

# A szoftver tesztelés alapjai

Micskei Zoltán, Majzik István

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

# Hol tartunk a félévi anyagban?

- Követelményspecifikáció ellenőrzése
- Ellenőrzések a tervezési fázisban
- Forráskód verifikáció technikái
- **Tesztelési módszerek és folyamatok**
  - **A szoftver tesztelés alapjai**
  - Specifikáció és struktúra alapú teszttervezési módszerek
  - Tesztelés a fejlesztés különböző fázisaiban
  - Teszt környezet és teszt automatizálás
  - Modell alapú tesztelés
- Validáció
  - GUI tesztelés
  - Stressz és robusztusság tesztelés, hibainjektálás

# A tesztelés definíciói (1)

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

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

- Általános definíció
- Kulcs: „evaluating product quality”
  - Quality: „the degree to which a system, component, or process meets specified requirements”

## 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.”

- Forrás: IEEE, "IEEE Standard for Software and System Test Documentation," *IEEE Std 829-2008*, 2008
- Kicsit specifikusabb
- Kulcs: végrehajtjuk a rendszert vagy komponenst

## 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.

– Forrás: International Software Testing Qualifications Board (ISTQB), URL: <http://istqb.org/>

– Sokkal bővebb definíció

– Kulcs:

- Tesztelés egy folyamat
- Statikus és dinamikus technikák is

# 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 szükséges
- Hibák csoportosulása (defect clustering)
  - A komponensek egy kis részében van a hibák nagy része
- Féregirtó paradoxon (pesticide paradox)
  - Teszt újra és újra futtatva egyre kevésbé hatékony
  - Minden módszer után marad még valami más típusú hiba
- A tesztelés függ a körülményektől (context dependent)
- Hibamentes rendszer téveszméje
  - Hibamentes tesztelfutás még nem jelent jó rendszert

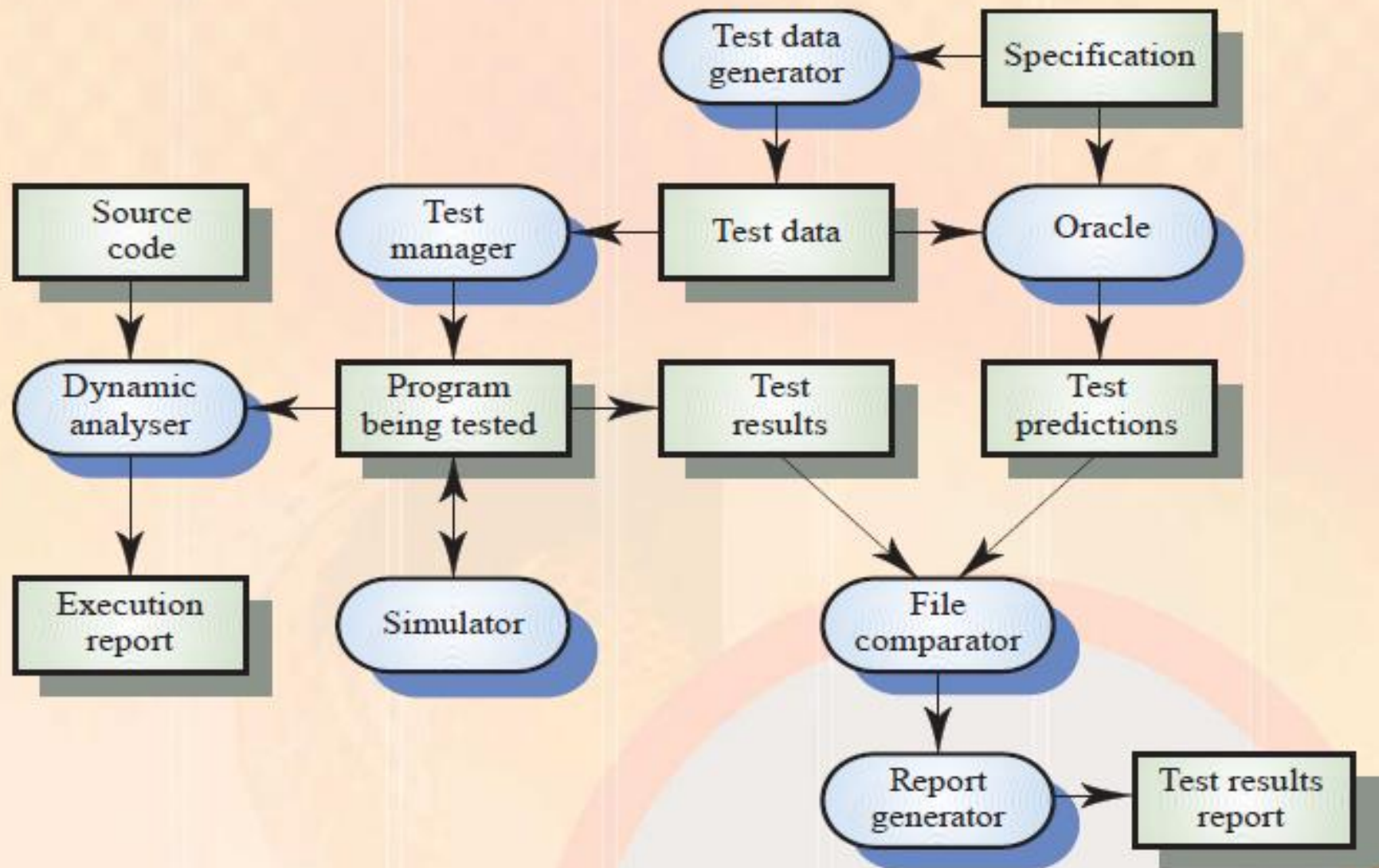
# Alapfogalmak



- **Teszteset (test case)**
  - Bemeneti értékek és végrehajtási előfeltételek
  - Várt eredmények és végrehajtási utófeltételek halmaza
- **Tesztkészlet (test suite)**
- **Orákulum (test oracle)**
  - Várt eredmények származtatása, összehasonlítása
- **Eredmény (verdict)**
  - Sikeres (pass), sikertelen (fail)
  - Nem meggyőző (inconclusive), hiba (error)
- **Tesztelés != Hibakeresés (debugging)**

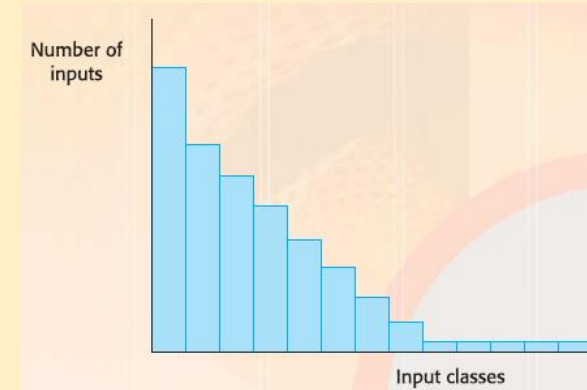


# Tesztelési környezet (eszközök)



# Alap problémák

- **Teszt kiválasztás (test selection)**
  - Milyen teszt bemeneteket használjunk?
  - Ld. működési profil szerepe
- **Kilépési feltétel (exit criteria)**
  - Meddig teszteljünk?
    - Specifikáció fedése, kód fedése, hibák fedése?
  - Ld. megbízhatóság előrejelzése
- **Orákulum**
  - Honnan lesz jó teszt orákulum?
- **Tesztelhetőség (testability)**
  - Megfigyelhetőség (observability)
  - Vezérelhetőség (controllability)

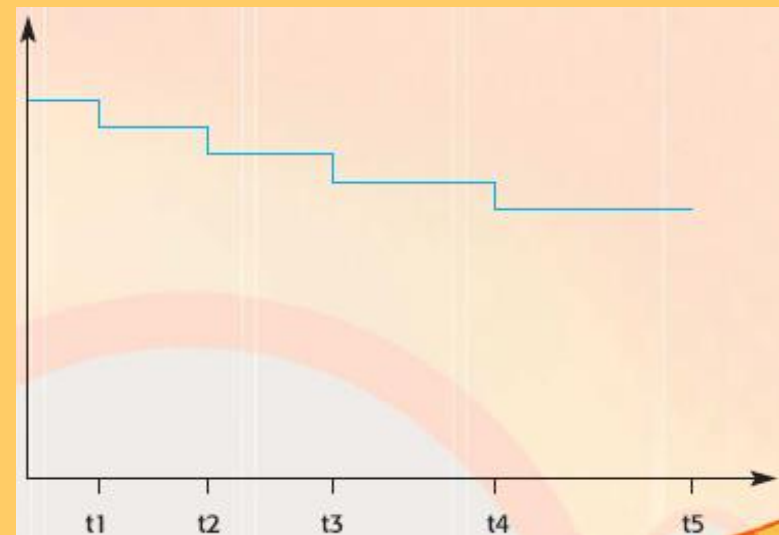
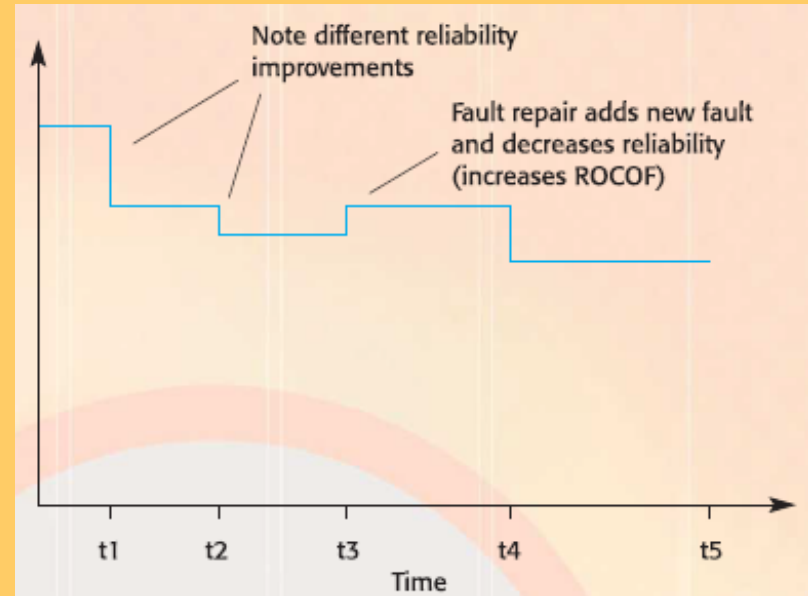


