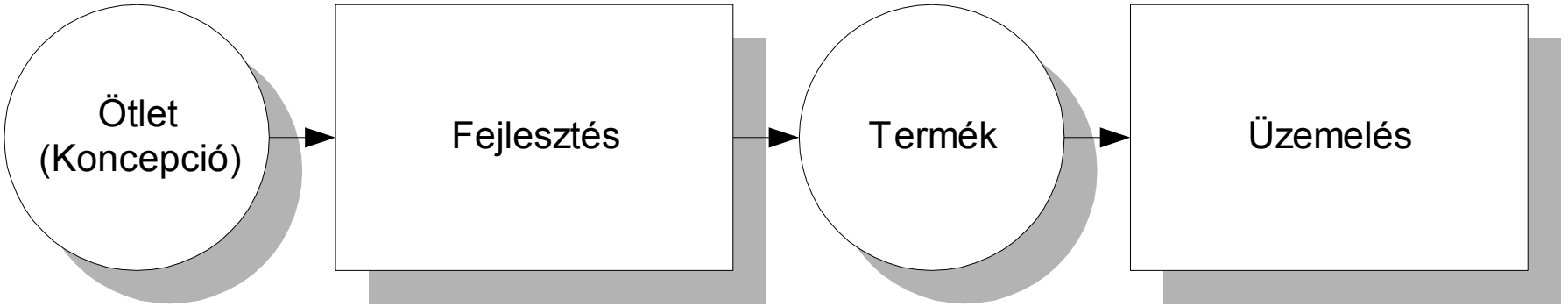
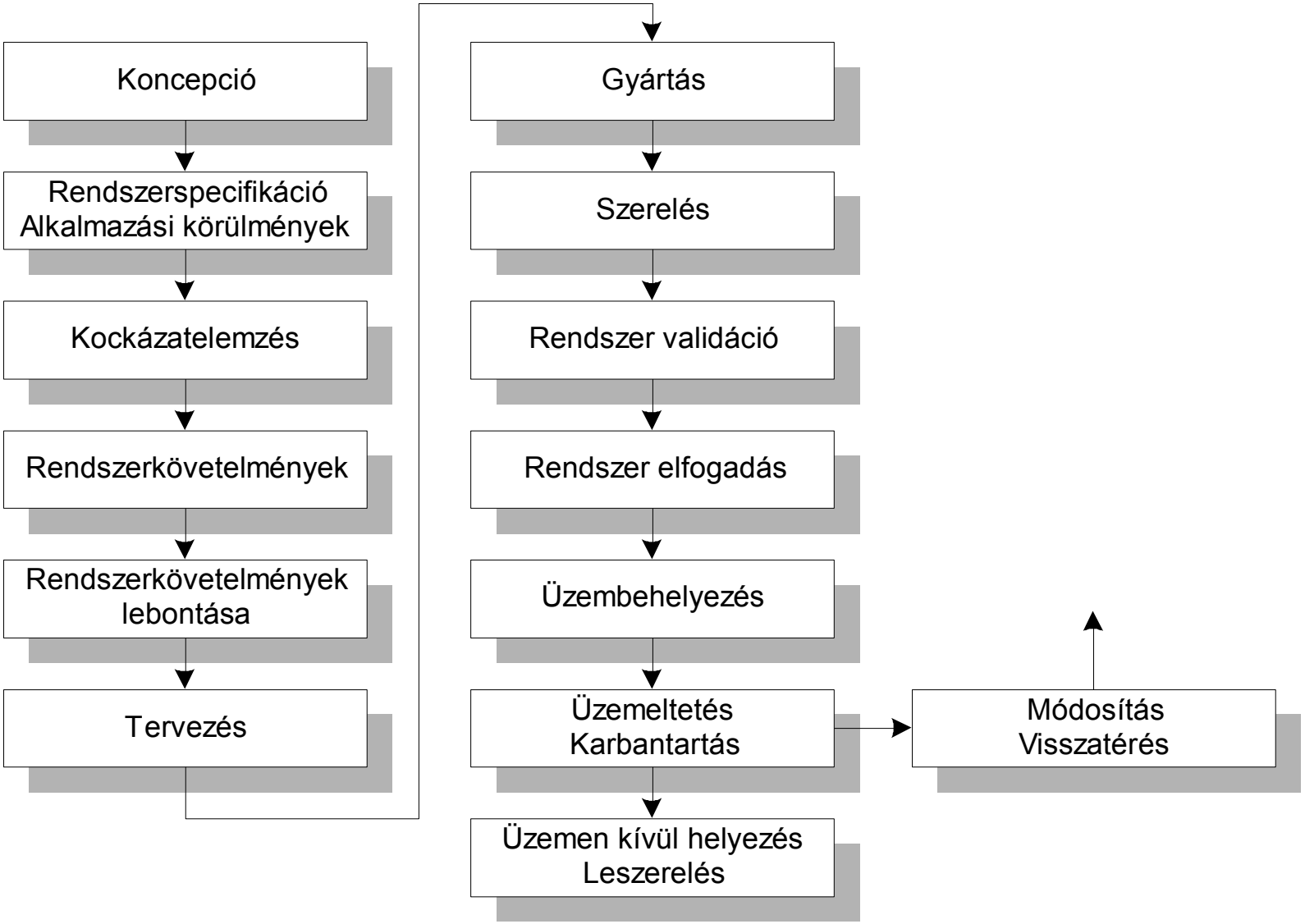
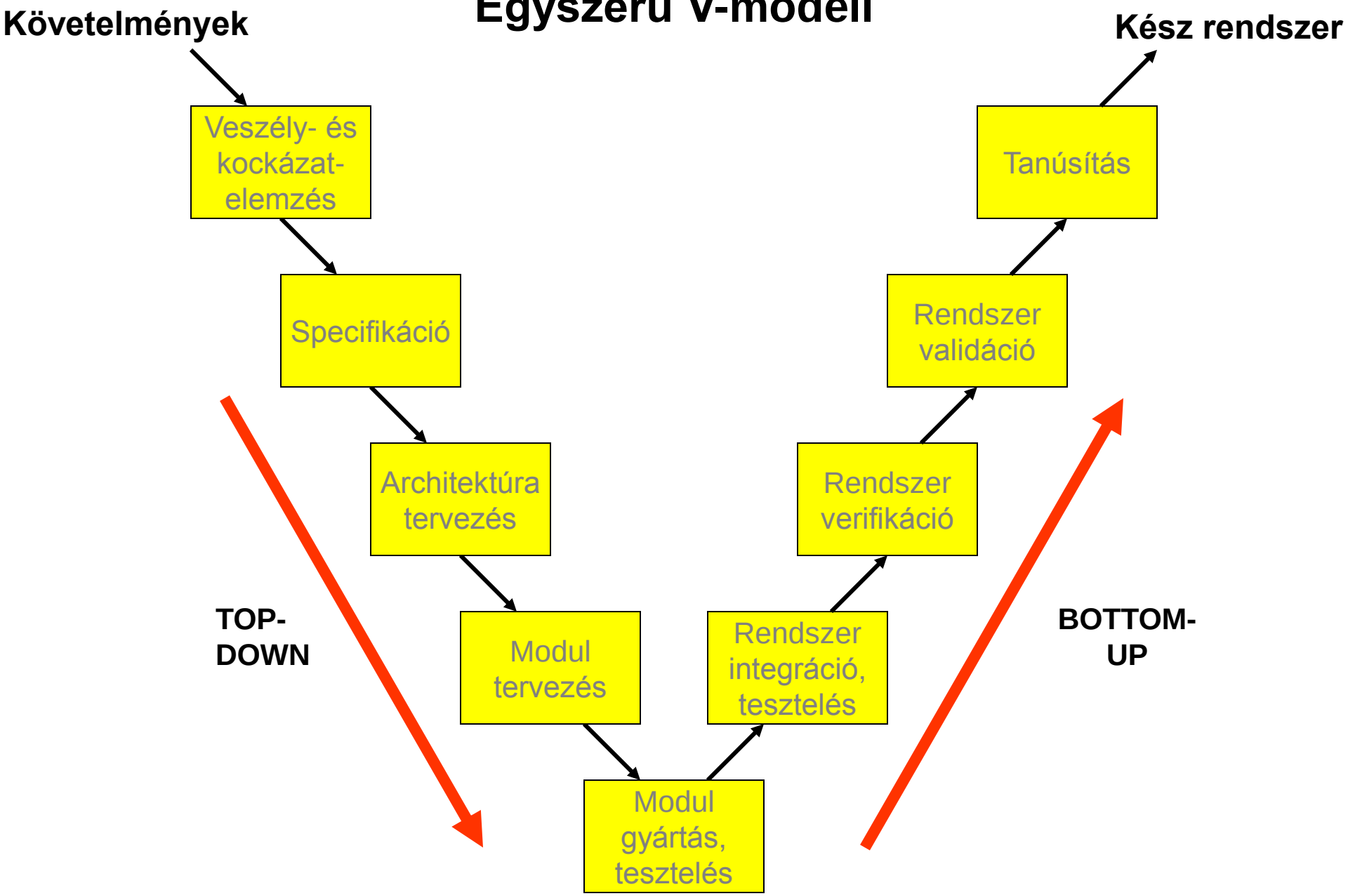
Egyszerű fejlesztési modell



