

Szoftverellenőrzési technikák (vimim148)

A szoftver tesztelés alapjai

Majzik István, Micskei Zoltán

<http://www.inf.mit.bme.hu/>

Utolsó módosítás: 2010.10.11.

Tartalom

- Követelményspecifikáció ellenőrzése
- Ellenőrzések a tervezési fázisban
- Forráskód verifikáció technikái
- **Tesztelési módszerek és folyamatok**
 - **A szoftver tesztelés alapjai**
 - Specifikáció és struktúra alapú tesztervezési módszerek
 - Integrációs és rendszer tesztek
 - Teszt környezet és teszt automatizálás
 - Modell alapú tesztelés
- Validáció
 - GUI tesztelés
 - Stressz és robusztusság tesztelés, hibainjektálás
- Hibakezelés, tesztelés dokumentálása

A tesztelés definíciói (1)

„Testing is an activity performed for evaluating product quality, and for improving it, by identifying defects and problems.” [SWEBOK]

- Általános definíció
- Kulcs: „evaluating product quality”

IEEE, Software Engineering Body of Knowledge (SWEBOK), URL:
<http://www.computer.org/portal/web/swebok/>

Quality: „the degree to which a system, component, or process meets specified requirements” (IEEE definition)

A tesztelés definíciói (2)

„ An activity in which a system or component is executed under specified conditions, the results are observed or recorded, and an evaluation is made of some aspect of the system or component.” [IEEE Std. 829]

- Kicsit specifikusabb
- Kulcs: végrehajtjuk a rendszert vagy komponenst

IEEE, "IEEE Standard for Software and System Test Documentation," *IEEE Std 829-2008*, 2008

A tesztelés definíciói (3)

„The process consisting of all lifecycle activities, both static and dynamic, concerned with planning, preparation and evaluation of software products and related work products to determine that they satisfy specified requirements, to demonstrate that they are fit for purpose and to detect defects. ” [ISTQB]

- Sokkal bővebb definíció
- Kulcs:
 - Tesztelés egy folyamat
 - Statikus és dinamikus technikák is

International Software Testing Qualifications Board (ISTQB), URL:
<http://istqb.dedicated.adaptavist.com/>

A tesztelés lehetséges céljai

A tesztelés lehetséges céljai

- Bizonyosságot szerezni a rendszer minőségi állapotáról
- Információ a döntéshozáshoz (pl. release / no release)
- Hibák keresése
- Hibák megakadályozása

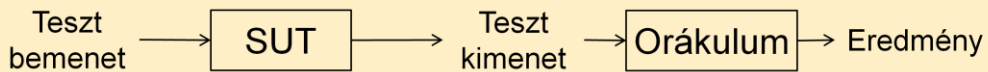
Mottók:

- Dijkstra: A tesztelés a **hibák jelenlétét**, és nem a hibamentességet tudja kimutatni.
- Hoare: A tesztelés egy **induktív bizonyítás** része:
Ha a program jól működik egy adott teszt adatra,
akkor várhatóan *hasonló adatokra* is jól működik.

Alapelvek (7 testing principles)

- Csak a hibák meglétét tudja kimutatni
- A kimerítő tesztelés legtöbbször lehetetlen
 - Hány teszt kéne egy 3 integer paraméterű függvényhez?
- Tesztelés már a korai fázisokban
- Hibák csoportosulása (Defect clustering)
 - A komponensek egy kis részében van a hibák nagy része
- Rovarirtó paradoxon (Pesticide paradox)
 - Teszt újra és újra futtatva egyre kevésbé „hatékony”
 - Minden módszer után marad még valami más típusú hiba
- A tesztelés kontextus függő
- Hibák hiánya tévedés (Absence-of-errors fallacy)
 - Hibamentes tesztelfutás még nem jelent jó rendszert

Alapfogalmak



- **teszteset (test case)**
 - bemeneti értékek, végrehajtási előfeltételek, várt eredmények és végrehajtási utófeltételek halmaza
- **tesztkészlet (test suite)**
- **orákulum (oracle)**
 - várt és az aktuális eredményeket összehasonlító forrás
- **eredmény (verdict)**
 - sikeres (pass), sikertelen (fail), nem meggyőző (inconclusive), hiba (error)
- **tesztelés != hibakeresés (debugging)**

SUT = System Under Test

Tesztkészlet: „rendszerre vagy rendszerkomponensekre készített tesztesetek halmaza, amelyben gyakran az egyik teszt utófeltétele egyben a másik teszt előfeltétele”

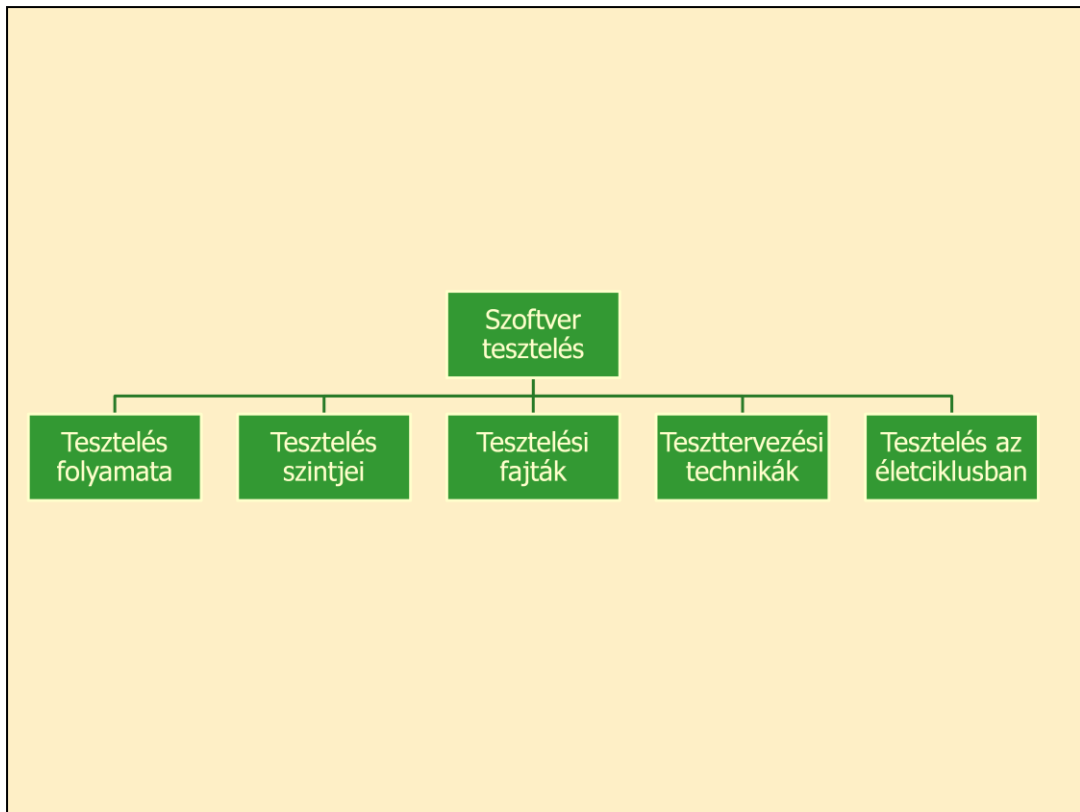
Verdict

-inconclusive: akkor lehet például, ha a tesztesetben megadjuk, hogy milyen válaszüzenet a helyes, milyen válaszüzenet a hibás, és a konkrét tesztelés során ezektől eltérő választ kapunk

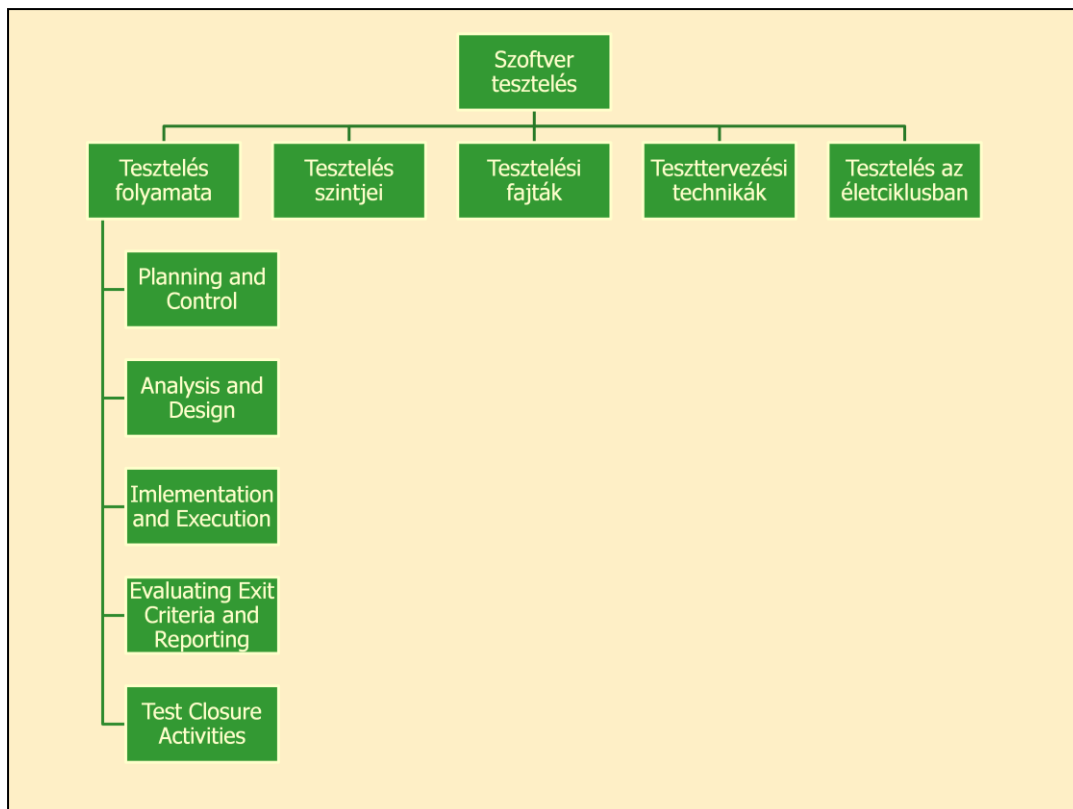
-error: a teszt rendszerben keletkezett a hiba, így nem tudjuk eldönteni, hogy milyen volt a SUT válasza

