előtt mindig a legfrissebb kiadást kell ellenőrizni. Ahol új kiadás készül, azt a megjegyzés jelzi.

D.1. Általános és módszertani szabványok

D.1. táblázat. Általános és módszertani szabványok

Szabvány	Tárgy	Fejezet
IEC 61508-1…7:2010 (IEC 2010a)	az E/E/PE biztonsági rendszerek funkcionális biztonsága (alapszabvány; a 3. kiadás 2027-re várható)	4., 13., 15.
IEC 61511-1…3:2016 (IEC 2016b)	biztonsági műszeres rendszerek a folyamatiparban	4., 11., 13.
ISO 31000:2018 (ISO 2018b)	kockázatmenedzsment, irányelvek	3.
IEC 31010:2019 (IEC 2019a)	kockázatfelmérési technikák	3.
ISO/IEC Guide 51:2014 (ISO/IEC 2014)	a biztonsági szempontok szabványokba foglalása	2.
IEC 60050-192:2015 (IEC 2015)	nemzetközi elektrotechnikai szótár: megbízhatóság	2., 12.
IEC 61882:2016 (IEC 2016c)	HAZOP-vizsgálat	5.
IEC 60812:2018 (IEC 2018)	FMEA és FMECA	6.
AIAG–VDA FMEA kézikönyv, 2019 (AIAG és VDA 2019)	az autóipari FMEA módszertana, intézkedési prioritás	6.
IEC 61025:2006 (IEC 2006a)	hibafa-elemzés	10.
IEC 62502:2010 (IEC 2010b)	eseményfa-elemzés	11.
IEC 61078:2016 (IEC 2016a)	megbízhatósági blokkdiagramok	9.
IEC 61165:2006 (IEC 2006b)	Markov-technikák	12.
MIL-STD-882E:2012 (U.S. Department of Defense 2012)	rendszerbiztonság (amerikai védelmi szabvány)	4.

Szabvány	Tárgy	Fejezet
ANSI/ISA-18.2:2016 (ISA 2016), EEMUA 191:2013 (EEMUA 2013)	riasztáskezelés	16.
GSN Community Standard v3, 2021 (Assurance Case Working Group 2021)	célstruktúra-jelölés a biztonsági érvelésekhez	17.

D.2. Ágazati szabványok

D.2. táblázat. Ágazati szabványok

Ágazat	Szabvány	Tárgy
gépipar	ISO 12100:2010 (ISO 2010)	a tervezés általános elvei, kockázatértékelés és -csökkentés
	ISO 13849-1:2023 (ISO 2023)	a biztonsági irányítórendszerek biztonsági részei, PL
	IEC 62061:2021 (IEC 2021)	a biztonsági irányítórendszerek funkcionális biztonsága, SIL
	ISO 10218-1/-2:2025 (ISO 2025a)	ipari robotok biztonsága
	ISO/TS 15066:2016 (ISO 2016)	kollaboratív robotok, biomechanikai határértékek
közúti járművek	ISO 26262:2018 (ISO 2018a)	funkcionális biztonság (a 3. kiadás előkészítés alatt)
	ISO 21448:2022 (ISO 2022)	a rendeltetésszerű funkció biztonsága (SOTIF)
	ISO/PAS 8800:2024 (ISO 2024)	biztonság és mesterséges intelligencia
	ISO/TS 5083:2025 (ISO 2025b)	automatizált vezetési rendszerek biztonsága
	ISO/SAE 21434:2021 (ISO/SAE 2021)	kiberbiztonsági mérnöki folyamat
	SAE J3016:2021 (SAE International 2021)	a vezetésautomatizálás szintjei és fogalmai
	EGAS felügyeleti koncepció, 6.0, 2015 (EGAS Workgroup 2015)	a motorvezérlés háromszintű felügyelete
vasút	EN 50126-1/-2:2017 (CENELEC 2017)	RAMS-folyamat
	EN 50129:2018 (CENELEC 2018)	biztonsági elektronikus jelzőrendszerek
	EN 50716:2023 (CENELEC 2023)	vasúti szoftverfejlesztés (az EN 50128 és az EN 50657 helyett)
polgári repülés	ARP4754B:2023 (SAE International 2023a)	repülőgépek és rendszereik fejlesztése
	ARP4761A:2023 (SAE International 2023b)	a biztonságértékelési folyamat

Ágazat	Szabvány	Tárgy
	DO-178C:2011 (RTCA 2011a), DO-333:2011 (RTCA 2011b)	fedélzeti szoftver, formális módszerek kiegészítése
orvostechnika	ISO 14971:2019 (ISO 2019)	kockázatkezelés
	IEC 62304:2006+A1:2015 (IEC 2006c)	az orvostechnikai szoftverek életciklusa
szoftver	MISRA C:2025 (MISRA 2025)	a C nyelv biztonságkritikus használata
kiberbiztonság	IEC 62443 sorozat (IEC 2009--2024)	az ipari automatizálás kiberbiztonsága
	IEC TR 63069:2019 (IEC 2019b)	a funkcionális biztonság és a kiberbiztonság összehangolása
mesterséges intelligencia	ISO/IEC TR 5469:2024 (ISO/IEC 2024)	funkcionális biztonság és MI-rendszerek
autonóm termékek	UL 4600, 3. kiadás, 2023 (UL Standards & Engagement 2023)	az autonóm termékek biztonsági értékelése

D.3. Jogszabályok és járműipari előírások

D.3. táblázat. Jogszabályok és járműipari előírások

Jogszabály	Tárgy	Alkalmazás
2012/18/EU irányelv (Seveso III) (Európai Parlament és Tanács 2012)	a veszélyes anyagokkal kapcsolatos súlyos balesetek veszélyének kezelése	hatályos
(EU) 2023/1230 rendelet (Európai Parlament és Tanács 2023)	gépek (a 2006/42/EK irányelv helyett)	2027. január 20-tól
(EU) 2024/1689 rendelet (Európai Parlament és Tanács 2024a)	mesterséges intelligencia (MI-rendelet)	fokozatosan; nagy kockázatú rendszerek: 2027. december 2-tól (III. melléklet), 2028. augusztus 2-tól (I. melléklet)
digitális omnibusz az MI-ről, 2026 (Európai Parlament és Tanács 2026)	az MI-rendelet határidőinek és egyes rendelkezéseinek módosítása	2026 júliusától hatályos
(EU) 2024/2847 rendelet (Európai Parlament és Tanács 2024b)	a digitális elemeket tartalmazó termékek kiberbiztonsága (CRA)	jelentési kötelezettség 2026. szeptember 11-től, teljes alkalmazás 2027. december 11-től
ENSZ EGB 13. sz. előírás (UNECE 2016)	fékrendszerek (haszonjárművek)	hatályos
ENSZ EGB 155. sz. előírás (UNECE 2021a)	kiberbiztonsági irányítási rendszer	minden új járműre 2024 júliusától (EU)
ENSZ EGB 157. sz. előírás (UNECE 2021b)	automatizált sávtartó rendszerek	hatályos



Pannon Egyetem
University of Pannonia

Biztonságkritikus rendszerek

Kézikönyv

Nyílt hozzáférésű tankönyv

https://complexsystemsintelligence.github.io/biztonsagkritikus-rendszerek/kezikonyv/