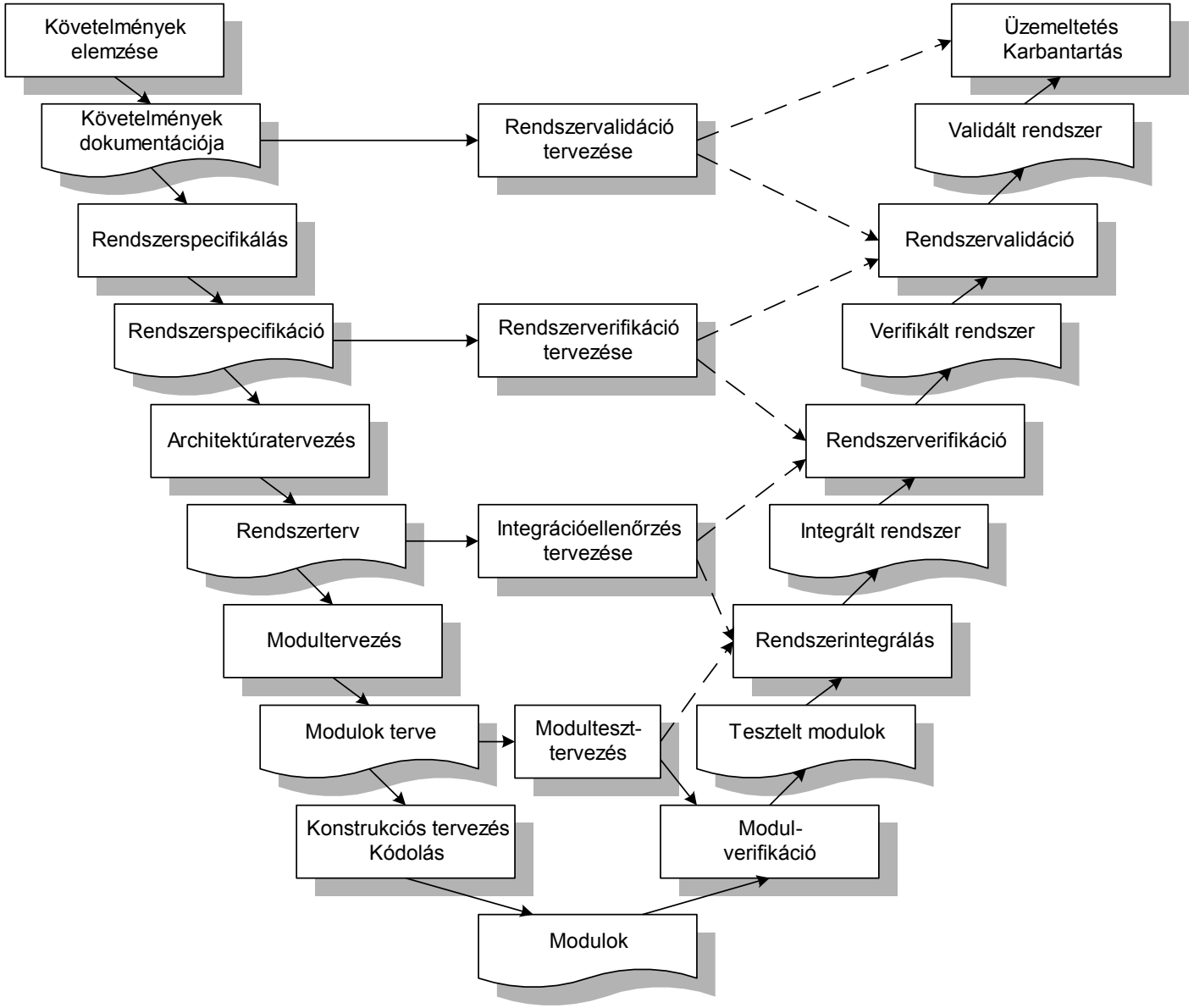
Fázismodell



Egyszerű V-modell



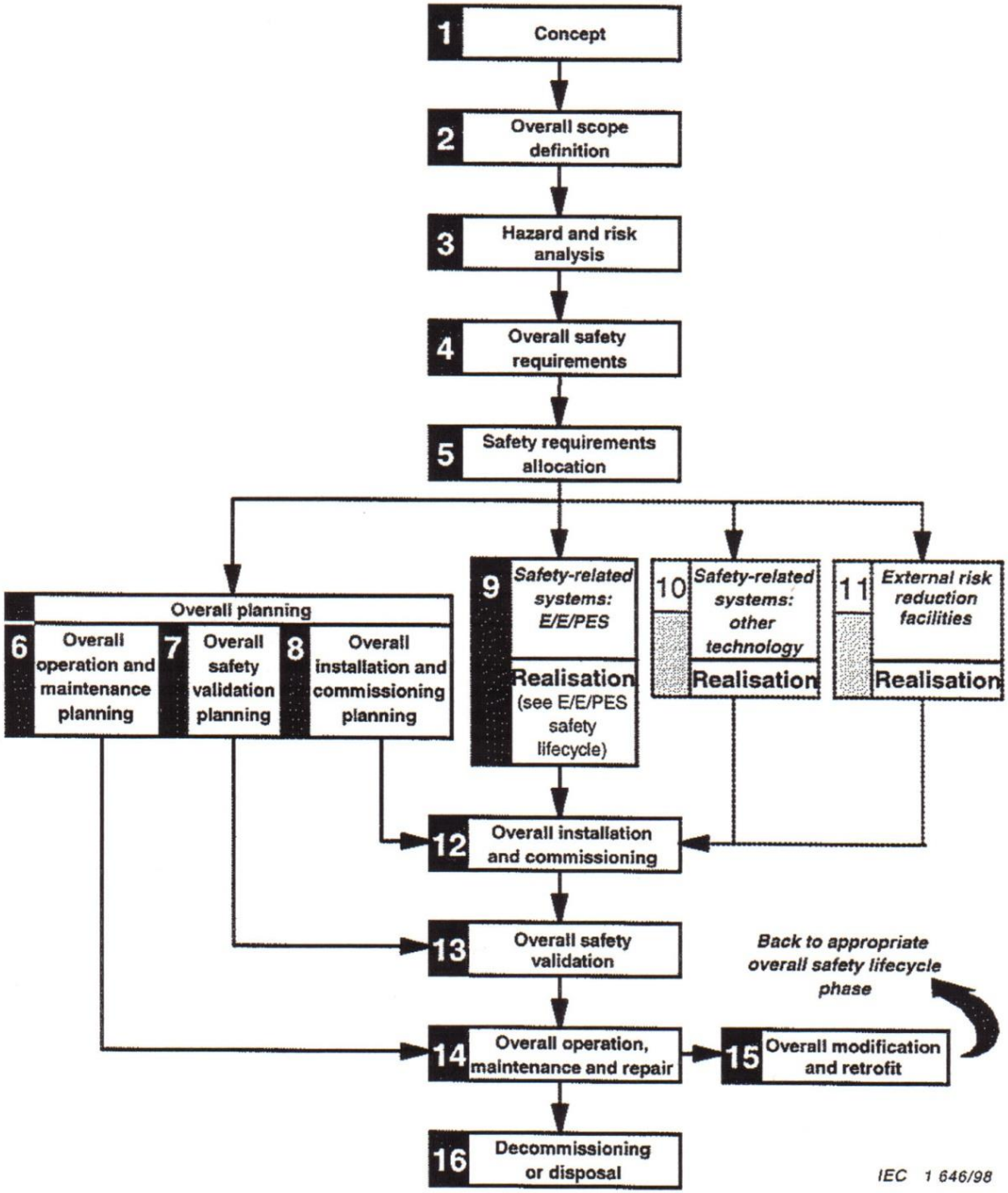
Bővített V-modell



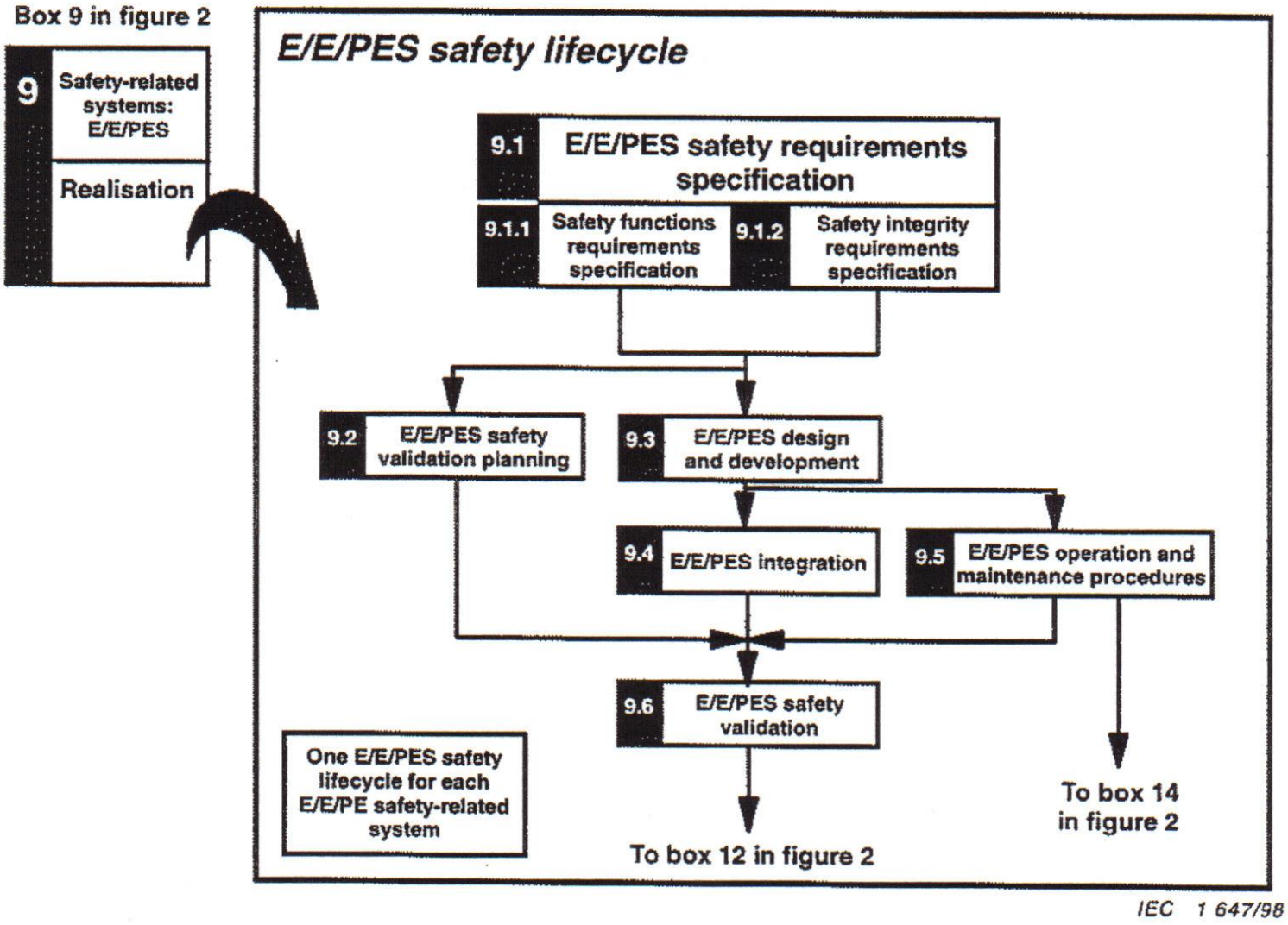
Bővített V-modell

- 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

