



# Szoftvertchnológia

## 2.Népszerű élelciklus modellek

BSc kurzus

Dr. Balla Katalin



# Tartalom

- Szoftverfejlesztési / technikai / műszaki folyamatok
- Népszerű élelciklus modellek
  - Vízesés, V-modell, spirál, iteratív, inkrementális



# Szoftverfejlesztési folyamatok

- Software Engineering Processes – magyarul nevezzük őket még technikai / műszaki / mérnöki folyamatoknak is. A továbbiakban **a szoftverfejlesztés műszaki folyamataiként** említjük őket.
- Általában minden szakember egyetért abban, hogy a szoftverfejlesztés az alábbi folyamatokat tartalmazza:
  - ☐ Követelmények fejlesztése és menedzsmentje
  - ☐ Tervezés
  - ☐ Kódolás
  - ☐ Tesztelés
  - ☐ Átadás
  - ☐ Karbantartás
- Ezzel együtt, nem létezik egyetlen, mindenki által, mindig és mindenütt elfogadott megközelítése a szoftvermérnöki / szoftverfejlesztési tevékenységeknek (lásd 1. előadás).
- A szoftverfejlesztési tevékenységeket általában szoftverfejlesztési életciklus modellekben írják le. (Software Development Life Cycle models (SDLC)).




# Szoftver életciklus

- Egy időperiódus, ami akkor kezdődik, amikor a szoftverterméket kigondoljuk és akkor fejeződik be, amikor a szoftvert már nem használják. A szoftver életciklus jellemzően a következő fázisokat tartalmazza: koncepció, követelmény, tervezés, megvalósítás, teszt, installáció és ellenőrzés, operáció és üzemeltetés, valamint időnként egy leállítási fázist.
  - Megjegyzés: a fázisok átfedhetik egymást, vagy akár ismétlődhetnek is
  - Forrás: Szoftvertesztelés egységesített kifejezéseinek gyűjteménye
  - [http://www.hstqb.com/images/d/d8/HTB-Glossary-3\\_2.pdf](http://www.hstqb.com/images/d/d8/HTB-Glossary-3_2.pdf)



# Szoftver életciklus modellek

- A szoftver életciklus tevékenységeit modellezzik.
- A szoftverfejlesztési modelleket a konkrét projekt és termék jellegzetességeinek függvényében testre kell szabni
- Mindig a projekt céljának, a készülő termék típusának, a termék működésével kapcsolatos kockázatoknak megfelelő életciklus modellt kell kiválasztani a szoftverfejlesztéshez.
  - Például, egy kis, belső adminisztráció segítő szoftvert másképpen kell elkészíteni, mint egy nagyobb, biztonságkritikus rendszert (pl. egy autó fékrendszerét vezérlő szoftvert). Az alkalmazott életciklus modell is különböző lesz.
  - ISTQB Foundation Level Syllabus alapján



# Szoftvermérnöki / szoftverfejlesztési módszertanok

- A szoftverfejlesztés tevékenységeit modellezzik.
- Megadják annak a módját, ahogyan a problémákat meg lehet oldani; az elvégzendő feladatok sorozatát is leírják.
- Irányt mutatnak , támogatás nyújtanak a munka során.
- Felhasználják a korábbról összegyűlt tapasztalatot hasonló problémák megoldására.
- Nagyobb valószínűséggel elérjük célunkat, ha követjük ajánlásait.
- Elnevezés:
  - Életciklus modell vagy módszertan ?



# Szoftver életciklus modellek

- A szoftverfejlesztésben alkalmazott életciklus modelleket a következő kategóriákba szokás sorolni:
  - Szekvenciális modellek
  - Inkrementális modellek
  - Iteratív modellek



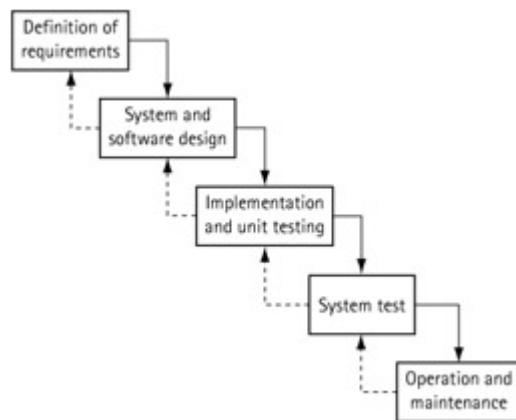
# Szekvenciális élekciklus modellek

- A szoftverfejlesztés folyamatát lineáris, szekvenciális folyamatként ábrázolják.
  - Ez azt jelenti, hogy minden fázis csak akkor kezdődhet, ha az előző fázis befejeződött; nem lapolódnak át a fázisok.
  - Két ismert példa: vízesés modell és V-modell.



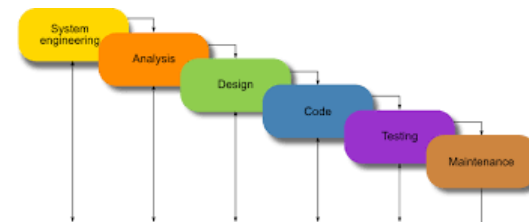
# Szekvenciális modellek

## Boehm klasszikus vízesés modellje



[Boehm76] **B. W. Boehm:** Software engineering.

*IEEE Transactions on Computers* C-25 (12) 1226–1241, 1976.



<https://airbrake.io/blog/sdlc/waterfall-model>



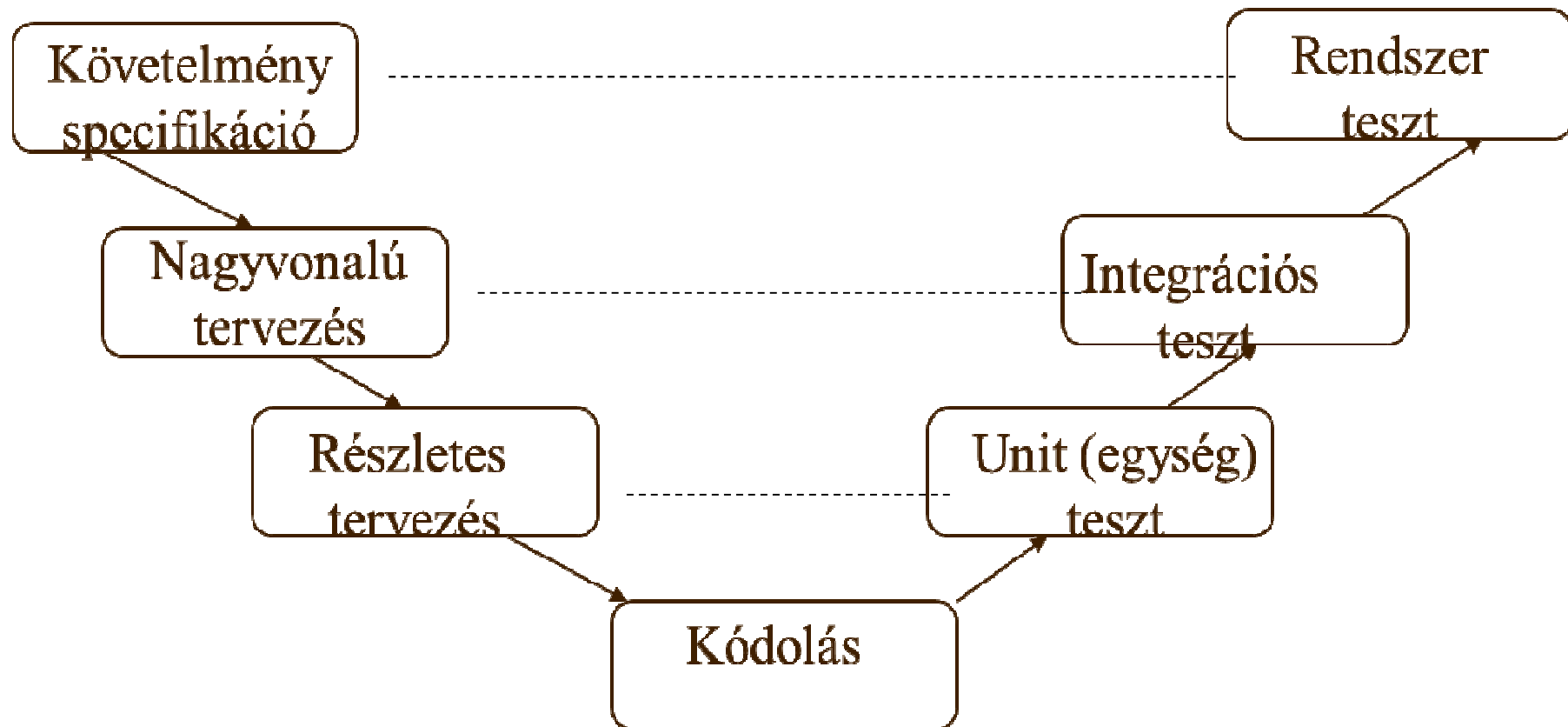
[https://www.youtube.com/watch?v=\\_ZKvvaZEFKE](https://www.youtube.com/watch?v=_ZKvvaZEFKE)