# A megbízhatóság előrejelzése

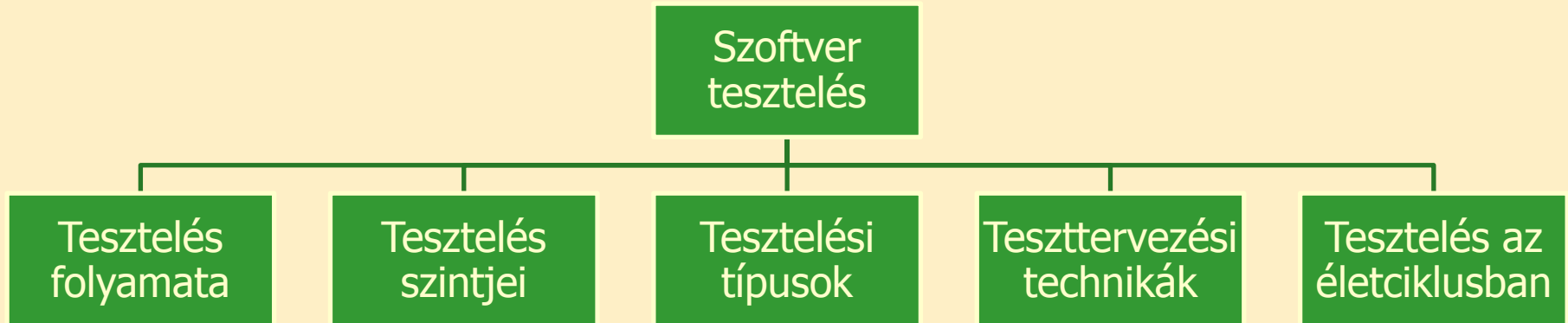
- Alapkérdés: A tesztelési folyamat során hogyan változik a szoftver megbízhatósága?
  - Hibák detektálása és javítása:  
Meghibásodási gyakoriság csökken
- Megbízhatóság növekedési modellek
  - Legegyszerűbb: Lépcsős függvény modell
    - Tökéletes hibajavítás
    - Azonos gyakoriságú hibák
  - Komplikáltabb: Véletlen ugrású lépcsős függvény
    - A hibajavítás újabb hibát bevihet
    - A javított hibák különböző gyakoriságúak
  - Folytonos modellek
    - Adott feltételek mellett alkalmazhatók (feladattípus, paradigma, komplexitás, fejlesztők, ...)
    - Mért hibajavítási adatok illesztése szükséges

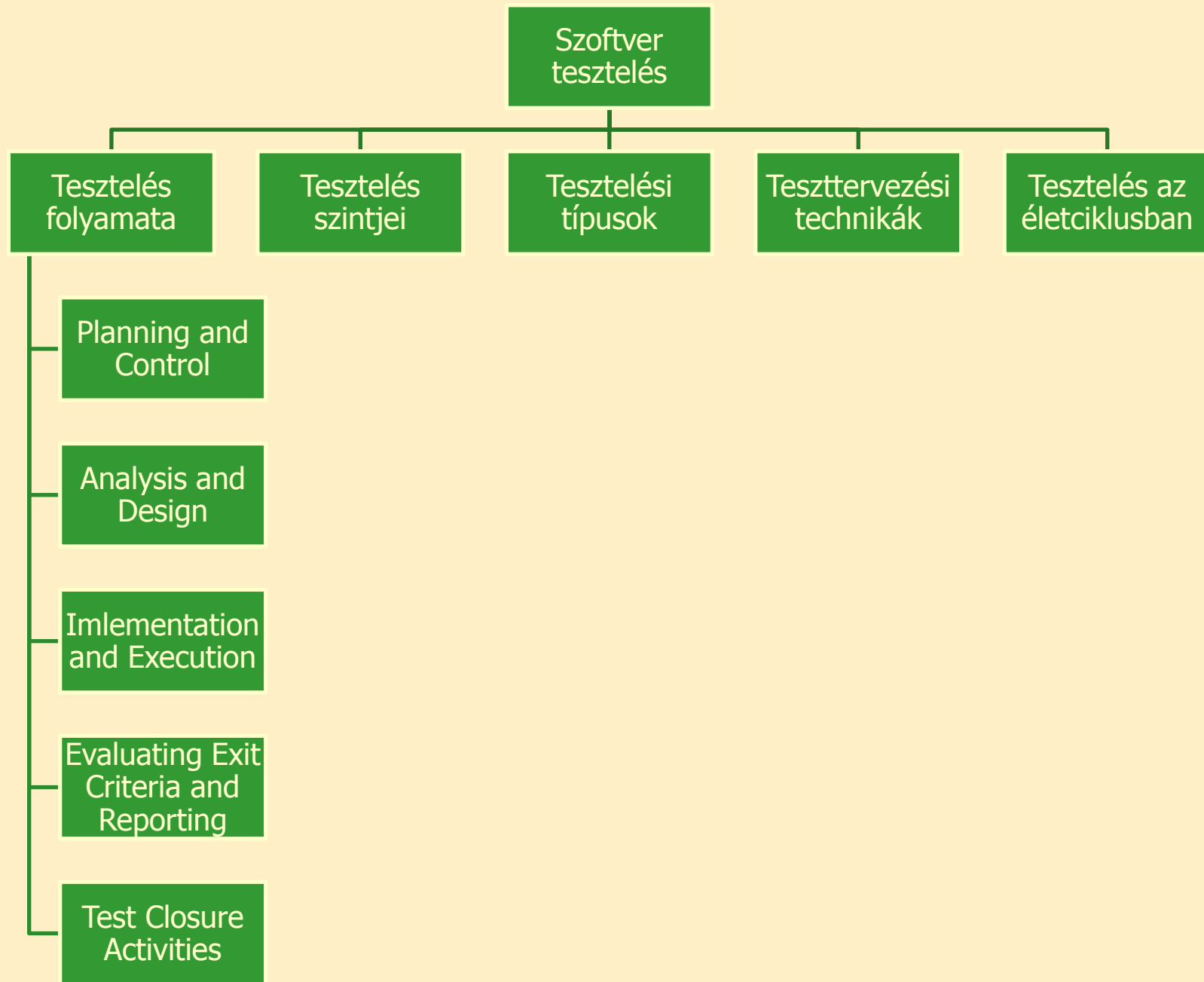
# A megbízhatóság

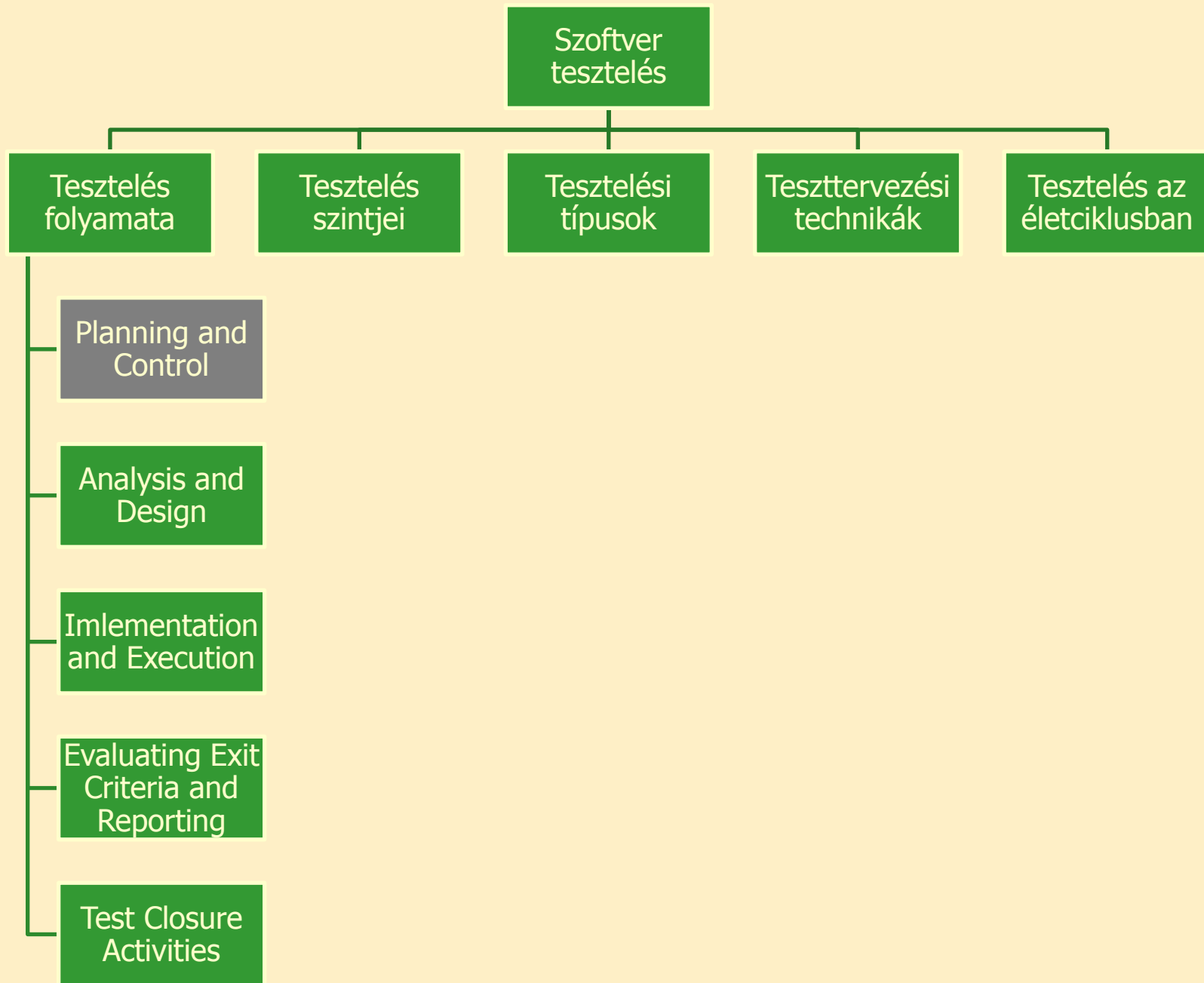
- Alapkérdés: A tesztelés során hogyan változik a szoftver megbízhatósága?
  - Hibák detektálása és javítása
  - Meghibásodási gyakoriság
- Megbízhatóság növekedése
  - Legegyszerűbb: Lépcsős modellek
    - Tökéletes hibajavítás
    - Azonos gyakoriságú hibák
  - Komplikáltabb: Véletlen hibák
    - A hibajavítás újabb hibákat okoz
    - A javított hibák különböző gyakoriságúak
  - Folytonos modellek
    - Adott feltételek mellett a szoftver megbízhatósága
    - Paradigma, komplexitás
    - Mért hibajavítási adatok