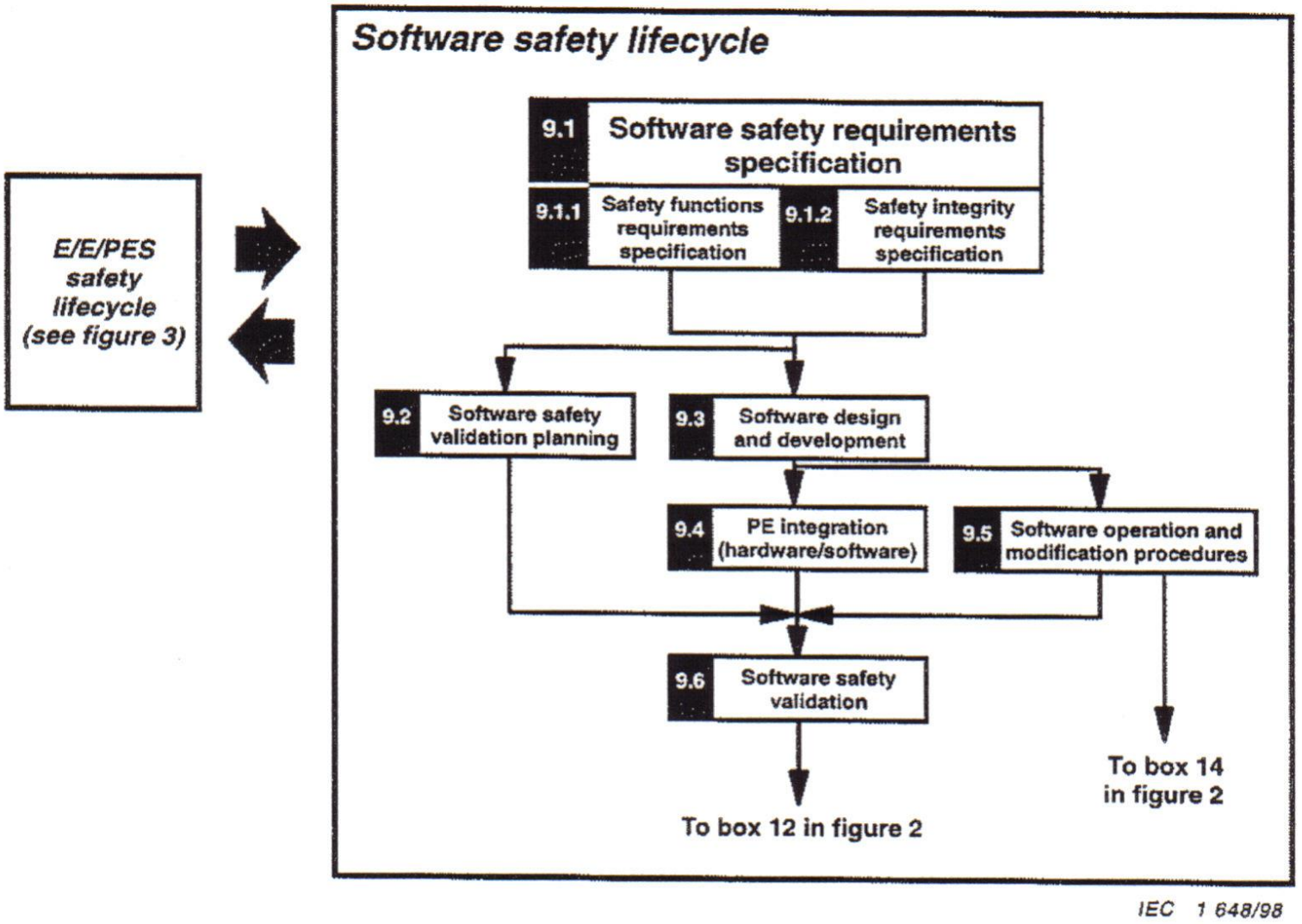
IEC 61508 biztonsági életciklus m.



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

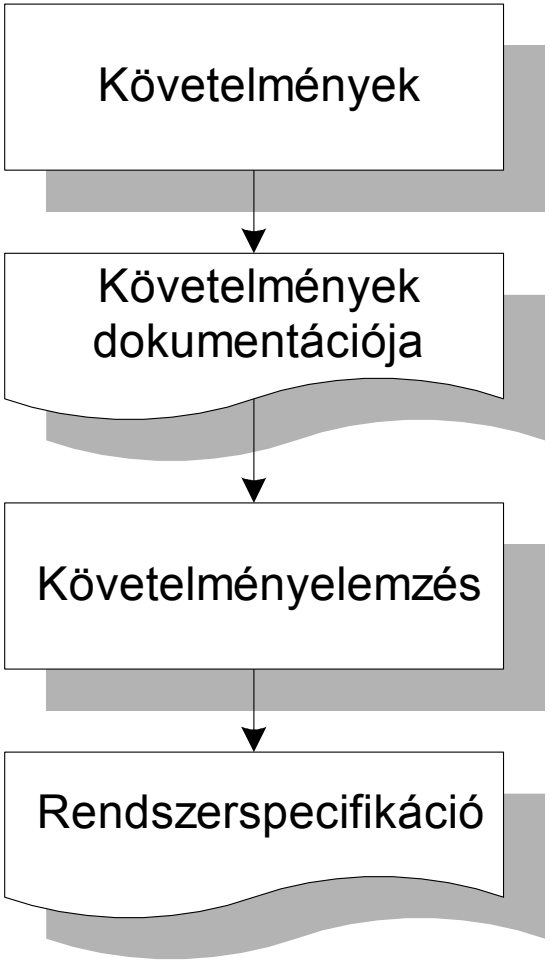
Fejlesztési módszerek, lépések

- Felhasználói követelmények
  - Funkcionális
  - Biztonsági
- Előzetes veszélyelemzés
- Specifikáció - a specifikáció animációja
- Top-level design
- Részletes tervezés
- A modulok megvalósítása és tesztelése
- Rendszer-integráció és rendszerteszt
- Engedélyezés