Alap problémák

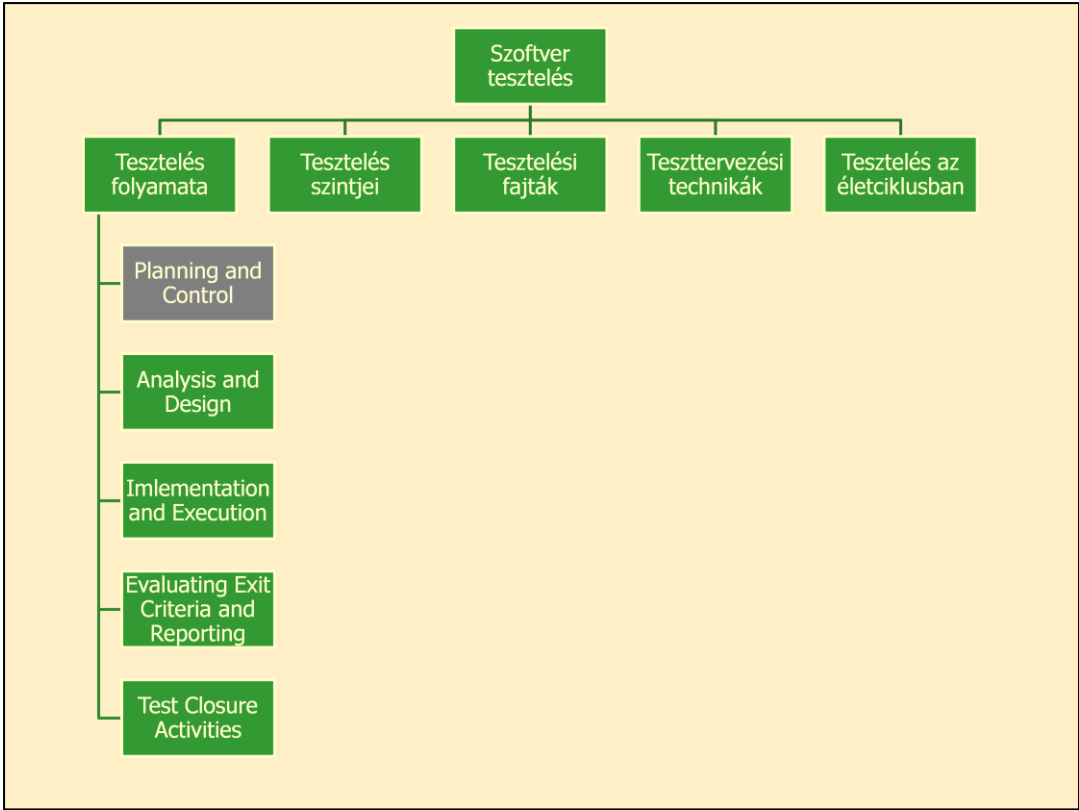
- **Teszt kiválasztás (test selection)**
 - Milyen teszt bemeneteket használjunk?
- **Kilépési feltétel (exit criteria)**
 - Meddig teszteljük?
- **Orákulum**
 - Honnan lesz jó teszt orákulumunk?
- **Tesztelhetőség (testability)**
 - Mennyire könnyű tesztelni a rendszert?
 - Megfigyelhetőség (observability)
 - Vezérelhetőség (controllability)



Fontosabb tesztelési fogalmak összegyűjtése és rendszerezése



Forrás: ISTQB Foundation Syllabus



Tesztelési stratégia

- Általános irányelvek
 - Milyen metodológiát,
 - milyen típusú tesztek,
 - milyen eszközöket,
 - ki fog használni,
 - milyen kilépési feltétellel,
 - milyen dokumentáció kell
 - ...
- Pl.:
 - extrém programozás
 - Modul & rendszer
 - JUnit & GUI Tester
 - Fejlesztő és teszt csapat
 - 90% utasítás lefedettség & minden használati eset

Teszt terv (test plan)

- Teszt stratégia leképezése az aktuális projektre
 - Tesztelési célok, irányelvek, környezet...
 - Feladatok szerepekhez rendelése
- Tesztelési fázisok definiálása
 - Fázisok hossza
 - Kilépési feltétel
 - Tesztelés minőségét hogyan fogjuk mérni
- Egyszerű példa: <http://bazman.tripod.com/>

IEEE 829 - Standard for Software and System Test Documentation

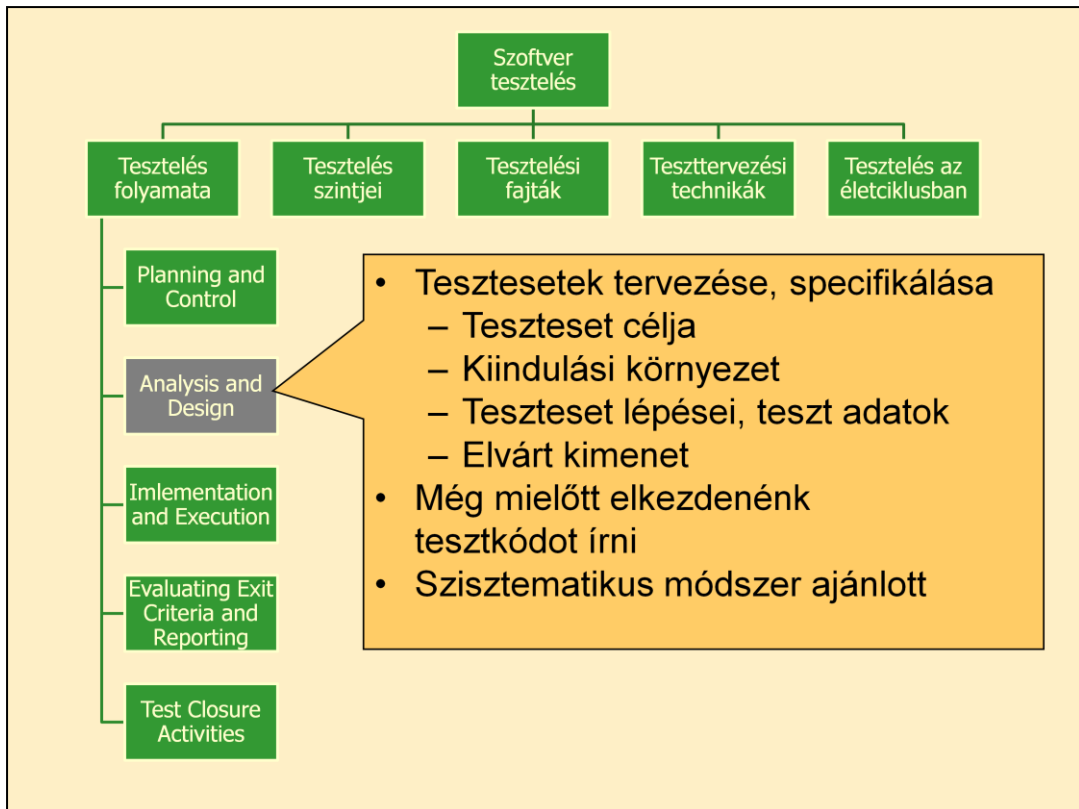
Level Test Plan Outline (full example)