# A tesztelési fogalmak áttekintése







# Tesztelési stratégia

- Általános irányelvek
  - Milyen metodológiát?
  - Milyen típusú tesztek?
  - Milyen eszközöket?
  - Ki fogja használni?
  - Milyen kilépési feltétellel?
  - Milyen dokumentáció kell?
  - ...
- Példák az irányelvekre:
  - Test-driven development
  - 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...
  - Tesztelendő funkciók
  - Erőforrások, szerepek, ütemezés
- Tesztelési fázisok definiálása
  - Fázisok hossza
  - Kilépési feltétel
  - Tesztelés minőségét hogyan fogjuk mérni
- Egyszerű példa: <http://bazman.tripod.com/>

# Teszt dokumentáció

## 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

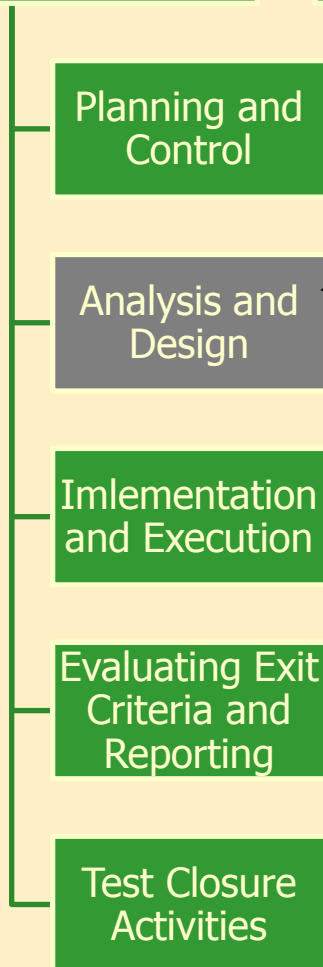
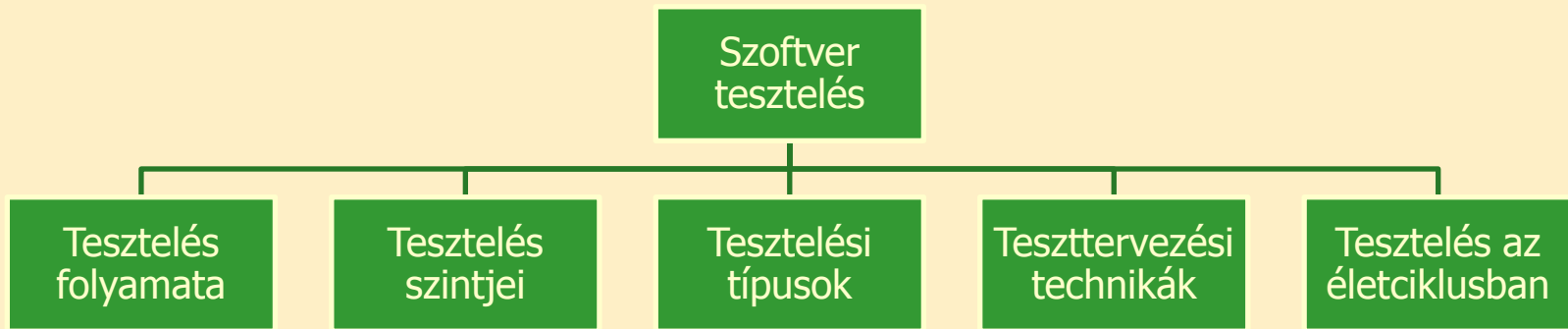
### 3. Test management

- 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)

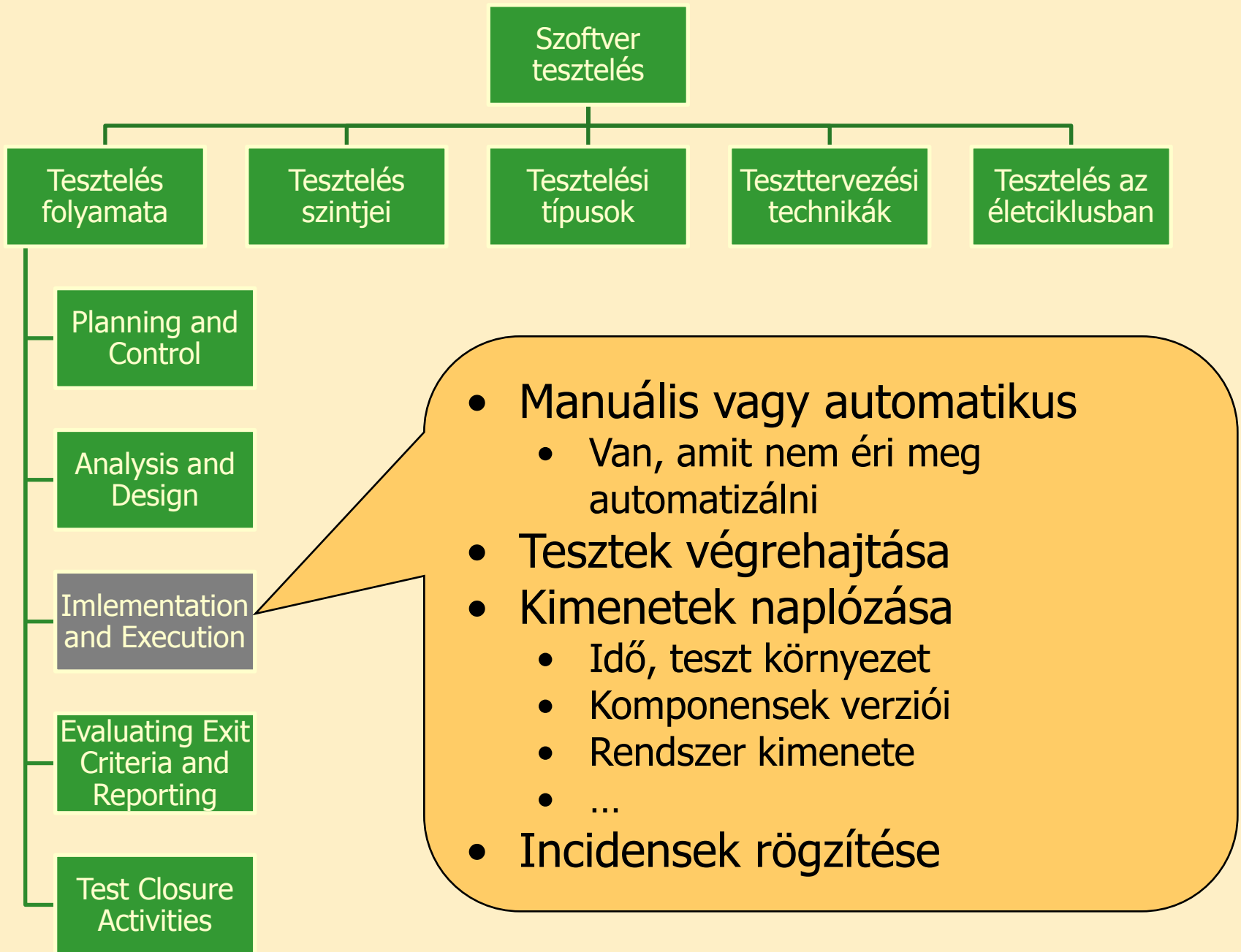
### 4. General

- 4.1 Quality assurance procedures
- 4.2 Metrics
- 4.3 Test coverage
- 4.4 Glossary
- 4.5 Document change procedures and history

- IEEE 829 - Standard for Software and System Test Documentation (1998)
  - **Test Plan** (SPACEDIRT: Scope, People, Approach, Criteria, Environment, Deliverables, Incidentals, Risks, Tasks)
  - **Test specifications**: Test Design, Test Case, Test Procedure Specifications
  - **Test reporting**: Test Item Transmittal Report, Test Log, Test Incident Report, Test Summary Report



- Mit lehet és kell tesztelni?
- Tesztesetek tervezése, specifikálása
  - Teszteset célja
  - Kiindulási környezet
  - Teszteset lépései, teszt adatok
  - Elvárt kimenet
- Mielőtt elkezdenénk tesztkódot írni
- Szisztematikus módszer ajánlott