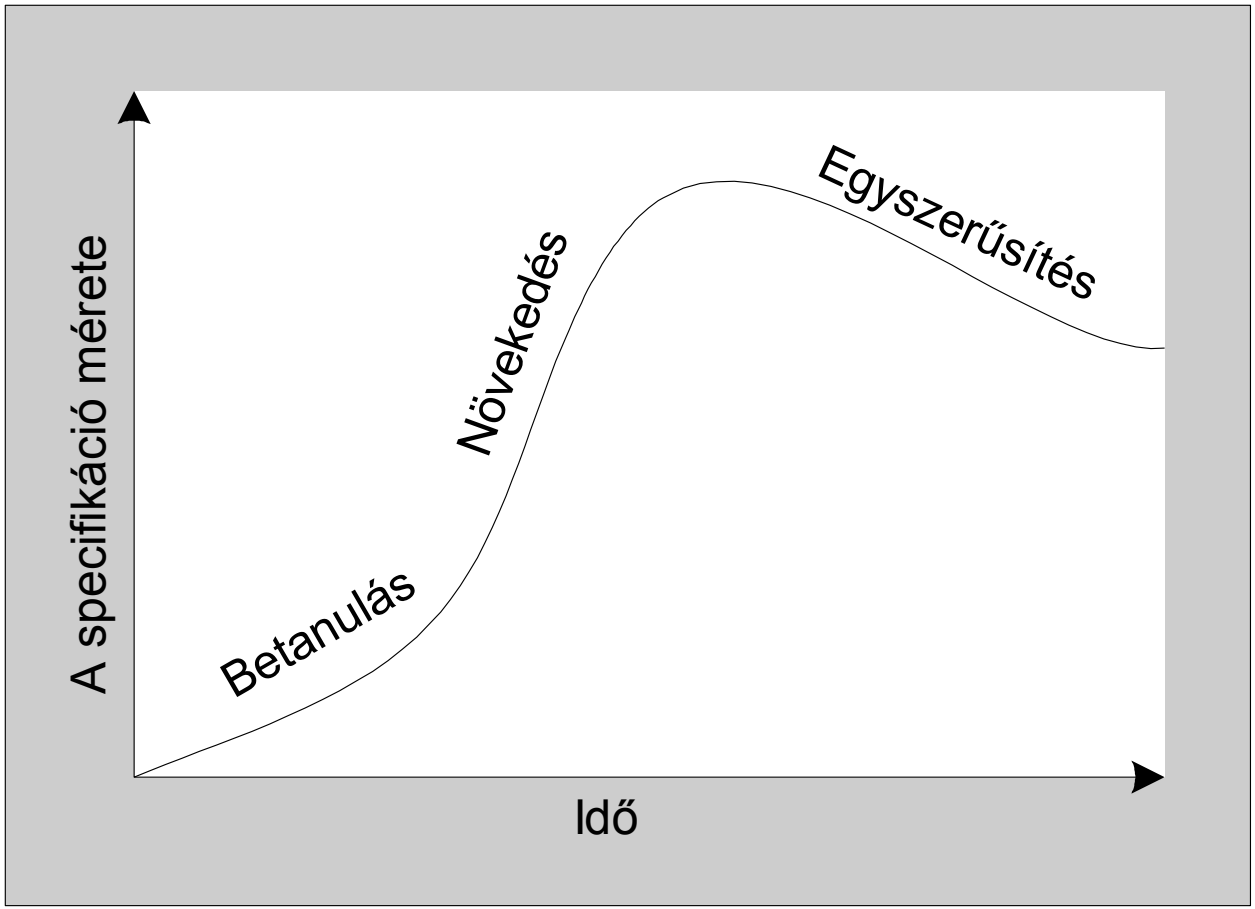
Specifikáció

- 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



A specifikáció kialakulása



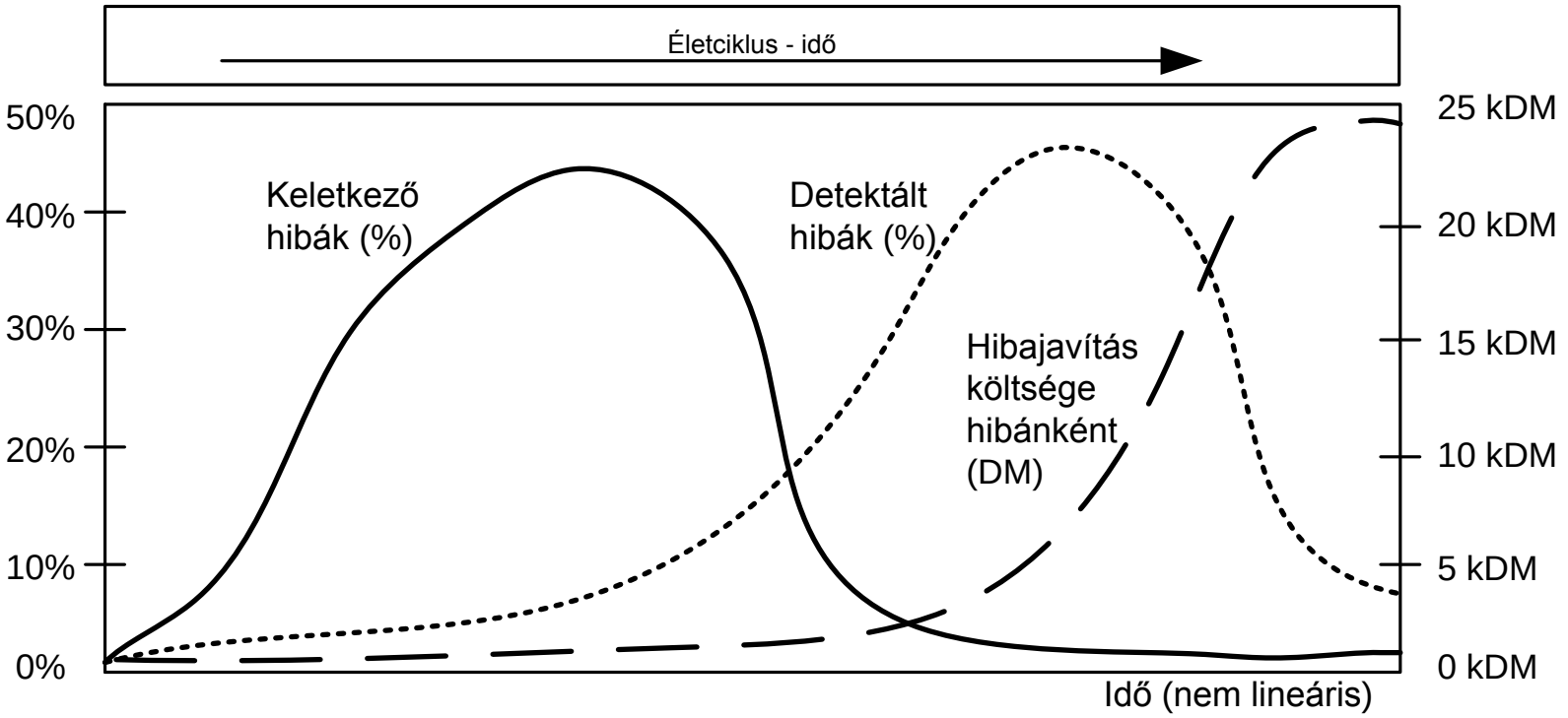
Az ideális specifikáció tulajdonságai

- 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

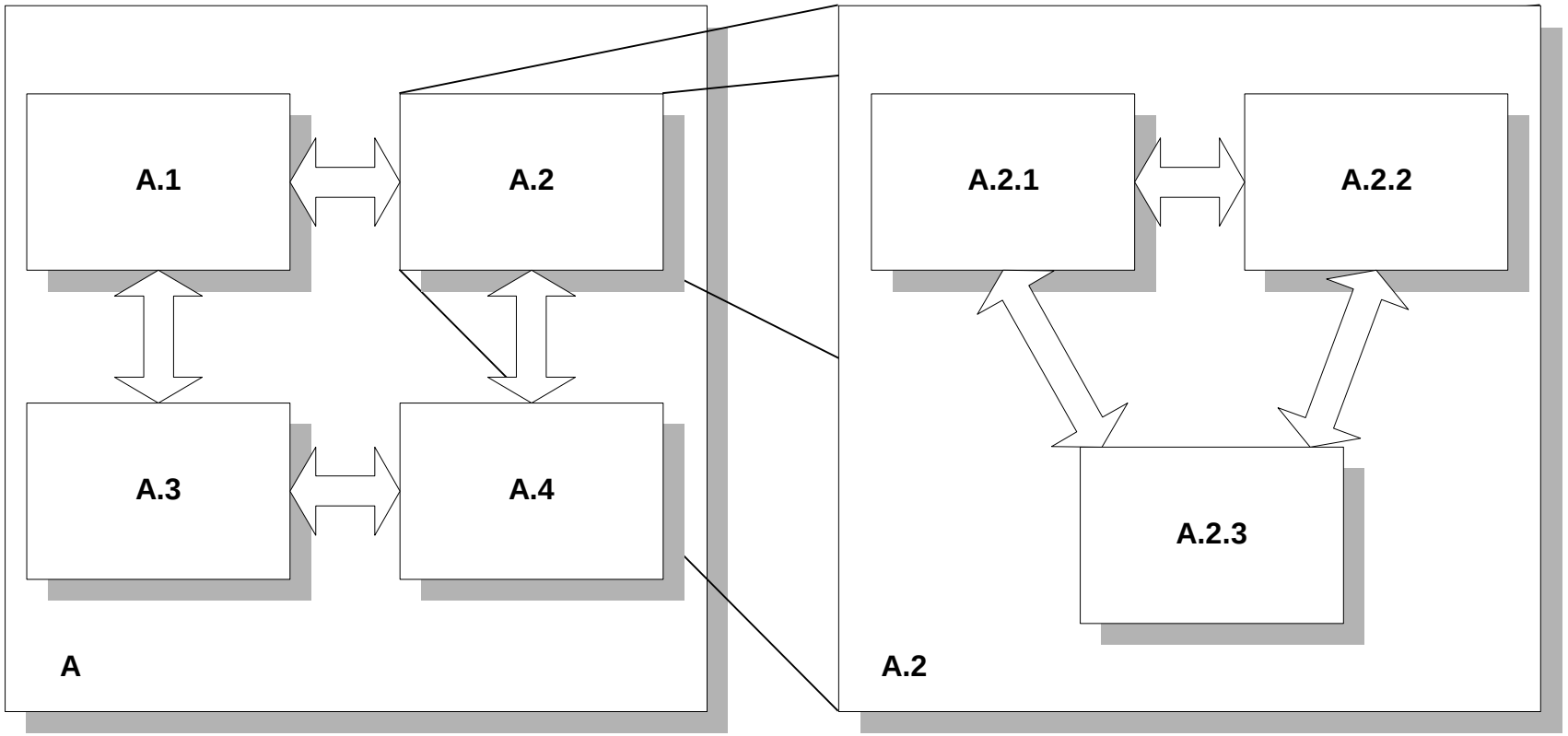
Hibák keletkezése és detektálása



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
- Modulok részletes tervezése
  - A dekompozíció gyakran iteratív (szubmodulok)

Dekompozíció



A modulok megvalósítása

- HW és SW modul implementáció
- Programnyelv választása
  - A programnyelv tulajdonságai
  - Fejlesztő eszközök elérhetősége
  - A fejlesztő csapat gyakorlottsága, tapasztalatai

Biztonsági folyamatirányító rendszerek szoftvere

Programozott irányítórendszerek

- Célgépek
  - Nincs operációs rendszer
  - Egyszerű szoftver
  - Pl. egy-chipes mikrokontrollerek
- Univerzális alkalmazású rendszerek
  - Moduláris hardver (általában kártya rendszerű)
  - Tagolt szoftverfelépítés

Tagolt szoftverfelépítés

KONFIGURÁCIÓ

Az adott alkalmazási hely konfigurációja; általában adatbázis

ALKALMAZÓI/FELHASZNÁLÓI SZOFTVER

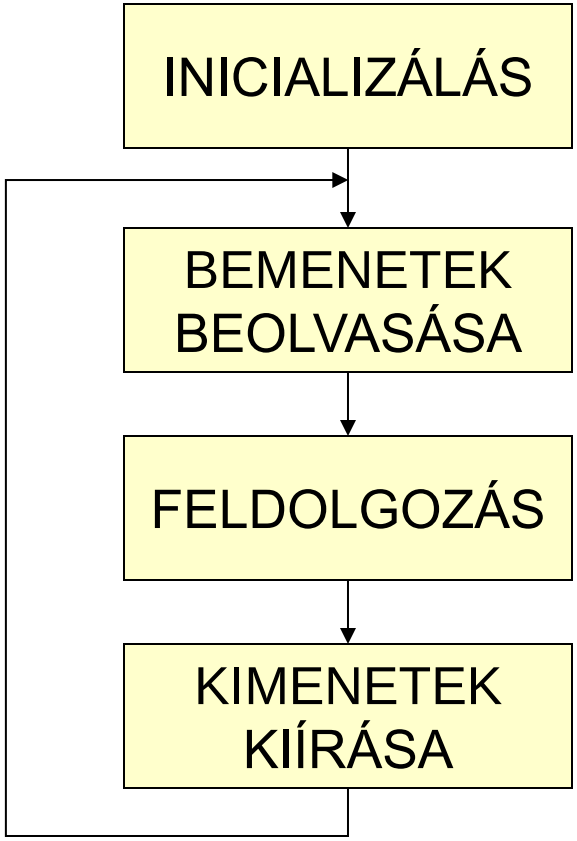
Alkalmazás-specifikus funkciók,
pl. általános jelzőlámpa vezérlés

OPERÁCIÓS RENDSZER

A folyamatirányítással kapcsolatos általános alapfunkciók