<https://medium.com/synapse-india/waterfall-model-of-software-development-a-sure-fire-practice-for-your-professional-software-needs-4c8997419800>

# Szekvenciális modellek

## ■ A V-modell



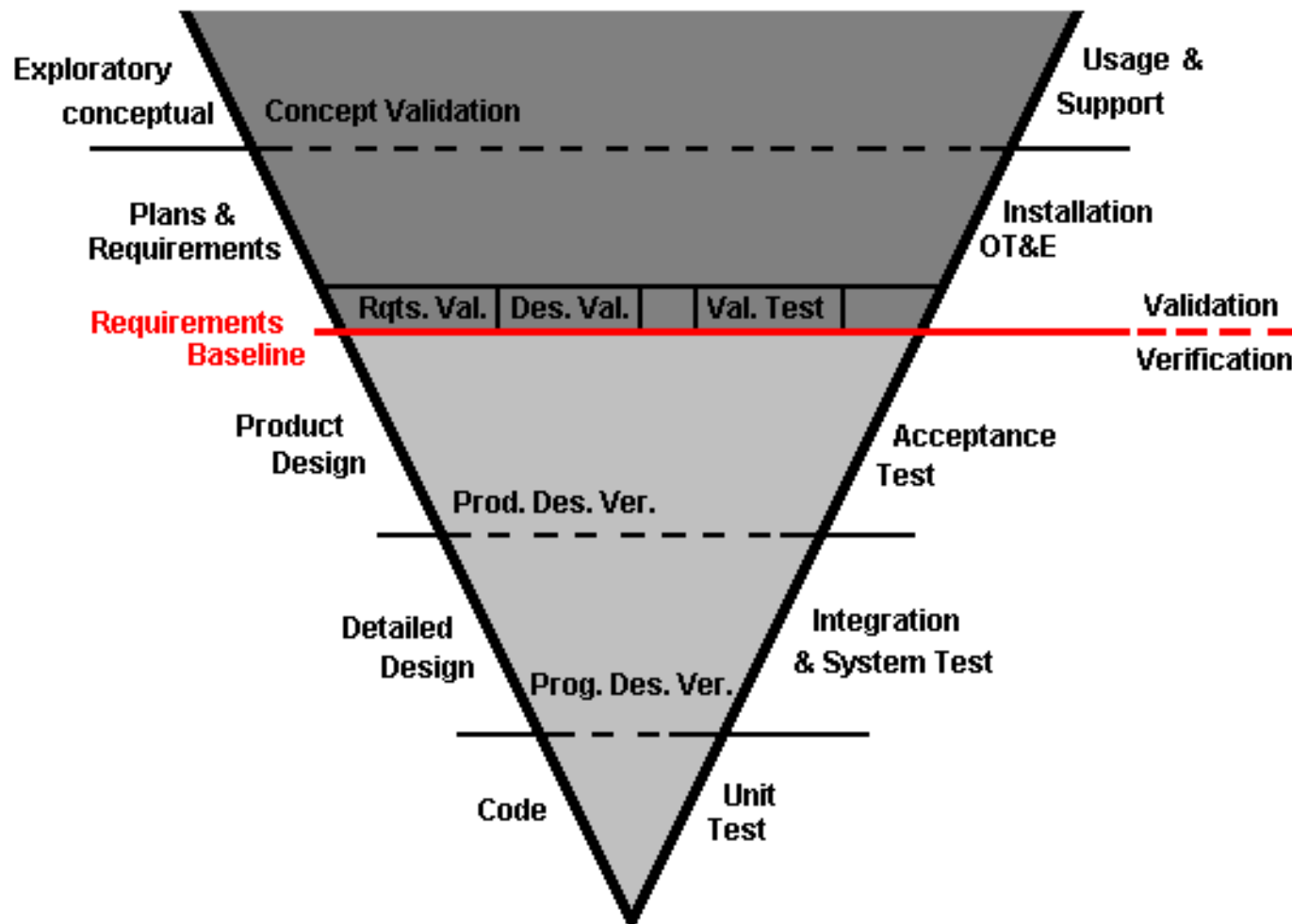
Egy adott fázisban megjelenített információkat felhasználjuk az adott fázishoz tartozó teszt esetek létrehozásában.



## Még több információ a V-modellről

- [/Myers, 1979/](#) The Art of Software Testing
  - The testing cycle has been structured to model the development cycle.
- [/Boehm, 1979/](#) Guidelines for Verifying and Validating Software Requirements and Design Specifications
  - "V-chart" which shows the context of verification and validation activities throughout the software lifecycle
- [/VM 1997/](#) V-Model 97, Lifecycle Process Model
  - Lifecycle Process Model -Developing Standard for IT Systems of the Federal Republic of Germany. General Directive No. 250. June 9
  - V-model: Regulations setting up all activities, products, and their logical interdependencies during the development and maintenance/modification of systems, realizing the system tasks predominantly by using IT, within the scope of the federal administration.
- [/Sommerville, 1999/](#) Software Engineering
  - V&V Process: is a whole life-cycle process. V&V must be applied at each stage in the software process.

# Még több információ a V-modellről



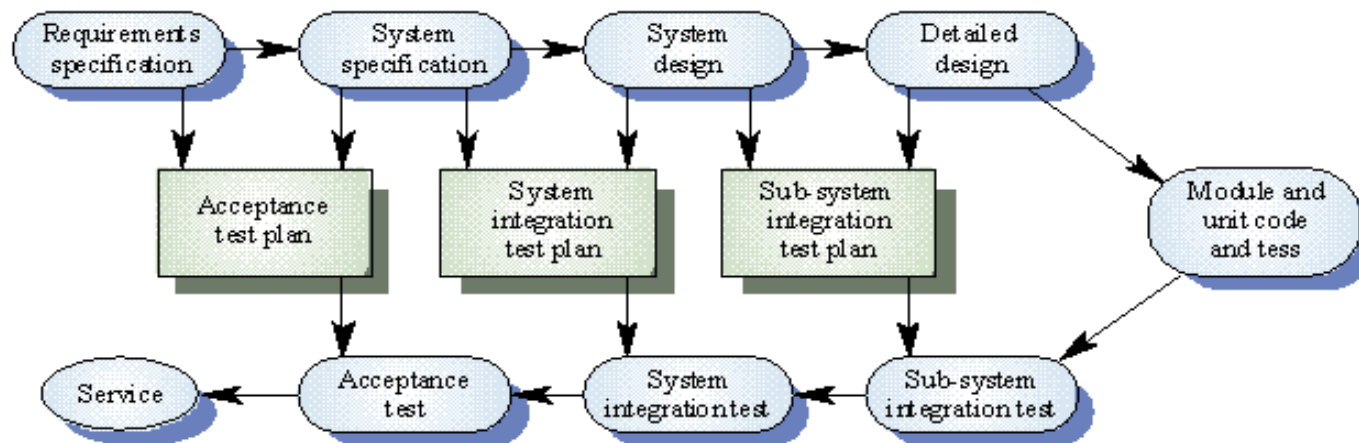
[Boehm, 1979/](#) Guidelines for Verifying and Validating Software Requirements and Design Specifications

Balla K.