Planning and Control

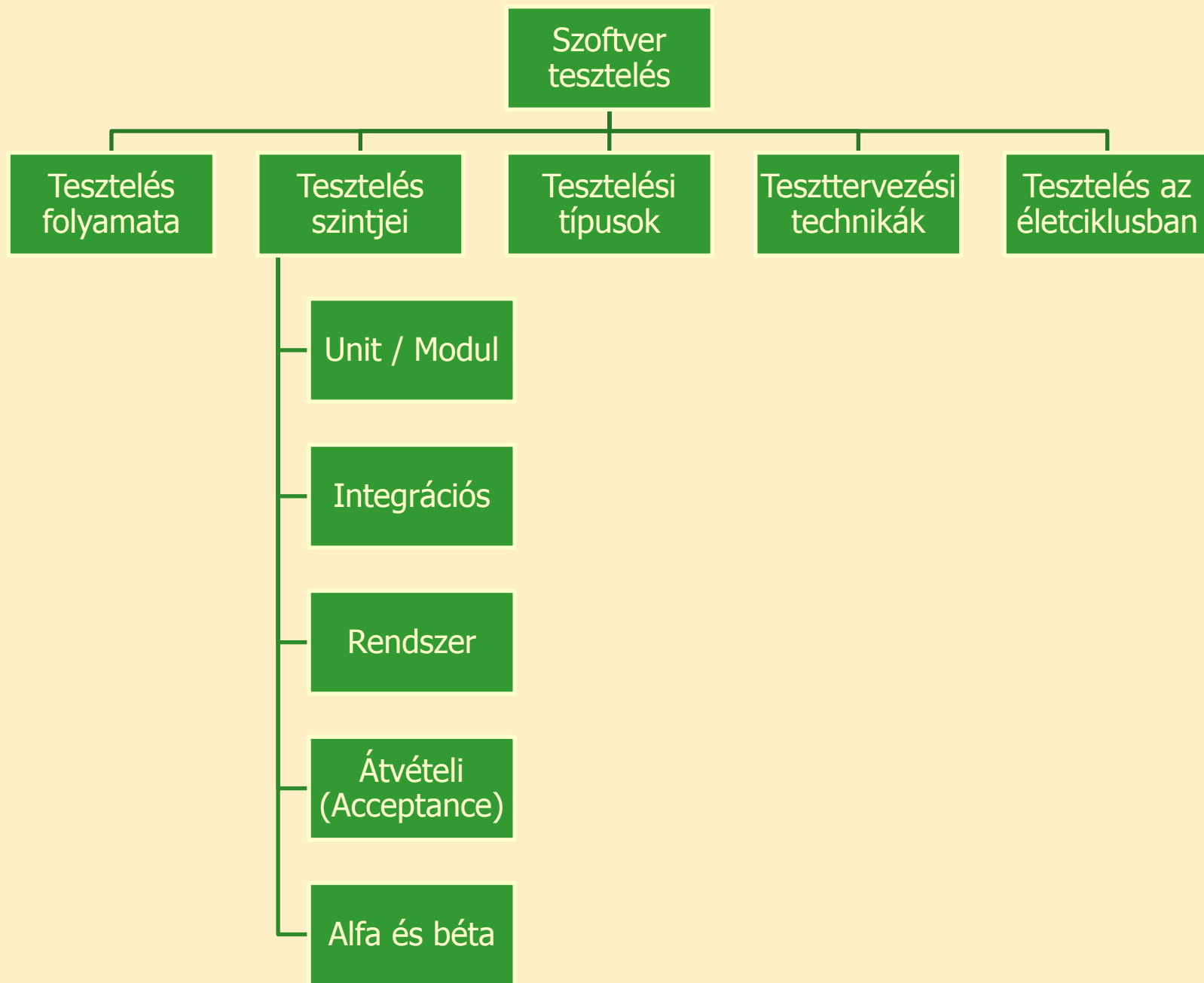
Analysis and Design

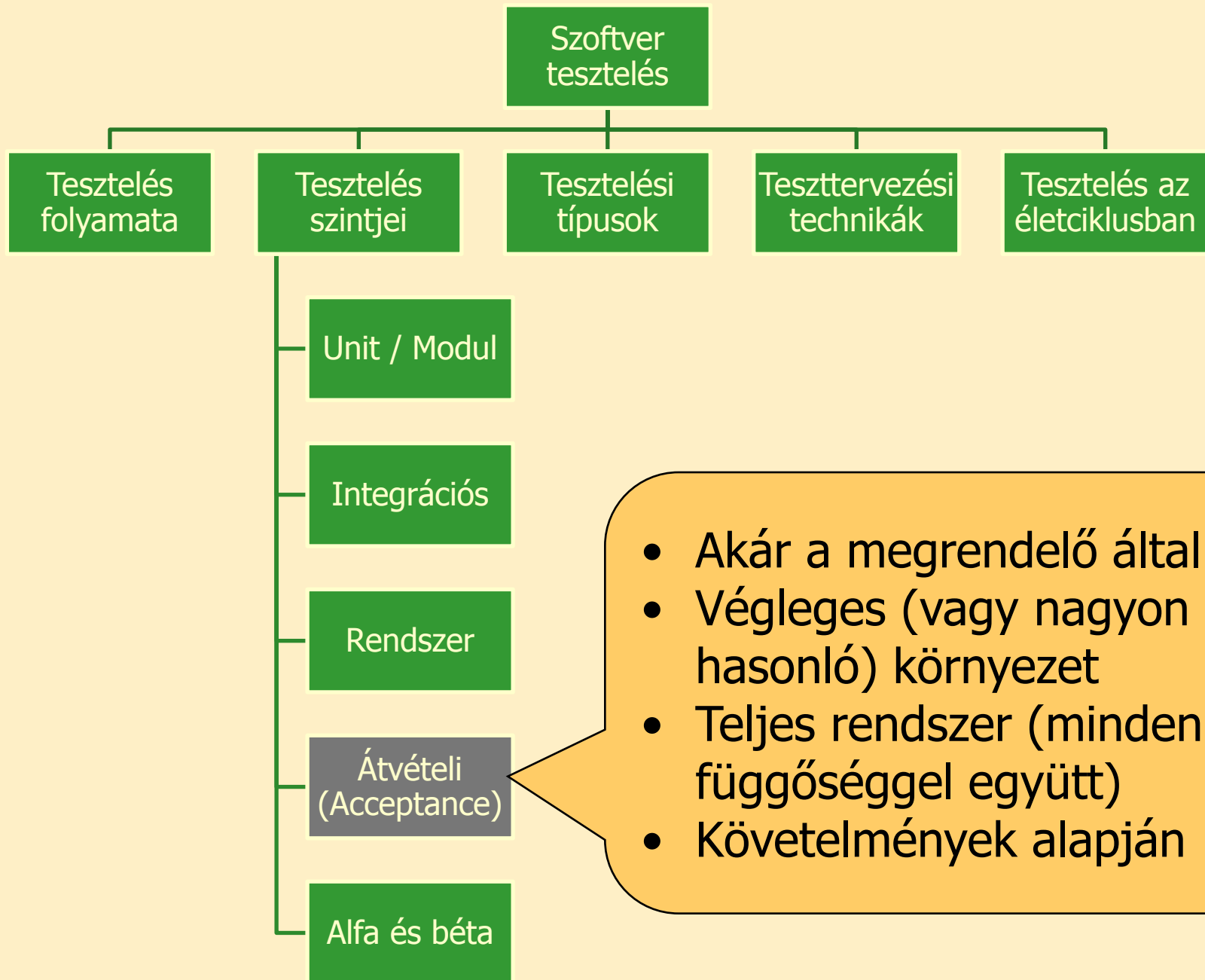
Implementation and Execution

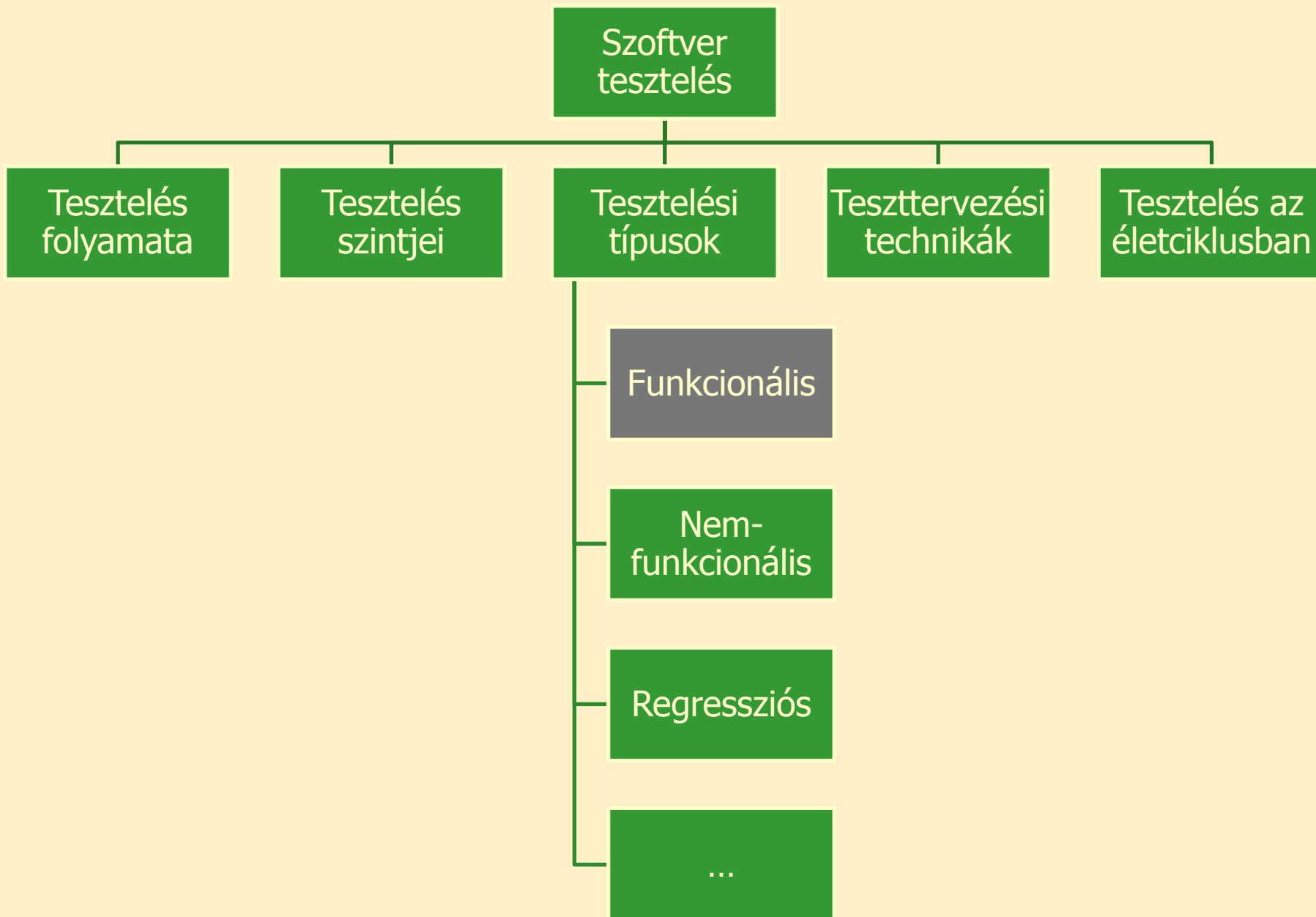
Evaluating Exit Criteria and Reporting

Test Closure Activities

