



DEBRECENI EGYETEM

TESZTELÉS A SZOFTVER ÉLETCIKLUSÁN ÁT

TESZTTÍPUSOK
KARBANTARTÁSI TESZT

TÁMOP-4.1.1.F-13/1-2013-0004

MUNKAERŐ-PIACI IGÉNYEKNEK MEGFELELŐ,

GYAKORLATORIENTÁLT KÉPZÉSEK, SZOLGÁLTATÁSOK A DEBRECENI
EGYETEMEN ÉLELMISZERIPAR, GÉPÉSZET, INFORMATIKA,
TURISZTIKA ÉS VENDÉGLÁTÁS TERÜLETEN
(MUNKAALAPÚ TUDÁS A DEBRECENI EGYETEM OKTATÁSÁBAN)

SZÉCHENYI 2020



MAGYARORSZÁG
KORMÁNYA

Európai Unió
Európai Szociális
Alap



BEFEKTETÉS A JÖVŐBE

2.3 Tesztípusok

Az alábbi négy szoftverteszt-típust különböztetjük meg:

- **Funkcionális teszt**
- **Nem funkcionális teszt**
- **Strukturális teszt**
- **Változásokhoz kapcsolódó teszt**



2.3.1 Funkcionális teszt

A funkcionális teszt a rendszer funkcionalitását vizsgáló teszt.

- Egy rendszer vagy komponens által megvalósított valamely viselkedést, működést funkciónak nevezzük .
- A funkciók meghatározása a követelményspecifikációkban a használati esetekben, vagy a működési specifikációban történhet.



2.3.1 Funkcionális teszt megközelítései

- Üzleti folyamatokon alapuló megközelítés
 - A műszaki teszttervezés az üzleti folyamatokra vonatkozó ismereteken alapszik.
- Követelmény alapú megközelítés
 - A műszaki teszttervezés alapját a rendszer követelmény specifikációja képezi.



2.3.2 Nem funkcionális teszt

A rendszer tesztelése nem funkcionális követelmények szerint.

Az ISO 9126 meghatároz bizonyos szoftver minőségi jellemzőket, mely nagy lépést jelent a szoftver minőség általános fogalma felé.

A szabvány hat minőségi jellemzőt ad, mely további aljellemzőkre bontható:

- Funkcionalitás (Funkcionális teszteléssel ál kapcsolatban)
- Megbízhatóság
- Használhatóság
- Hatékonyság
- Karbantarthatóság
- Hordozhatóság



TÁMOP-4.1.1.F-13/1-2013-0004

MUNKAERŐ-PIACI IGÉNYEKNEK MEGFELELŐ, GYAKORLATORIENTÁLT KÉPZÉSEK,
SZOLGÁLTATÁSOK A DEBRECENI EGYETEMEN ÉLELMISZERIPAR, GÉPÉSZET,
INFORMATIKA, TURISZTIKA ÉS VENDÉGLÁTÁS TERÜLETEN
(MUNKAAALAPÚ TUDÁS A DEBRECENI EGYETEM OKTATÁSÁBAN)

2.3.2 Nem funkcionális jellemzők: Megbízhatóság

- megbízhatóság
 - érettség (robosztusság)
 - hibatűrés
 - visszaállíthatóság
 - megfelelés



2.3.2 Nem funkcionális jellemzők: Használhatóság

- használhatóság
 - érthetőség
 - tanulhatóság
 - működtethetőség
 - megjelenés
 - megfelelőség



2.3.2 Nem funkcionális jellemzők: Hatékonyság

- hatékonyság
 - időbeli viselkedés (teljesítmény)
 - erőforrás-kihasználtság
 - megfelelőség



2.3.2 Nem funkcionális jellemzők: Karbantarthatóság

- karbantarthatóság
 - elemezhetőség
 - változtathatóság
 - stabilitás
 - tesztelhetőség
 - megfelelőség



2.3.2 Nem funkcionális jellemzők: Hordozhatóság

- hordozhatóság
 - adaptálhatóság
 - telepíthetőség
 - koegzisztencia
 - helyettesíthetőség
 - megfeleléség