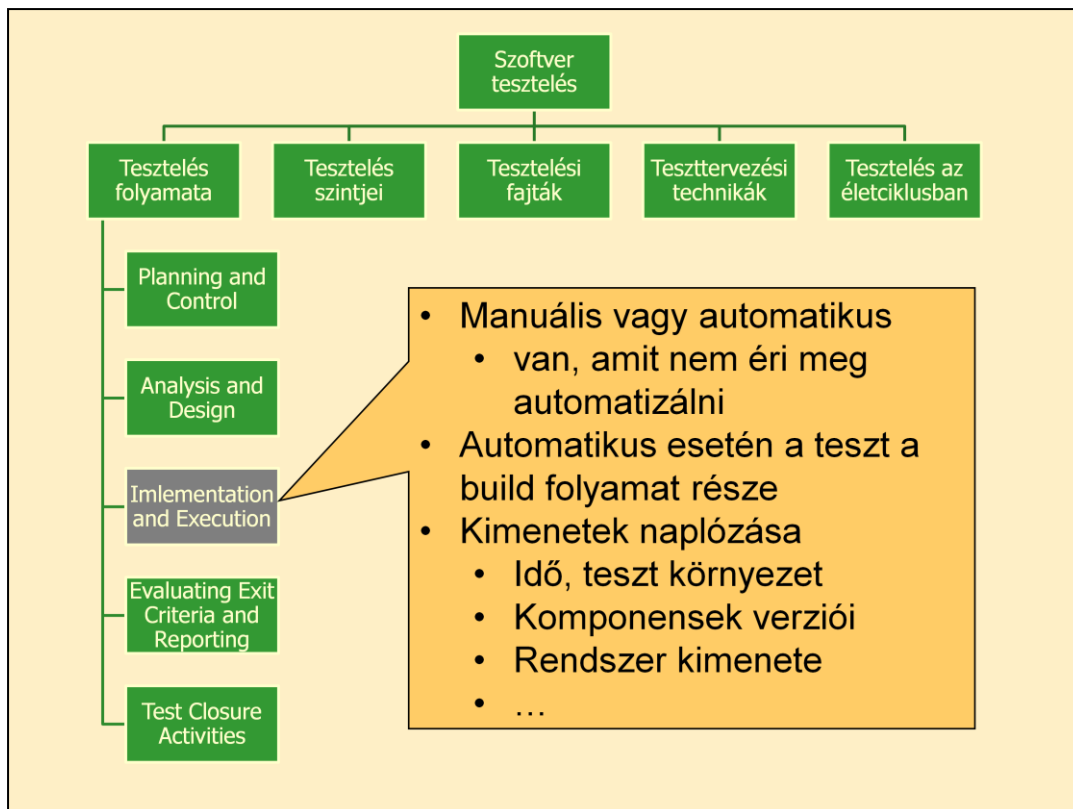
| |
|---|
| 1. Introduction |
| 1.1. Document identifier |
| 1.2. Scope |
| 1.3. References |
| 1.4. Level in the overall sequence |
| 1.5. Test classes and overall test conditions |
| 2. Details for this level of test plan |
| 2.1 Test items and their identifiers |
| 2.2 Test Traceability Matrix |
| 2.3 Features to be tested |
| 2.4 Features not to be tested |
| 2.5 Approach |
| 2.6 Item pass/fail criteria |
| 2.7 Suspension criteria and resumption requirements |
| 2.8 Test deliverables |
| 3. Test management |
| 3.1 Planned activities and tasks; test progression |
| 3.2 Environment/infrastructure |
| 3.3 Responsibilities and authority |
| 3.4 Interfaces among the parties involved |
| 3.5 Resources and their allocation |
| 3.6 Training |
| 3.7 Schedules, estimates, and costs |
| 3.8 Risk(s) and contingency(s) |
| 4. General |
| 4.1 Quality assurance procedures |
| 4.2 Metrics |
| 4.3 Test coverage |
| 4.4 Glossary |
| 4.5 Document change procedures and history |

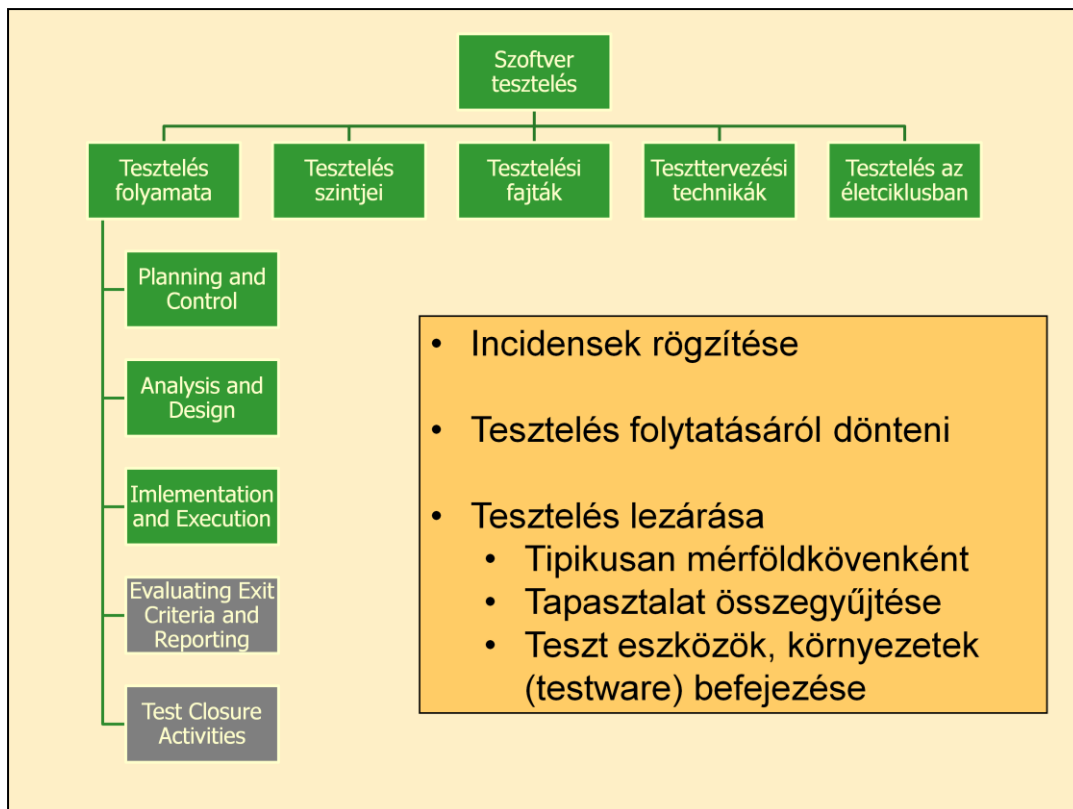
- Master Test Plan
- Level Test Plan
- Integritási szintek

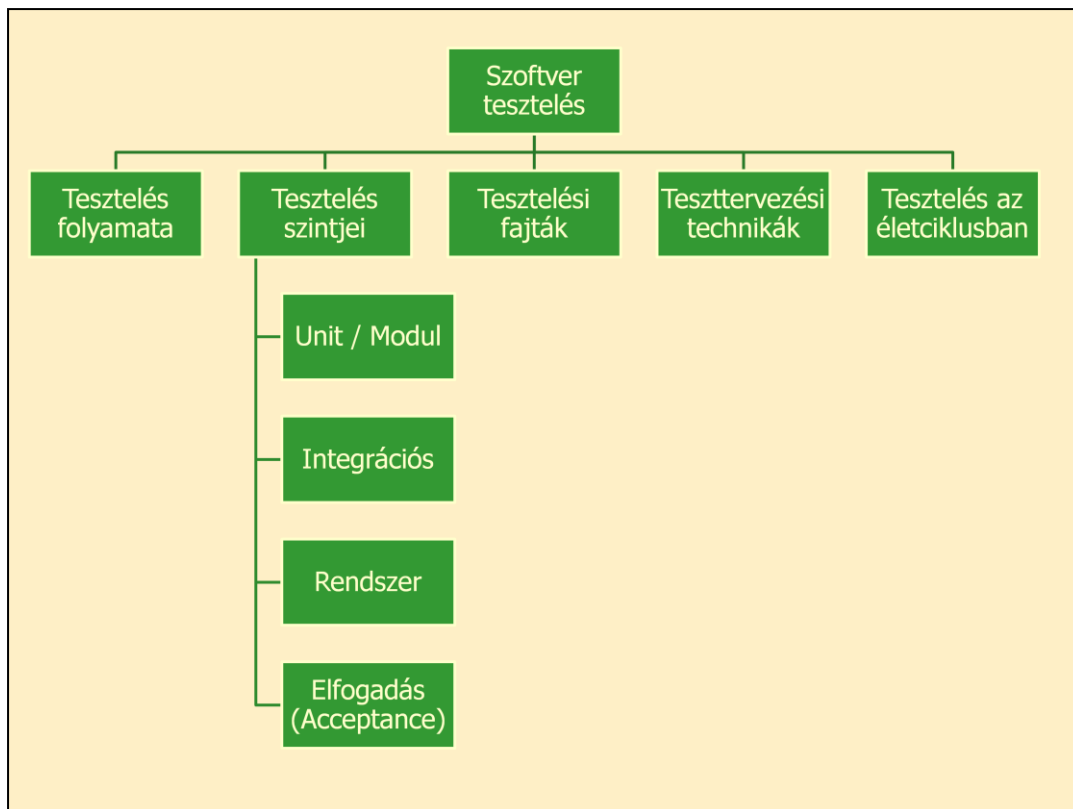
- Célok, hatókörök, felelőségek definiálása