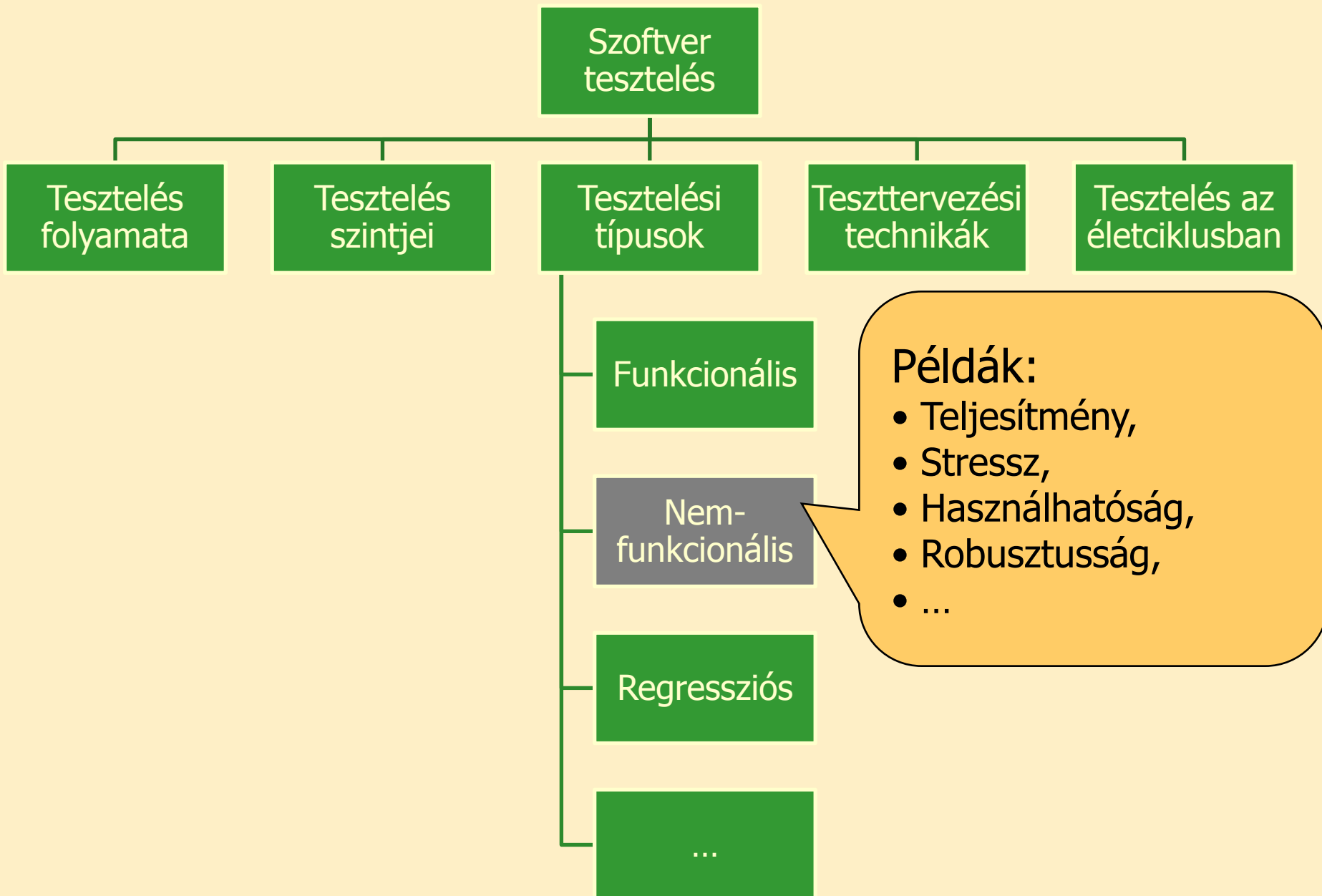
- Döntés a tesztelés folytatásáról
- Összefoglaló jelentés
- Tesztelés lezárása
  - Tipikusan mérföldkövenként
  - Tapasztalat összegyűjtése
  - Teszt eszközök, környezetek (testware) befejezése

