NGB_IN003_1 SZE 2014-15/2 (13)

Szoftver-minőségbiztosítás

Szoftverminőség és formális módszerek

Formális módszerek

- Formális módszer ≠ formalizált módszer(tan)
- Formális eljárások alkalmazása a fejlesztésben
  - nincs olyan formális eljárás, ami egy komplex rendszer minden aspektusát leírná
  - struktúrális és viselkedési leírások
  - tervezés, terv verifikálás

Formális módszerek (folyt.)

- Formális eljárások
  - közös koncepciók
    - rögzített **szintaxis, szemantika**
    - **matematikai eszközök**
    - eltérő jelölés
- absztrakt matematikai modellek a rendszer állapotáról
- egyre részletesebb modellek
  - -> **implementáció** (absztrakció csökken)
  - a **specifikáció finomítása**

Formális módszerek (folyt.)

- Szigorú módszerek a rendszertervezésben és fejlesztésben
- Szimbólikus (matematikai) logikai konstrukciók => formálisak (reprezentációk, eszközök)
- Rendszerrel szembeni bizalom erősítése
  - implementáció
  - követelmények
- Formális modellek létrehozása, formális követelményeknek megfelelőség mechanikus **bizonyítása** (formális végrehajtás)

Formális módszerek felhasználása

- Más elemzési és tervezési módszerek kiegészítésére
- Bugok felderítése (kódban, specifikációban)
- Fejlesztési idő csökkentése (tesztelés)
- Bizonyos rendszertulajdonságok biztosítása
- Automatizálás lehetővé tétele
- Fejlesztés minőségbiztosítása

Alkalmazási területek

- real time adatbázisok
- hardver tervezés
- biztonságkritikus alkalmazások (orvosi, nukleáris, haditechnikai, közlekedési)

Motiváció

- A technológiai rendszerek kis hibái katasztrófákhoz vezethetnek
- A mindenütt jelenlévő szoftverek hibái egyre több területen okoznak meghibásodásokat
- A szoftverminőség egyre jelentősebb gazdasági és jogi probléma

Szoftverrendszerek problémái

- A szoftverek nem folytonos függvényként működnek
- A HW redundancia nem segít a szoftverhibákon
- Az alrendszerek nem szeparálhatók tisztán
- A költséghatékonyság fontosabb a megbízhatóságnál
- A megbízható szoftverek tervezése nem kiforrott

Szoftverek korrektségének biztosítása

- Általános stratégia: tesztelés
- Más megközelítések: szoftver folyamat minőségbiztosítás, módszertanok

- Tesztelés
  - szoftverhibák ellen: specifikált működés ellenőrzése
  - külső hibák tűrésére: hiba-injektálás, hibaterjedés ellenőrzése

