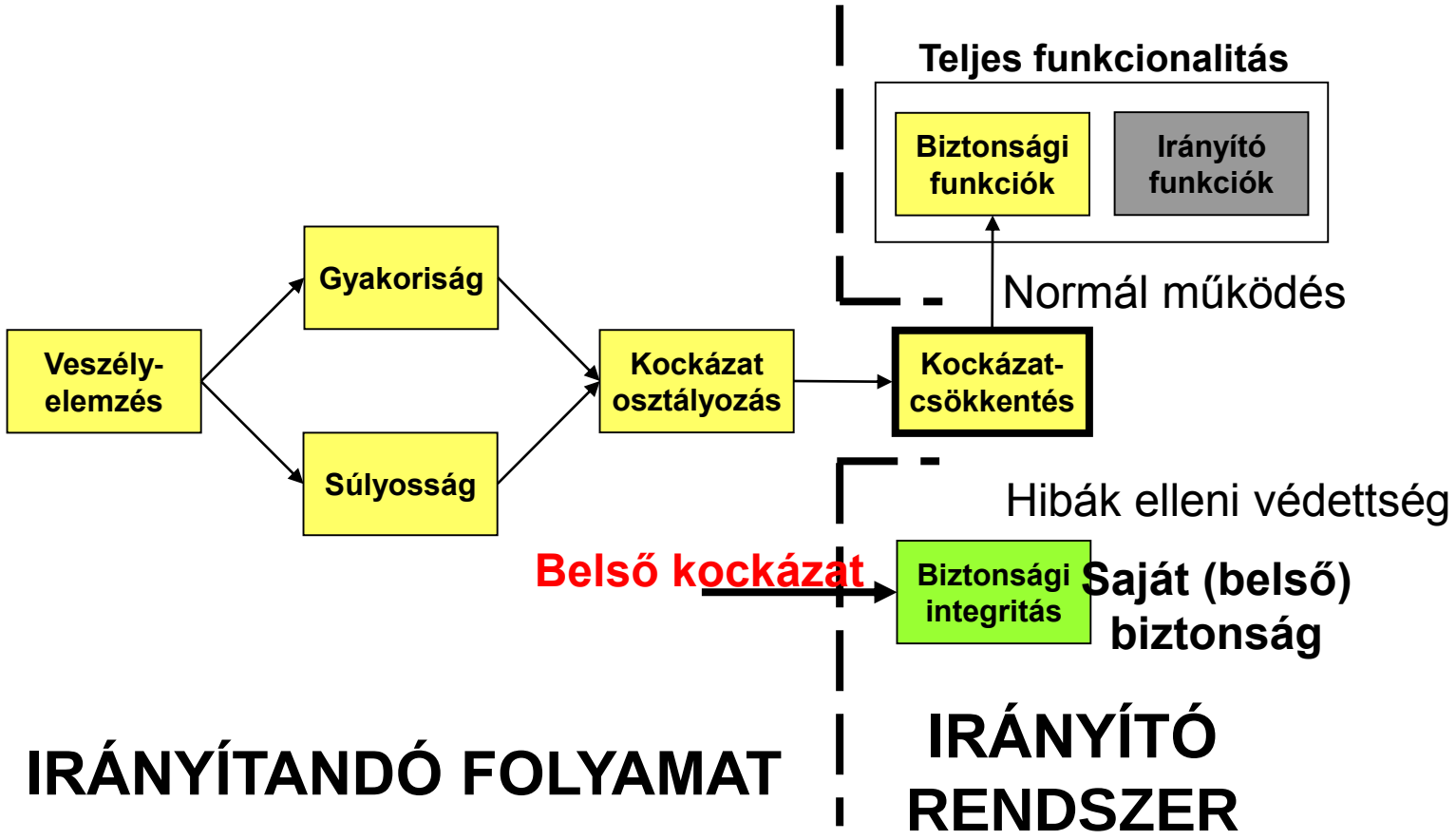
Biztonsági funkciók – Biztonsági integritás



A biztonsági rendszerek belső veszélyforrásai

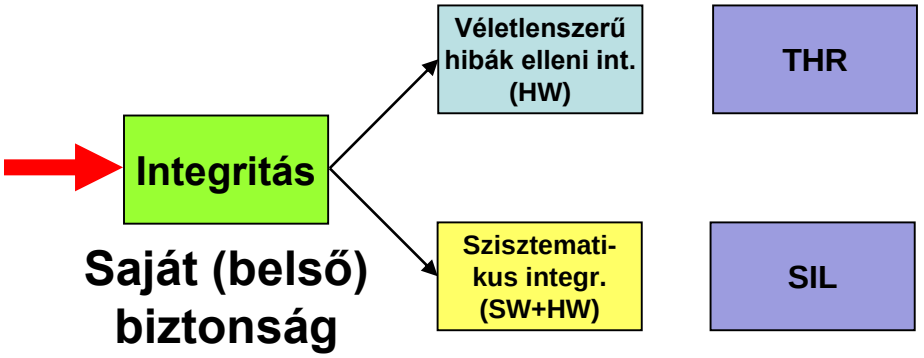
Szisztematikus hibák

- A rendszer **létrehozása során** elkövetett emberi hibák, amelyek
- a rendszer üzemelése során helytelen működést okoznak.
- Specifikációs hibák, tervezési hibák, gyártási hibák, szoftverhibák stb.
- Fellépési gyakoriság nem adható meg.

Véletlenszerű meghibásodások

- A rendszer **üzemelése során** fellépő meghibásodások.
- Fellépési gyakoriságuk megadható.
- A fellépési gyakoriságot befolyásolja az üzemmód, környezeti hatások, túlterhelés stb.

Biztonsági integritási szintek és hibakezelés



Eltűrhető veszélyeztetési ráta
Tolerable Hazard Rate
[1/h]

Biztonságintegritási szint
Safety Integrity Level

A szisztematikus hibák elleni védelem

- Személyi függetlenségek
  - ellenőrizhetőség
- Megfelelő módszerek alkalmazása
  - hibaelkerülés
- A fejlesztési/tervezési/gyártási folyamat szabályozása → életciklus modellek
  - követhetőség, ellenőrizhetőség, áttekinthetőség

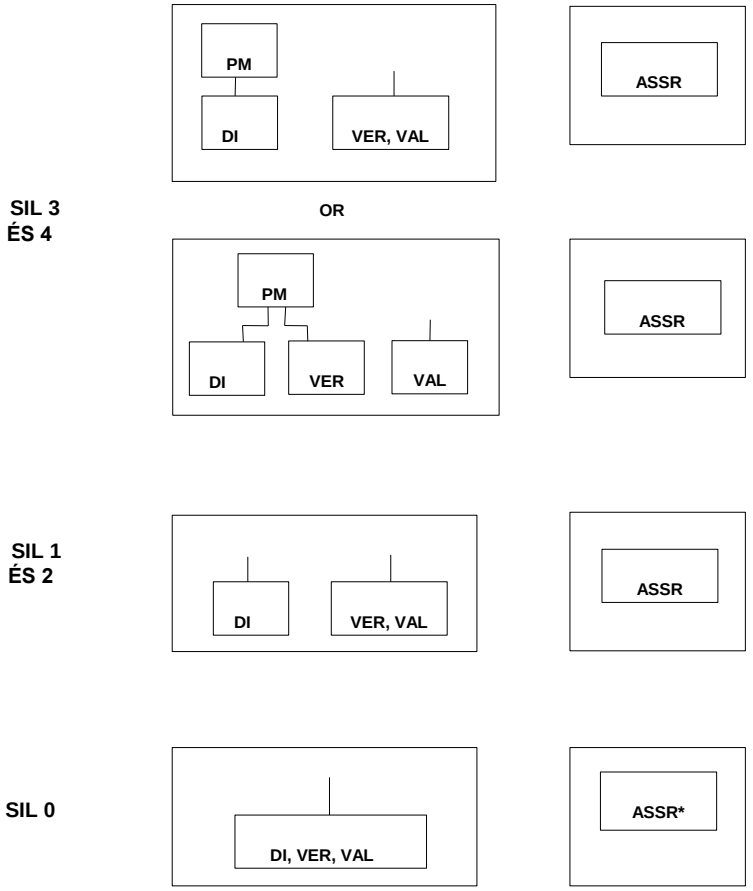
SIL-intézkedések, példa (EN 50129)