- Konkrét teszteset még nincsen









Tesztelési környezet: Egy modul tesztelése

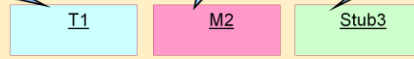
Teszt végrehajtó

- modul hívása
- Teszt kiértékelő
- kimenet figyelése

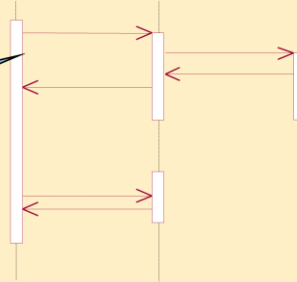
Tesztelendő modul

Teszt csonk

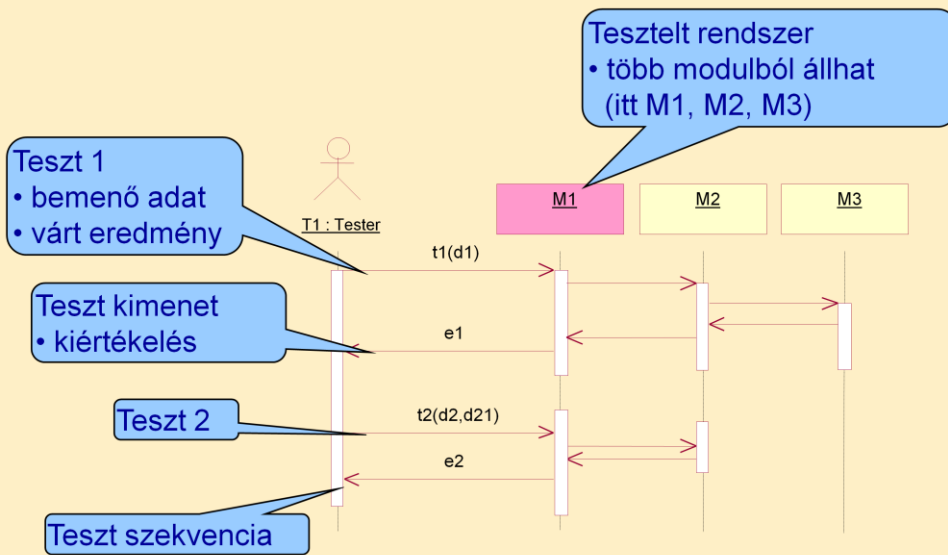
- korlátozott funkcionalitás

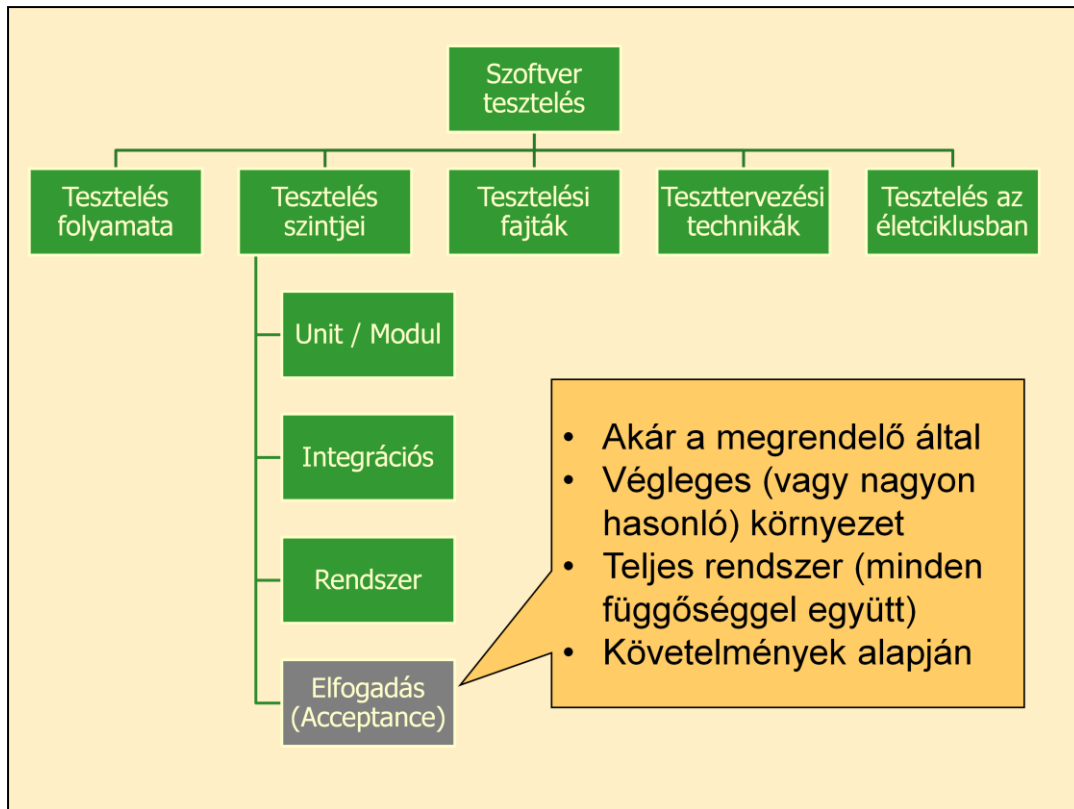


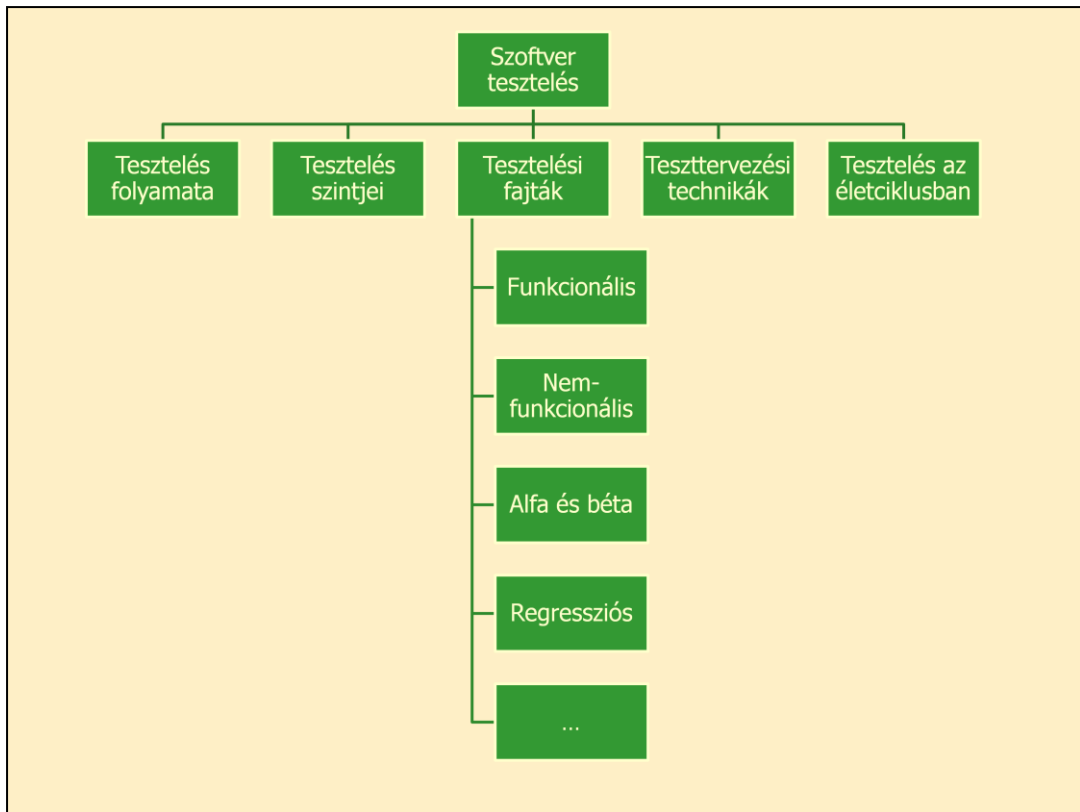
Teszt program
vagy teszt script

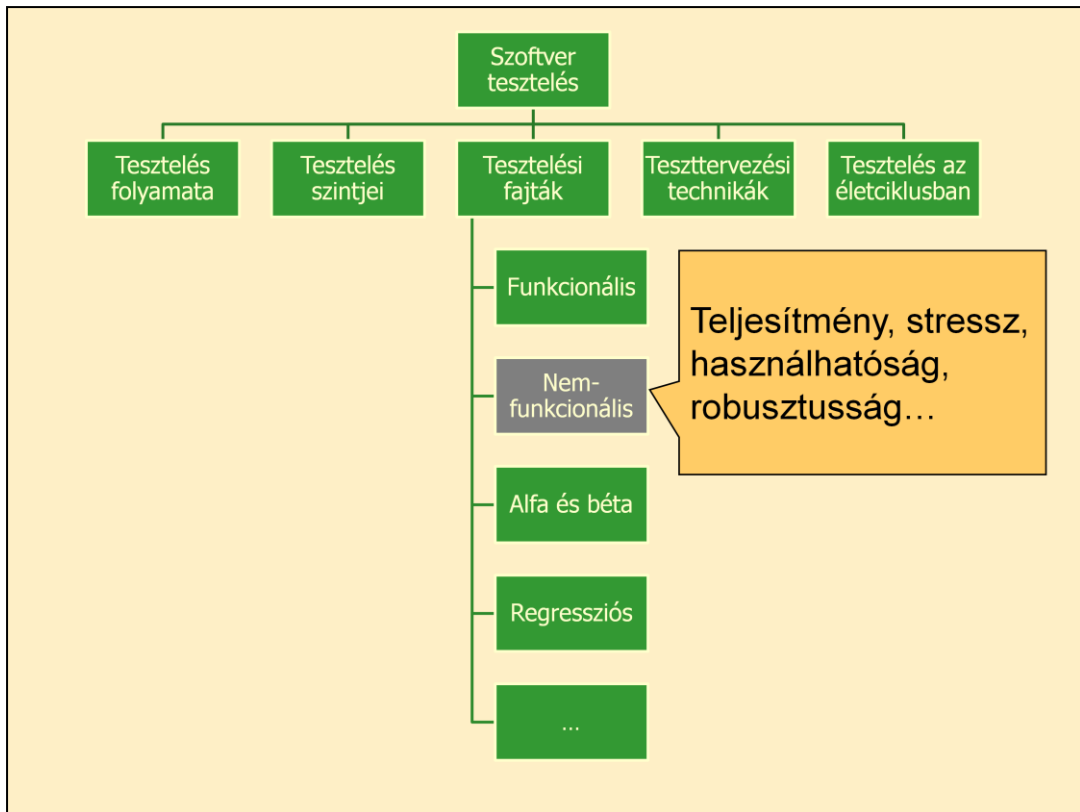


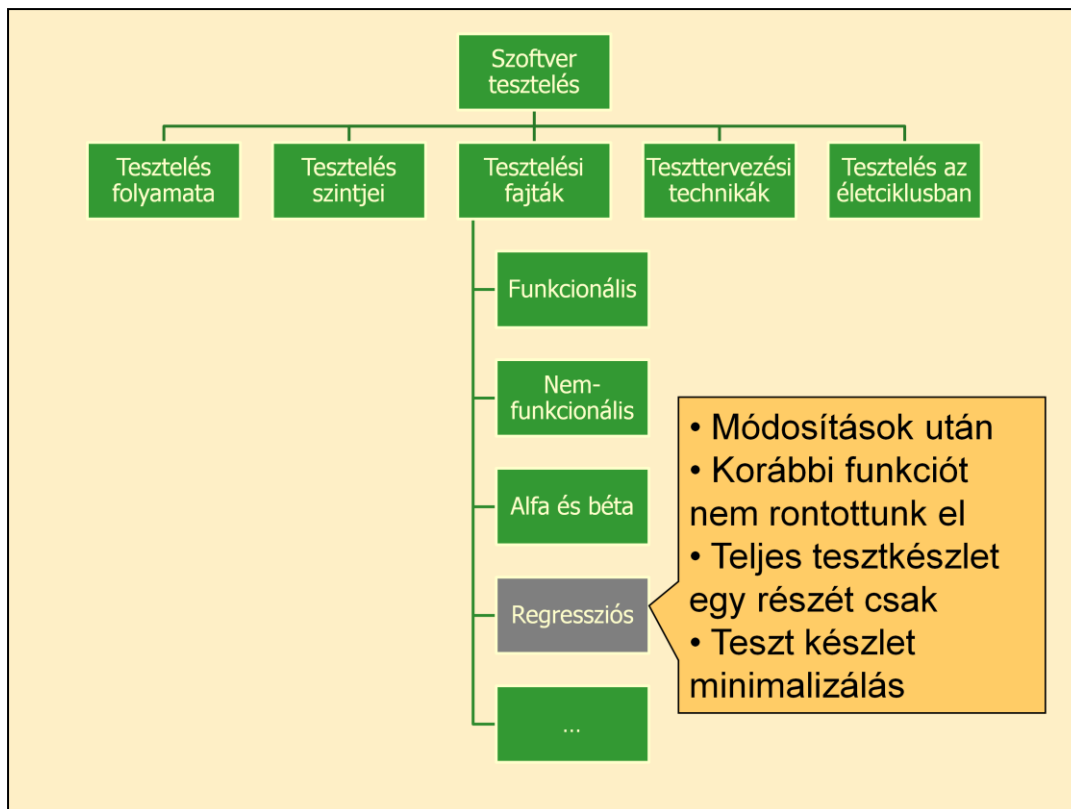
Tesztelési környezet: Rendszer tesztelése