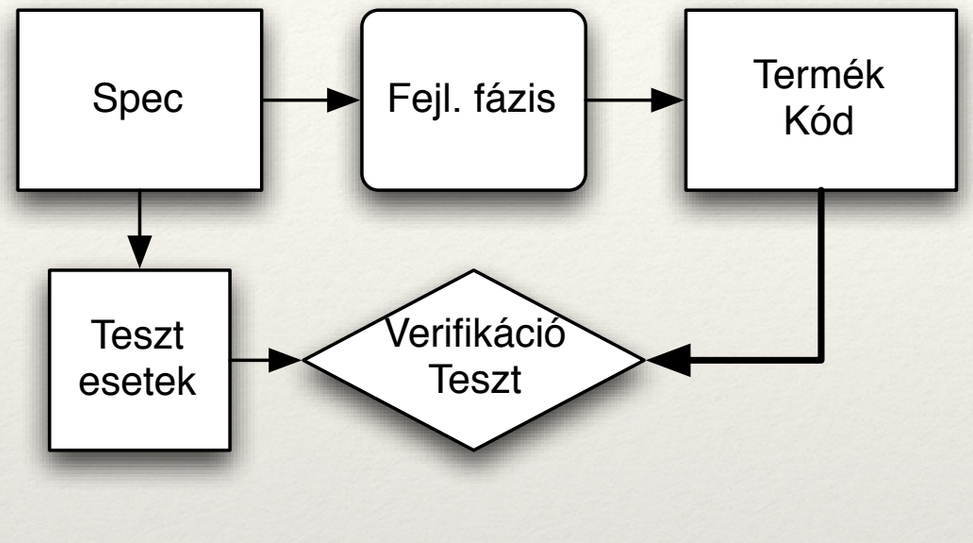
A tesztelés korlátai

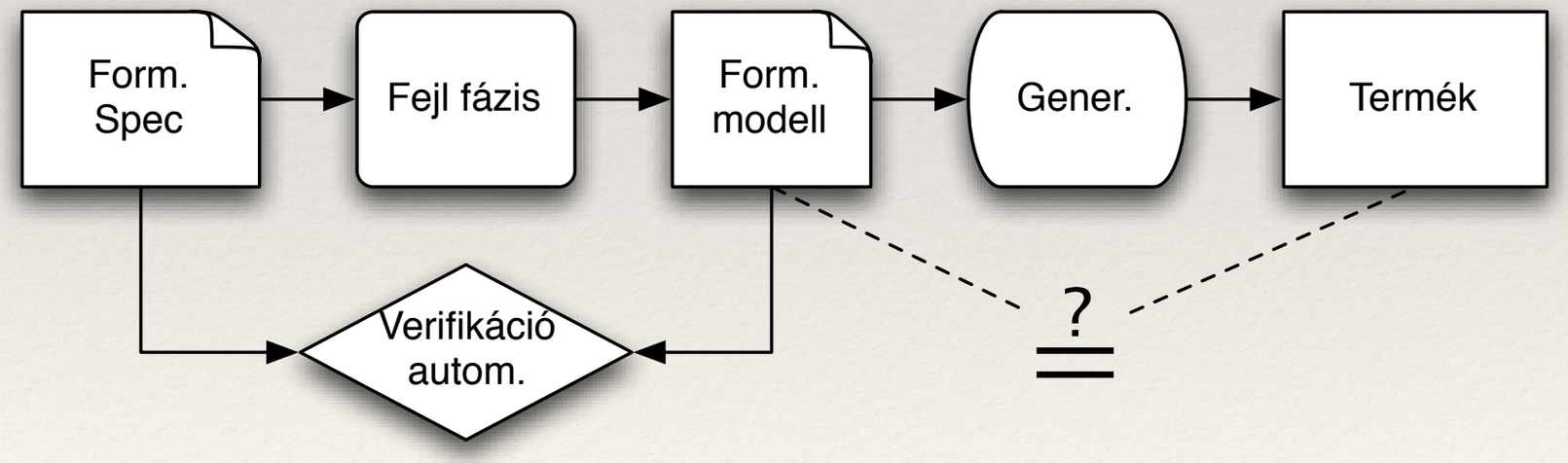
- A tesztelés célja hibák felfedése (javításra) és nem a hibátlanság igazolása
  - kimerítő tesztelés lehetetlen
- A tesztesetek reprezentativitása központi kérdés
  - váratlan, ritka esetek?
- A tesztelés költséges
  - tesztesetek tervezése, végrehajtása

Formális módszerek és tesztelés

- Tesztelési kihívások
  - Tesztesetek előállítása
    - random (hibafelfedés?, olcsó)
    - intelligens szintetizálás (kézi, munkaigényes)
    - automatikus (formalizált spec. szükséges)
    - lefedési kérdések
  - Végrehajtás (megfigyelés)
- Formális módszerek (model alapú) alk. **teszteset generálásra**

Formális módszerek és tesztelés





Specifikálás -követelmények

- Elvárt jellemzők
  - biztonsági jellemzők (pl. kölcsönös kizárás)
  - élőségi jellemzők (pl. éhezés elkerülése)
  - konkurencia, elosztottsági jellemzők (pl. holtpont mentesség)
  - nem funkcionális követelmények (pl. futási idő, memória használat, használhatóság)

Specifikálás

- A teljes viselkedési specifikálás igényelné:
  - a kód teljesíti a **funkcionalitására** vonatkozó előírásokat
  - **adatkonzisztencia** és invariancia előírásokat
  - **modularitás**, egységbezárási kérdések leírását
  - fejlesztési fázisok termékei közötti ekvivalencia és finomítási viszonyok leírását