Technikák/Intézkedések	SIL 1	SIL 2	SIL 3	SIL 4
1. A biztonságorientált és nem biztonságorientált rendszerek szétválasztása	R: jól meghatározott interfészek a biztonságorientált és nem biztonságorientált rendszerek között		HR: jól meghatározott interfészek a biztonságorientált és nem biztonságorientált rendszerek között és interfész-elemzés	
2. Grafikus leírás beleértve pl. blokkdiagramokat	HR		HR	
3. Strukturált specifikáció	HR: manuális, hierarchikus szétválasztás alfeladatokra, interfészleírások		HR: hierarchikus szétválasztás formális módszerek alkalmazásával, automatikus konzisztencia-ellenőrzés, finomítás a funkcionális szintig	
4. Formális vagy félformális módszerek			R: számítógéppel támogatott	
5. Számítógéppel támogatott specifikációs eszközök		R: eszközök kiválasztása bármely konkrét tervezési módszer előnyben részesítése nélkül	R: modellorientált eljárások hierarchikus felosztással, minden objektum, kapcsolatainak, közös adatbázisának, automatikus konzisztencia-ellenőrzésének leírása	
6. Ellenőrzőlisták	R: előkészített ellenőrzőlisták minden biztonsági életciklus-fázisra		R: előkészített részletes ellenőrzőlisták minden biztonságorientált életciklus-fázisra	
7. Veszélynapló	HR: A Veszélynaplót fel kell fektetni és karban kell tartani a rendszer teljes életciklusa során			

SIL-intézkedések, példa (EN 50129)

Technikák/Intézkedések	SIL 1	SIL 2	SIL 3	SIL 4
1. A biztonsági szervezet tagjainak képzése	HR: Kezdeti képzés minden biztonságorientált tevékenységnél		HR: Minden biztonságorientált tevékenységgel kapcsolatban ismétlődő képzés vagy a tevékenység rendszeres teljesítése	
2. A résztvevők személyi függetlensége	lásd a 6. ábrát: a függetlenség megszervezése			
3. A biztonsági szervezet személyzetének képesítése (lásd az 1. sz. megjegyzést) 4. (lásd a megjegyzést)	HR: műszaki oktatás vagy elegendő tapasztalat		HR: magasabb szintű műszaki oktatás vagy szélesebb körű tapasztalat	

Személyi függetlenség (példa)



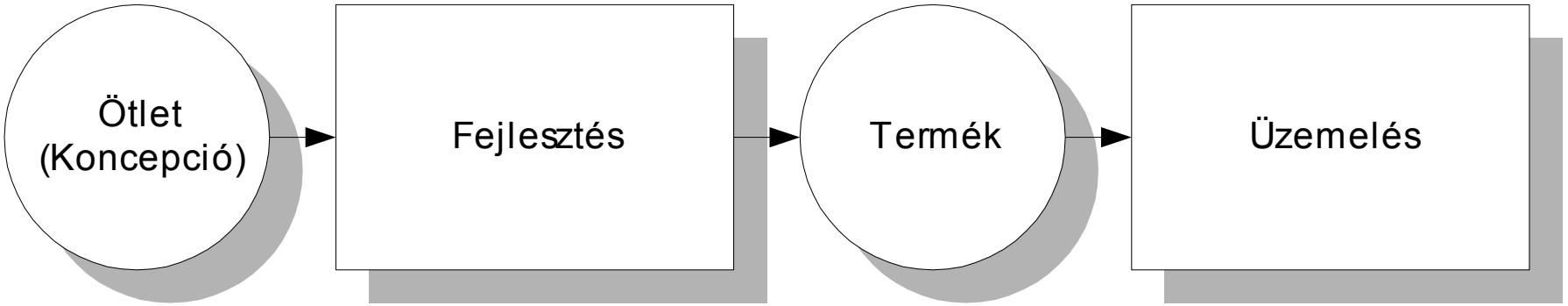
Magy.:
- PM = Project Manager
- DI = Tervező, megvalósító
- VER = Verifikáló
- VAL = Validáló
- ASSR = Asszesszor

= lehet egyazon személy

=lehet egyazon szervezet

* =SIL0-ra csak akkor kell asszesszor, egy átfogó rendszer biztonsá-gára lehet hatással

Egyszerű fejlesztési modell



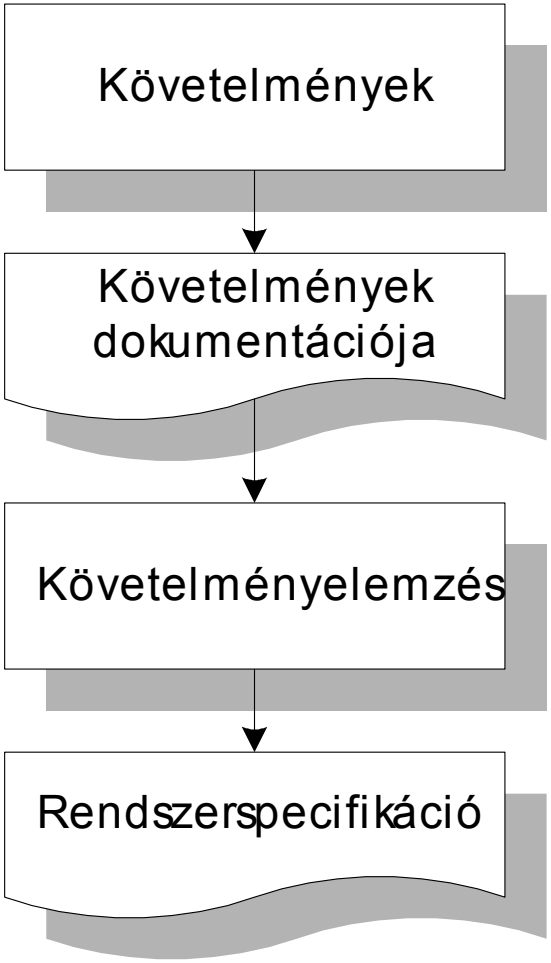
Fejlesztési módszerek, lépések

- Felhasználói követelmények
  - Funkcionális
  - Biztonsági
- Előzetes veszélyelemzés
- Specifikáció
- Top-level design
- Részletes tervezés
- A modulok megvalósítása és tesztelése/verifikáció
- Rendszer-integráció és rendszerteszt / validáció
- Engedélyezés

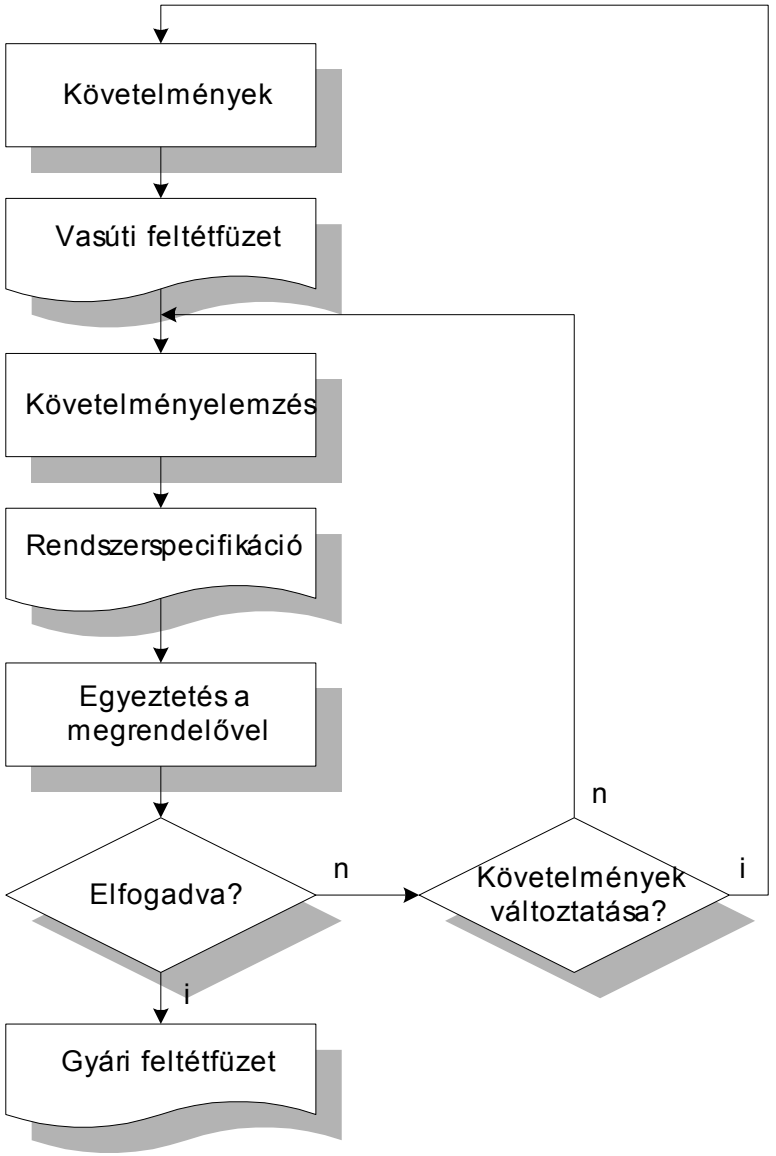
Specifikáció

- A rendszer működésének leírása
  - Funkciók
  - Együttműködés más rendszerekkel
  - Operátori kapcsolatok
  - Biztonsági jellemzők
    - Tervezési „kényszerek”
- Konzultációk a megbízó és a szállító között
- A szerződéses kapcsolat alapja
- A fejlesztési folyamat végén bizonyítani kell, hogy az eredmény minden tekintetben megfelel a specifikációnak (és remélhetőleg a megbízói követelményeknek)