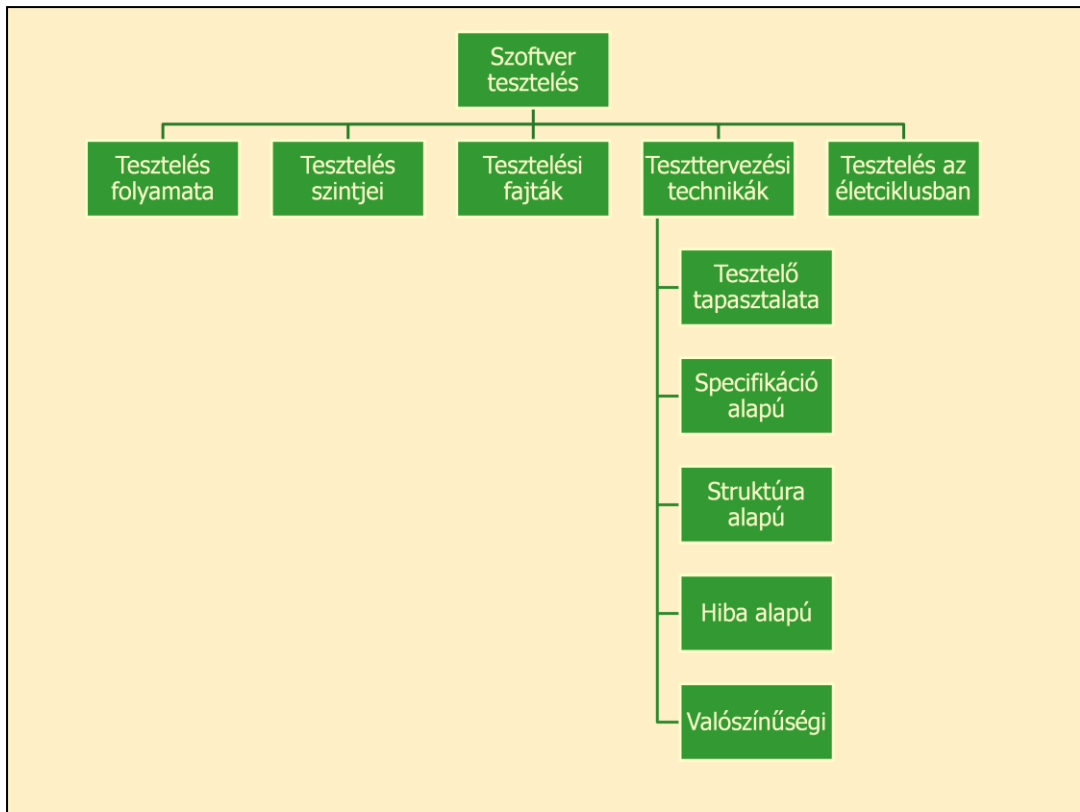


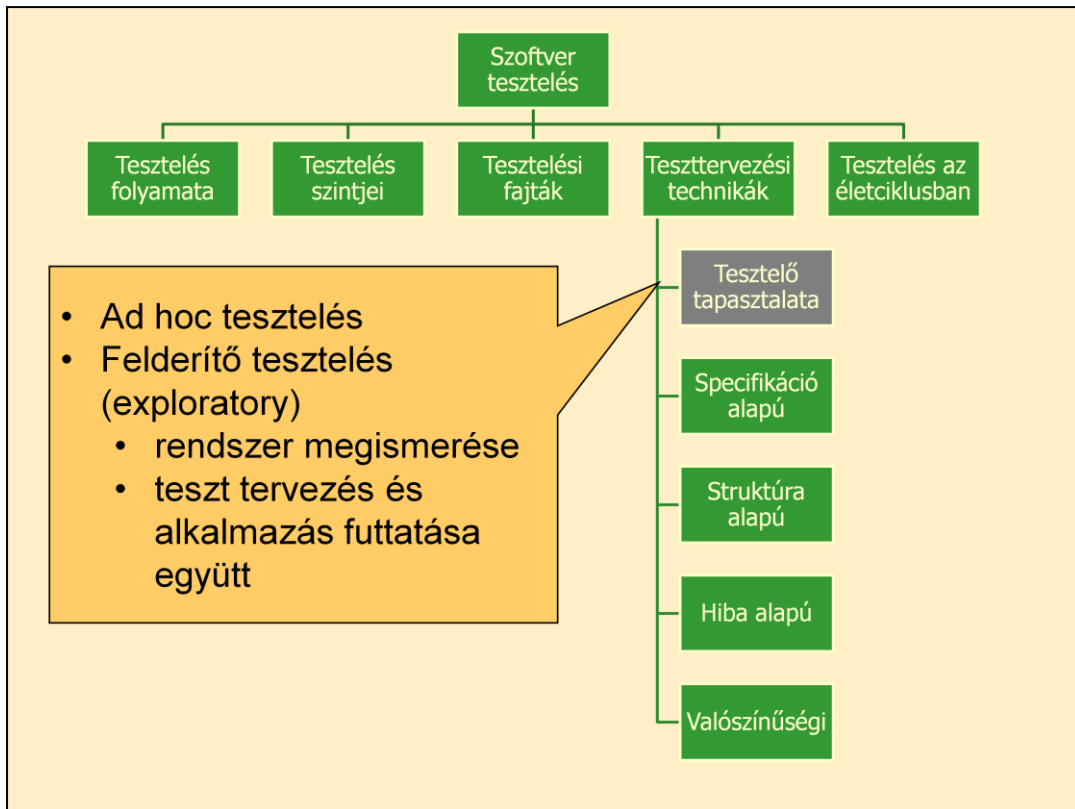


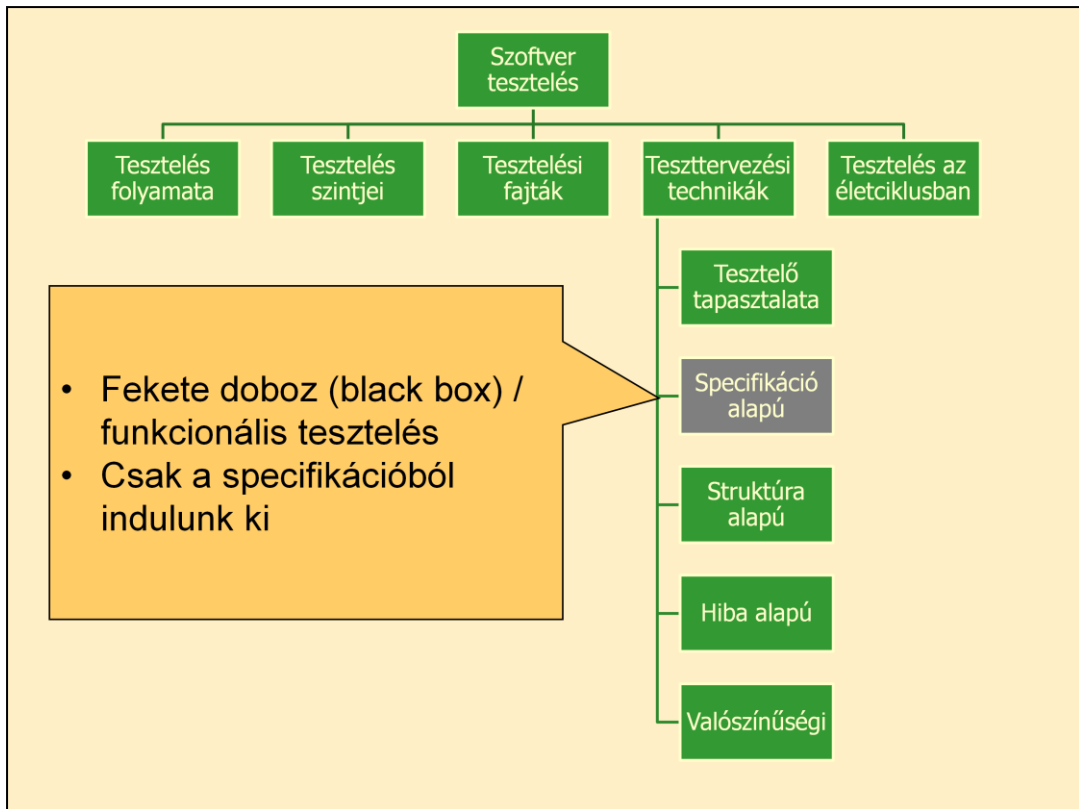


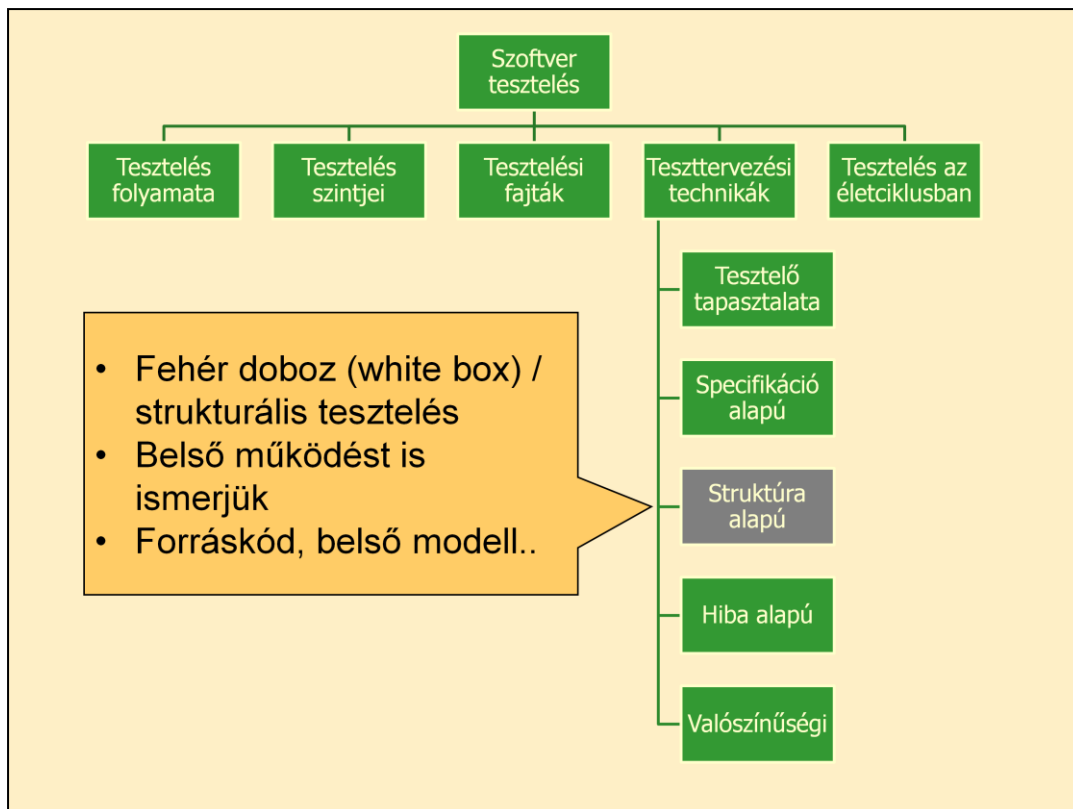
Regressziós teszteléssel kapcsolatban jó áttekintő cikkek:

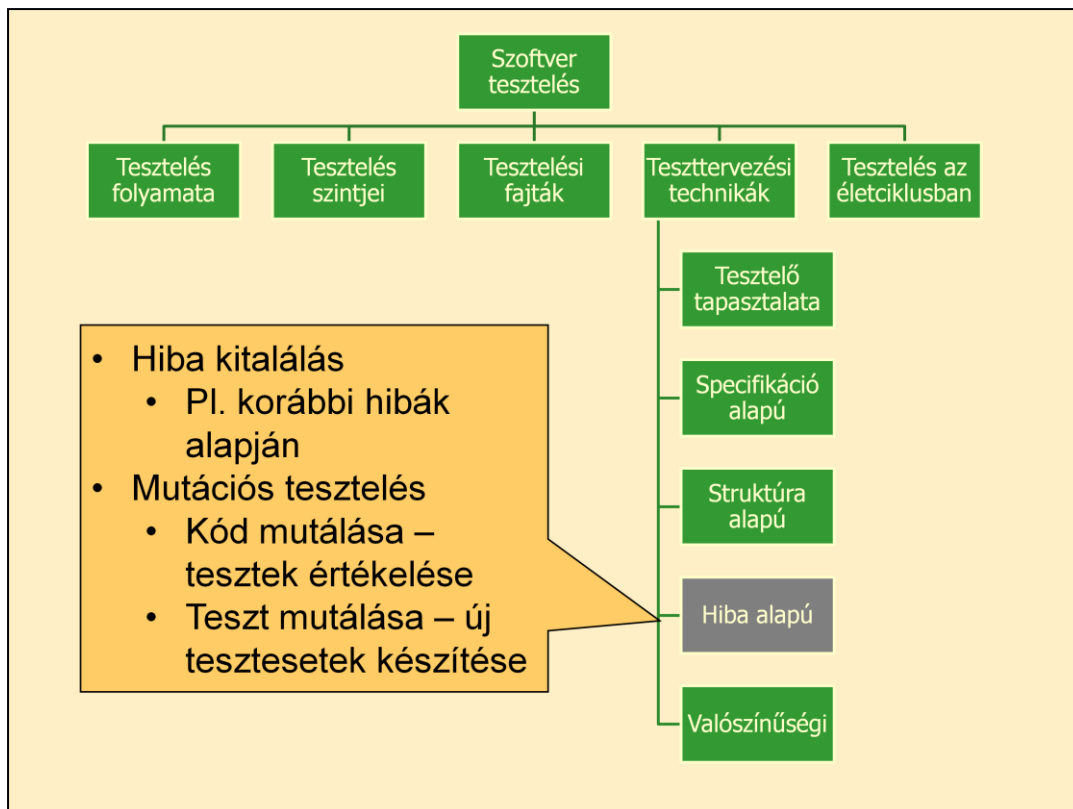
- Rothermel, G. and Harrold, M.J., "Analyzing Regression Test Selection Techniques," IEEE Trans. Software Eng., vol. 22, no. 8, pp. 529-551, Aug. 1996.
- Rothermel, J., Untch, R. H., Chu, C., Harrold, M.J., "Prioritizing Test Cases For Regression Testing," IEEE Transactions on Software Engineering, vol. 27, no. 10, pp. 929-948, October, 2001.