Formális módszerek alkalmazása

- A **formális bizonyítás helyettesíthet** sok (akár végtelen) **tesztesetet**
- A formális módszerek **javítják a specifikálás minőségét**
- A formális módszerek **garantálhatnak** bizonyos rendszer-jellemzőket

Formális módszerek alkalmazása

- Nem alkalmasak a teljes rendszer korrektségének bizonyítására
- Nem helyettesítik teljes egészében a tesztelést
  - nem natív kód szintjén működnek
  - sok nem formalizálható jellemző van
- Nem helyettesítik a megfelelő tervezési eljárásokat

Formalizálás

- A valóság leképezése
  - formális követelmény specifikációra
    - túl egyszerűsítés
    - nem teljesség
- formális működési/végrehajtási modellre
  - modellezés (leképezés) pontossága

Rendszerek leírása

- Leírási szintek
  - Absztrakt szint
    - Véges állapottér
    - Automatikus bizonyítás elvben lehetséges
    - Az egyszerűsítés elkerülhetetlen
  - Konkrét szint
    - Nem véges és komplex adatszerkezetek
    - Valódi programozási nyelvi modellek
    - Automatikus bizonyítások általában lehetetlenek

Automatikus tételbizonyítás

- Automatikus bizonyítás (batch mode)
  - a bizonyításközben nem kell interakció
  - eszközök paramétereket be kell hangolni
    - formális specifikálás kézzel
- Félautomatikus bizonyítás (interaktív)
  - bizonyítás közben kellhet interakció
  - ismerni kell az eszköz belső működését
  - a bizonyítást ellenőrzi az eszköz

Modellellenőrzés

- Annak az automatikus ellenőrzése, hogy egy modell rendelkezik-e bizonyos jellemzőkkel (pl. holtpont mentesség)
- Ipari alkalmazás
  - HW verifikálás
  - SW verifikálás
  - vezérlő rendszerek, protokollok
  - absztrakciók ellenőrzése

Kihívások

- Tudásgát a formális módszerek alkalmazására
  - erősen absztrakt
  - munka-ráfordítás igényes
  - néhány területen nincsenek eszközök
    - diszkrét~folytonos idő
    - időváriáns, nem lineáris rendszerek

Modellezés - Occam borotvája

- Heurisztika - a legegyszerűbb, még megfelelő modellt kell választani
- Absztrakciók
  - irreleváns részletek mellőzése -> fontos jellemzők maradnak
  - további redukció a komplexitás csökkentésére
- Elfogadott, bevált modellek alkalmazása

Szoftverrendszerek modellezése

- Különböző rendszer nézetek
  - funkcionális / relációs / szekvenciális nézetek
    - a rendszer mint input / output reláció, függvény
    - absztrakt adattípusok a rajtuk értelmezett műveletekkel (szekvenciákkal)
  - struktúrális nézetek
    - komponensek és kapcsolataik
    - OO modellezés