# Még több információ a V-modellről

- V&V Process: is a *whole life-cycle process*. V&V must be applied at each stage in the software process. Has two principal objectives:
  - The discovery of defects in a system.
  - The assessment of whether or not the system is usable in an operational situation.

([Sommerville, 1999/](#) )



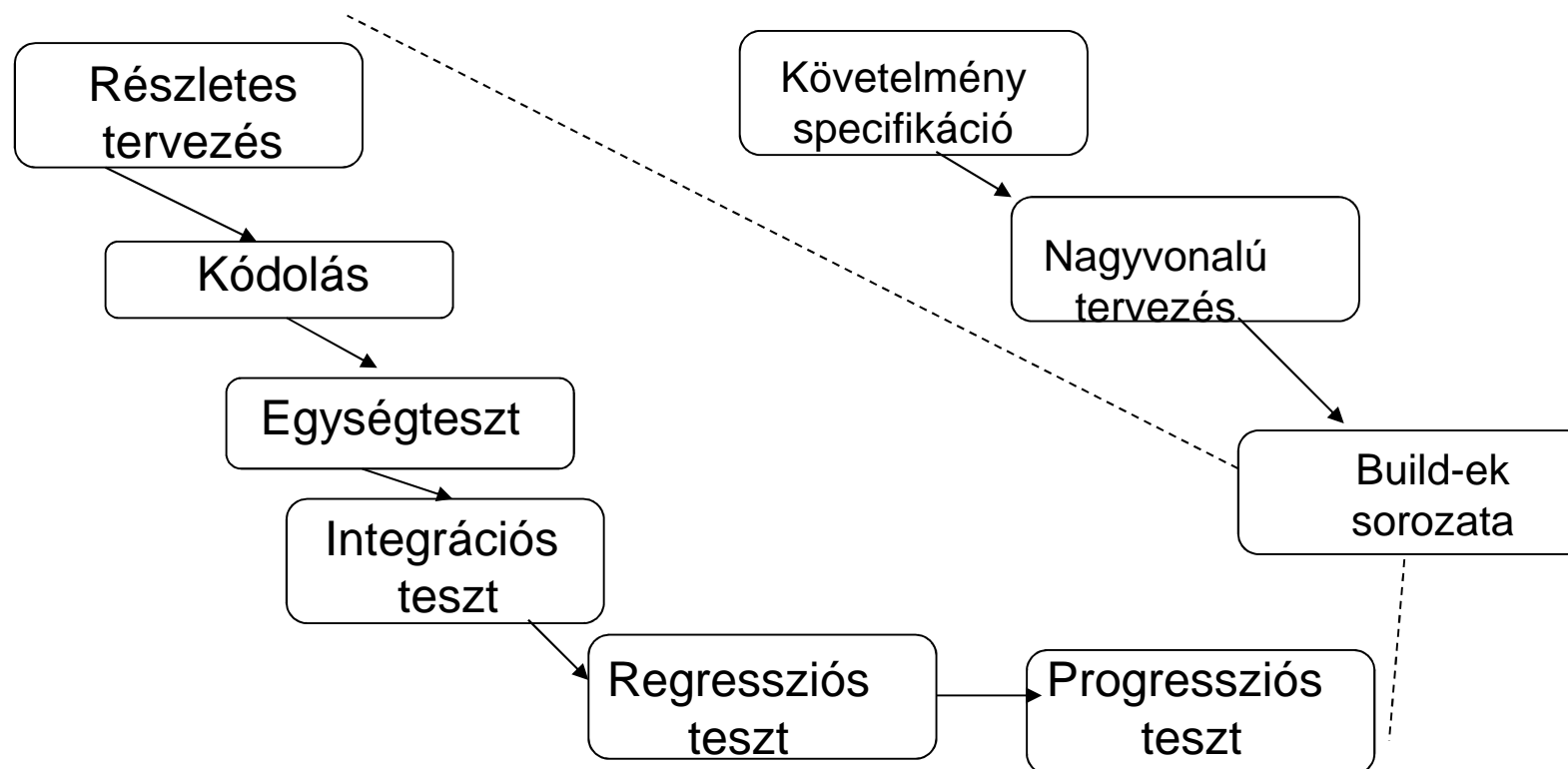


# Inkrementális modellek

- A követelményeket kisebb csoportokra bontják, és a követelményelemzés tervezés ,kódolás, tesztelés erre a kisebb követelménycsoportra történik, rövidebb idő alatt. A rövidebb fejlesztési ciklusokat inkrementumoknak nevezzük.
- Példák inkrementális modellekre: RUP, RAD, Spiral.
  - A termék “build”-jei inkrementálisan fejlődnek.
  - A felhasználó hamarabb kap visszajelzést.