2.3.2 Nem funkcionális tesztek

- Teljesítmény teszt (Performance Testing)
 - Terheléses teszt (Load Testing)
 - Stresszteszt (Stress Testing)
 - Kitartási teszt (Soak / Endurance Testing)
- Csúcs teszt (Spike Testing)
- Mennyiségi teszt (Volume Testing)
- Használhatósági teszt (Usability Testing)
- Nemzetköziesség teszt (Internationalization - i18n & Localization - l10n Testing)
- Biztonsági teszt (Security Testing)
- Skálázhatósági teszt (Scalability Testing)
- Visszaállíthatósági teszt (Recovery Testing)
- Hordozhatósági teszt (Portability Testing)
- Kompatibilitási teszt (Compatibility Testing)
- Hatékonysági teszt (Efficiency Testing)
- Megfelelési teszt (Contract Acceptance Testing)
- Szabályossági teszt (Compliance Testing / Regulation Acceptance Testing)
- Linearitás vizsgálat
- Megbízhatósági teszt (Reliability Testing)
- Regressziós teszt (Regression Testing)
- Strukturális teszt (Structural Testing)
- Karbantartási teszt (Maintenance Testing)
- Hatáselemzés (Impact Analysis)



TÁMOP-4.1.1.F-13/1-2013-0004

MUNKAERŐ-PIACI IGÉNYEKNEK MEGFELELŐ, GYAKORLATORIENTÁLT KÉPZÉSEK,
SZOLGÁLTATÁSOK A DEBRECENI EGYETEMEN ÉLELMISZERIPAR, GÉPÉSZET,
INFORMATIKA, TURISZTIKA ÉS VENDÉGLÁTÁS TERÜLETEN
(MUNKAALAPÚ TUDÁS A DEBRECENI EGYETEM OKTATÁSÁBAN)

2.3.2 Nem funkcionális tesztek: Teljesítmény teszt

A teljesítmény teszt célja a rendszer válaszütemének, sebességének, viselkedésének vizsgálata bizonyos típusú terhelés mellett.

- Az egyik legelterjedtebb nem funkcionális teszt.
- Magas szinten történik, az integrált rendszeren hajtjuk végre.
- Folyamatos rendszer monitorozás, diagnosztikai naplófájlok készítése.



2.3.2 Nem funkcionális tesztek: Terheléses teszt

A terheléses tesztek célja, hogy bizonyos terhelés —ez lehet konkurens felhasználószám, tranzakciószám vagy kérés szám— mellett keressük a rendszer szűk keresztmetszeteit.

- Szűk keresztmetszet lehet (pl):
 - Adatbázis
 - CPU
 - Memória
 - Külső rendszerrel való kommunikáció



2.3.2 Nem funkcionális tesztek: Stresszteszt

A stresszteszt célja, hogy megmutassa: a rendszer stabilitása az elvárásoknak megfelel. Jóval a normál működés feletti terhelést adunk a rendszernek, és figyeljük az egyes elemeket.

- Keressük a leggyengébb láncszemet a rendszerben, azaz addig növeljük a terhelést, míg a rendszer összeomlik vagy használhatatlanná válik.
- Figyeljük a rendszer összeomlásának mikéntjét, így nyerve tapasztalatot egy esetleges későbbi összeomláshoz



2.3.2 Nem funkcionális tesztek: Kitartási teszt

A kitartási teszt is egyfajta terheléses teszt, melynek célja, hogy egy jól meghatározott terhelés alatt tartsa a rendszert hosszú időn keresztül.

- Célja belátni, hogy a rendszer megfelelően viselkedik akkor is, ha huzamosabb ideig folyamatosan használat alatt áll, azaz nincsenek üresjárat időszakok.
- Ellenőrizhető (pl):
 - a rendszer gyorsabban / lassabban válaszol-e bizonyos idő után
 - a gyorsítótárak működését
 - figyelhetjük a memória szivárgást
 - a garbage collector működése



2.3.2 Nem funkcionális tesztek: Csúcsteszt

A csúcsteszt esetében a rendszert folyamatos normál terhelést követően hirtelen nagyságrendekkel nagyobb terheléssel árasztjuk el, majd a terhelést ugyan ilyen hirtelen megszüntetjük.