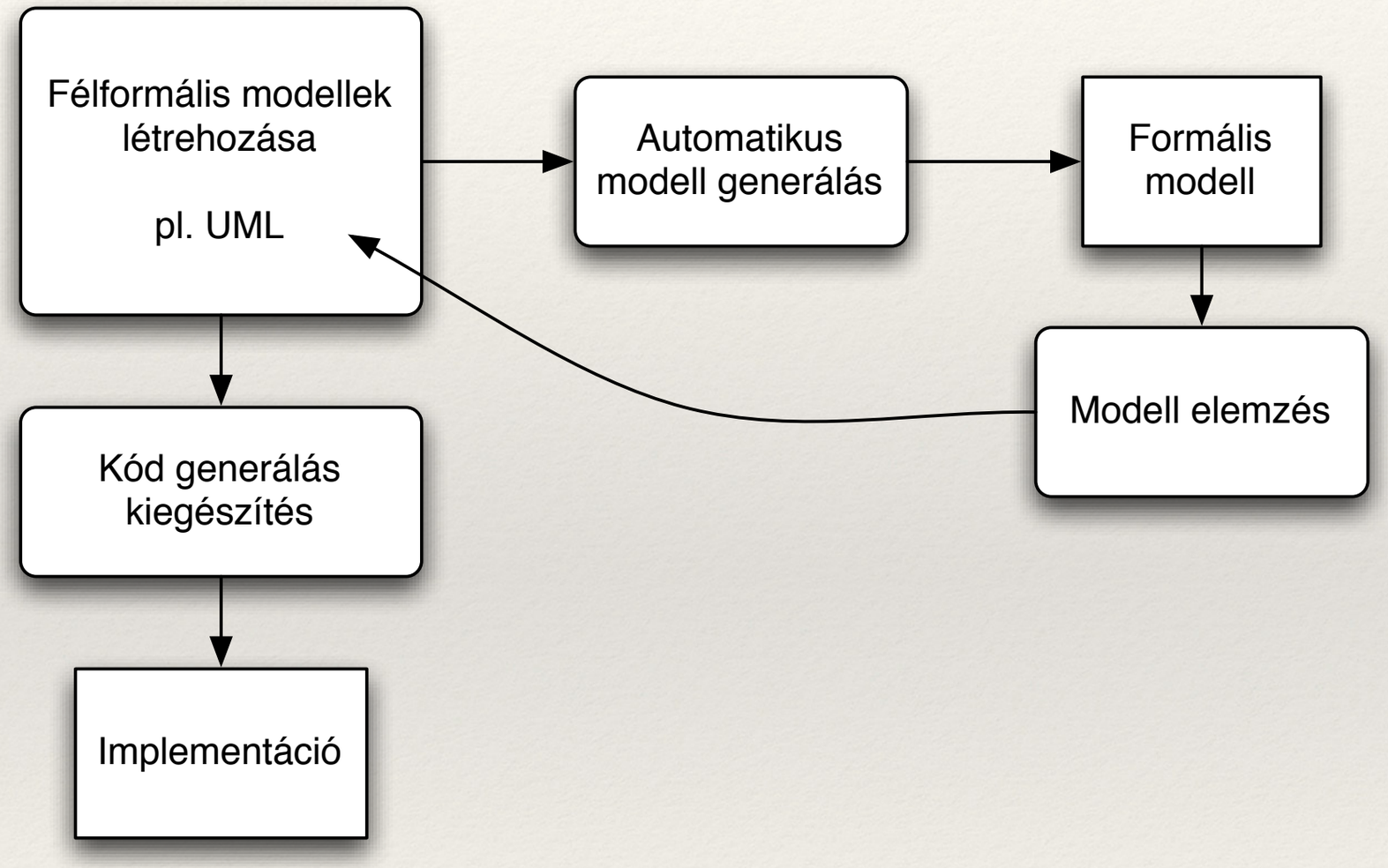
Szoftverrendszerek modellezése

- Viselkedési modellek
  - reaktivitás
  - konkurencia / elosztottság
    - interakció ismeretlen környezettel
- Modellezési szempontok, kritériumok
  - fontos jellemzők
  - funkcionális korrektség, terminálás, holtpontok, éhezés