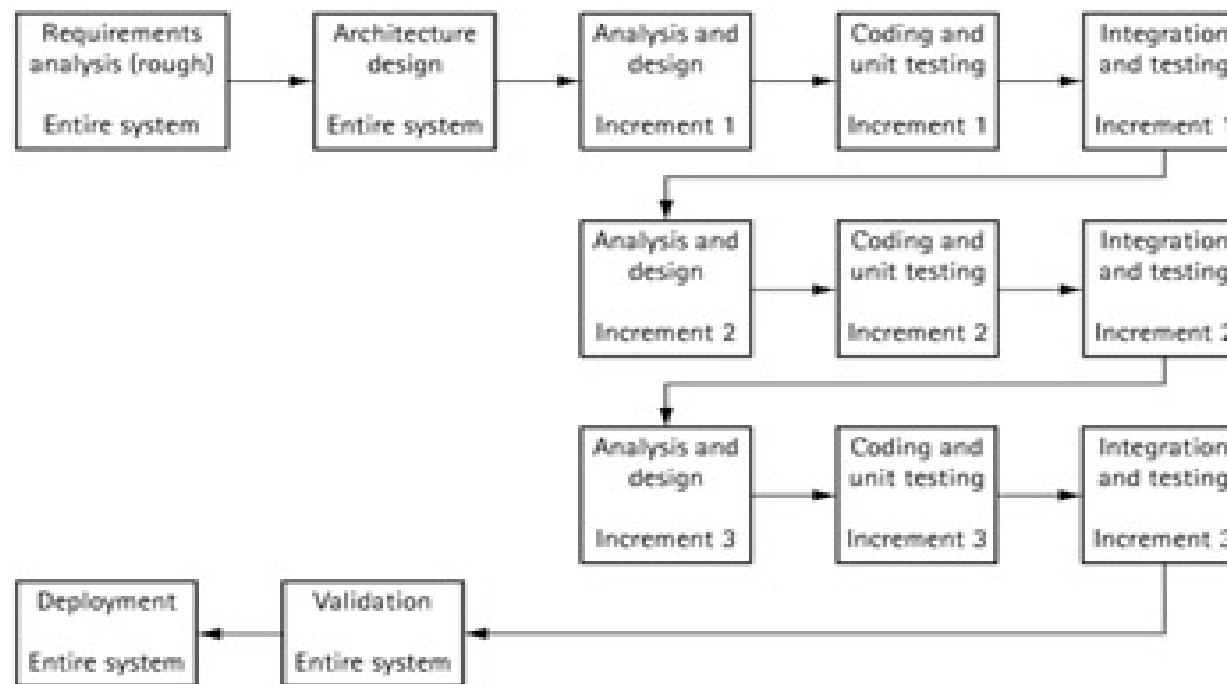
# Inkrementális modellek

## Példák



# Inkrementális modellek

## Példák



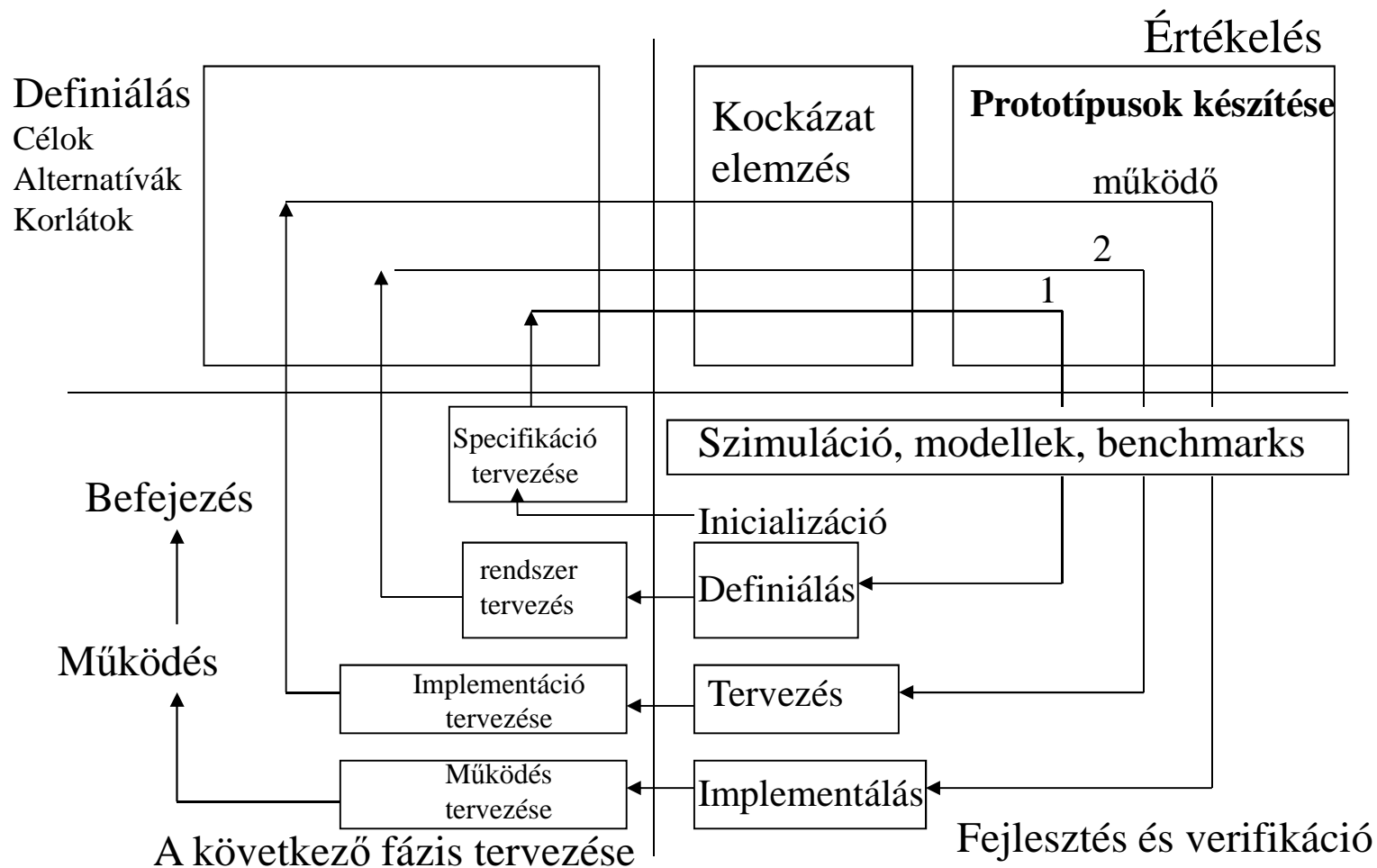
[Jacobson99] **Ivar Jacobson, Grady Booch, and James Rumbaugh:**  
*The Unified Software Development Process*, Object Technology Series.