- A folyamat közben folyamatosan vizsgáljuk a válaszütemeket:
 - a terhelés alatt
 - a terhelés megszüntetése után



2.3.2 Nem funkcionális tesztek: Mennyiségi teszt

A mennyiségi tesztekkel az a célunk, hogy a rendszer működését nagy adatbázison, nagy rekordszám mellett vizsgáljuk.

- Előfordulhat ugyanis, hogy hiányos vagy nem megfelelő indexelés esetén az alkalmazás akkor válik majd használhatatlanná, mikor annak adatbázisa elért egy bizonyos méretet
- Adatgeneráló technikák alkalmazására
 - Véletlenszerű
 - Szabályszerű



2.3.2 Nem funkcionális tesztek: Használhatósági teszt

A használhatósági teszt célja, hogy megmutassa a rendszerről azt, hogy mennyire felhasználó barát, mennyire könnyen használható.

- A fekete doboz technikák közé soroljuk.
- Alapvetően három kérdésre keressük a választ egy adott rendszer kapcsán:
 - Mennyire egyszerű használni?
 - Mennyire egyszerű tanulni?
 - Mennyire kényelmes a felhasználó számára?
- A felhasználói felületre és a gép-ember interakciókra helyezik a hangsúlyt



2.3.2 Nem funkcionális tesztek: Használhatósági teszt

A használhatósági tesztek az alábbi szempontokat veszik figyelembe:

- **Tanulhatóság**
 - Mennyire egyszerű egy kezdő felhasználó számára feladatokat elvégezni úgy, hogy először látja a rendszert?
- **Hatékonyság**
 - Milyen gyorsan tud egy tapasztalt felhasználó feladatokat elvégezni?
- **Emlékezetőség**
 - Képes-e egy felhasználó azonnal folytatni a hatékony munkát, miután hosszabb idő elteltével újra találkozik a rendszerrel, vagy újra kell tanulnia a kezdetektől?
- **Hibás használat**
 - Hány hibát ejt egy felhasználó, milyen súlyosságúak ezek a hibák, és milyen gyorsan képesek orvosolni a hibát?
- **Elégedettség**
 - Mennyire szereti a felhasználó használni a rendszert?



TÁMOP-4.1.1.F-13/1-2013-0004

MUNKAERŐ-PIACI IGÉNYEKNEK MEGFELELŐ, GYAKORLATORIENTÁLT KÉPZÉSEK,
SZOLGÁLTATÁSOK A DEBRECENI EGYETEMEN ÉLELMISZERIPAR, GÉPÉSZET,
INFORMATIKA, TURISZTIKA ÉS VENDÉGLÁTÁS TERÜLETEN
(MUNKAAALAPÚ TUDÁS A DEBRECENI EGYETEM OKTATÁSÁBAN)

2.3.2 Nem funkcionális tesztek: Nemzetköziesség teszt

Az I18n Testing elsősorban azt vizsgálja, hogy a rendszer képes-e több nyelven kommunikálni a felhasználóval, azaz le van-e fordítva több nyelvre. Az L10n Testing pedig a rendszernek azt a képességét ellenőrzi, hogy különböző országok szokásainak, szabályainak megfelelően működik-e.

- A két fogalmat együtt olykor *globalization* névvel illetik.



2.3.2 Nem funkcionális tesztek: Nemzetköziesség teszt

Internationalization:

- Képernyő feliratainak fordítása megtörtént.
- Képernyőn megjelenő karakterek kódolása megfelelő (Unicode).
- A rendszer output dokumentumai megfelelő nyelven jelennek meg.
- A képernyőn megjelenő feliratok különböző nyelven, különböző hosszúsággal nem rontják el a képernyő elrendezését.
- Szám, dátum formátumok az adott nyelvnek megfelelőek.
- Balról jobbra és jobbról balra történő írás támogatása.
- Kis-, nagybetűk megfelelő kezelése.
- Film és videó anyagok fordítása hangsáv vagy felirat segítségével.