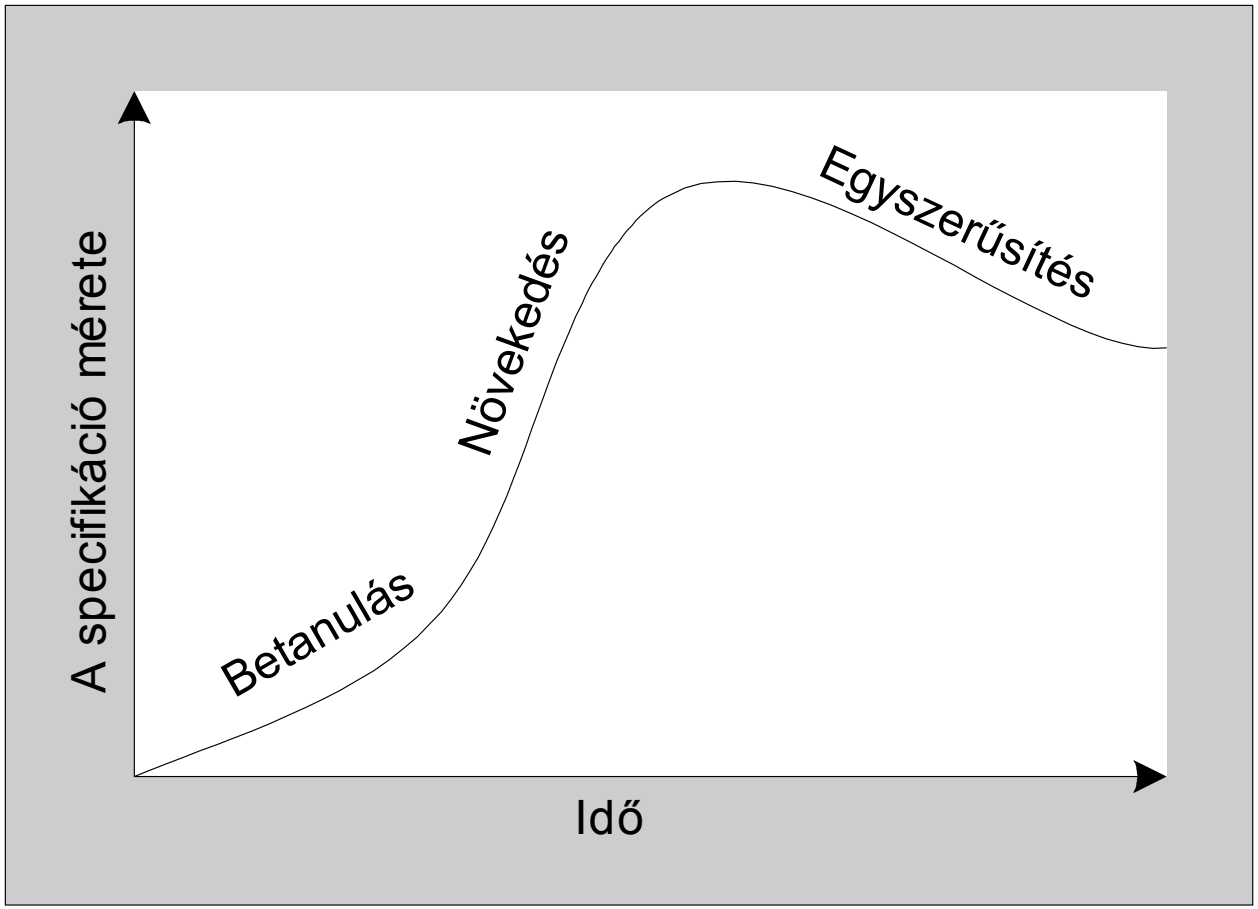
Specifikáció



Iteratív specifikációs modell



A specifikáció kialakulása



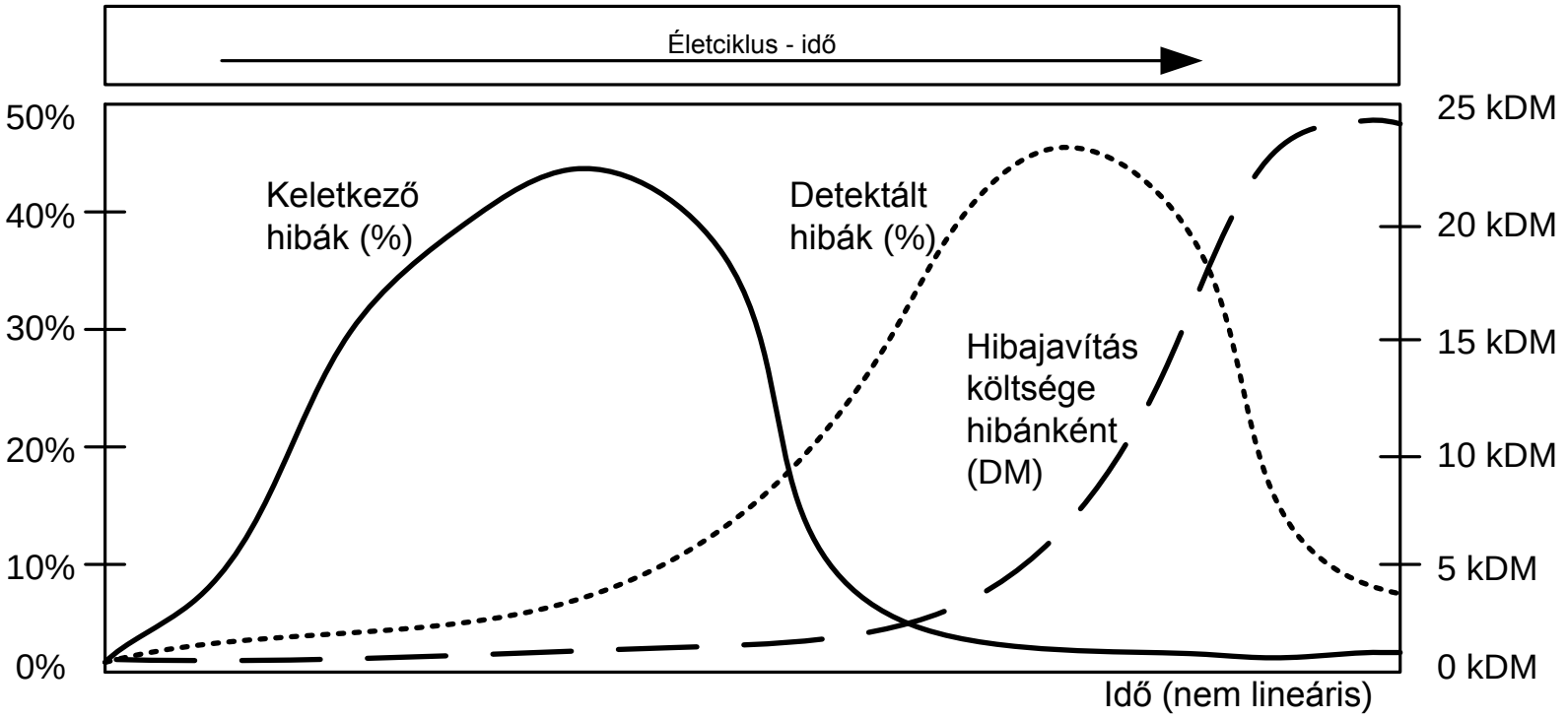
Az ideális specifikáció tulajdonságai

- Korrekt
- Teljes (nemcsak normál körülményekre)
- Konzisztens (ellentmondásmentes)
- Féreérthetetlen (természetes nyelvek!)
  - A természetes nyelven írt specifikációk helyességének ellenőrzése nehéz
- Lehetőségek
  - Strukturált szerkezet (félformális)
  - Formális matematikai módszerek