Az operációs rendszer feladatai

A folyamatirányító rendszerek szoftvere általában ciklikus működésű:



A biztonsági operációs rendszer többletfeladatai

- Többcsatornás működés támogatása
  - Adatcsere (beolvasás után, kiírás előtt)
  - Összehasonlítás
  - Csatornák szinkronizálása
  - Diszkrepancia-analízis / időablak
- Futás közbeni tesztek
  - Minden ciklusban v. ritkábban
  - Kommunikáció tesztelése
  - stb.

Cél: elsődlegesen hardverhibák feltárása,
ritkábban szoftverhibák feltárása.

Szoftverek megbízhatósága

- A szoftverek működőképességének valószínűsége **független az időtől** (amennyiben nem változtattuk meg).

- Szoftverek esetében nem beszélhetünk meghibásodásról:
  - A szoftver megbízhatóságát „csak” az eredeti, szisztematikus, specifikációs, tervezési és megvalósítási hibák csökkentik.
  - A szoftvert tároló alkatrész meghibásodása hardver-meghibásodás.
  - DE! Emberi beavatkozással egy jó szoftvert is el lehet rontani, például:
  - **Újra-bekerülési hiba**: átlagosan minden harmadik hibajavítással újabb hibát idézünk elő.

- A szoftver akkor működőképes – és így biztonságos –, ha a követelményeket (specifikáció) jól fogalmaztuk meg, és a megvalósítás (implementáció) is helyes.

Követelmények a biztonsági szoftverekkel szemben

Cél: **hibamentesség**.