Boston: Addison-Wesley, 1999



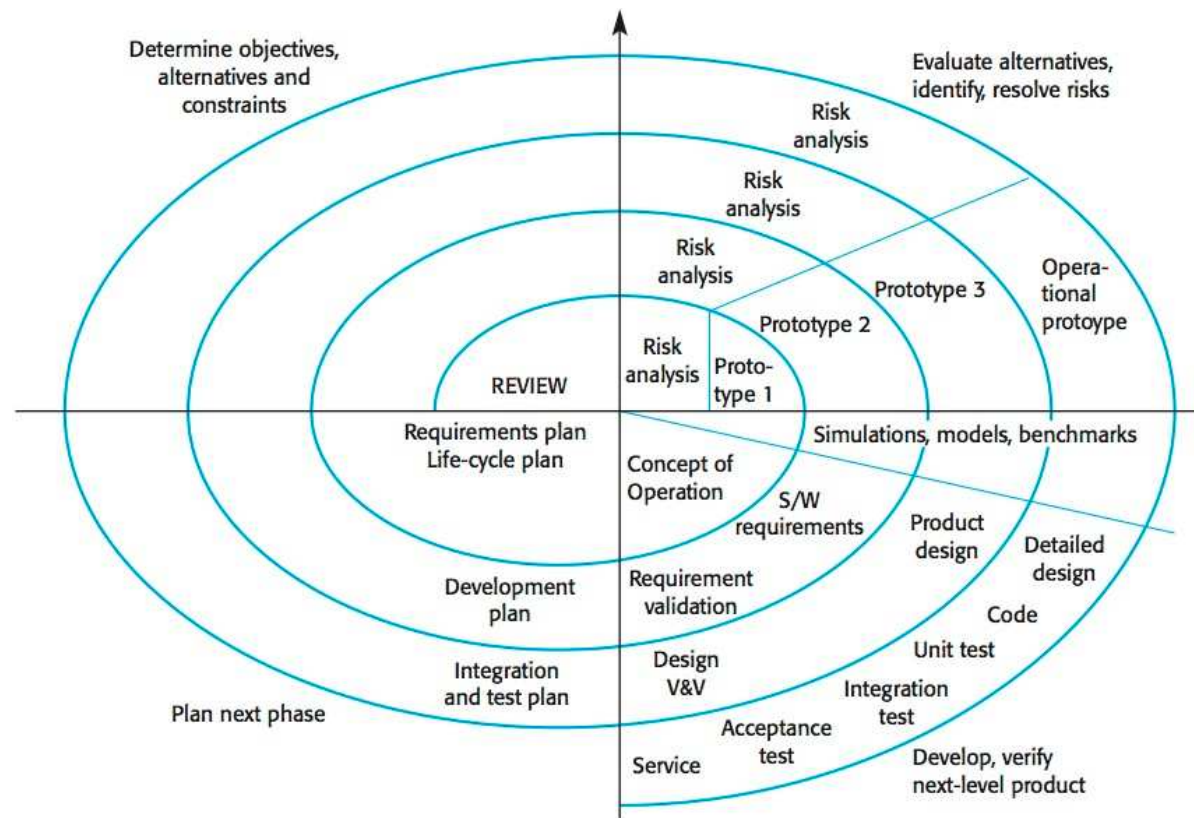
# Inkrementális modellek

## Spirál modell



# Inkrementális modellek

## Boehm spirál modellje



<http://iansommerville.com/software-engineering-book/web/spiral-model/>



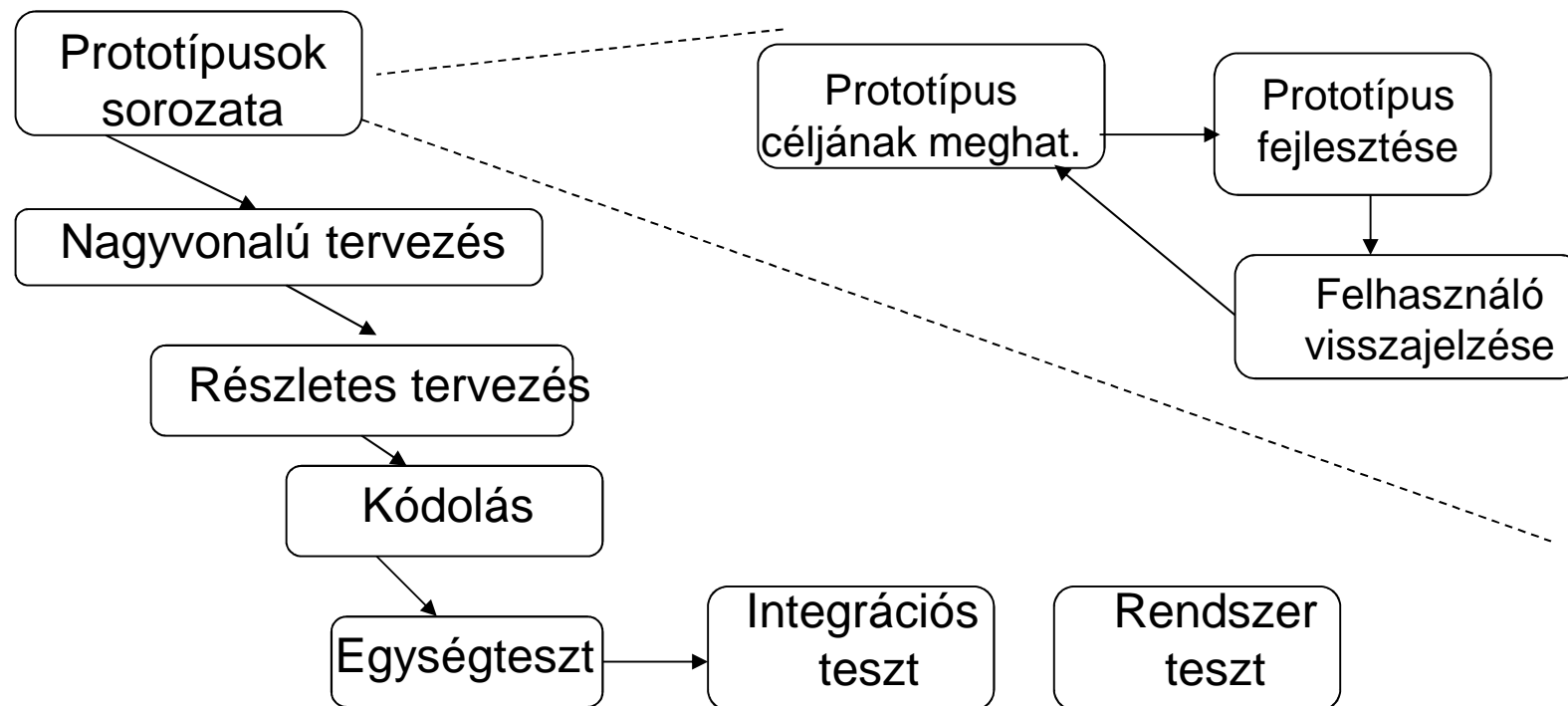
# Iteratív modellek

- Az iteratív modellek ciklikusak; előbb a követelményeknek egy kisebb (al)csoportját implementáljuk egy előre meghatározott időintervallum alatt – melyet iterációnak nevezünk.
  - Iteratív modellek tipikusan az **Agilis megközelítések** (Scrum, Kanban, XP, Lean) – lásd később, ebben az előadásban.

# Iteratív modellek

## ■ Evolúciós fejlesztés („rapid prototyping”)

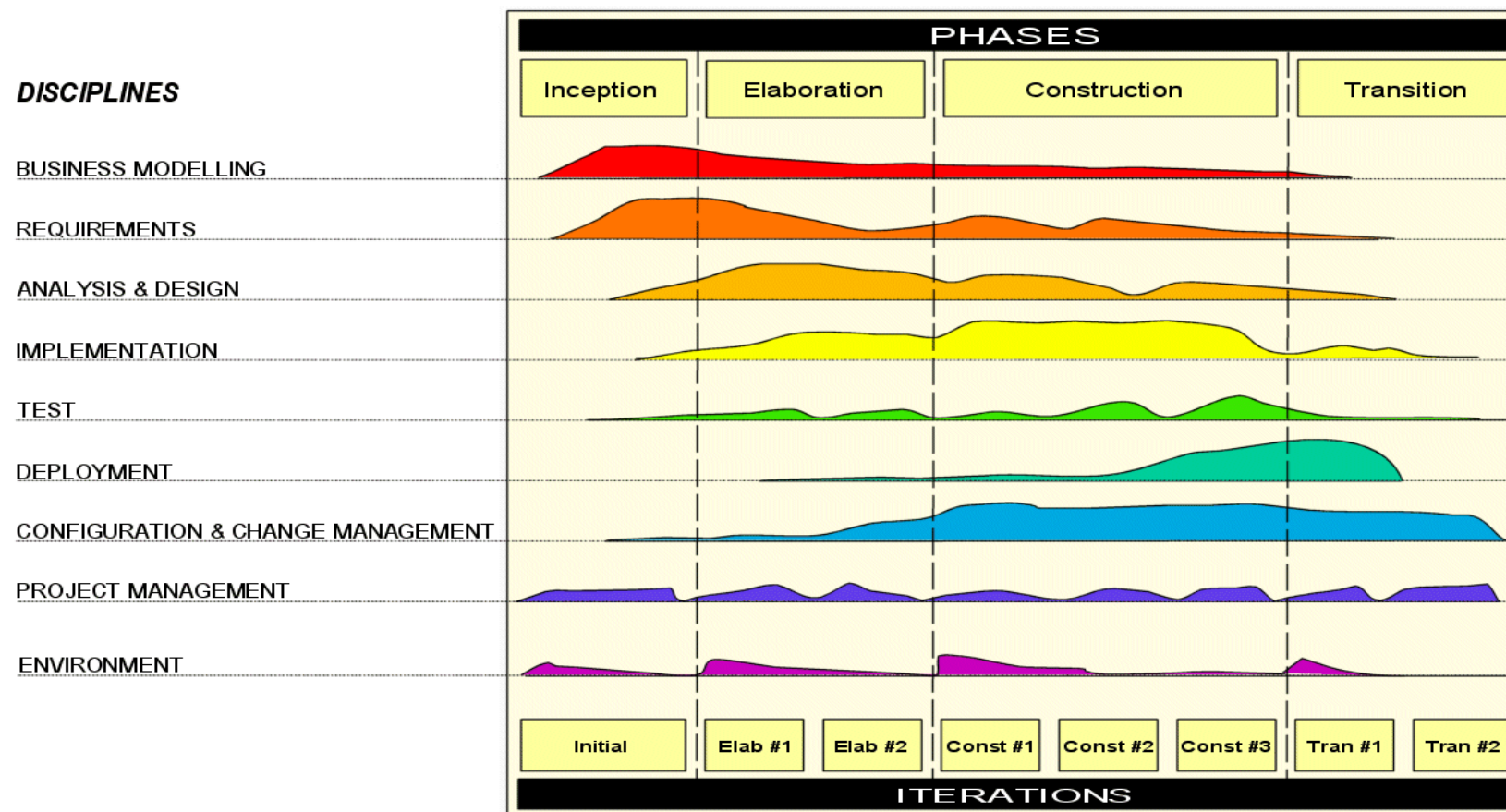
- Csak az első build-et definiálják, majd, a felhasználó prioritásainak alapján ezt egészítik ki újabb elemekkel