A specifikáció hibái

- A biztonságkritikus rendszerek egyik legnagyobb problémája a hibás specifikáció
  - A felhasználói követelmények meg nem felelősége
  - A specifikáció nem felel meg a felhasználói követelményeknek
- A specifikációs hibák gyakran csak a kész rendszer vizsgálatakor derülnek ki, amikor a hibajavítás már igen költséges

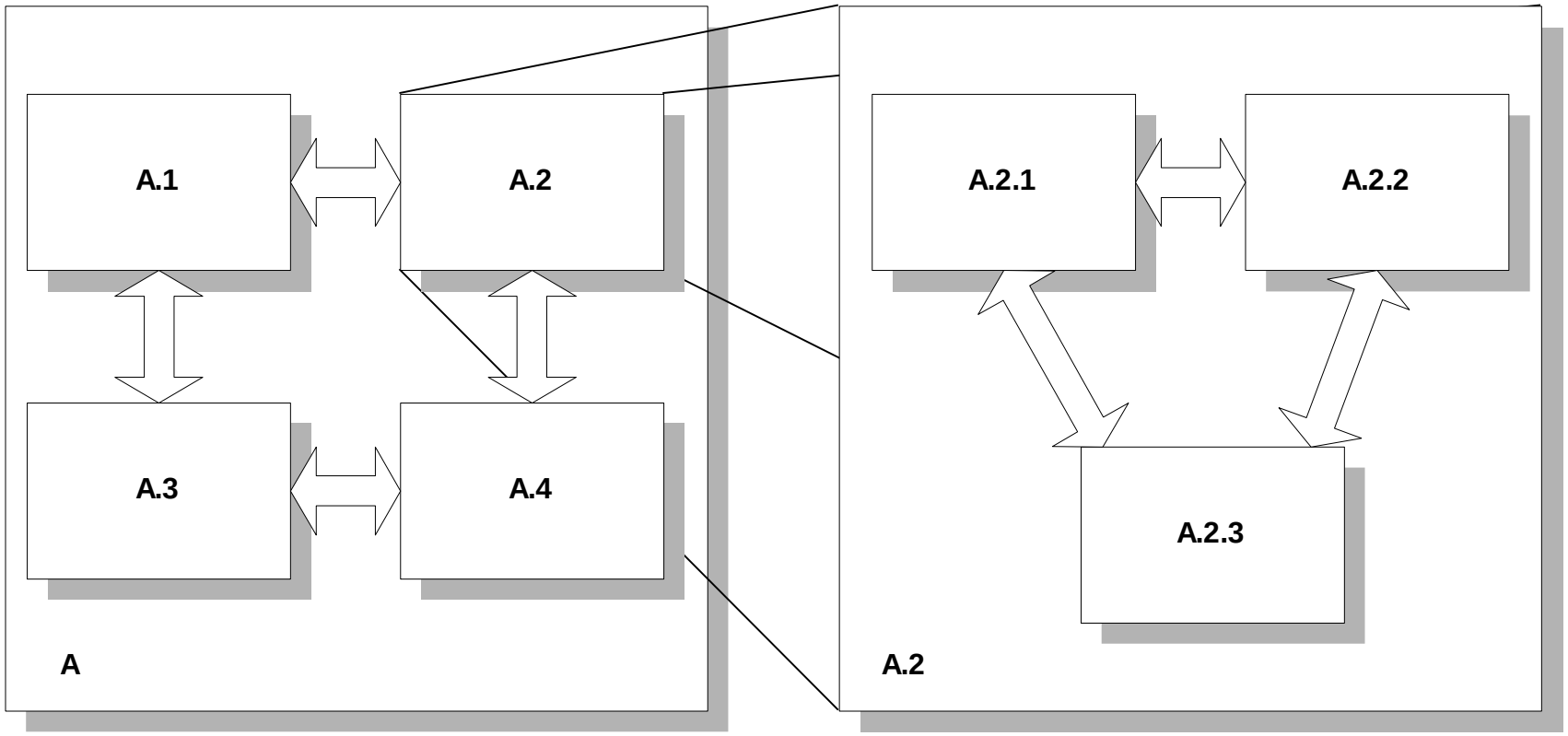
Hibák keletkezése és detektálása



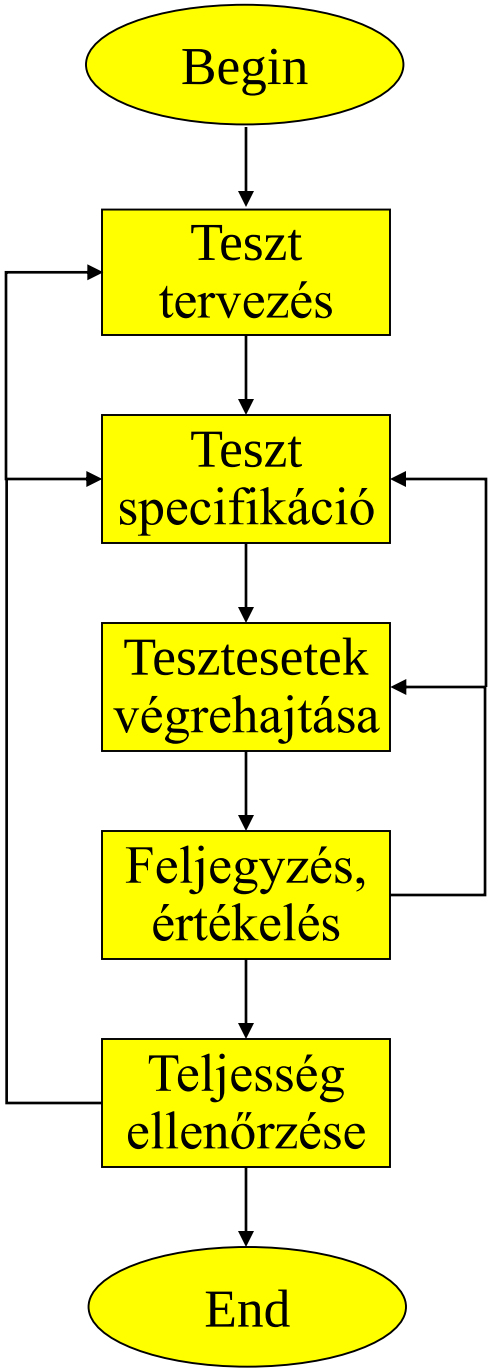
Top-level design - Detailed design
Magasszintű tervezés – Részl. terv.

- Rendszerfunkciók szétbontása
  - Hardver
  - Szoftver
- Architektúrák kidolgozása (HW és SW)
  - Modulokra bontás (hierarchikus struktúra)
  - Modulkapcsolatok meghatározása (interfész)
  - Meghatározni a modulok
    - Funkcióit
    - Biztonsági jellemzőit
  - Lényeges SW adatsruktúrák meghatározása
- Követelmények lebontása az architektúra elemeire
- Modulok részletes tervezése
  - A dekompozíció gyakran iteratív (szubmodulok)

Dekompozíció



A TESZTELÉS FOLYAMATA



Tesztelési terv, teszt-esetek

- Tesztelési terv
  - A teszt-eseteket meghatározó technika megadása
  - A tesztelési eljárás megfelelőségét értékelő módszerek
  - A tesztelendő rendszer fejlesztését és teszt-tervezését végzők függetlensége
  - A teszt-környezet
  - A tesztelés teljességének kritériumai
- Teszt-esetek meghatározása
  - A tesztelendő rendszer kezdeti állapota
  - A bemenetek
  - A várt válaszok