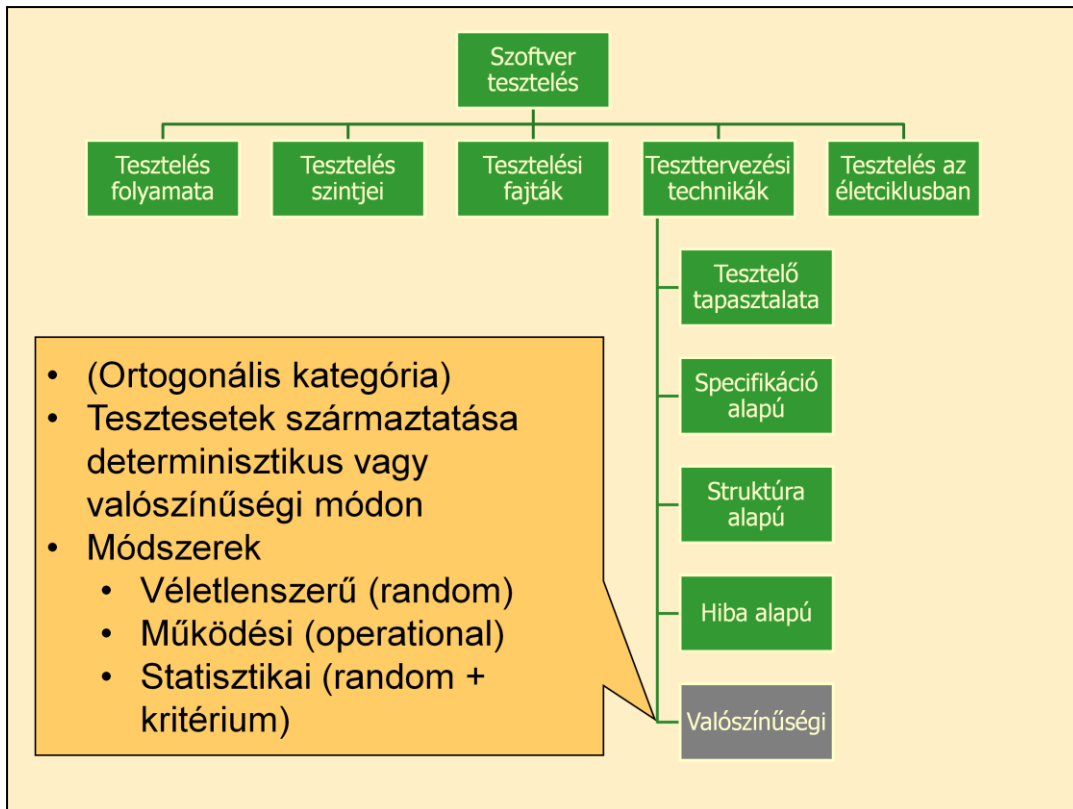


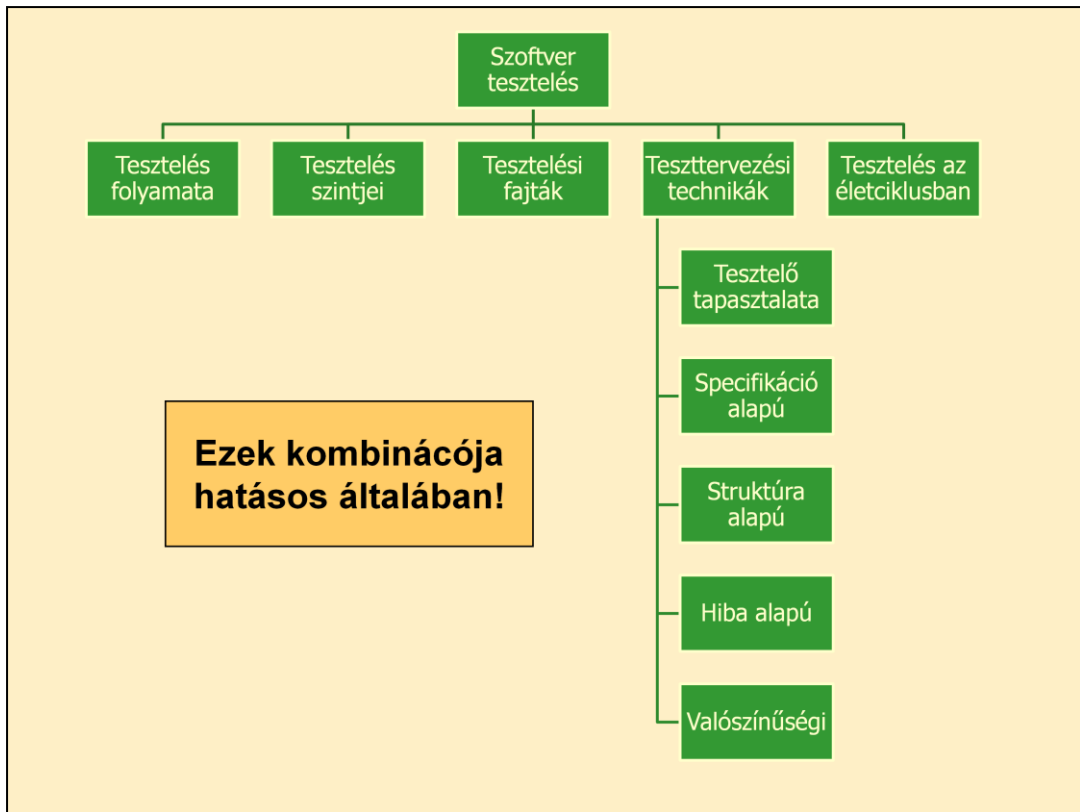


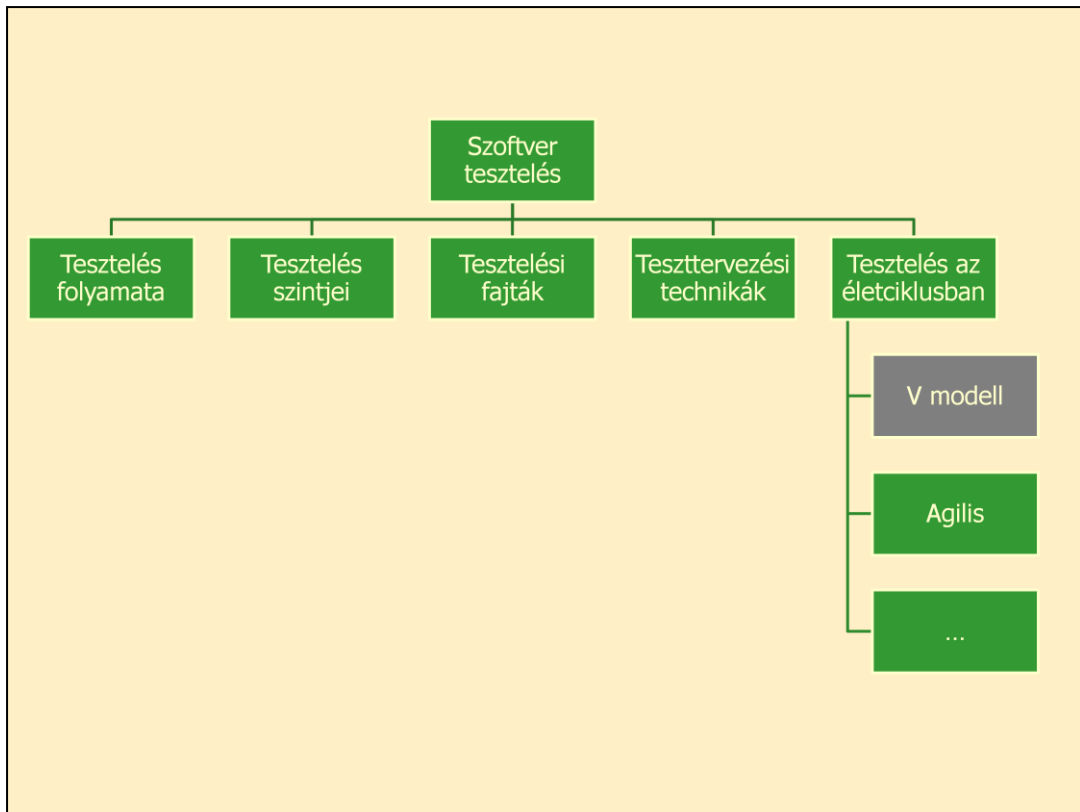




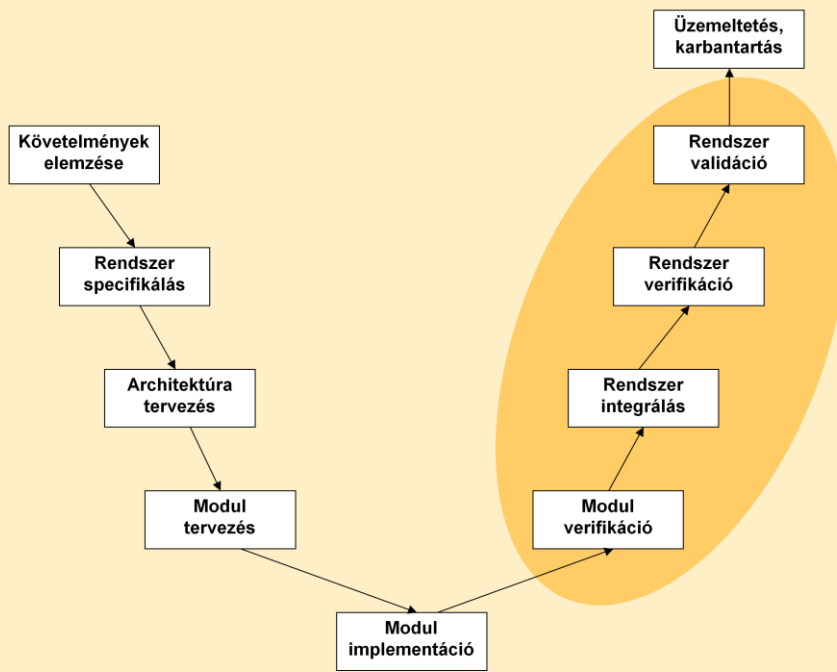
Hiba alapú (fault based)



