

# Szoftvertesztelés

## Jellemzők

- A szoftver tesztelés során ellenőrizzük, hogy a program úgy működik-e ahogy elvárjuk.
- Tesztelés nélkül nem tudjuk, hogy azt csinálja-e, amit csinálnia kéne
- Visszajelzés a minőségről, lehetőség a javításra.
- A tesztelést érdemes a fejlesztés elején bevezetni:
  - Ez meghatározza azt is, hogy hogyan fogjuk írni a kódot.
  - Minél korábban találjuk meg a hibát, annál olcsóbb és egyszerűbb kijavítani.

## Szerepkörök

### Megrendelő

- A kiadás előtt az alfa/béta tesztelésben vesz részt.
- Az átadási és átvételi teszteléseket végzi.

### Üzleti elemző / tervező

- Funkcionális specifikációt ír, ami alapján a teszt terv készül.
- Ellenőrzi a teszt tervet.

### Fejlesztő

- Megírja a kódot, amit tesztelni lehet.
- Átnézi a teszt tervet.
- Unit tesztek készítését.

### Tesztelő (mérnök)

- Teszt tervet készíti.
- Teszt eseteket ír le és végrehajtja őket.
- Jóváhagyja a kiadni kívánt terméket.

### Végfelhasználó (end user)

- A felhasználó, akit a ki nem javított hibák érintenek.
- Részt vehet béta tesztelésben vagy átvételi tesztben.

## Folyamata

**Manuális tesztelés:** Egy ember végig kattint az alkalmazáson. Lehet ad-hoc, forgatókönyv mentén stb.

**Automatikus tesztelés:** A gép végzi, nagy rendszereknél szükség van automatikus tesztelésre.

## Ellenőrzés és validáció

- **Validáció (validation):** „A megfelelő szoftvert készítjük?”
- **Ellenőrzés (verification):** „Jó irányban haladunk a szoftverrel?”
- A végfelhasználói teszt a validációról szól, a többi teszt az ellenőrzésről.

## Tesztelési stratégia

- Megadja, hogy melyik tesztet milyen módszerrel végezzük:
  - Manuális / automata
  - **White / Grey / Black Box**
  - Unit teszt, integrációs teszt, stb.
- Melyik teszt keretrendszert, milyen eszközöket használunk majd (NUnit, Visual Studio, stb.)
- Milyen előfeltételei vannak a tesztek végrehajtásának: infrastruktúra, architektúra, input adatok, stb.

## Tesztelési módszerek

### White Box / fehérdobozos teszt

A tesztelő ismeri a szoftver belső felépítését és algoritmusait és felhasználja az információkat.

- API tesztelés.
- Kódlefedettség vizsgálat.
- Hibakezelés vizsgálat.

### Black Box / feketedobozos teszt

Nem használja a belső működésről ismerteket.

- Bemenetekre adott válaszok vizsgálata.
- Szélső érték elemzés.
- Alpha/beta tesztelés.
- „Dogfooding”: saját termékek és szolgáltatások használata.

### Grey Box / szürkedobozos teszt

A tesztelő ismeri a szoftver belsejét de felhasználói/fekete doboz szinten tesztel

- Integrációs teszt: interfész szinten.
- Reverse engineering.

## A tesztelés szintjei

### Unit test / egységteszt

**Típus:** White / Grey Box

- A kód egy nagyon kis részét ellenőrzi, jellemzően White Box.
- Ez egy komponensteszt, vagyis a rendszer önálló részeit vizsgálja.
- A legkisebb szoftveres egységre fókuszál.
- A metódusokat kezeli: kimenet-bemenetre fókuszál (jó-e a visszatérési érték).
- Automatizált, gyors és ismételhető.
- **Modul teszt:** Általában a nem funkcionális részeket teszteli.
  - Memóriaszivárgás
  - Sebesség
  - Szinkronizáció
  - Szűk keresztmetszet (bottleneck)

### Integration test / integrációs teszt

**Típus:** White / Grey Box

- A komponensek közti interfészt ellenőrzi.
- Tipikusan iteratív folyamat amíg a teljes rendszer nincs ellenőrizve.
- A komponensek együttműködését teszteli:
  - A komponensek közötti kapcsolatot.
  - Az összeillesztéskor keletkező hibát.
- Unit tesztek után szokott lenni.
- Interfészek definiálásánál kezdődik el a tesztelés.
- Hibakeresése nehezkesebb.

### System test / rendszerteszt

**Típus:** Black Box

- A rendszereket önmagukban ellenőrzi a követelmények tükrében.
- Az integrációs teszt után szokott következni.
- A rendszerteszt a már kész szoftverterméket teszteli.
  - Megfelel-e a követelményeknek (elsősorban funkcionális teszt).
  - Megfelel-e a rendszertervnek.
- Van, hogy független cég végzi.
- Gyakran manuális.

### System integration test / rendszer integrációs teszt

**Típus:** Black Box

- Külső rendszerekkel való együttműködést ellenőrzi.
- Garantálja, hogy mindegyik rendszer átment a rendszerteszten.

## Regression test / regressziós teszt

**Típus:** White / Grey Box

- Olyan teszt, ami a korábbi hibák újbóli felbukkanását jelzi.
- Tipikusan a hibajavítás során készülnek.
- Korábban megjelent hibát vagy forgatókönyvet tesztel újra.
- Tipikusan valamilyen változtatás után végzik: jó-e még a rendszer?
  - Amikor fejlesztésnél írunk egy új funkciót, akkor azt szeretnénk, hogy a már meglévő funkciók ne sérüljenek.
  - Ezt segítik az automata unit tesztek.

## User acceptance test / felhasználói átvételi-elfogadási teszt

**Típus:** Black Box

- *Átvételi folyamathoz tartozó teszt.*
- A végfelhasználó végzi saját környezetében, az integrációs és rendszertesztek után.
- Felhasználói köre projektfüggő.
- Tipikusan ennek eredménye alapján kerül kiállításra a teljesítés igazolás.

## Alpha, beta test / Alfa, béta teszt

**Típus:** Black Box

- *Átvételi folyamathoz tartozó teszt.*
- Nem-kész verzió ellenőrzése belső vagy meghívott külső felhasználók által.
- **Alfa teszt:** meghívott végfelhasználók végzik, nem a fejlesztőcsapat.
- **Béta teszt:** végfelhasználók szűk csoportja végzi.

## Safety and load tests / Biztonsági és terheléses tesztek

**Típus:** Black Box

- A biztonsági tesztet szakosodott cég végzi.
- A terheléses teszt során a várható terhelésre adott választ vizsgáljuk.