2.3.2 Nem funkcionális tesztek: Nemzetköziesség teszt

Localization:

- Különböző pénznemek támogatása.
- Telefonszám formátumok támogatása.
- Cím formátumok támogatása.
- Mértékegységek támogatása.



2.3.2 Nem funkcionális tesztek: Biztonsági teszt

A biztonsági tesztek azt vizsgálják, hogy a rendszer mennyire sebezhető rosszindulatú támadás vagy használat esetén.

- E témakörbe tartozik adatok védelme és a hálózati környezet biztonságossága, illetve a rendszer adatbázisaiban tárolt érzékeny adatok titkosítása, mint például a jelszavak.



2.3.2 Nem funkcionális tesztek: Biztonsági teszt

Az alábbi fogalmakat kell lefednünk biztonsági tesztekkel:

- Titoktartás (Confidentiality)
 - A rendszerből kikerülő információk védelmében olyan megoldásokat kell alkalmazni, hogy a jogosultakon és/vagy címzetteken kívül senki más ne férjen hozzájuk, ne használhassa jogosulatlanul olyan célra, mely ártalmas lehet.
- Sértetlenség (Integrity)
 - A fogadó fél számára lehetővé teszi, hogy ellenőrizhesse a megkapott információ helyességét. A fogadó fél lehet a felhasználó és a rendszer is. A módszerek titkosítás helyett algoritmusokat használnak (checksum, hash, Luhn), mivel az információ már a bevitelkor is sérülhetett - pl.: hibás gépelés.
- Hitelesítés (Authentication)
 - Egy személy vagy másik rendszer azonosságát, egy objektum eredetét, az elnevezés és a tartalom megfeleltetését kell biztosítani a rendszernek.



2.3.2 Nem funkcionális tesztek: Biztonsági teszt

- **Meghatalmazás (Authorization)**
 - A rendszernek képesnek kell lennie eldönteni azt, hogy a rendszert használónak engedélyezett-e az adott művelet, használhatja-e az adott szolgáltatást.
- **Elérhetőség (Availability)**
 - A rendszerhez jogosan hozzáférők számára elérhető a kért információ, amikor arra szükségük van.
- **Letagadhatatlanság (Non-repudiation)**



2.3.2 Nem funkcionális tesztek: Biztonsági teszt

Ahhoz, hogy egy rendszerről elmondhassuk azt, hogy biztonsági teszten esett át a következő, részben egymásra épülő lépéseket kell végrehajtani:

- **Felderítés (Discovery)**
 - A rendszer egészét vizsgáljuk, és keresünk az egyes komponensek esetleges elévült verzióit, melyek sérülékenységet okozhatnak.
- **Sebezhetőségi vizsgálat (Vulnerability Scan)**
 - Automatikus teszteszközökkel keresünk ismert, gyakori biztonsági hibákat.
- **Sebezhetőségi értékelés (Vulnerability Assessment)**
 - Az előző két lépés értékelése történik az adott környezet és rendszer esetén. Keressük és eltávolítjuk az automatikus értékelés eredményeként kapott álpozitív részeket a jelentésekből, és csak a releváns sebezhetőségi tételek maradnak.



2.3.2 Nem funkcionális tesztek: Biztonsági teszt

- Biztonsági értékelés (Security Assessment)
 - Manuális ellenőrzéssel győződünk meg a sebezhetőségi értékelés tételeiről a rendszer teljes terjedelmén. Az adott biztonsági hiba előidézése után, nem folytatunk, azt kihasználva, további gyengeségek keresését, a rendszerbe való mélyebb behatolást.
- Behatolási teszt (Penetration Test)
 - A behatolási teszttel egy rosszindulatú felhasználó ténykedését szimuláljuk. Lehet fehér doboz és fekete doboz megközelítéssel is tesztelni, attól függően, hogy rendelkezésre állnak-e a rendszer részletei, forráskódja, vagy nem. Az előző eredmények alapján a rendszer mélyebb részeihez próbálunk hozzáférni a talált sebezhető pontokon. Behatolási tesztek esetén sok múlik a tesztelő tudásán és képességein, korábbi tapasztalatain.