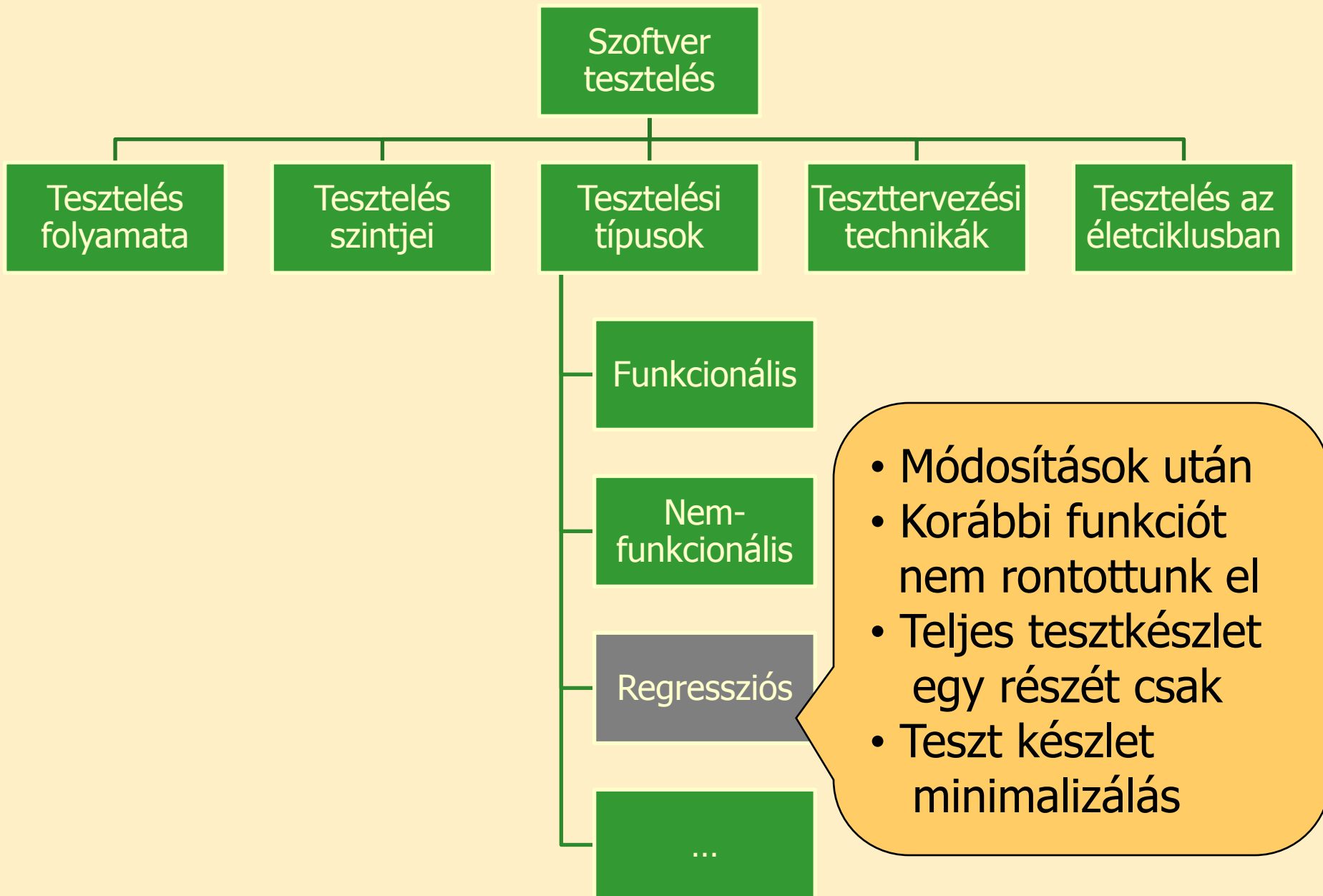


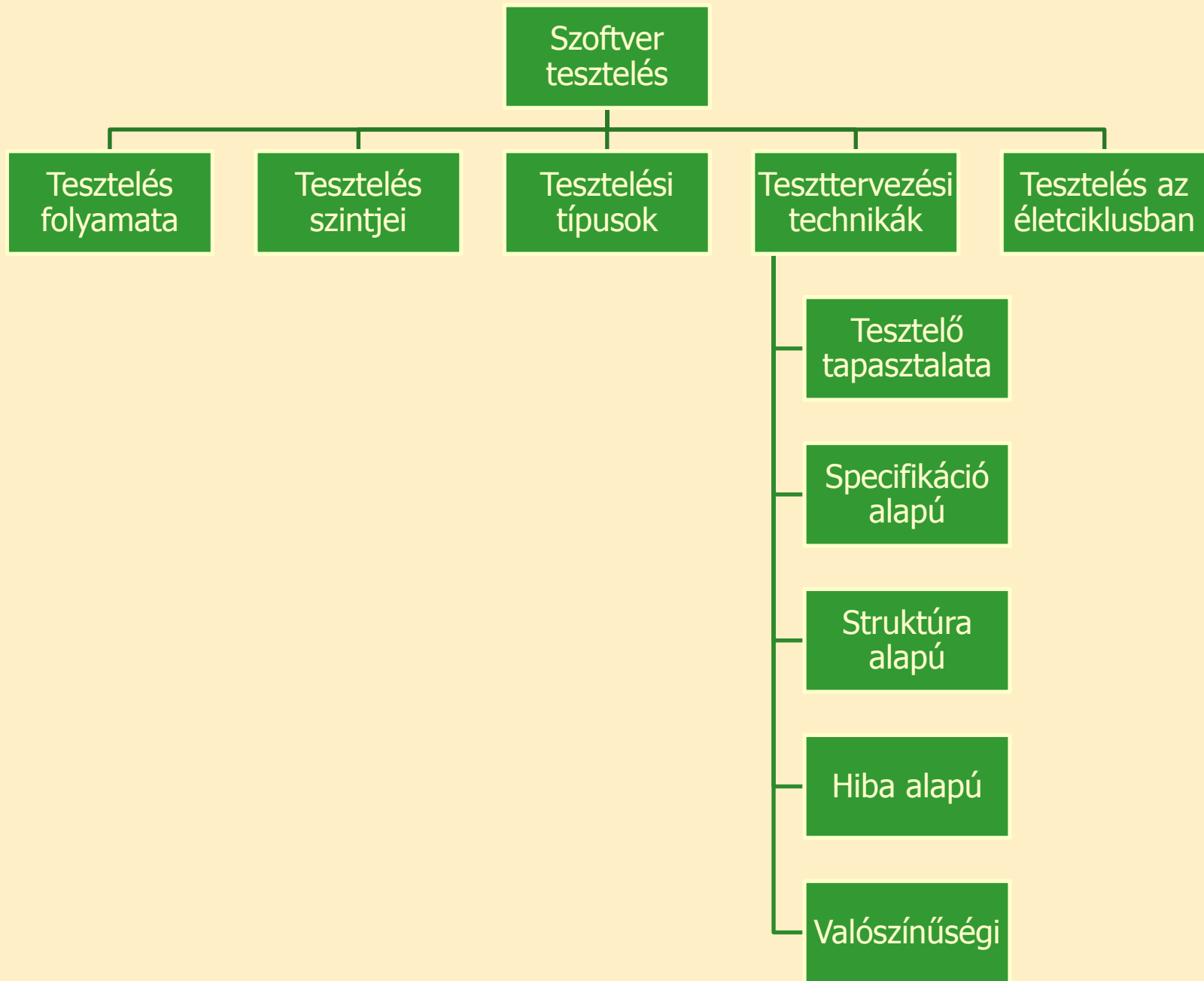












# Szoftver tesztelés

Tesztelés  
folyamata

Tesztelés  
szintjei

Tesztelési  
típusok

Teszttervezési  
technikák

Tesztelés az  
életciklusban

Tesztelő  
tapasztalata

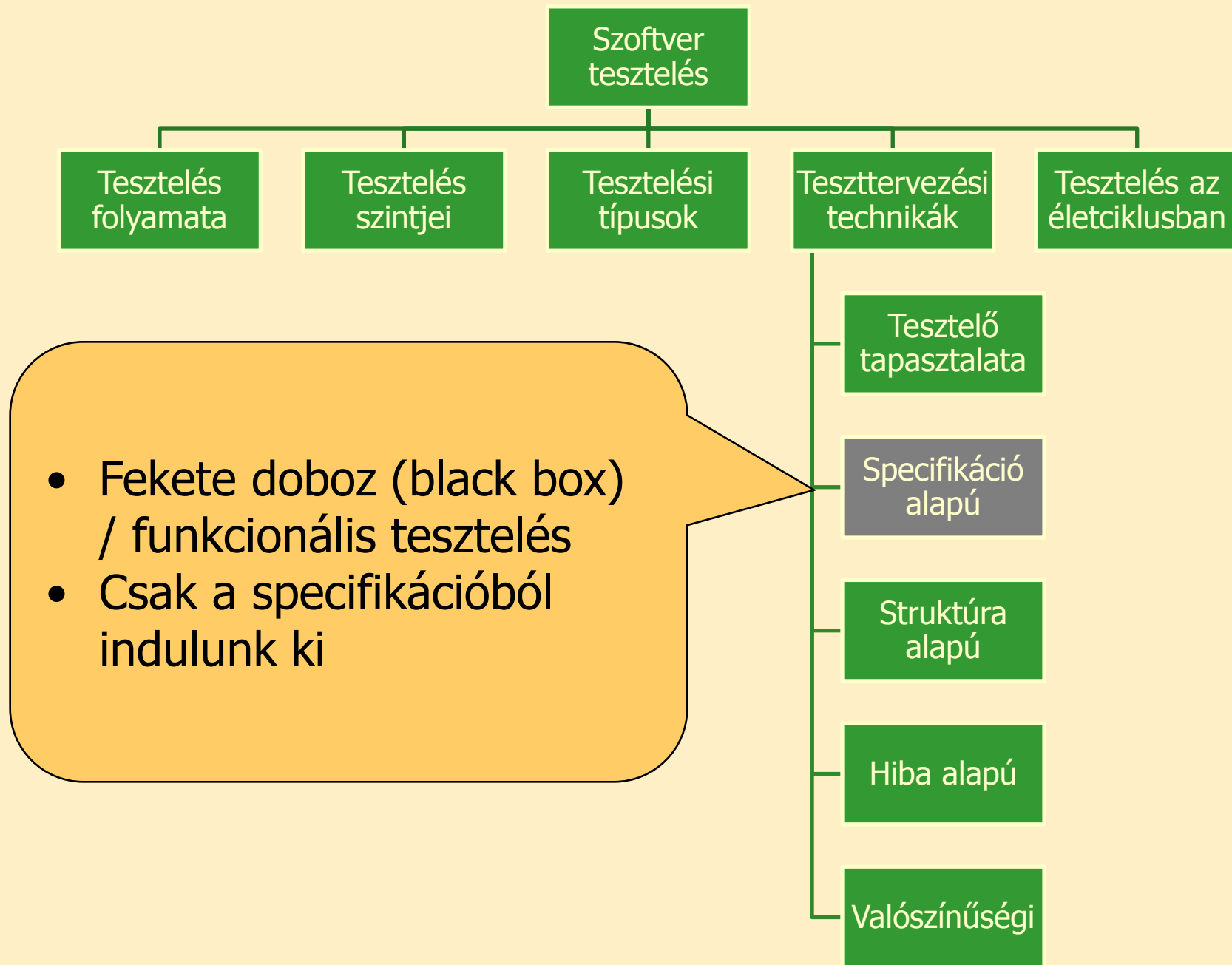
Specifikáció  
alapú

Struktúra  
alapú

Hiba alapú

Valószínűségi

- Ad hoc tesztelés
- Felderítő tesztelés (exploratory)
  - Rendszer megismerése
  - Teszt tervezés és alkalmazás futtatása együtt



# Szoftver tesztelés

Tesztelés  
folyamata

Tesztelés  
szintjei

Tesztelési  
típusok

Teszttervezési  
technikák

Tesztelés az  
életciklusban

Tesztelő  
tapasztalata

Specifikáció  
alapú

Struktúra  
alapú

Hiba alapú

Valószínűségi

- Fehér doboz (white box) / strukturális tesztelés
- Belső működést is ismerjük
- Forráskód, részletes terv, belső modell ismert