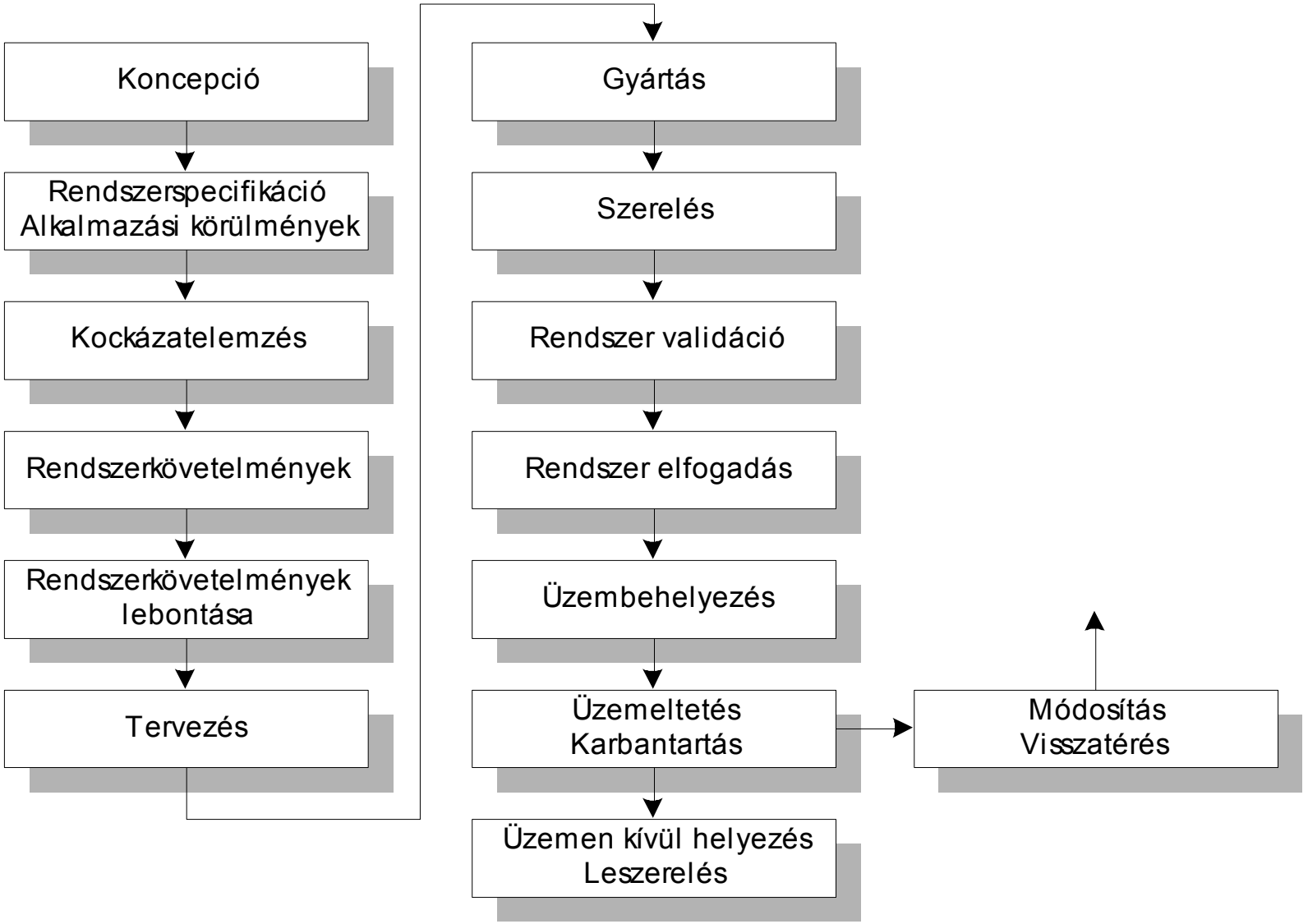
Rendszer integráció

- Progresszív integráció - hagyományos
  - A modulok kis csoportját (minimális rendszer) tesztelik, a hibákat javítják
  - Fokozatosan újabb modulokkal bővítenek, tesztelnek, javítanak
  - Az egyszerű kezdés és a kis lépésekben való bővítés miatt egyszerű a hibadetektálás és a diagnózis
  - Hátrány: A teljes rendszer jellemzői csak az integráció befejeztével vizsgálhatók - az ilyen funkciókkal kapcsolatos hibák késői, drága feltárása, javítása
- „Big bang” módszer
  - Tesztelés csak az integráció befejeztét követően
  - Feltételezés: a modulok kialakítása és tesztelése megfelelő volt
  - Előny: a durva követelmény- vagy specifikációs hibák viszonylag korán kiderülnek, javításuk kevésbé költséges
  - Hátrány: a tesztelendő rendszer bonyolultsága miatt a tesztelés feladata jóval nehezebb

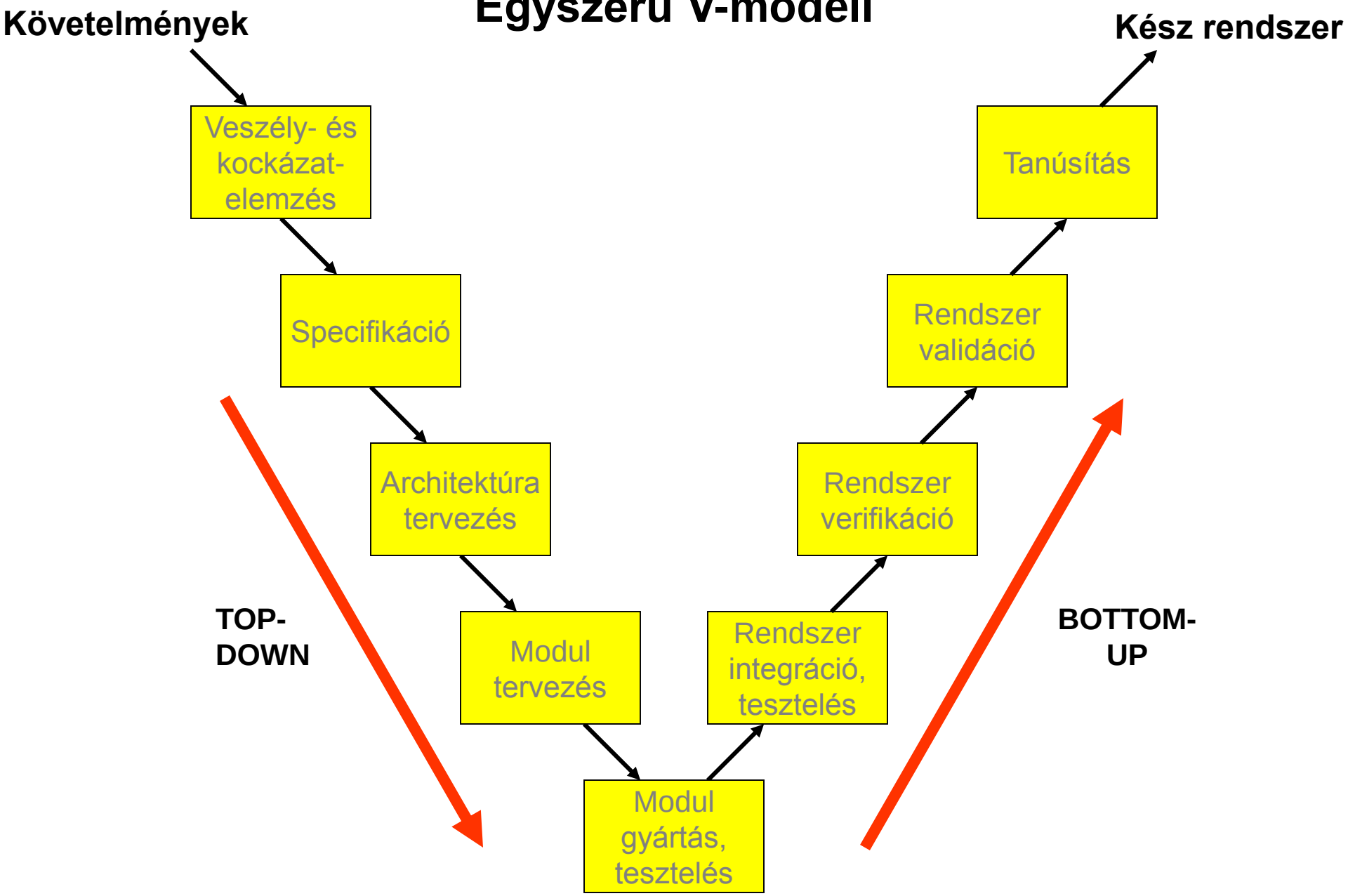
Rendszerteszt, engedélyezés

- A rendszer integrációt követően
  - A teljes rendszer megfelel a specifikációnak
  - Dinamikus és statikus módszerek kombinációja
  - Szimulált vagy valós környezetben
- Független tanúsító (az engedélyezéshez)
  - A fejlesztés valamennyi fázisát megfelelő gondossággal és kompetenciával hajtották végre
  - Dokumentálni kell
    - Valamennyi munkafázist
    - A tesztelés részleteit és eredményeit
  - A tanúsítás folyamatát már a projekt elején tervezni kell
  - A szabványok, irányelvek megadják az egyes fejlesztési fázisokban szükséges dokumentációk listáját

