

NGB_IN003_1 SZE 2014-15/2 (11)

Szoftver- minőségbiztosítás

Tesztautomatizálás

A tesztelés kivitelezése

Tesztelési feladatok

- ❖ Detektálatlan maradék hibák számának csökkentése
 - ❖ hatásosan és hatékonyan
 - ❖ megfelelő gazdálkodás a felhasznált erőforrásokkal
- ❖ Elérés eszközei:
 - ❖ **hatékony tesztesetek**
 - ❖ **tesztelés automatizálása**

A szoftverhibák hatása

- ❖ A felhasználókra nézve
 - ❖ emberi életek veszélyeztetése
 - ❖ alapvető szervezeti / üzleti funkció meghiúsítása (nem helyettesíthető)
 - ❖ a teljes rendszer hibás működését okozza
 - ❖ üzleti alkalmazások hibás működése
 - ❖ személyi alkalmazások hibás működése
 - ❖ felhasználók számára kényelmetlenséget okoz

A szoftverhibák hatása (folyt.)

- ❖ A fejlesztő szervezetre nézve:
 - ❖ Anyagi veszteség
 - ❖ kártérítés, kötbér, közvetlen bevétel kiesés, karbantartási költségek
 - ❖ Nem vagyoni veszteség
 - ❖ **presztizs-veszteség**, piacvesztés (elmaradó megrendelések)

Szoftver-kockázatok

- ❖ A kockázat szintjét meghatározó tényezők
 - ❖ A rendszer / alkalmazás
 - ❖ mérete
 - ❖ komplexitása
 - ❖ új és újra felhasznált kód aránya
 - ❖ A fejlesztői
 - ❖ szakmai kvalitás
 - ❖ tapasztalat
 - ❖ domén támogatás rendelkezésre állása

A teszttervek változtatása

- ❖ A tesztterveket a kár mértéke és a kockázatok együttes értékelése alapján kell meghatározni
- ❖ A teszttervek változhatnak:
 - ❖ Erőforrások nem állnak rendelkezésre
 - ❖ A tesztelés időigénye túllépi a határidőket
 - ❖ A kockázatok vagy a tesztelési erőforrásigény megítélése megváltozik

Tesztek végrehajtása

- ❖ Fejlesztő csapat (egység- és integrációs tesztek)
 - ❖ saját kód
 - ❖ más fejlesztő csapat kódja
- ❖ Független (belső vagy külső) tesztelő csapat (rendszer tesztek)
- ❖ Több tesztelő csapat

A tesztelés befejezése

- ❖ Teszttervek végrehajtásának lezárulása
- ❖ Beültetett hibák meghatározott arányú felderítése után
- ❖ Duális, független teszt csapatok eredménye alapján
- ❖ Erőforrások elhasználásával

A tesztesetek forrásai

- ❖ Valódi esetekből véletlenszerű mintájából
- ❖ Szintetizált teszt esetek
 - ❖ specifikáció feltételeiből, paramétereiből

	Véletlen minta	Szintetikus tesztesetek
Teszteset meghatározás erőforrás igénye	alacsony, (kimeneti eredmények rendelkezésre állnak)	magas, (kimenetek meghatározása a paramétereiből)
Tesztesetek száma	relatív magas	relatív alacsony
Tesztvégrehajtás erőforrás igénye	magas	alacsony
Hiba-feltérési hatékonyság	relatív alacsony	relatív magas

Rétegzett mintavételezés

- ❖ Jelentős hatékonyságnövelés a mintavételes teszteset-előállításnál
- ❖ Irányított mintavételezés
 - ❖ teszt populáció lebontása alcsoportokra
 - ❖ nagyobb mintaszám a potenciálisan nagyobb hibafeltárási képességű alcsoportokból

A jó teszteset

- ❖ Jó hibadetektáló képesség
- ❖ Kis redundancia más tesztesetekkel
- ❖ Az érdekes területre koncentrálnak
- ❖ Más területek minimális használata
- ❖ Sem nem túl egyszerű és sem nem túl bonyolult
- ❖ Lehetővé teszi a hiba izolálását és azonosítását

Tesztautomatizálás

Automatizálási motiváció

- ❖ Feltételezett
 - ❖ költségcsökkentés
 - ❖ rövidebb teszt-végrehajtási idő
 - ❖ alaposabb tesztelés
 - ❖ megnövelt pontosság
 - ❖ jobb riportolás és statisztikai értékelés

Automatizálható tesztek

- ❖ Kód auditálása (statikus analízis)
- ❖ Lefedettségmentő monitorozás
- ❖ Funkcionális tesztek
- ❖ Terheléses tesztek
- ❖ Teszt menedzsment