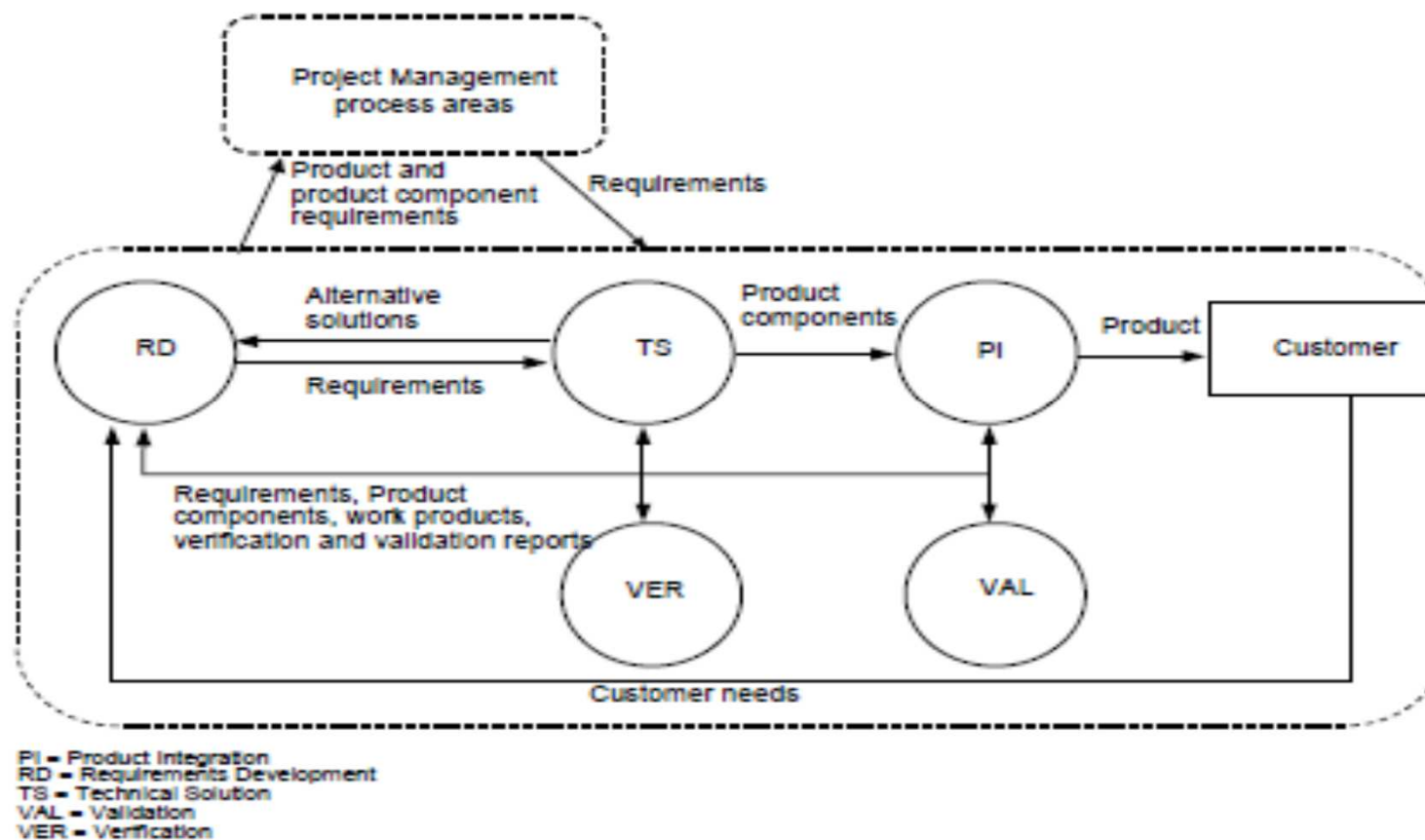


# Egyéb életciklus modellek / módszertanok

RUP: iteratív, inkrementális, use-case vezérelt  
(támogatja az OO fejlesztést)



# Szoftvermérnöki folyamatok a CMMI-ben

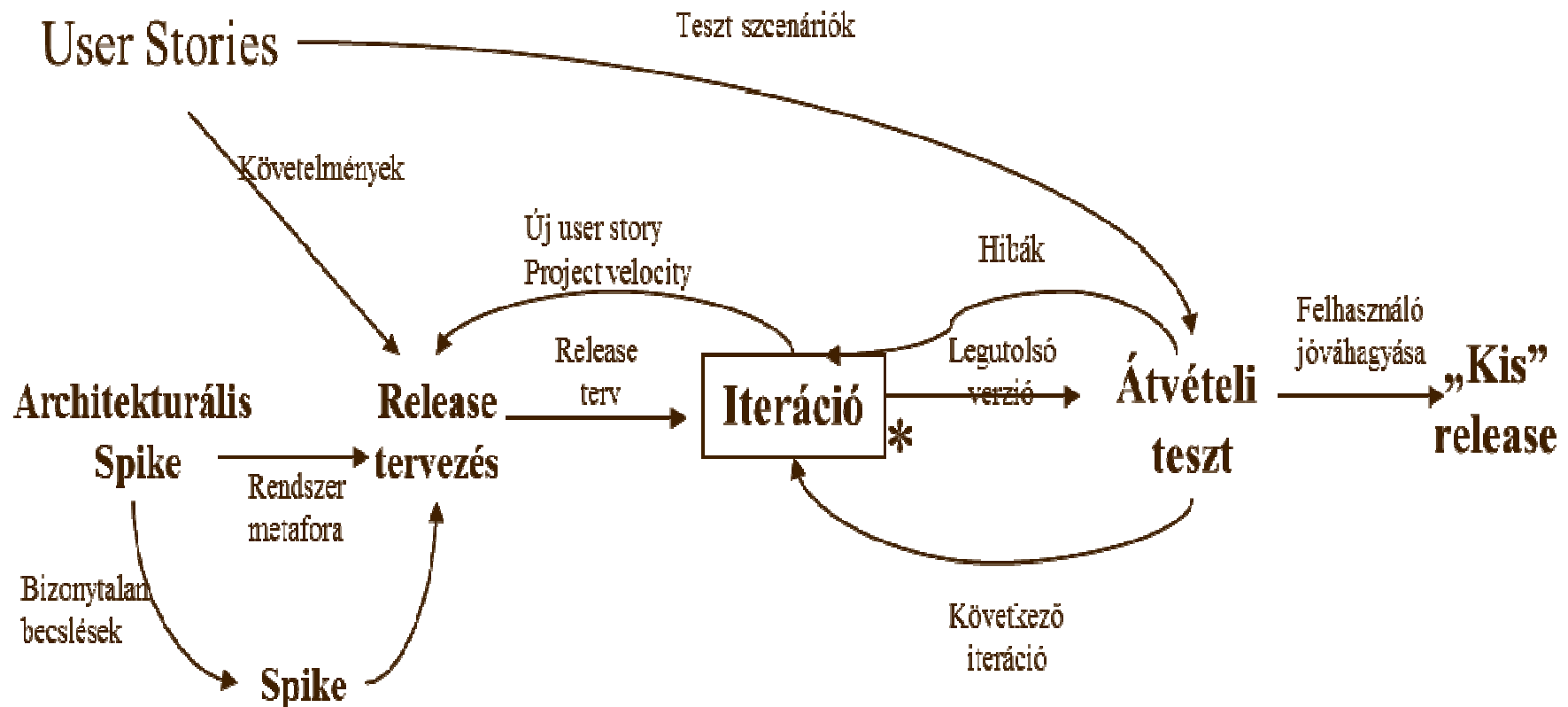




# Extrém programozás

- Gyors, változékony, rugalmas projektek módszertan
- Könnyedén és rugalmasan kezeli a változásokat
- A következő tényezőkre épít:
  - ☐ Egyszerűség
  - ☐ Kommunikáció
  - ☐ Visszajelzés
  - ☐ Bátorság (új dolgok kipróbálásában)
- Un. „user story”-kra épít
- Un. „Spike”-okat (megoldásokat) használ ismétlődő feladatokra
- A tervek elkészülése után rövid időtartalommal, iteratív módon történik a fejlesztés
- Ha a teszt elfogadásra került, kész az adott verzió
- Együtt van ügyfél, menedzser és programozó
  - ☐ **Megj: Életciklus modell, vagy programozási módszer? Mindkettő lehet. Itt azért említjük, mert a szoftverfejlesztéshez kapcsolódó folyamatokat ír le, sorrendet javasol a végrehajtásukra.**

