



Rendszer és szoftvertesztelés

Varjasi Norbert
norbert.varjasi@sze.hu



Elvárás: a rendszerek hibamentessége

- **Szolgáltatási szint szerződések (SLA)**
 - pl extrém magas sla: „Öt kilences” szolgáltatás: 99,999% (5 perc/év kiesés)
- **Biztonságkritikus rendszerek:**
 - Szabvány előírások a hibák gyakoriságára
 - Biztonságintegritási szintek(SIL) szerint

Verifikáció és validáció

Verifikáció

-igazolás

– jól terveztem?

- Fejlesztési fázisok
- Objektív, ellenőrizhető
- Jó terv?
- Jó megvalósítás?

Validáció

– érvényesítés

- jó programot készítettem?

Eredmények ellenőrzése

Kész rendszer <> felhasználói
elvárások

Követelményeket jól
fogalmazták meg?

Alapelvek (7 testing principles)

1. A tesztelés csak a hibák jelenlétét mutatja meg.

- Kimutatható, hogy hibás, vagy sem
- Teszteléssel csökkenthető a hiba esélye
- Ha nincs hiba, attól még nem biztos hogy jó a sw

Alapelvek (7 testing principles)

2. Teljes teszt nem lehetséges

- Minden előfeltételt lehetetlen tesztelni (lásd még: ciklomatikus komplexitás)
- Pl: 15 bemenet, 5-5 lehetséges érték $\rightarrow 5^{15}$ teszteset
- Helyette: priorizálás, kockázatok elemzése

Alapelvek (7 testing principles)

3. Korai teszt

- Az élelciklus elején kezdeni
- Specifikáció+dokumentáció is tesztelhető, nem csak a SW
- Meghatározni a teszt célját és kereteit

Alapelvek (7 testing principles)

4. Hibák csoportosulása (defect clustering)

- Tesztelési idő véges
- A hibák nagy része egy-egy kisebb részben koncentrálódik
- Fókusz a legvalószínűbb hibahelyekre
- (határértékek, bemenetek, felhasználó)

Alapelvek (7 testing principles)

5. Féregirtó paradoxon (pesticide paradox)

- Ugyanazon tesztet többször futtatva nem jön elő új hiba
- Ugyanakkor bizonyos hibák még megmaradnak
- Megoldás:
 - tesztek elemzése, újratervezés
 - Más módszerek bevezetése

Alapelvek (7 testing principles)

6. Körülményektől függő tesztelés (content dependent)

A rendszer tulajdonságaitól függ a tesztelés

- Időkeretek
- Jellegtől függő: SLA, kritikusság

Alapelvek (7 testing principles)

7. Hibátlan rendszer téveszme

- Ha nem találtunk hibát, vagy minden teszt sikeres, (verifikáció rendben) attól még a lehet, hogy a validáció elbukik.
- Felesleges minden teszt, ha a megrendelő nem azt kapja, mint amit kért :(

Fogalmak

- **Teszt eset (test case)**
- **Teszt készlet (test suite)**
- **Orákulum - a várt eredmények elemzése Eredmény (verdict)**
- **Sikeres (pass) vagy sikertelen (fail)**
- **Teszt tervezés (test plan)**
- **Teszt specifikáció**
- **Teszt jelentés (report)**

Megj: a tesztelés != hibakeresés

Tesztelési folyamat

- 1) Tesztelés tervezés lépései**
- 2) Tesztelési szintek**
- 3) Tesztelés típusai**
- 4) Teszt tervezés technikák**
- 5) Tesztelés és életciklus**

1. Tesztelés tervezés lépései

1.1: stratégia alkotás

- Milyen módszerrel, mennyit, mit, milyen kimeneti követelményekkel. Mit mérünk?

1.2: teszt analízis és terv

- Tesztesetek, környezet, elvárt eredmények

1.3: megvalósítás

- Manuális/automata, naplózás, verziók, incidensek

1.4 kiértékelés, kilépési feltétel

- Döntés, jelentések, mérföldkövek

2. Tesztelési típusok

2.1: egységteszt (unit test)

2.2: moduláris teszt

2.3: integrációs teszt

2.4: rendszerteszt - end-to-end

2.5: elfogadási (acceptance) - éles környezeti teszt

2.6: alfa és béta teszt

2.7 egyéb tesztek:

- **biztonság, véletlen tesztek, stressz, teljesítmény, kompatibilitási, telepítési/helyreállítási teszt, ...**

3. Tesztelés típusai

3.1: Funkcionális

- Ekvivalencia partíciók
- Határérték analízis
- Ok-okozati analízis
- Kombinatorikus módszerek
- Véges automaták

3.2: Nem funkcionális (használhatóság, robosztusság)

3.3: Regressziós tesztek - folyamatosan bővülő eszkéskészlettel

4. Teszt tervezés technikák

- 4.1: Ad-hoc, fejlesztői tapasztalat alapú
(rendszer működési elemzése, tipikus hibák alapján)
- 4.2: Specifikáció alapján – pl black box
- 4.3: Stuktúra alapján – pl white boks
Ismert a működés/forráskód/db
- 4.4: Felmerülő hibák alapján – sejtés/tipp alapú,
„majd szólnak, ha gond van”,
- 4.5: Valószínűségi

5. Tesztelés és életciklus

- **Vízesés modell**
- **V-modell**
- **Iteratív modellek**
- **Agilis módszertanok**
 - TestDrivenDevelopment,
 - DesignDrivenTesting,
 - BehaviourDrivenDevelopment