Automatizált funkcionális tesztek

- ❖ Teszt szkriptek létrehozása (manuális írás vagy rögzítés)
- ❖ Regressziós tesztelés
 - ❖ létező tesztkészlet végrehajtása minimális ráfordítással
 - ❖ output komparátor (gold fájlok előző tesztfutásokból)

Regressziós tesztelési problémák

- ❖ A legtöbb hibát a tesztek tervezésekor lehet megtalálni
- ❖ A regressziós tesztek "mindig" ugyanazt tesztelik
- ❖ A programok nem öregednek és romlanak el
- ❖ Más tesztek is megtalálhatják a "mellékhatásokat"
- ❖ Az automatizálás csökkenti a tesztelési változatosságot
- ❖ Csak a tesztbe programozott dolgokat verifikálja

Terheléses tesztek

- ❖ Maximális terhelés környezet kialakítása
 - ❖ valódi terhelési viszonyok
 - ❖ **események / tranzakciók és frekvenciáik**
 - ❖ virtuális felhasználók / események
 - ❖ valódi események / tranzakciók rögzítése
- ❖ végrehajtás nagy mennyiségben, tervezett eloszlással

Tesztmenedzsment

- ❖ Manuális tesztmenedzsment megközelítésnél magasabb szintű szolgáltatások
 - ❖ nagyobb formalitás
 - ❖ interoperabilitás a tesztkörnyezet többi elemével, integráció
- ❖ Teszttervezés, eredmény-dokumentálás
 - ❖ teszteset nyilvántartás, hibakövetés, monitorozás
- ❖ Tesztvégrehajtás
 - ❖ konfiguráció menedzsment, automatizált tesztvégrehajtás, eredmény-rögzítés
- ❖ Javítások követése

Automatizált tesztelés - előnyök/hátrányok

❖ Előnyök

- ❖ Pontos teljesítmény követés
- ❖ Pontos eredmény logok
- ❖ Tesztvégrehajtás kisebb emberi erőforrással
- ❖ Rövidebb tesztelési periódus idő
- ❖ Jobb regressziós tesztelés
- ❖ Manuális tesztelésnél szélesebb körű tesztelés

❖ Hátrányok

- ❖ Magasabb bevezetési költség
- ❖ Magasabb előkészítési költség
- ❖ Nem minden fajta tesztelés automatizálható

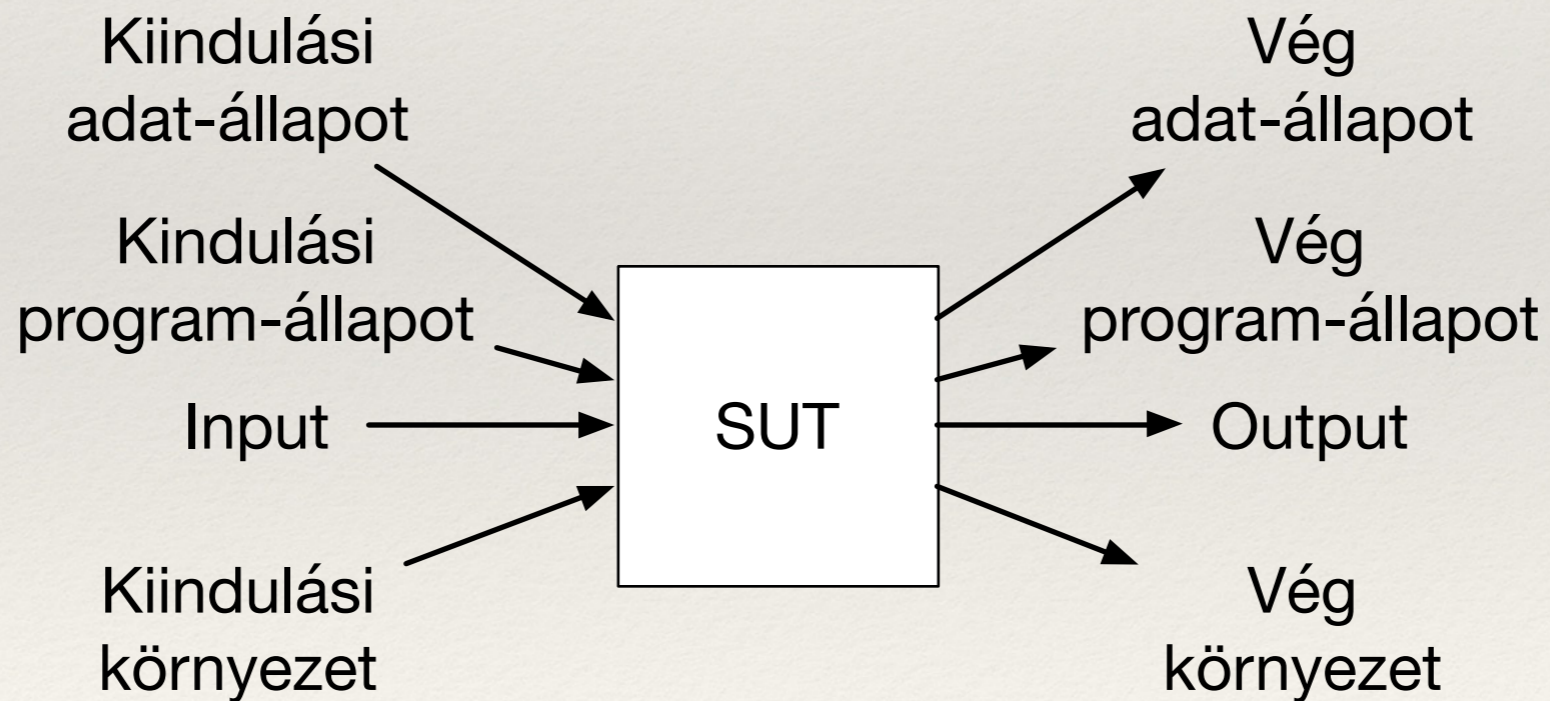
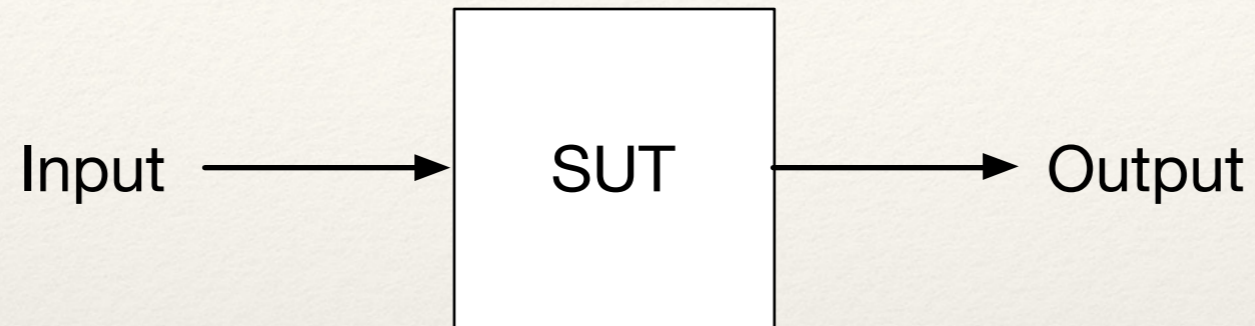
Az automatizált tesztelés jellemzői