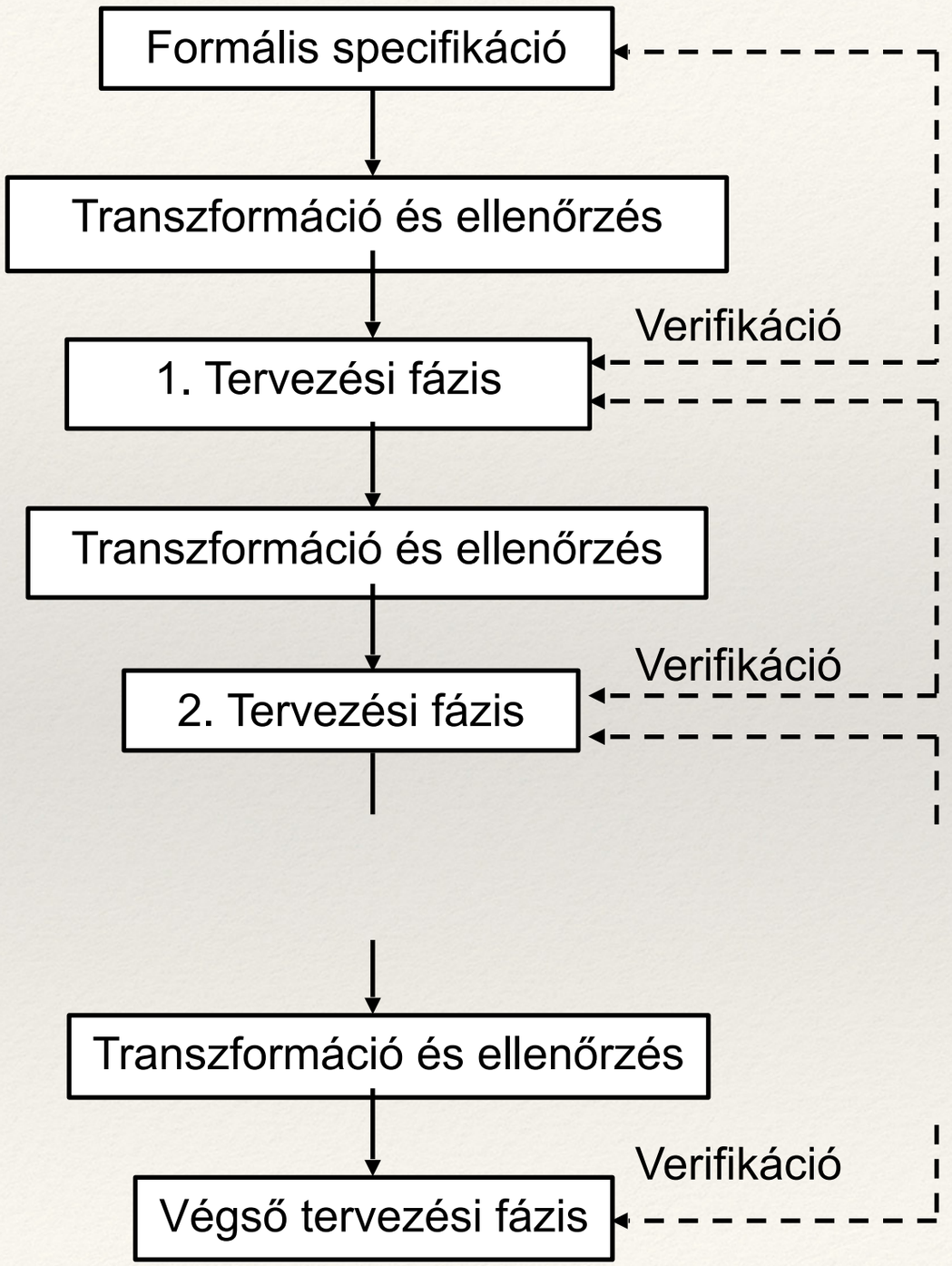
Szoftverrendszerek modellezése

- Konkurens/Interaktív rendszerek
  - nem determinisztikusan előforduló események
  - állapot átmenet rendszerek
    - számítás = állapotok szekvenciája átmenetek mentén
    - állapot = rendszer (globális) állapot
    - átmenetek = állapotok változása
- végrehajtási modellek
  - lineáris (számítási utak)
  - elágazó (számítási fák)