# Szoftver tesztelés

Tesztelés  
folyamata

Tesztelés  
szintjei

Tesztelési  
típusok

Teszttervezési  
technikák

Tesztelés az  
életciklusban

Tesztelő  
tapasztalata

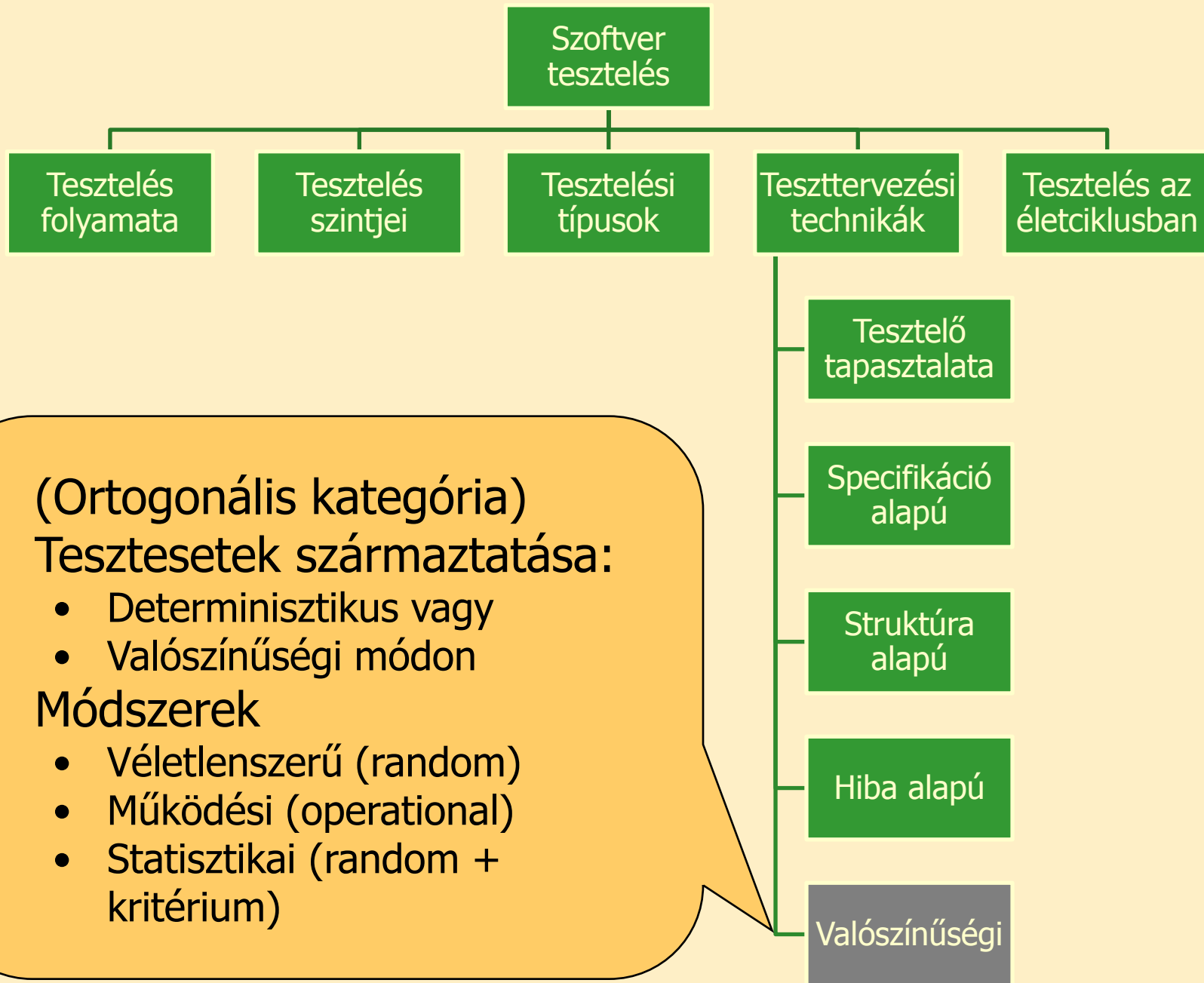
Specifikáció  
alapú

Struktúra  
alapú

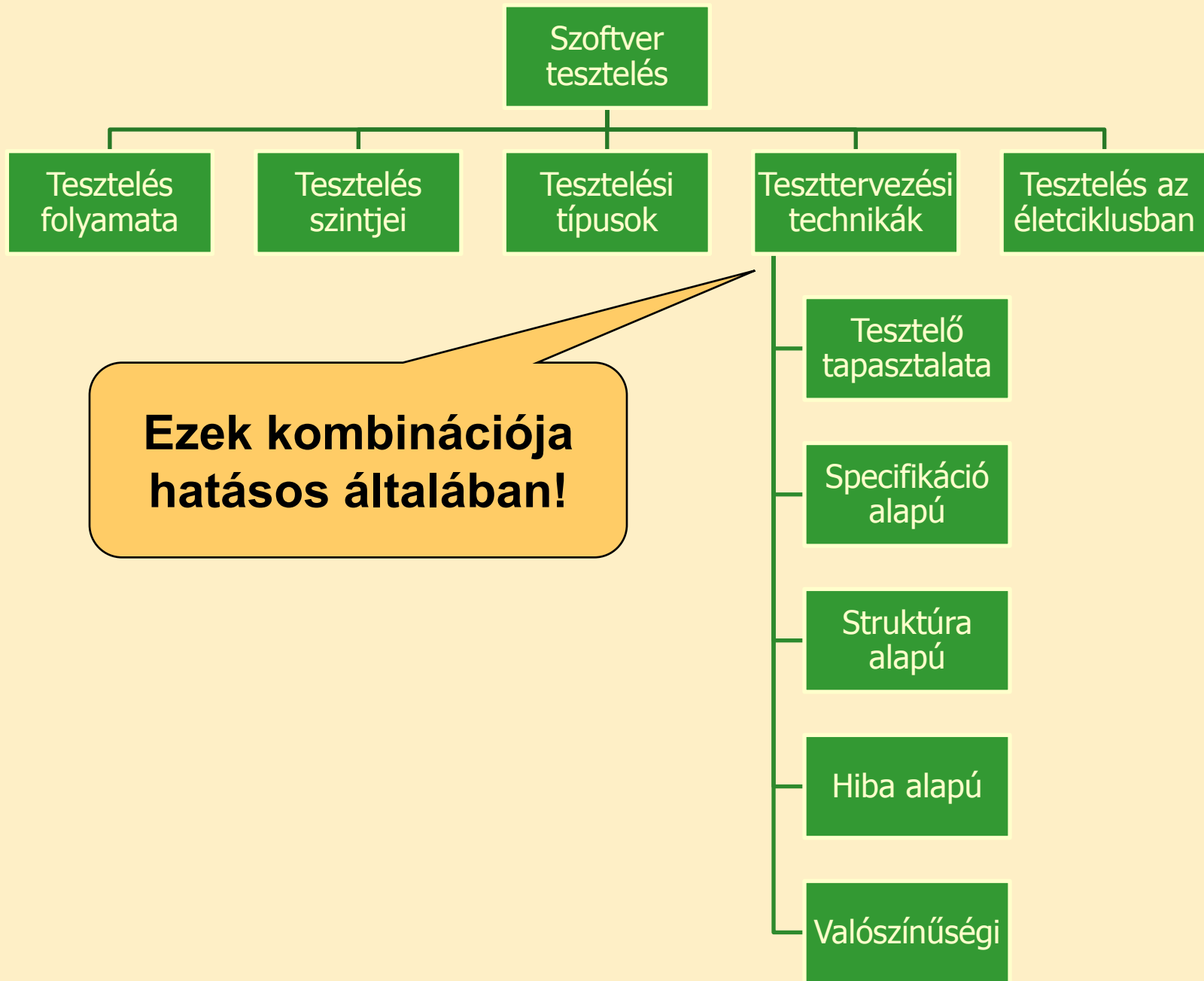
Hiba alapú

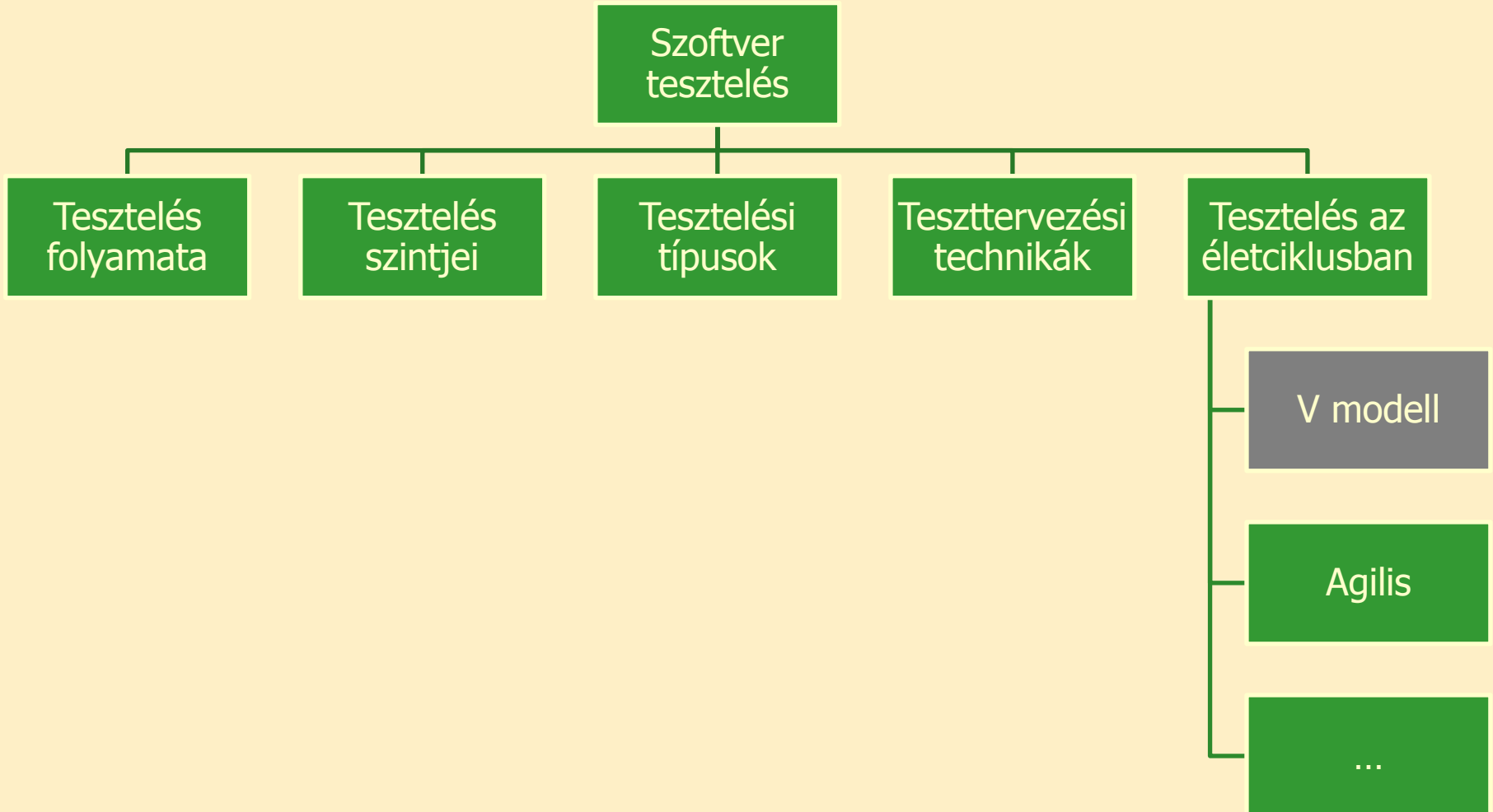
Valószínűségi

- Hibasejtés (error guessing)
  - Pl. korábbi hibák alapján
- Mutációs tesztelés
  - Kód mutálása
    - tesztek értékelése
  - Teszt mutálása
    - új tesztesetek készítése