2.3.2 Nem funkcionális tesztek: Biztonsági teszt

- Biztonsági Audit (Security Audit)
 - Egy magas szintű, kockázatokat feltáró folyamat, mely a korábbi lépésekre épít, illetve ellenőrzi azok meglétét, folyamatait.
- Biztonsági Felülvizsgálat (Security Review)
 - Verifikációs folyamat, mely vizsgálja, hogy rendszerben alkalmazták-e az iparági és a cég belső biztonsági előírásait. Minden lehetséges eszközt felvonultat, mely a rendszer kódjának, dokumentációjának, diagramjainak, architektúrájának átvizsgálását lehetővé teszi.



2.3.2 Nem funkcionális tesztek: Skálázhatósági teszt

E tesztel azt mérjük, hogy a rendszer mennyire képes a növekvő igényeknek megfelelni. A növekedés igénye jelentkezhet terhelésben, tranzakció számban, konkurens felhasználók számában, átviteli sebesség mértékében vagy adatmennyiség növekedés formájában.

- Azt vizsgáljuk, hogy a rendszer beállításait tudjuk-e úgy módosítani, hogy az ilyen fajta változásoknak eleget tegyen.



2.3.2 Nem funkcionális tesztek: Visszaállíthatósági teszt

A visszaállíthatósági teszt azt vizsgálja, hogy a rendszer képes-e helyreállni, miután valamilyen külső hibából eredően lehetetlenné vált a működése.

- Például vizsgálhatjuk, hogy ha egy hálózat kapcsolat hirtelen megszűnik - akár adatátvitel közben -, mi történik, ha újra visszakapcsoljuk a hálózatot.
- Vizsgálhatjuk azt is, hogy mi történik, ha áramkimaradás következik be, vagy szabálytalanul állítjuk le a szervert, majd újraindítjuk.
- Az is érdekes lehet, hogy a kliensek hogy reagálnak egy esetleges szabályos szerver leállításra és újraindításra.



2.3.2 Nem funkcionális tesztek: Hordozhatósági teszt

A hordozhatósági teszteket akkor és olyan alkalmazások esetén végezzük, amikor elvárás az, hogy több különböző platformon működőképesek legyenek.

A hordozhatósági tesztek az alábbiakat foglalják magukba:

- Telepíthetőség
 - Azt vizsgálja, hogy a rendszert hogyan lehet telepíteni egy célplatformra. Kitér a tárhely ellenőrzésre, előfeltételek teljesülésére, a telepítés különböző módozataira, testre szabhatóságára, újra telepítésre és eltávolításra.



2.3.2 Nem funkcionális tesztek: Hordozhatósági teszt

- Kompatibilitás és együttélés
 - Egymástól független rendszerek esetén akarjuk belátni, hogy nem befolyásolják egymás működését. Különös figyelem fordítunk erre olyan esetekben, ha mindketten használják ugyanazt a komponenst.
- Adaptálhatóság
 - Az mutatja meg, hogy a rendszer minden platformon ugyanolyan működést produkál-e, és nincs szükség a normális interakciókon kívül egyéb műveletre platform specifikus módon.
- Helyettesítés
 - Vizsgáljuk, hogyha egy rendszeren belül kicserélünk egy komponenst egy vele kompatibilis specifikációval rendelkező komponensre, akkor a működés nem változik.



2.3.2 Nem funkcionális tesztek: Kompatibilitási teszt

Olyan esetben beszélünk kompatibilitási tesztről, amikor egy konkrét platformon vizsgáljuk az alkalmazás működését.