Formális módszerek a gyakorlatban



Formális fejlesztési folyamat



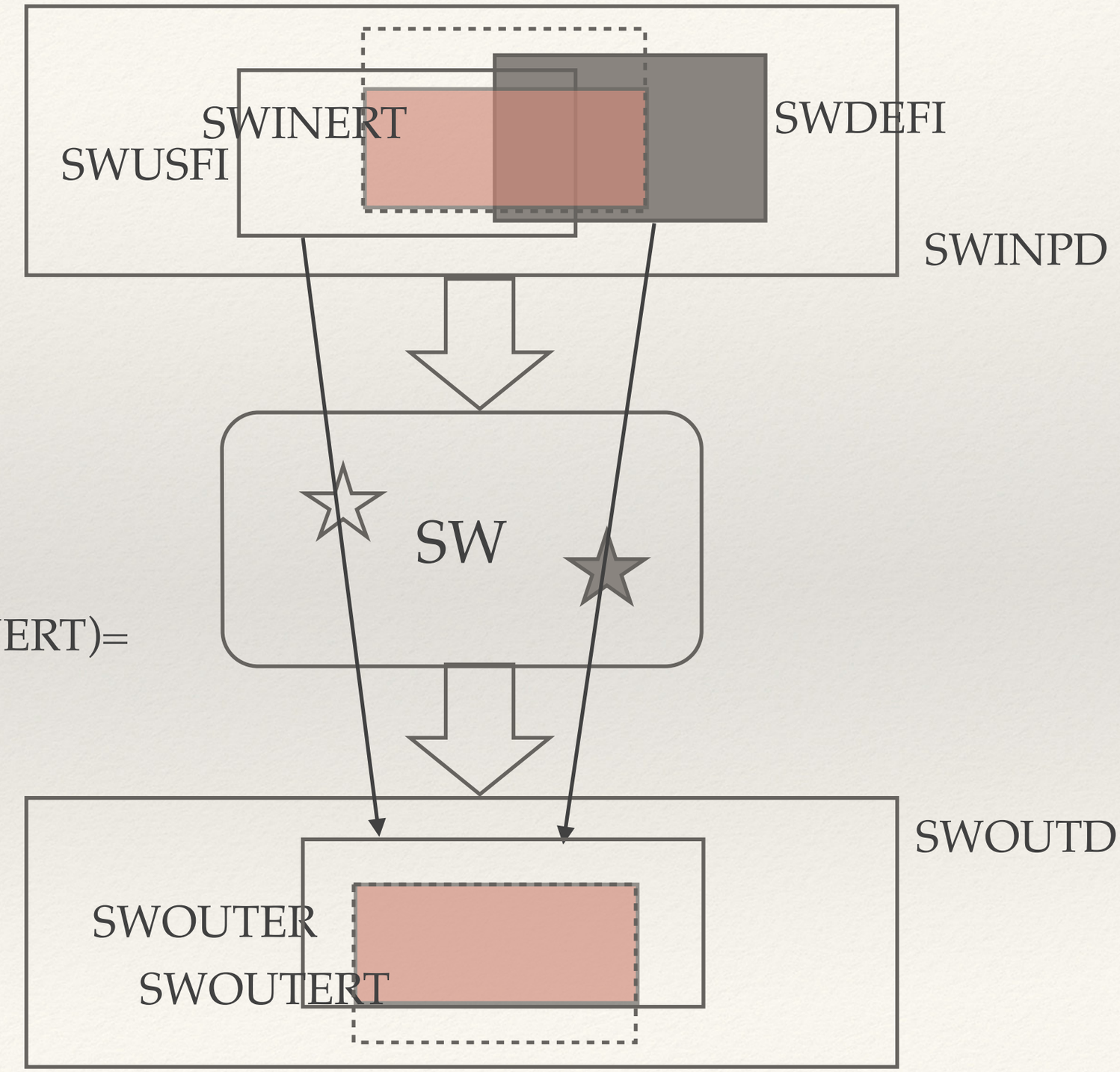
Sziray modellje

- Halmaz (input-output) leképezési (szoftver) modell a szoftver-verifikáció és validáció viszonyainak szemléltetésére (2004)
- A kétféle minőségbiztosítási megközelítés fontosságának kiemelése
- Szoftver hibaforrások megkülönböztetése (specifikálás, fejlesztés)
- Verifikáció (transzformációs lépések)
- Validáció (felhasználói igények kielégítése)

Sziray modellje (folyt.)

SWOUTER1=
SWM(SWUSFI∪SWDEFI)