Szoftver-megbízhatóságot növelő módszerek

- Jól strukturáltság
- Moduláris felépítés
- Áttekinthetőség – szükséges az ellenőrzéshez is
  - Modulonként kevés be/kimenet (lehetőleg 1-1) → könnyű tesztelni
  - Jól definiált interfészek
  - Feltétel nélküli ugrások (GOTO) kerülése
  - Tesztelhetőség kialakítása – „tesztelés-barát tervezés”
- Jól dokumentáltság
  - Funkciók leírása
  - Interfészek leírása
- Nem-biztonsági részek: arra kell ügyelni, hogy a nem-biztonsági rész semmilyen módon ne legyen hatással a biztonsági részekre – **visszahatásmentesség**.

- Programozási technikák
  - Top-down
  - Bottom-up
  - Az előző kettő kombinációja
- Programozási koncepciók
  - **Defenzív programozás**: számítok arra, hogy a programozás során hibákat fogok elkövetni.
    - Passzív ellenőrzések: pl. ellenőrző összegek, hihetőség-vizsgálat
    - Aktív ellenőrzések: adatáramlástól független ellenőrzés, így tesztelhetők a ritkán aktív lefutási ágak is.
  - CASE: Computer Aided Software Engineering
    - Automatikus programgenerátorok: a generátor program helyességét kell bizonyítani.

Szoftverírás

- Programnyelv, fejlesztői környezet
  - Olyan programozási nyelv szükséges, amely a valósidejű (real-time) működést támogatja.
  - Programozási nyelv szintje
    - Alacsony szintű programnyelv (pl. Assembly):
      - gépközeli, rugalmas,
      - de a szoftveríró nincs rákényszerítve a strukturált programozásra.
    - Magas-szintű programnyelv:
      - a nyelv szabályai rákényszerítenek a biztonságos programírás szabályaira,
      - de nem gépközeli, ezért a folyamat-vezérlést nehezebb programozni,
      - fordítóprogram (compiler) szükséges: ennek helyes működését is igazolni kell!
  - Programnyelv választásának kritériumai
    - A programozó mennyire jártas az adott nyelven való programozásban?
    - Mennyi tapasztalat van az adott programnyelvvel?
    - Van-e elterjedt, nagy valószínűséggel hibamentes fordítóprogram?
    - Mekkora az adott programnyelv/fejlesztőkörnyezet támogatottsága (pl. szimulációs háttér)?

Szoftver tesztelés

- Az eredeti hibák kiküszöbölésének leggyakoribb eljárása a **hibaeltávolítás**. Lépései:
  - tesztelés
  - diagnózis
  - javítás.

- **Tesztelés**
  - Statikus: a rendszer működtetése nélküli tesztelés. Ez végrehajtható
    - a rendszeren magán vagy
    - a rendszer alkalmas modelljén végrehajtott vizsgálatokkal.

    Formái:
    - statikus analízis (program-átvizsgálás, szimbolikus végrehajtás, adatfolyam analízis stb.)
    - helyességbizonyítás (induktív bizonyítás)

Szoftver tesztelés

- Dinamikus tesztelés: a rendszer működtetése révén
  - Tesztbemenetek kiválasztásának kritériumai
    - A tesztelés célja szerint
      - » konformitás tesztek: a specifikáció teljesítésének vizsgálata
      - » hibakereső tesztek: hibák feltárására
    - Rendszermodell szerint
      - » funkcionális teszt (black box, feketedoboz): csak a be/kimenetek viselkedését vizsgálom.
      - » strukturális teszt (white/glass box, „fehér/üveg doboz”): a teljes belső szerkezet figyelembevételével tesztel.
      - » kombináció (grey box, „szürke doboz”): nagy vonalakban (nem részletekbe menően) teszi láthatóvá a tesztelendő egység belső struktúráját is.
  - Tesztbemenetek generálása
    - Determinisztikus tesztek: a tesztmintákat előre meghatározzák
    - Valószínűségi (véletlen vagy statisztikus) tesztek: a tesztmintákat valószínűségi eloszlás alapján választják ki.

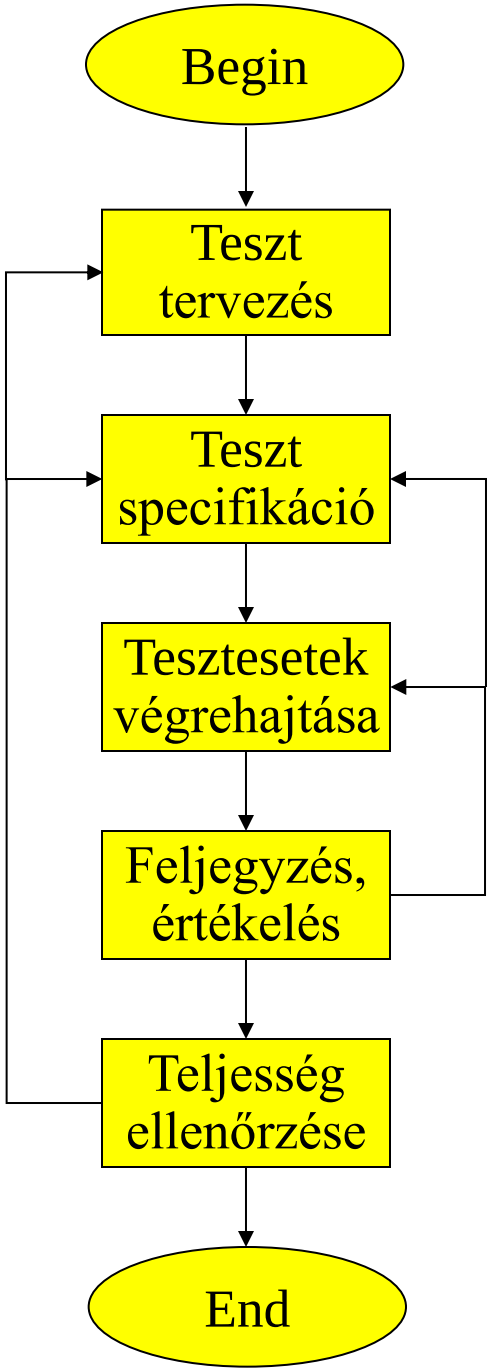
Szoftver tesztelés

- A tesztkimenetek figyelésével dönthető el, hogy a teszt-feltételek teljesültek-e.
  - Összes tesztkimenet figyelése
  - A tesztkimenetek kompakt reprezentációja

- Referencia
  - Kimeneti eredmények szimulálása
  - Referenciarendszer („golden unit”)
  - Specifikáció
  - Prototípus
  - Másik implementáció

- Tesztelési fokozatok
  - Modulteszt
  - Több modul kapcsolatának tesztelése
  - Integrációs teszt: teljes kapcsolatrendszer + kommunikáció
  - Rendszerteszt: célhardveren való tesztelés
  - Átvételi teszt: a megrendelő végzi
  - Javítás utáni teszt

A TESZTELÉS FOLYAMATA



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

Megbízhatóság az informatikai rendszerekben

Az információ

- Minden intelligens rendszer „hajtóanyaga”
- Az információ minőségi jellemzői
  - Sértetlenség
  - Biztonság
  - Adatvédelem
  - Titkosság
  - Hitelesség
  - Rendelkezésre állás
  - Archiválhatóság
  - Könnyű kereshetőség
  - Könnyű kezelhetőség

Sértetlenség (Integrity)

- Az információt eredeti tartalmában és teljességében megőrzi, veszteség, módosítás és hozzáadás nélkül,
- bármilyen adatkezelési és transzformációs folyamat során.