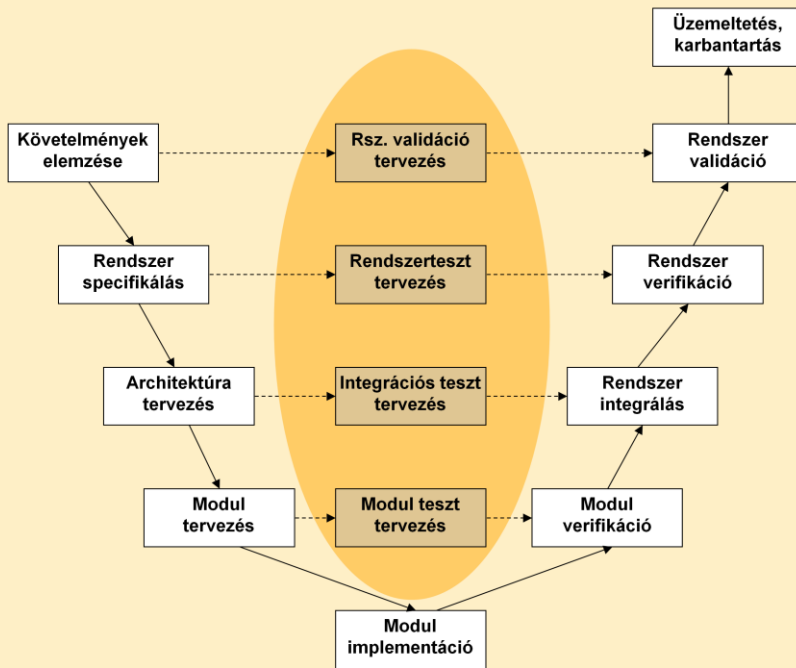


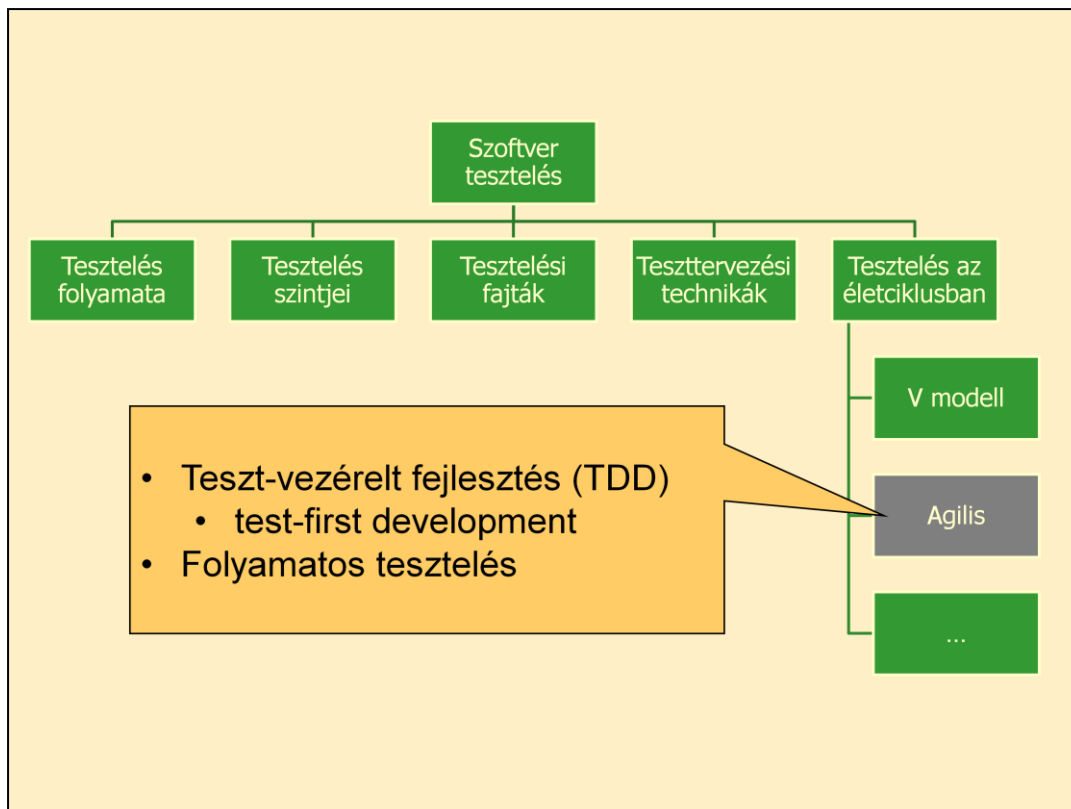


Ismétlés: Tesztelés a V modellben



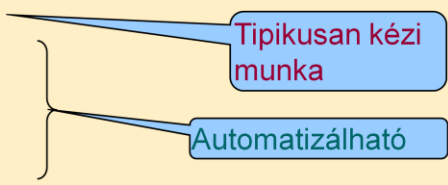
Ismétlés: A tesztelés tervezése





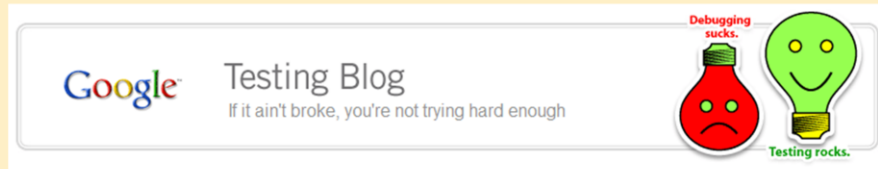
Ambler, Scott (2010). "Agile Testing and Quality Strategies: Discipline over Rhetoric". <http://www.ambysoft.com/essays/agileTesting.html>.

Tesztelés gyakorlati kérdései

- Tesztelés a fejlesztési költség több mint 50%-a!
 - **teszt adatok generálása**
 - **teszt kód írása**
 - **tesztek futtatása**
 - **eredmények kiértékelése**

The diagram shows a list of four testing tasks. A blue callout box labeled 'Tipikusan kézi munka' (Typically manual work) points to 'teszt adatok generálása'. A second blue callout box labeled 'Automatizálható' (Automatable) points to a bracket that groups the remaining three tasks: 'teszt kód írása', 'tesztek futtatása', and 'eredmények kiértékelése'.
- Az automatizálás alapja: Pl. UML modell
 - osztálydiagram: modul interfészek
 - **teszt vezérlők és teszt csonk váz generálása**
 - szekvencia diagram: modulok együttműködése
 - **teszt szekvencia**
- Lásd majd a modell-alapú tesztelés előadást

Testing @ Google



- „Testing on the Toilet“
- [Test Engineering at Google: It's Not QA](http://googletesting.blogspot.com/2007/01/introducing-testing-on-toilet.html)

<http://googletesting.blogspot.com/2007/01/introducing-testing-on-toilet.html>

Testing @ Microsoft

- Software Developer Engineer in Test (SDET)
- Fejlesztőivel egyenrangú karrierút
 - Tesztelő nem belépő pozíció
 - Teszt menedzser nem előléptetés, hanem külön út
- Kb. ugyanannyi tesztelő, mint fejlesztő
- 10 éves támogatási ciklus az OS verziókhöz
 - Megéri automatizálni
- „How we test software at Microsoft”, Microsoft Press, ISBN 0735624259, 2008.



További információ

- International Software Testing Qualifications Board (ISTQB), URL: <http://istqb.dedicated.adaptavist.com/>
 - ISTQB Glossary of Testing Terms
 - Foundation Level Syllabus (2010)
 - Magyarul is: <http://www.hstqb.com/index.php?title=Downloads>
- IEEE, Software Engineering Body of Knowledge (SWEBOK), URL: <http://www.computer.org/portal/web/swebok/>
 - Chapter 5: Software Testing
- IEEE, Software and Systems Engineering Vocabulary (SE VOCAB), URL: http://pascal.computer.org/sev_display/
 - Definíciók kereshető jegyzéke

Összefoglalás

Szoftver tesztelés