# Extrém programozás



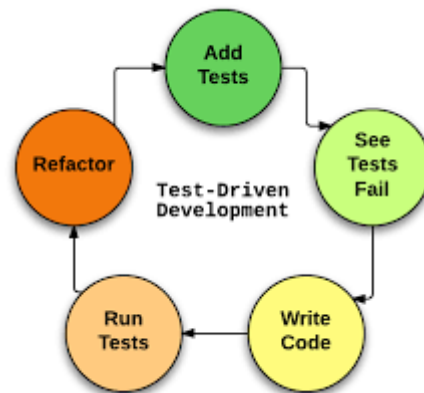
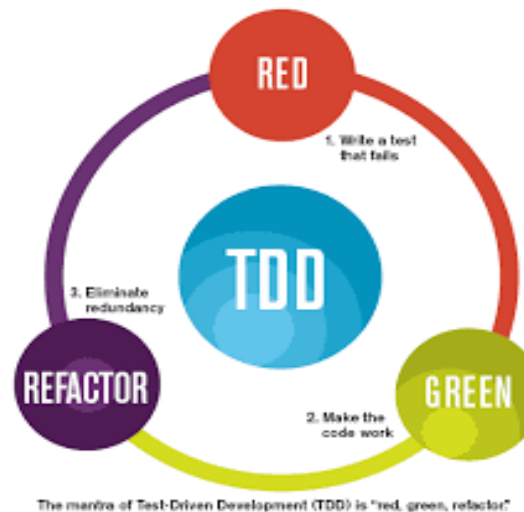
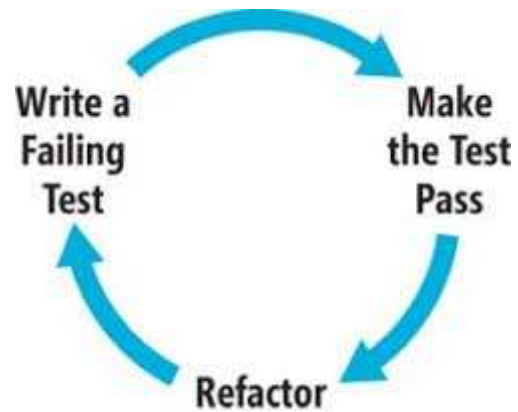




# Tesztvezérelt fejlesztés / Test Driven Development (TDD)

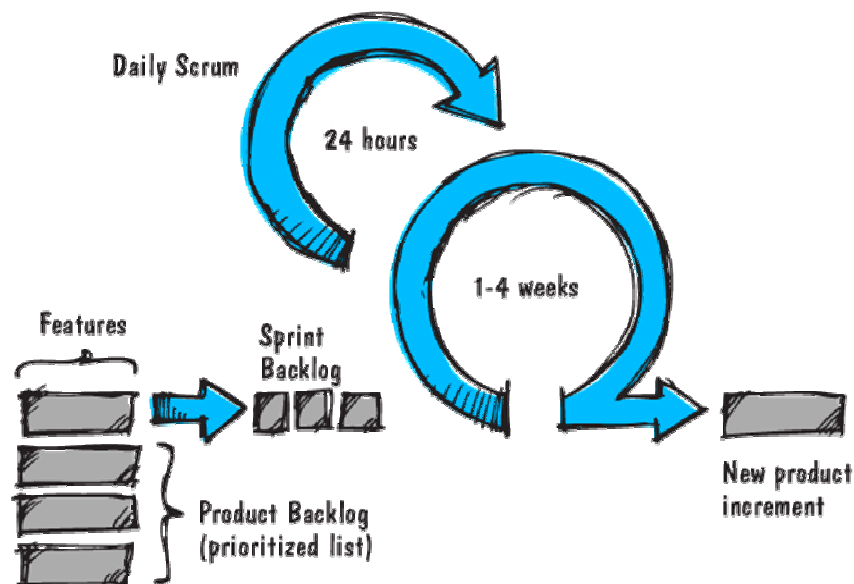
- A TDD-t általában az extrém programozásban és az agilis megközelítésekben alkalmazzák
  - Lásd 7. előadás
    - A programozó először a tesztekét írja meg
    - A tesztek előbb sikertelenek lesznek
    - Fokozatosan elkészül a tesztekhez tartozó kód
    - A tesztekét újabb elemekkel egészítik ki
    - Megj: Életciklus modell, vagy tesztelési módszer? Mindkettő lehet. Itt azért említjük, mert a szoftverfejlesztéshez kapcsolódó folyamatokat ír le, sorrendet javasol a végrehajtásukra.

# Test Driven Development



# Scrum

- Gyakran emlegetik élelciklus modellként; tulajdonképpen nem (csak) az!
- Sikeres agilis eszköz
- Alapvetően egy menedzsment eszköz, melynek vannak a szoftverfejlesztéshez kapcsolódó elemei
- Itt említjük, mert ajánl a szoftverfejlesztésben alkalmazható élelciklus modellt is





# Szoftver élelciklus modellek

- Mindegyik szoftverfejlesztési élelciklus modell (többé-kevésbé) ugyanazokból a folyamatokból épül fel
- A teljes „menü”, melyekből a folyamatokat kiválaszthatjuk, szabvánnyá vált
- ISO /IEC 12207: Information Technology - Software life cycle processes
  - *The international standard ISO 12207 is a globally accepted standard for software lifecycle processes. Though not suited for the direct application in a concrete project, it offers a frame that national standards or corresponding process details to be integrated in order to achieve a standard that can be used in an actual case. Furthermore, the ISO 12207 standard includes definitions that can be applied as a basis for a common terminology, even in national standards.*
- „This is the Process for Building Software In the 21st Century!”
  - <http://www.12207.com/>
  - <http://www.12207.com/12207-news.html>
- Teljesen kompatibilis a V-modellel



# Szoftver élelciklus modellek

- ISO/IEC 12207:2008 : Systems and software engineering -  
- Software life cycle processes.

<https://www.iso.org/standard/21208.html>

- Most (2017 őszén), fejlesztési fázisban egy újabb verzió:

- ☐ ISO/IEC/IEEE FDIS 12207: Systems and software engineering –  
Software life cycle processes

- ☐ <https://www.iso.org/standard/63712.html>

Korábban:

- ☐ [ISO/IEC 12207:1995](#)

Information technology -- Software life cycle processes

- ☐ [ISO/IEC 12207:1995/Amd 1:2002](#)