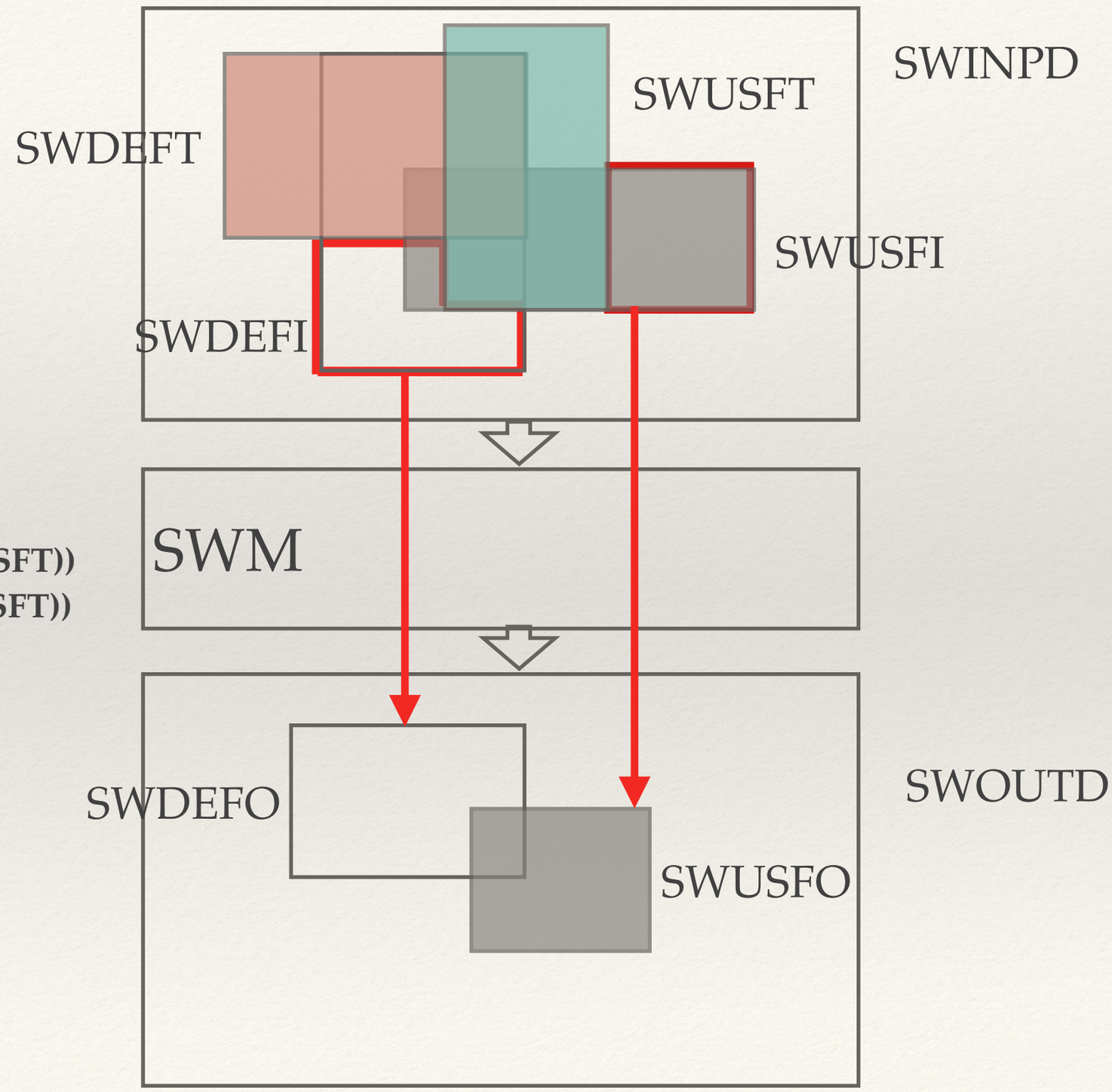
Tesztelés után:
SWOUTER2=
SWM((SWUSFI∪SWDEFI)\SWINERT)=
SWOUTER1\SWOUTERT



Sziray modellje (folyt.)

Verifikálás és validálás

SWOUTD=SWM(SWINPD)
Nem detektált hibák:
SWDEFO=SWM(SWDEFI\(SWDEFT∪SWUSFT))
SWUSFO=SWM(SWUSFI\(SWDEFT∪SWUSFT))



Sziray modellje (folyt.)

- Formális módszerek alkalmazásakor:
  - SWDEFI=Ø
  - SWDEFT=Ø
  - SWDEFO=Ø
  - => SWUSFO=SWM(SWUSFI\SWUSFT)
  - azaz, formális módszerek alkalmazása esetén is szükségesek a validációs tesztek