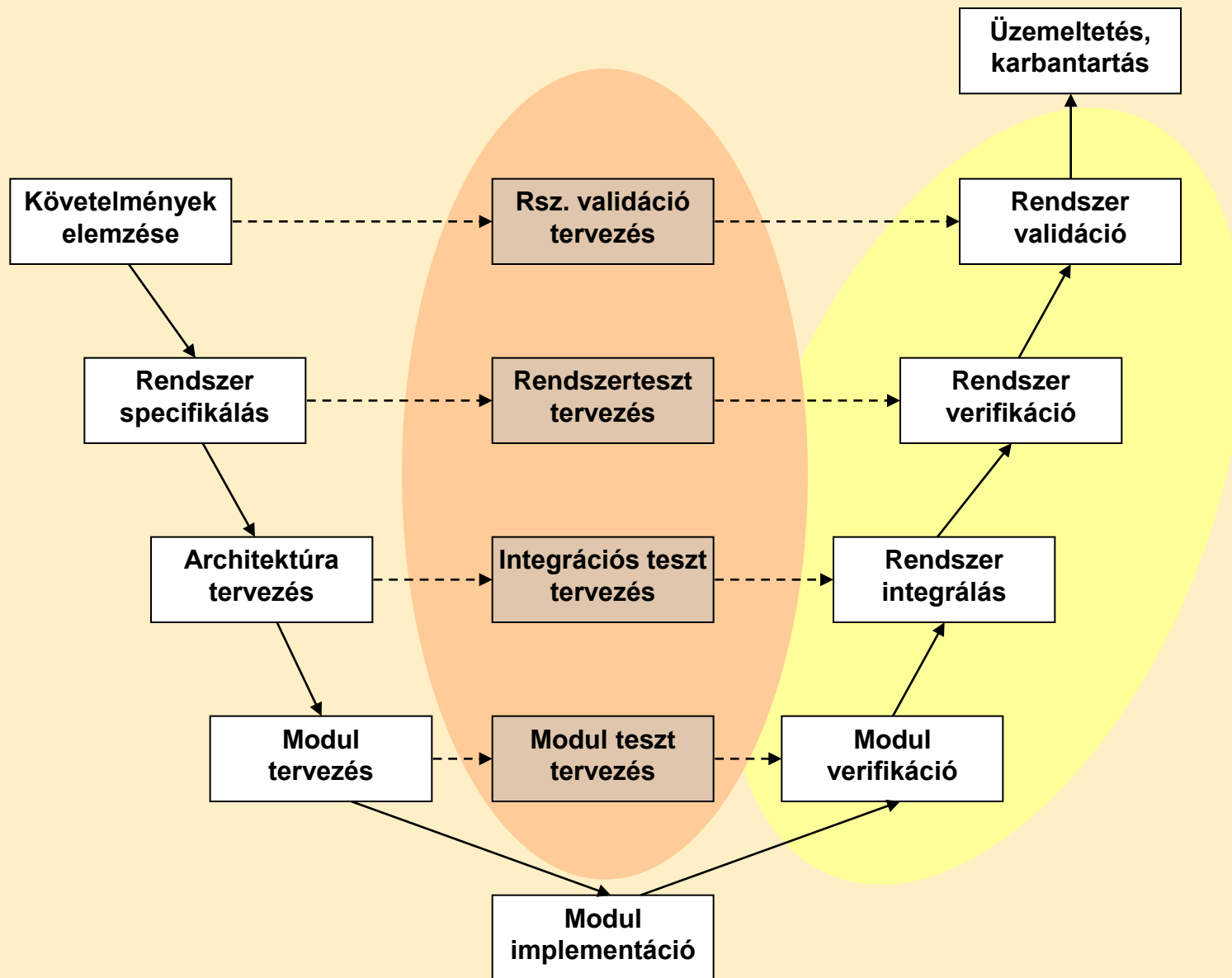


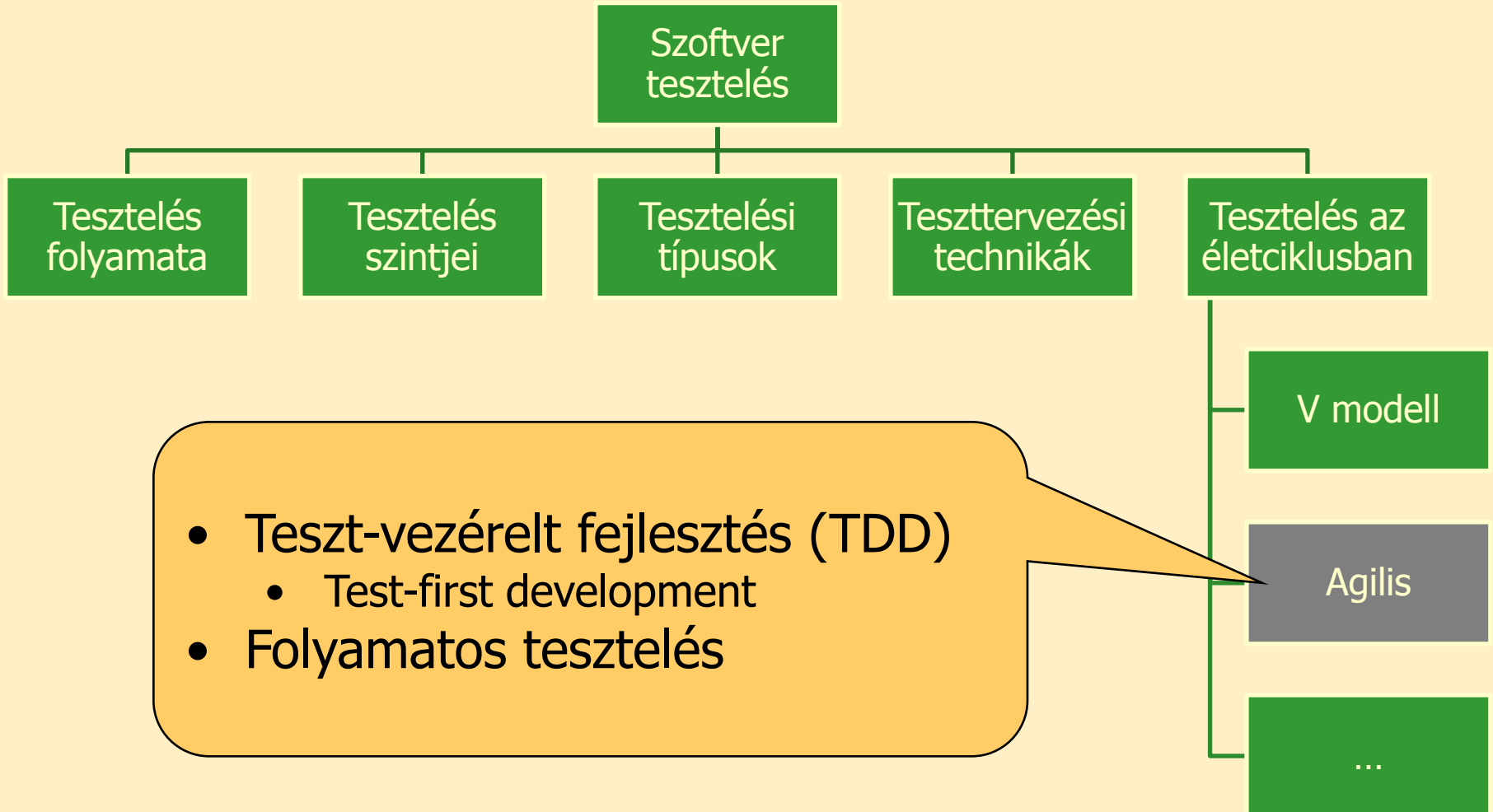






# Ismétlés: A tesztelés tervezése

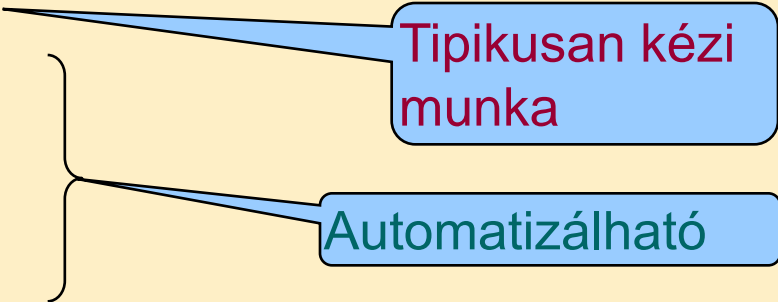




# Tesztelés gyakorlati kérdései

- Tesztelés a fejlesztési költség több mint 50%-a!

- Teszt adatok generálása
- Teszt kód írása
- Tesztek futtatása
- Eredmények kiértékelése



Tipikusan kézi munka

Automatizálható

- Az automatizálás alapja: pl. UML modell

- Osztálydiagram: modul interfészek
  - Teszt vezérlők és teszt csomópont váz generálása
- Szekvencia diagram: modulok együttműködése
  - Teszt szekvencia származtatása

- Lásd majd a modell-alapú tesztelés előadást!

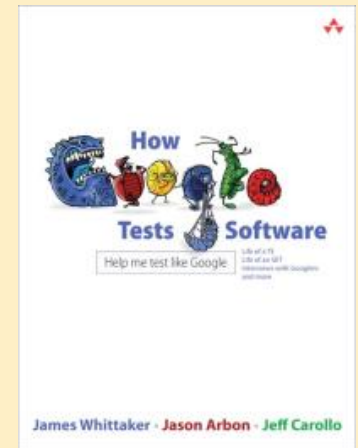
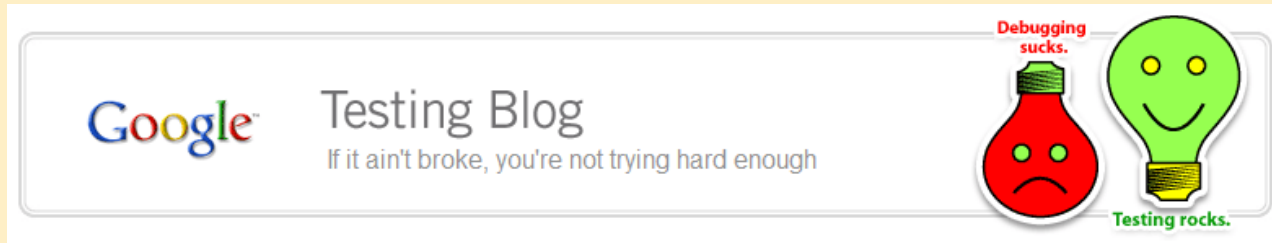
# Testing @ Microsoft

- Software Developer Engineer in Test (SDET)
- Kb. ugyanannyi tesztelő, mint fejlesztő
- Fejlesztőivel egyenrangú karrierút
  - Tesztelő nem belépő pozíció
  - Teszt menedzser nem előléptetés, hanem külön út
- 10 éves támogatási ciklus az OS verziókhoz
  - Megéri automatizálni a tesztelést

„How we test software at Microsoft”, Microsoft Press, ISBN 0735624259, 2008.



# Testing @ Google



- Software Engineer in Test (SET) és Test Engineer (TE) szerepek
- „The burden of quality is on the shoulders of those writing the code.”
- „Ne vegyünk fel túl sok tesztelőt”

# További információk

- International Software Testing Qualifications Board (ISTQB), URL: <http://istqb.org/>
  - ISTQB Glossary of Testing Terms
  - Foundation Level Syllabus (2011)
  - 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/](http://pascal.computer.org/sev_display/)
  - Definíciók kereshető jegyzéke



# Összefoglalás

## Szoftver tesztelés