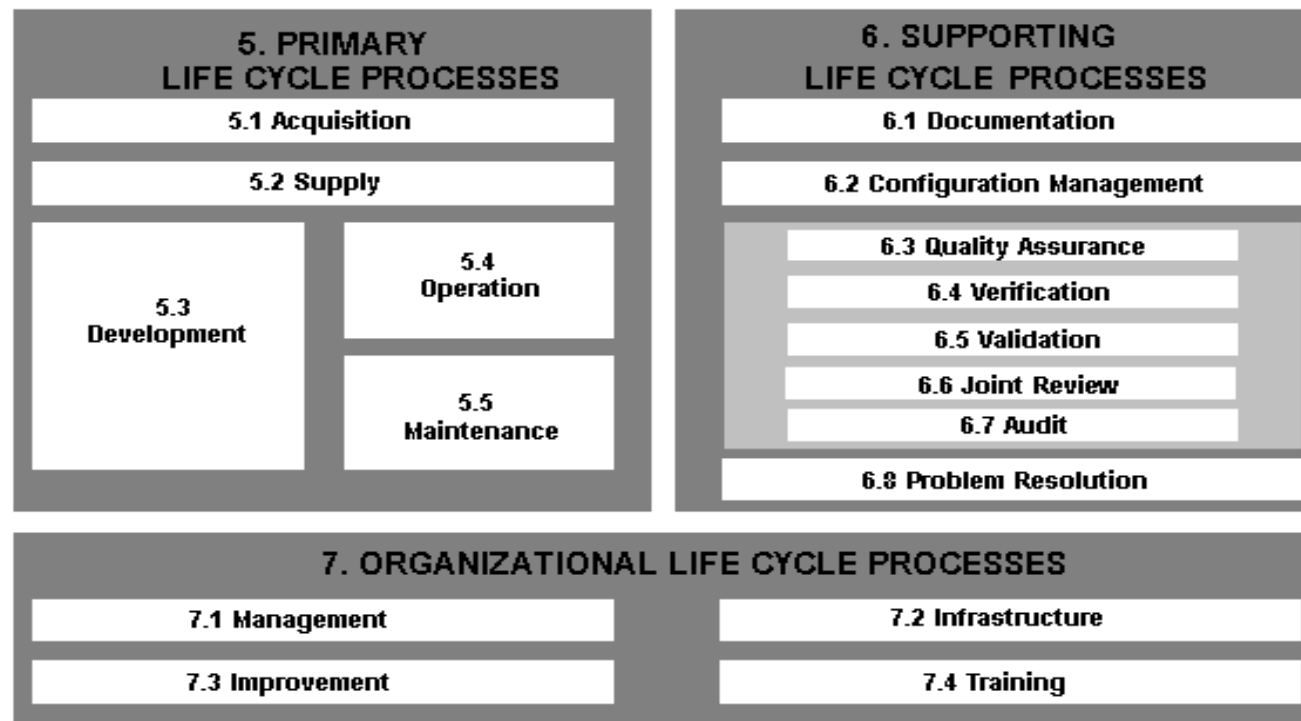


# Szoftver életciklus folyamatok

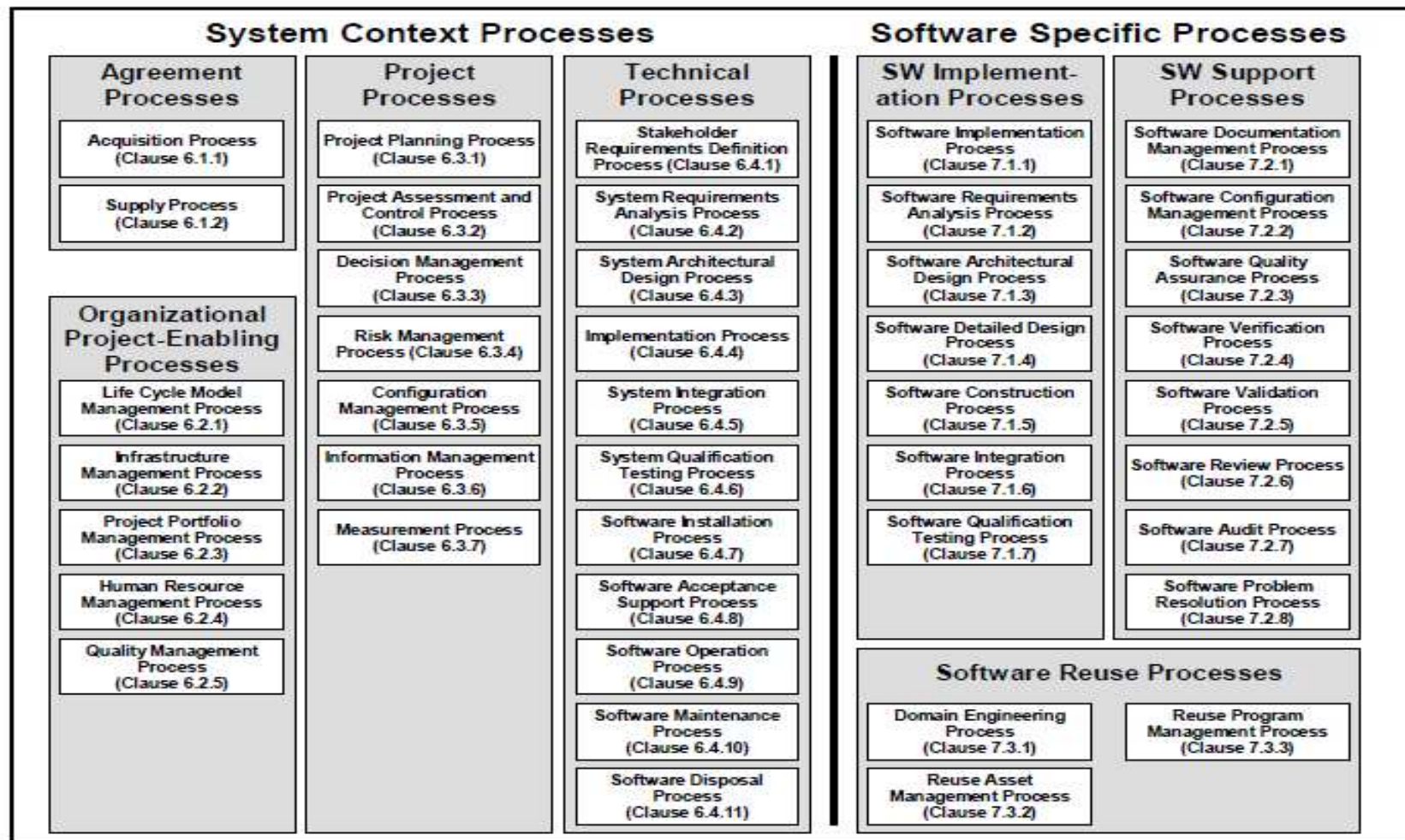
- Korábbi, magyar verzió:
- MSZ ISO/IEC 12207, 2000. május
  - Informatika. Szoftveréletciklus - folyamatok
- Egységes fogalmi keretet hoz létre a szoftveréletciklus-folyamatokra olyan jól meghatározott terminológiát használva, amely a szoftveripar számára kiindulásul szolgálhat. Olyan folyamatokat, tevékenységeket és feladatokat tartalmaz, amelyek szoftvert tartalmazó rendszerek, különálló szoftvertermékek és szoftverszolgáltatások beszerzése, valamint szoftvertermékek szállítása, fejlesztése, üzemeltetése és karbantartása során alkalmazandók. A „szoftver” kifejezés kiterjed a förmver szoftverrészére is.
  - Olyan folyamatot is tartalmaz, amelyet a szoftveréletciklus-folyamatok létrehozására, ellenőrzésére és javítására lehet használni.
  - Alkalmazható rendszerek, valamint szoftvertermékek és szoftverszolgáltatások beszerzésénél, szoftvertermékek és förmver szoftverrészének szállítása, fejlesztése, üzemeltetése és karbantartása során, függetlenül attól, hogy azt a szervezeten belül vagy kívül hajtják végre

# Szoftver élelciklus folyamatok

- Az ISO 12207 szerint a következő folyamatok lehetségesek a szoftverfejlesztés során:



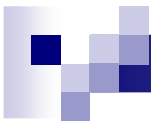
# ■ ISO/IEC 12207: 2008. Systems and software engineering- Software life cycle



2018.10.03.

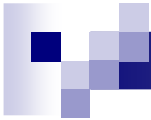
Figure 1 — Life Cycle Process groups





# Az ISO 12207

- Legújabb verzió:
- ISO/IEC/IEEE 12207:2017: Systems and software engineering -- Software life cycle processes
  - <https://www.iso.org/standard/63712.html>
  - Testreszabási útmutatók
    - Harmonizálják az ISO/IEC 33004 szabvánnyal



# Miről volt szó...

- Szoftverfejlesztési / technikai / műszaki folyamatok
- Népszerű élelciklus modellek
  - Vízesés, V-modell, spirál, iteratív, inkrementális