Biztonság (Safety)

- Az információ olyan tulajdonsága, hogy az azt felhasználó rendszer működése nem vezet veszélyeztető állapothoz (életveszély, egészségi kár, anyagi kár, környezeti ártalom).

Adatvédelem/adatbiztonság (Security)

Az információ védelme

- véletlenszerű vagy
- illetéktelen szándékos
  - hozzáféréstől,
  - felhasználástól,
  - módosítástól,
  - megsemmisítéstől vagy
  - elérhetetlenné tételtől.

Titkosság (Privacy)

- Garancia az információ tulajdonosa számára, hogy
- az információt kizárólag arra a célra használják, amire ő szánta.

Hitelesség (Credibility)

- Garancia az információ felhasználója számára,
- hogy a kapott információ az illetékes kibocsátótól származik, és
- az információ kibocsátását szabályosan engedélyezték,
- tartalmának helyességét ellenőrizték.

Rendelkezésre állás (Timeliness)

- A kívánt információ kellő időben való elérhetősége

Könnyű kezelhetőség

- Az információ könnyű érthetősége és feldolgozhatósága akár
- a számítógépes munkaállomás személyzete, akár
- automatikus eljárás számára.

Könnyű visszakereshetőség

- Az információ azon tulajdonsága, hogy könnyen megtalálható valamely adatbázisban.

Archiválhatóság

- Az információ alkalmassága arra, hogy visszakereshető módon tárolják egy adatvédelmi célból létesített különleges adattárban.

Informatikai megbízhatóság

- További fogalmak
  - Veszélyeztetés
  - Támadás
  - Informatikai kockázat

Veszélyeztetés

- Veszélyeztetés vagy fenyegetés alatt a megbízhatóság potenciális megsértését értjük, beleértve a rendszer hozzáférés-védelmének sérülését is.
- Objektív veszélyeztetés
  - Természeti (term. jelenségek, katasztrófák)
  - Fizikai (elektromágneses sugárzás)
  - Műszaki (véletlen meghibásodás)

Veszélyeztetés

- Szubjektív veszélyeztetés
  - Nem szándékos
    - Emberi gondatlanság a rendszer életciklusának bármely szakaszában,
    - Nem megfelelően kiképzett személyzet
    - A felhasználó hibás tevékenysége
  - Szándékos
    - A belső környezetből (egy jogosított felhasználó jogaival való visszaélés)
    - Külső környezetből (pl. hacker behatolása)

Támadás

- **Sebezhető hely:** olyan rendszerrész, amelyen keresztül a külső környezetből kiinduló veszélyeztetések kedvezőtlen hatással lehetnek a rendszerre.
- Támadás: a sebezhető helyeknek a rendszerösszetevők szándékos károsítása céljából történő felhasználása.
- Támadás hatására: az összetevőkben (információ, adat, HW, SW) károk keletkezhetnek.
- Ez a rendszer működésének megakadályozásához és illetéktelen jogosultság megszerzéséhez vezethet.

Kockázat

- Annak valószínűsége, hogy valamely meghatározható veszélyeztetés támadja a rendszert a sebezhető helyeken keresztül.
- Kockázat csökkentése: sebezhető helyek számának csökkentése → helyes biztonságpolitika
  - szabályok, eszközök, intézkedések

Biztonsági IT rendszerek kialakítása

- Problémák
  - Minden konkrét rendszer más → sem modellt, sem a kialakítási javaslatot, sem a biztonság megvalósításának módját nem lehet teljesen szabványosítani.
  - Az IT rendszerek összetettek, dinamikusak, interaktívak és rendszerint hibák is vannak bennük.
  - Az IT rendszer környezetének működése bizonytalan.

Biztonsági IT rendszerek kialakítása

- A megvalósított rendszer kompromisszum az elérni kívánt biztonság és a megvalósítás költségei között.
- A kialakítás során fontos szerepe van a következőknek (módszerek, stratégiák):
  - sebezhető helyek felderítése
  - a sebezhetőség mérséklése
  - veszélyeztetés modellezése
  - támadások osztályozása
  - kockázatbecslés
  - tesztelés, verifikáció, validáció

A biztonsági IT rendszer megvalósításának lépései

1. Biztonsági cél meghatározása
2. Veszélyeztetési modell megalkotása
3. Felelősség meghatározása
4. Kockázatok elemzése
5. Biztonsági intézkedések kidolgozása
6. Jóváhagyás
7. Implementáció

A biztonság megvalósításának eszközei

- A biztonsági funkciók biztonsági eszközök révén valósíthatók meg.
- Négy kulcsterület:
  - Hitelesítés (authentication)
  - Vizsgálat (audit)
  - Hozzáférés-ellenőrzés (access control)
  - Titkosítás (encryption)

Hitelesítés

- A rendszer azonosítja a felhasználót, vagy felhasználó csoportot és
- verifikálja a felhasználók azonosságát.
- Alapelvek
  - Tudáson,
  - Tulajdonságon és
  - Sajátosságon alapuló

Tudáson alapuló hitelesítés

- Név+jelszó
- Jelszó biztonsági követelményei
  - hosszúság, számok, egyéb karakterek, ne legyen szokásos szó, gyakori módosítás.
- Előny
  - többnyire szoftveresen is megvalósítható
- Hátrány
  - nagy sebezhetőség a jelszó elfogásával
  - gyakori változtatás → rövid élettartam
  - a jelszó mintáját el kell helyezni a rendszerben.

Tulajdonságon alapuló hitelesítés

- Kiegészítés a tudáson alapuló hitelesítéshez.
- Chipkártya
  - A hitelesítéshez szükséges információt a kártya tartalmazza → nagyfokú biztonság
  - Nagy memóriakapacitás, nehéz másolni, további funkciók lehetősége
  - Speciális hardver szükséges

Tulajdonságon alapuló hitelesítés

- Mágneskártya
  - Kis kapacitás
  - Olcsó
  - Könnyen másolható, módosítható

Sajátosságon alapuló hitelesítés

- Biometrikus módszer
- A felhasználó fizikai tulajdonságain alapul
- Biztonságos
- Nem hamisítható, nem duplikálható, nem lehet elfelejteni
- Módszerek: ujjlenyomat, tenyér geometria, arc felismerés, aláírás dinamika, retina, szivárványhártya, hangfelismerés
- Nagy hardverigény, költséges (az előzetes mintafelvétel miatt is).

Biztonsági audit

- Az adatintegritás megtartására és a hozzáférés fizikai és logikai vezérlésére szolgáló eszköz.
- Az IT rendszer letapogatása és aktivitásainak feljegyzése.
- Visszamenőleges elemzés

Hozzáférés ellenőrzése

- A felhasználóknak a munkájukhoz szükséges hozzáférési jogok kiutalása.
- Kinek milyen hozzáférése van az információkhoz.

Titkosítás

- Akkor szükséges, ha a szándékos támadások nem zárhatók ki.
- Nyilvános átviteli hálózatok alkalmazása.