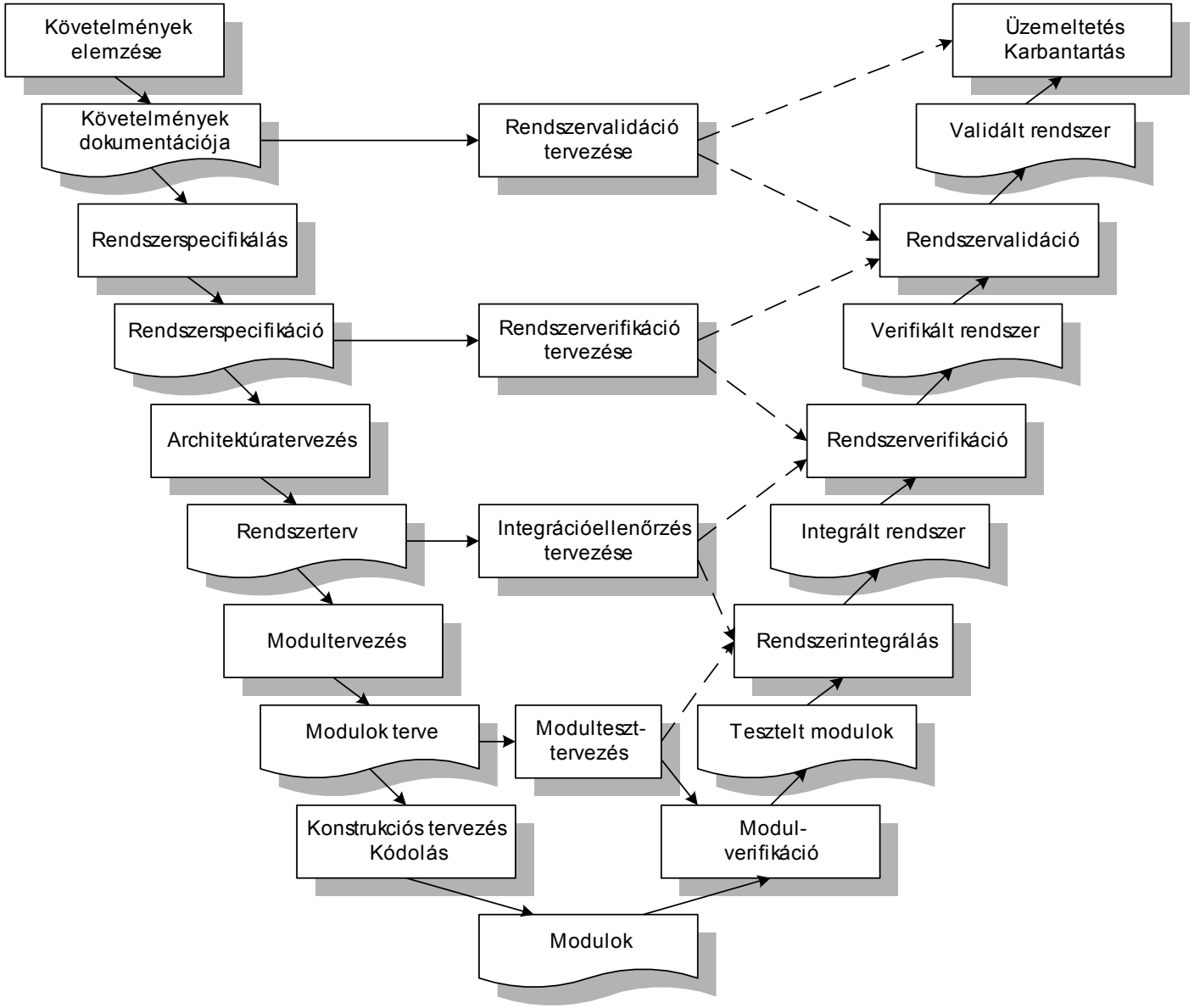
Fázismodell



Egyszerű V-modell



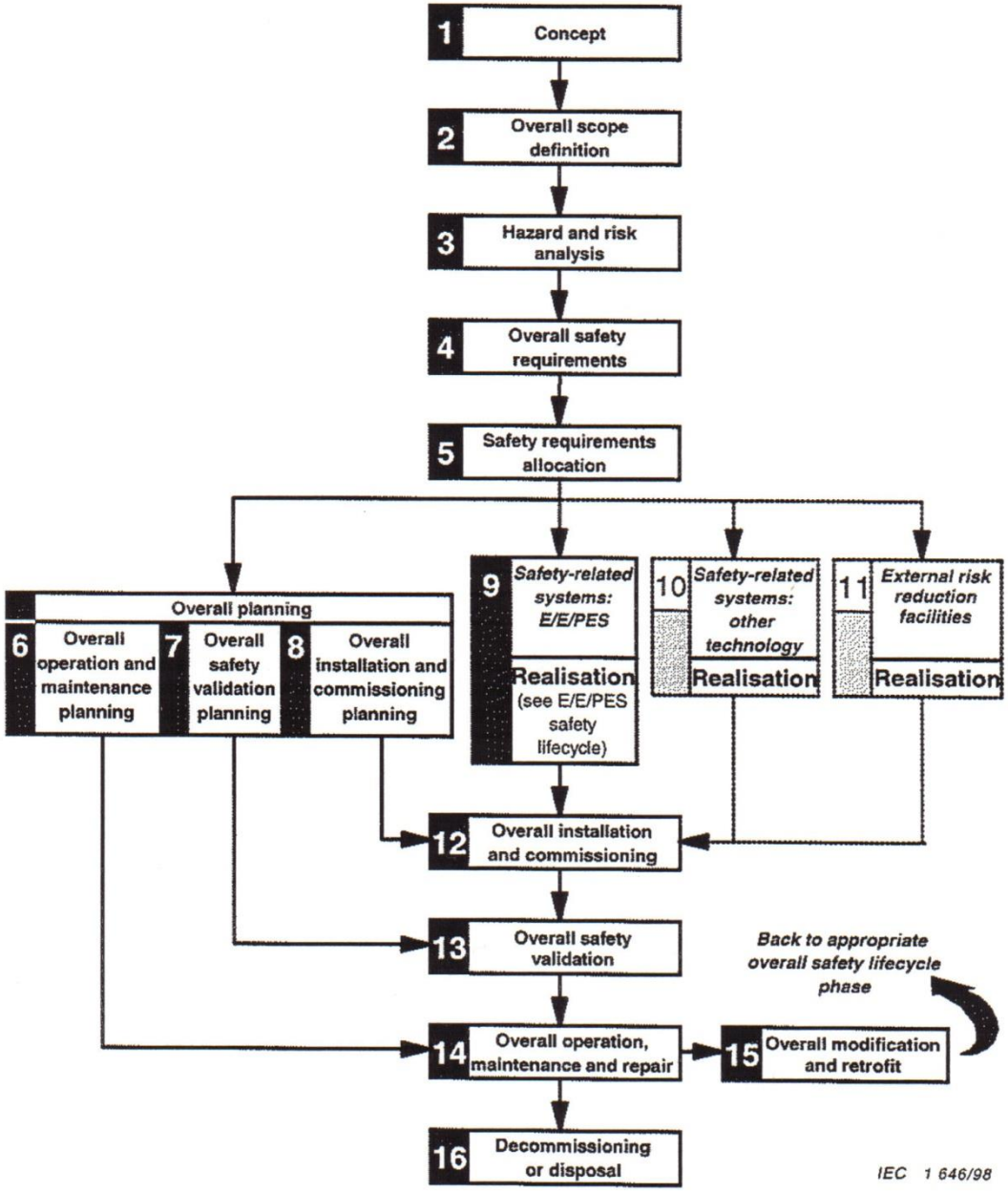
Bővített V-modell



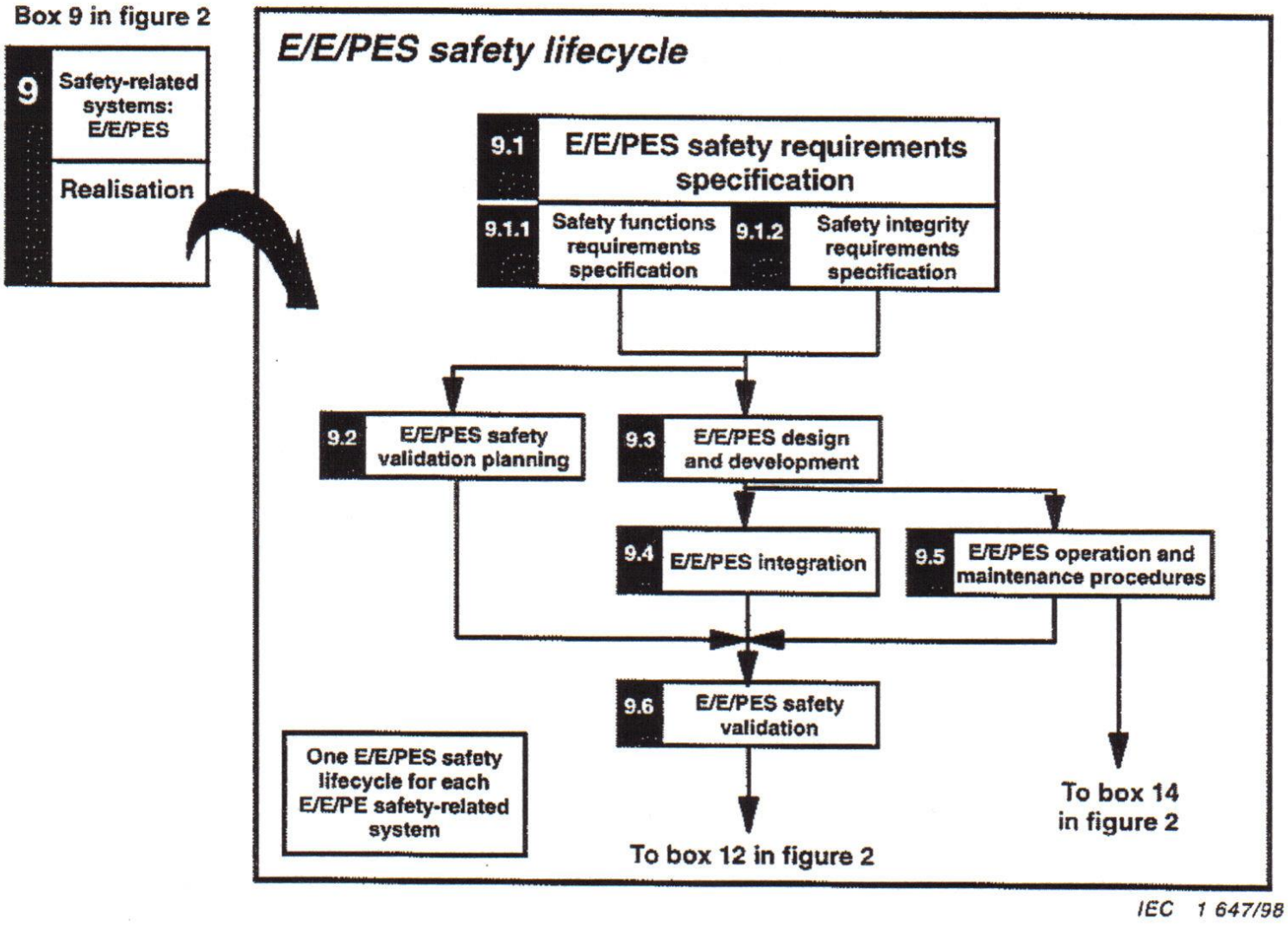
Bővített V-modell

- Többlet-információ
  - Az egyes fázisok „terméke” (dokumentáció stb.)
  - A fázisok közötti információáramlás