- Legtöbbször a hordozhatósági teszt szinonimájaként találkozunk.
- Az adott platformon certifikálhatjuk a rendszert, ha megfelelt a kompatibilitási teszteknek.
- Szükséges lehet a visszafelé kompatibilitás vizsgálata.



2.3.2 Nem funkcionális tesztek: Hatékonysági teszt

A hatékonysági tesztek célpontja egy-egy funkció. Azt vizsgálják, hogy milyen mennyiségű kód és teszt szükséges ahhoz, hogy az adott funkcionalitást biztosítani tudjuk.

Néhány aspektus, amiket vizsgálhatunk:

- Hány igényt elégítünk ki a rendszerrel?
- Milyen mértékben sikerült teljesíteni a specifikációban megfogalmazottakat?
- Milyen ráfordítást jelent a rendszer fejlesztése?

A tesztelés hatékonyságát is mérhetjük:

- Tesztek hatékonysága: (megtalált hibák az egység+integrációs+rendszer tesztekben) / (megtalált hibák az egység+integrációs+rendszer+megfelelési tesztekben)

Tesztelés hatékonysága: (megtalált hibák / lejelentett hibák)*100



TÁMOP-4.1.1.F-13/1-2013-0004

MUNKAERŐ-PIACI IGÉNYEKNEK MEGFELELŐ, GYAKORLATORIENTÁLT KÉPZÉSEK,
SZOLGÁLTATÁSOK A DEBRECENI EGYETEMEN ÉLELMISZERIPAR, GÉPÉSZET,
INFORMATIKA, TURISZTIKA ÉS VENDÉGLÁTÁS TERÜLETEN
(MUNKAALAPÚ TUDÁS A DEBRECENI EGYETEM OKTATÁSÁBAN)

2.3.2 Nem funkcionális tesztek: Megfelelési teszt

A rendszer kapcsán megkötött szerződésekben foglaltaknak való megfelelést vizsgáljuk.

- A rendszerteszt szinten is vizsgálni a szerződésnek való megfelelést.



2.3.2 Nem funkcionális tesztek: Szabályossági teszt

**Iparági szabályoknak, törvényi előírásoknak való
megfelelést ellenőrzünk.**

Például:

- Egy számlázó alkalmazásnak meg kell felelni bizonyos törvényi környezetnek.
- Az ilyen jellegű megfogalmazások magukban hordozzák a folyamatos jogkövetés igényét, mely azon túl, hogy az alkalmazás rendszeres felülvizsgálatát és esetleges módosítását jelenti, időről időre szakértő bevonását is követelheti meg.



2.3.2 Nem funkcionális tesztek: Linearitás vizsgálat

E teszt célja, hogy az implementált algoritmusokról belássa, hogy nem exponenciális időben futnak akkor, ha a paramétereik száma vagy azok értéke lineárisan növekszik és az elvárás az, hogy a futás idő is maximum lineárisan változzon.

- Nem hatékonyak az olyan megoldások, melyek futásideje exponenciálisan változik.
- Linearitás vizsgálatot azoknál a metódusoknál érdemes elvégezni, melyek futási ideje kiugróan nagy más metódusokhoz képest.
- Tekinthetjük rá úgy, mint alacsony szintű teljesítmény tesztre is, mely a helytelen kódolásból eredő hibákat keresi.



2.3.2 Megbízhatósági teszt

A megbízhatósági teszt célja, hogy a rendszerrel elvégezhető feladatokról – lehetőleg mindről – megmutassa, hogy hosszú időn át képesek determinisztikusan és konzisztensen működni adott környezetben.

A megbízhatósági tesztípus magába foglalja, használja az alábbi tesztípusokat:

- Funkcionális tesztek
- Terheléses tesztek
- Regressziós tesztek



2.3.3 Strukturális teszt

- Egy komponens struktúrájának vagy a rendszer felépítésének tesztje.
- Gyakran használatos módszer a teszt alaposságának mérésére.
- A strukturális tesztmegközelítéseket használhatjuk rendszer-, rendszerintegrációs vagy átvételi szinteken.
- Komponens és komponensintegrációs teszteknel vannak kódlefedettség mérését támogató eszközök.
- „Fehérdoboz teszt” vagy „üvegdoboz” teszt

Jellemzően fehérdoboz technikákat használunk.