- ❖ Manuális teszteléssel nem detektálható hibák megtalálása
- ❖ Ember számára nem látható információ monitorozása
- ❖ SUT stimulálása API-n keresztül
- ❖ Nagymennyiségű tevékenység végrehajtása
- ❖ Pszeudo random események => kontrollált változatosság
- ❖ Nagymennyiségű kimenet összehasonlítása

Téves elképzelések az automatizálásról

- ❖ Minden teszt automatizálható
- ❖ Az automatizálás rögtön kifizetődik
- ❖ Létező manuális tesztek könnyen automatizálhatók
- ❖ Nincs bejáratási idő
- ❖ Rögzítés / visszajátszás regressziós tesztelés hatékony
- ❖ Mindenre alkalmas az adott eszköz

Egyszerű és kiterjesztett tesztmodell



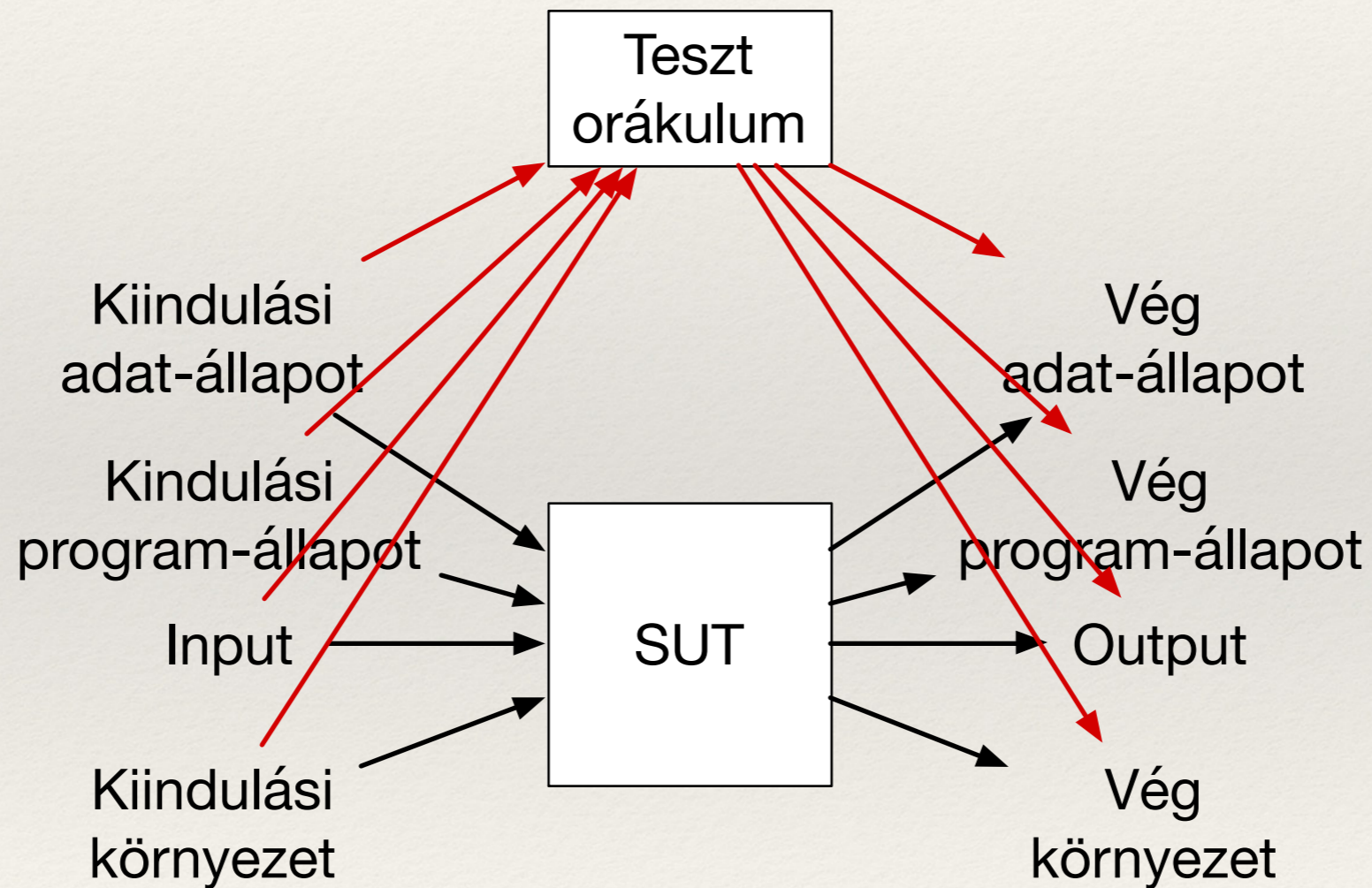
Problémás GUI tesztautomatizálás

- ❖ A GUI teszt eszközöket a termékhez és környezethez kell igazítani
- ❖ A GUI változásai tönkreteszik a teszteket
 - ❖ ki kell várni a GUI stabilizálódását
- ❖ A téves riasztások drágák
- ❖ humán elemzés szükséges
- ❖ A tesztkarbantartás általában nehézkes

Teszt orákulum

- ❖ Kétféle megközelítés
 - ❖ Mi a helyes kimenet adott bemenet esetén?
 - ❖ A szoftver sikeresen vette-e a tesztet?
- ❖ Determinisztikus orákulum
 - ❖ az eltérés programhibát jelez
- ❖ Valószínűségi orákulum
 - ❖ eltérés esetén valószínűleg hiba van

Teszt orákulum (folyt.)



Teszteredmények

	Nincs hiba	Hiba van
A teszteredmény OK	Korrekt	Nem detektált hiba
A teszteredmény NOT OK	Fals riasztás	Detektált hiba

Az orákulum teljessége

- ❖ Input lefedés
- ❖ Output lefedés
- ❖ Funkció lefedés
- ❖ Hibatípusok

- ❖ Kihívások
 - ❖ karbantarthatóság
 - ❖ költségek
 - ❖ időbeli viselkedés

Az orákulum pontossága

- ❖ Mennyire hasonló a SUT-hoz?
 - ❖ aritmetikai pontosság
 - ❖ statisztikai hasonlóság
- ❖ Mennyire független a SUT-tól?
 - ❖ algoritmusok
 - ❖ platform, alprogramok, könyvtárak
- ❖ Mennyire kiterjedt?

Automatizálási fejlesztés

- ❖ Az automatizálás szoftverfejlesztés
- ❖ A regressziós tesztelés automatizálása drága
- ❖ A karbantarthatóság létfontosságú
- ❖ A tesztelési követelményekhez igazodóan kell fejleszteni

A tesztautomatizálás - programozás

- ❖ De a legtöbb eszköznél:
 - ❖ beábrázolt konstansok vannak
 - ❖ nem modulárisak a szkriptek
 - ❖ nincs verzió kezelés
- ❖ A létrehozott szkriptek
 - ❖ nem dokumentáltak
 - ❖ nincs követelmény elemzés