- Még ez is erősen egyszerűsített
  - Nem mutatja az iterációkat (bonyolult lenne)
    - Fázisokon belül
    - Fázisok között
  - Nem mutatja
    - Az egyes fázisokban párhuzamosan (pl. HW+SW) és
    - a több fázison keresztül folytatandó tevékenységeket

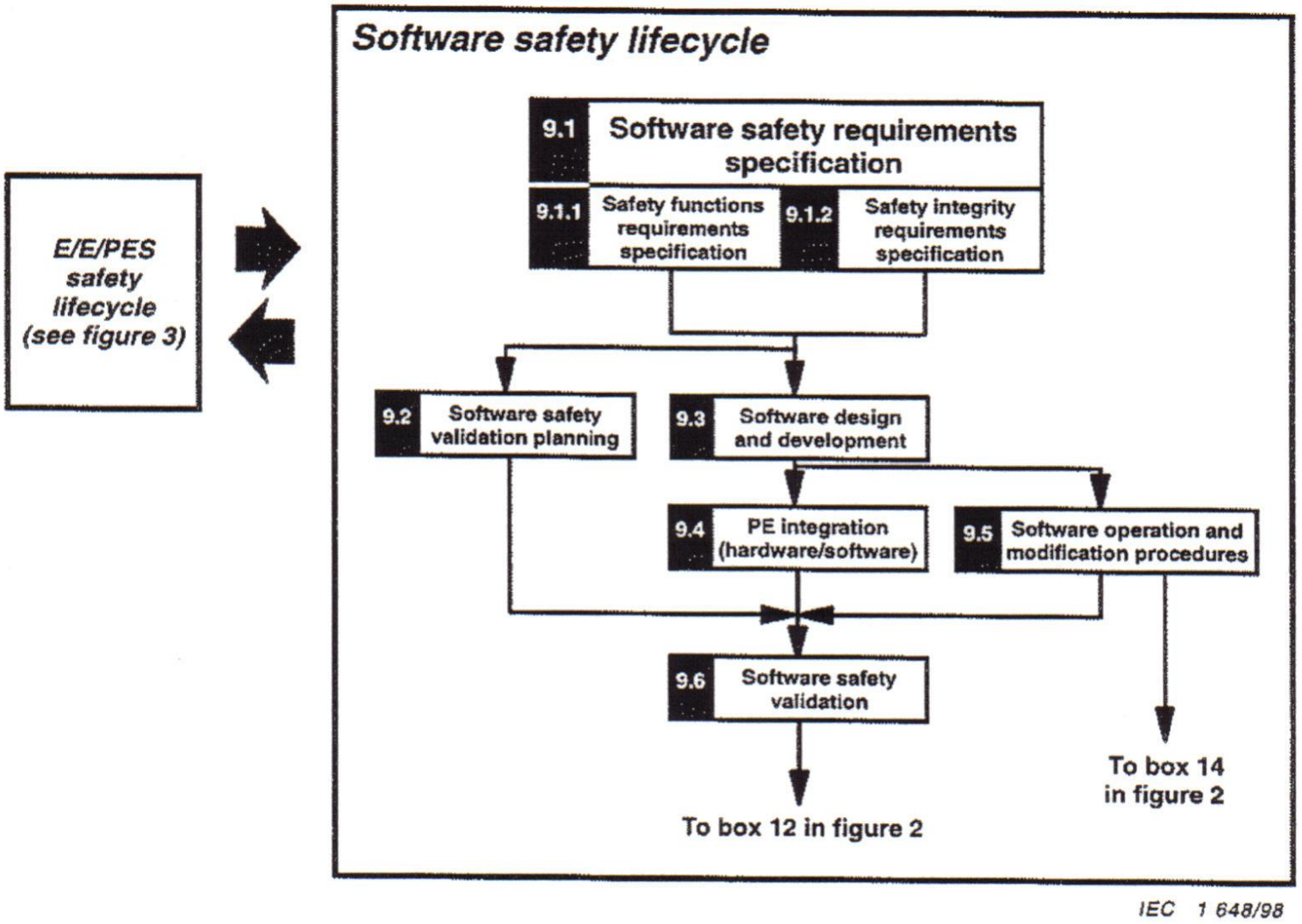
IEC 61508 biztonsági életciklus m.