2.3.4 Változásokhoz kapcsolódó teszt

- Ezeknek a teszteknek a célja, hogy a hibajavítást követően, a bekövetkezett változásokat és azok esetleges mellékhatásait tesztelje.
- Bármilyen változtatás magában hordozza a lehetőségét annak, hogy a szoftver funkcionális, nem funkcionális és strukturális jellemzői megváltozzanak.
- Két típusa van:
 - **Ellenőrző teszt**
 - **Regressziós teszt**
- Ezek tulajdonképpen magukba foglalják az összes többi tesztípust.



2.3.4 Változásokhoz kapcsolódó teszt: ellenőrző teszt

Az ellenőrző teszt végrehajtása során, egy korábban megtalált programhiba kijavítását követően, —a programhibától megtisztított verzión— újra végrehajtásra kerülnek azok a tesztek, melyek a segítségével megtaláltuk a programhibát.

- Ezzel bizonyosodhatunk meg arról, hogy a programhiba maradéktalanul ki lett javítva.
- Fontos, hogy ugyanazokkal a tesztekkel teszteljünk.
- A javítás okozhat újabb hibákat a szoftver más részein. → regressziós teszt



2.3.4 Változásokhoz kapcsolódó teszt: regressziós teszt

A regressziós teszt célja annak ellenőrzése, hogy az elvégzett módosítások (szoftver, környezet) nem okoznak-e —nem várt— mellékhatásokat. A rendszer továbbra is megfelel azoknak a követelményeknek, melyeknek a módosítások előtt is megfelelt.

- Valójában anti-regressziós teszt. Azt vizsgáljuk, hogy a rendszer nem fejlődött-e vissza.
- Az ellenőrző teszthez hasonlóan olyan tesztek kerülnek végrehajtásra, melyeket korábban már végre hajtottunk.
- Az ellenőrző tesztel ellentétben most azok a tesztek kerülnek végrehajtásra, melyek korábban sikeresek voltak.



2.4 Karbantartási teszt

Karbantartási tesztnek nevezzük a karbantartást követően a működő rendszeren végrehajtott tesztet.

- Karbantartásnak nevezzük az alkalmazás átadását követően végrehajtott módosításokat.
 - Hibajavítások
 - Új és megváltozott követelményekhez való alkalmazkodás
 - Teljesítmény (vagy egyéb nem funkcionális jellemző) feljavítása
- Különbözik a karbantarthatósági tesztől!



Irodalom

- Dorothy Graham, Erik van Veenendaal, Isabel Evans, Rex Black, **A szoftvertesztelés alapjai** - Az ISTQB alapszintű ismeretek egyetlen könyvben összegyűjtve, ALVICOM Kft., 2010
- ISTQB® Certified Tester Foundation Level - Hivatalos magyar nyelvű tanterv Alapszintű képesítés, Verzió: 3.02, 2012.02.28.





DEBRECENI EGYETEM

KÖSZÖNÖM A FIGYELMET!

TÁMOP-4.1.1.F-13/1-2013-0004

MUNKAERŐ-PIACI IGÉNYEKNEK MEGFELELŐ,

GYAKORLATORIENTÁLT KÉPZÉSEK, SZOLGÁLTATÁSOK A DEBRECENI
EGYETEMEN ÉLELMISZERIPAR, GÉPÉSZET, INFORMATIKA,
TURISZTIKA ÉS VENDÉGLÁTÁS TERÜLETEN
(MUNKAALAPÚ TUDÁS A DEBRECENI EGYETEM OKTATÁSÁBAN)

SZÉCHENYI 2020



MAGYARORSZÁG
KORMÁNYA

Európai Unió
Európai Szociális
Alap



BEFEKTETÉS A JÖVŐBE