

# C++ programok egységtesztelése googletest segítségével (GKxB\_INTM006)

Dr. Hatwágner F. Miklós

Széchenyi István Egyetem, Győr

2019. július 8.

Tesztelés célja: a hibákat megtalálni üzembe helyezés előtt

Tesztelés alapelvei

- ❶ A tesztelés bizonyos hibák jelenlétét jelezheti (ha nem jelzi, az nem jelent automatikusan hibamentességet)
- ❷ Nem lehetséges kimerítő teszt (a hangsúly a magas kockázatú részeken van)
- ❸ Korai teszt (minél hamarabb találjuk meg a hibát, annál olcsóbb javítani)
- ❹ Hibák csoportosulása (azokra a modulokra/bemenetekre kell tesztelni, amelyekre a legvalószínűbben hibás a szoftver)
- ❺ Féregirtó paradoxon (a tesztesetek halmazát időnként bővíteni kell, mert ugyanazokkal a tesztekkel nem fedhetünk fel több hibát)
- ❻ Körülmények (tesztelés alapossága függ a felhasználás helyétől, a rendelkezésre álló időtől, stb.)
- ❼ A hibátlan rendszer téveszméje (A megrendelő elsősorban az igényeinek megfelelő szoftvert szeretne, és csak másodsorban hibamenteset; verifikáció vs. validáció)

## Tesztelési technikák

### Fekete dobozos (black-box, specifikáció alapú)

A tesztelő nem látja a forrást, de a specifikációt igen, és hozzáfér a futtatható szoftverhez. Összehasonlítjuk a bemenetekre adott kimeneteket az elvárt kimenetekkel.

### Fehér dobozos (white-box, strukturális teszt)

Kész struktúrákat tesztelünk, pl.:

- kódsorok,
- elágazások,
- metódusok,
- osztályok,
- funkciók,
- modulok.

Lefedettségi: a struktúra hány %-át tudjuk tesztelni a tesztesetekkel?

Egységteszt (unit test): a metódusok struktúra tesztje.

A tesztelés szintjei:

- ❶ komponensteszt (egy komponens tesztelése)
  - ❶ egységteszt
  - ❷ modulteszt
- ❷ integrációs teszt (kettő vagy több komponens együttműködése)
- ❸ rendszerteszt (minden komponens együtt)
- ❹ átvételi teszt (kész rendszer)

Kik végzik a tesztelést?

1-3 Fejlesztő cég

4 Felhasználók

Komponensteszt

- fehér dobozos teszt
- egységteszt
  - bemenet → kimenet vizsgálata
  - nem lehet mellékhatása
  - regressziós teszt: módosítással elronthattunk valamit, ami eddig jó volt → megismételt egységtesztek
- modulteszt
  - nem funkcionális tulajdonságok: sebesség, memóriaszivárgás (memory leak), szűk keresztmetszetek (bottleneck)

## Integrációs teszt

- Komponensek közötti interfészek ellenőrzése, pl.
  - komponens - komponens (egy rendszer komponenseinek együttműködése)
  - rendszer - rendszer (pl. OS és a fejlesztett rendszer között)
- Jellemző hibaokok: komponenseket eltérő csapatok fejlesztik, elégtelen kommunikáció
- Kockázatok csökkentése: mielőbbi integrációs tesztekkel

Rendszerteszt: a termék megfelel-e a

- követelmény specifikációnak,
- funkcionális specifikációnak,
- rendszertervnek.

Gyakran fekete dobozos, külső cég végzi (elfogulatlanság)

Leendő futtatási környezet imitációja

## Átvételi teszt, fajtái:

- alfa: kész termék tesztelése a fejlesztőnél, de nem általa (pl. segédprogramok)
- béta: szűk végfelhasználói csoport
- felhasználói átvételi teszt: minden felhasználó használja, de nem éles termelésben. Jellemző a környezetfüggő hibák megjelenése (pl. sebesség)
- üzemeltetői átvételi teszt: rendszergazdák végzik, biztonsági mentés, helyreállítás, stb. helyesen működnek-e



Tesztelésről általában:

Ficsor Lajos, Kovács László, Kusper Gábor, Krizsán Zoltán: Szoftvertesztelés  
ISTQB CTFL Syllabus 2018