IEC 61508 - megvalósítás



IEC61508 - szoftver



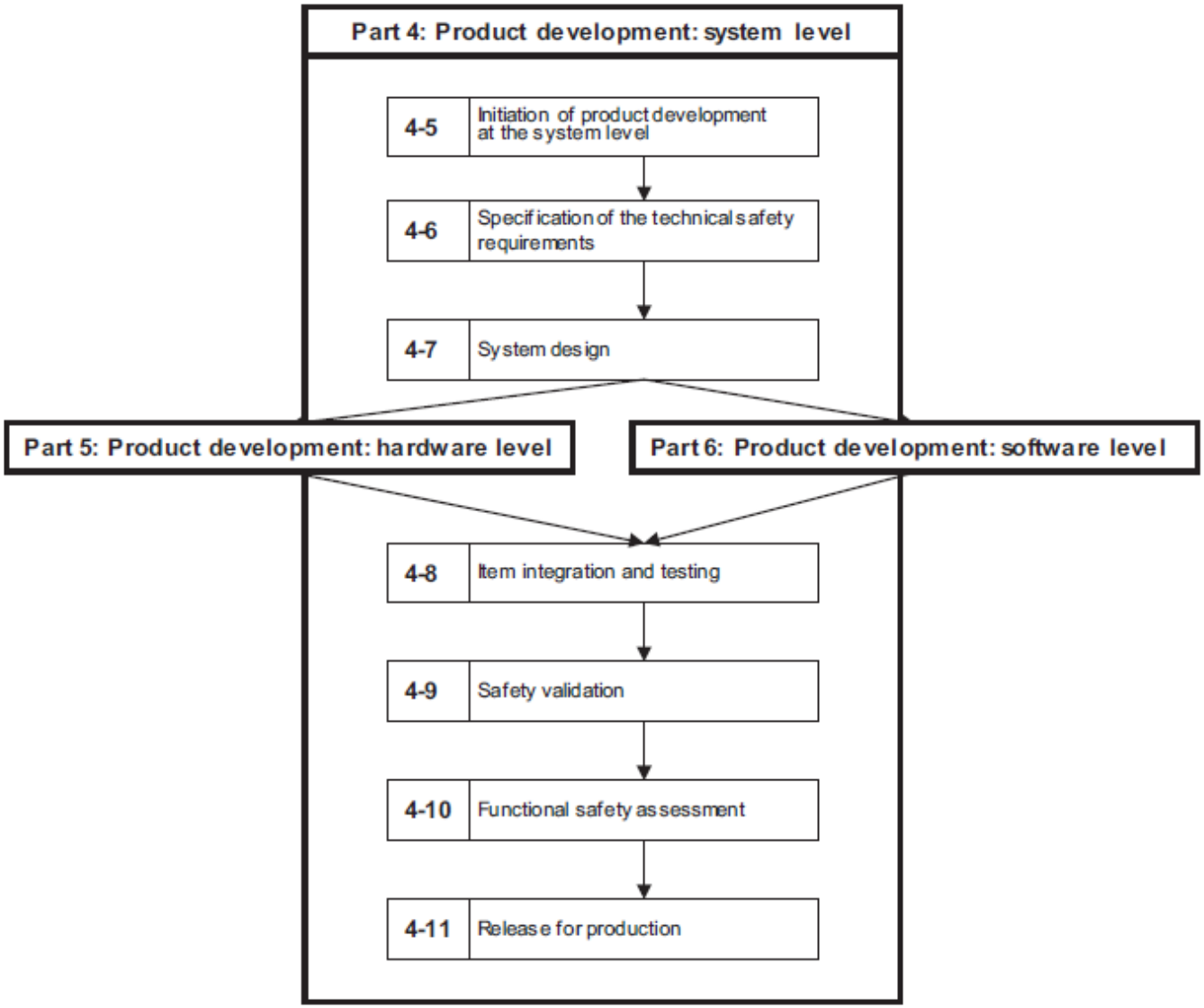
61508 életciklus modell

- A teljes (nemcsak a fejlesztési) életciklus valamennyi tevékenysége
  - kezdve a koncepciós fázistól
  - mindaddig, amíg a rendszer használatra alkalmatlanná nem válik
- Minden fázishoz tartozik biztonsági célú tevékenység, pl.
  - Veszély- és kockázatelemzés
- A fejlesztést követő fázisok is befolyásolják a biztonságot
- Megvalósítási módok
  - Villamos/elektronikus/programozható technológiák
  - Egyéb (mechanikai, hidraulikai stb.) technológiák
  - Külső (rendszeren kívüli) kockázatcsökkentési lehetőségek
  - A biztonság érdekében mindig a legegyszerűbbet kell választani!

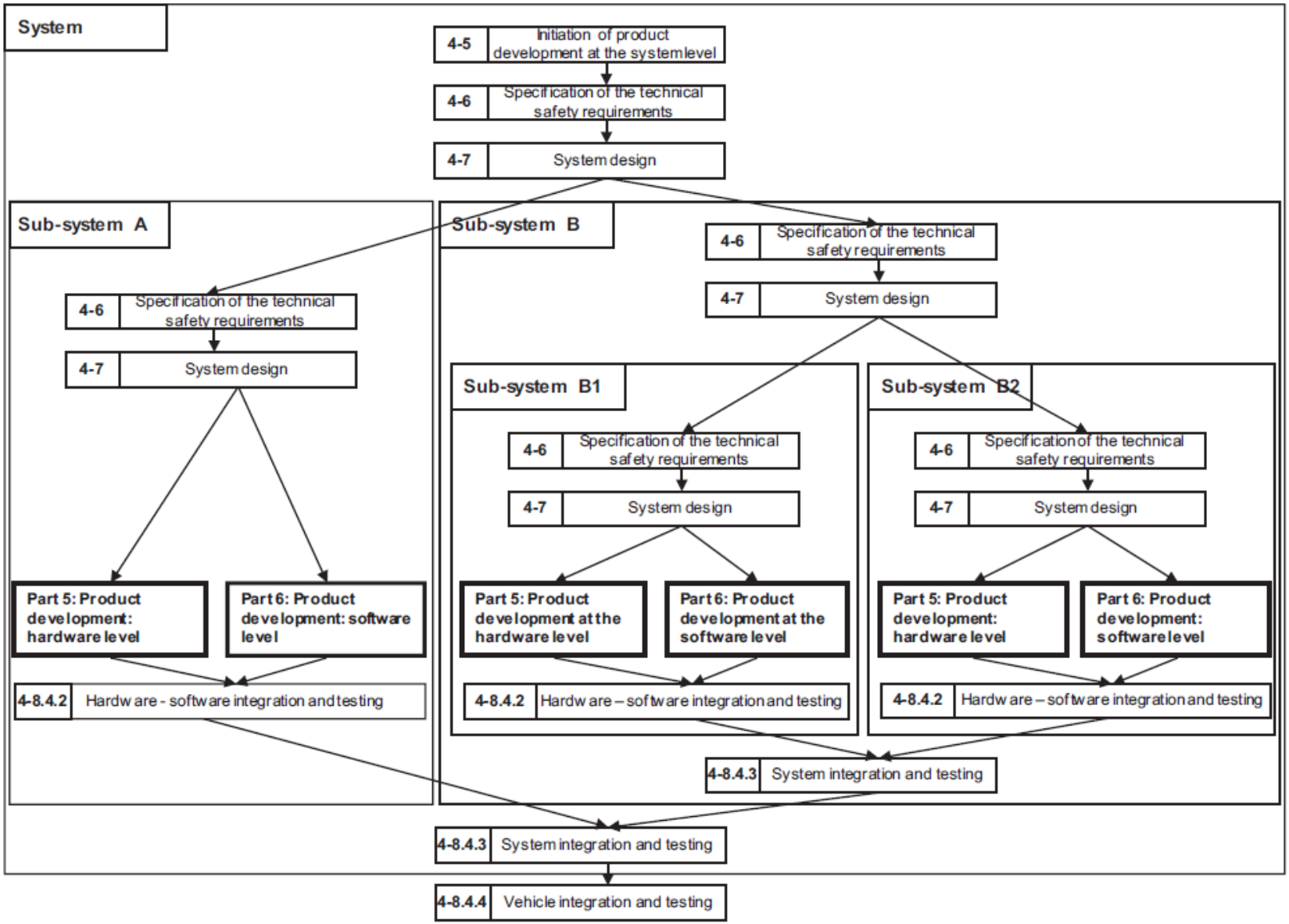
61508 életciklus modell

- Párhuzamos tervezési tevékenységek a későbbi fázisok számára
- Ez a modell is egyszerűsített, pl.
  - Nem mutatja az értékelési és verifikációs tevékenységeket - ezek minden fázisnál szükségesek a továbblépés előtt
- A modell bármely integritási szint esetén használható
  - Az egyes fázisokban szükséges tevékenységek azonban a SIL-től függően erősen eltérhetnek

ISO 26262


Part 4: Product development: system level
4-5
Initiation of product development at the system level
4-6
Specification of the technical safety requirements
4-7
System design
Part 5: Product development: hardware level
Part 6: Product development: software level
4-8
Item integration and testing
4-9
Safety validation
4-10
Functional safety assessment
4-11
Release for production

ISO 26262


System
4-5 Initiation of product development at the system level
4-6 Specification of the technical safety requirements
4-7 System design
Sub-system A
4-6 Specification of the technical safety requirements
4-7 System design
Part 5: Product development: hardware level
Part 6: Product development: software level
4-8.4.2 Hardware - software integration and testing
Sub-system B
4-6 Specification of the technical safety requirements
4-7 System design
Sub-system B1
4-6 Specification of the technical safety requirements
4-7 System design
Part 5: Product development at the hardware level
Part 6: Product development at the software level
4-8.4.2 Hardware – software integration and testing
Sub-system B2
4-6 Specification of the technical safety requirements
4-7 System design
Part 5: Product development: hardware level
Part 6: Product development: software level
4-8.4.2 Hardware – software integration and testing
4-8.4.3 System integration and testing
4-8.4.3 System integration and testing
4-8.4.4 Vehicle integration and testing