

Fehér Krisztián

Szoftvertesztelési alapismeretek

BBS-INFO Kiadó, 2016.

Minden jog fenntartva! A könyv vagy annak oldalainak másolása, sokszorosítása csak a szerző írásbeli hozzájárulásával történhet.

A könyv nagyobb mennyiségben megrendelhető a kiadónál:
BBS-INFO Kiadó, 1630 Bp. Pf. 21. Tel.: 407-17-07

A könyv megírásakor a szerző és a kiadó a lehető legnagyobb gondossággal járt el. Ennek ellenére, mint minden könyvben, ebben is előfordulhatnak hibák. Az ezen hibákból eredő esetleges károkért sem a szerző, sem a kiadó semmiféle felelősséggel nem tartozik, de a kiadó szívesen fogadja, ha ezen hibákra felhívják figyelmét.

Papírkönyv: ISBN 978-615-5477-31-7
E-book: ISBN 978-615-5477-32-4

Kiadja a BBS-INFO Kft.
1630 Budapest, Pf. 21.

Felelős kiadó: a BBS-INFO Kft. ügyvezetője

Nyomdai munkák: Biró Family Nyomda

Felelős vezető: Biró Krisztián

TARTALOMJEGYZÉK

1. Gondolatok a szoftvertesztelésről.....	10
1.1. Mi a szoftvertesztelés?	10
1.2. Miért van egyáltalán szükség szoftvertesztelésre?	11
1.3. Kik a szoftvertesztelők és hogyan dolgoznak?.....	15
1.4. A tesztelés célja.....	17
1.5. A tesztelés emberi vonatkozásai	18
1.6. A tesztelés anyagi előnyei	18
2. A tesztelés alapfogalmai.....	21
2.1. A hiba fogalma	21
2.2. Teszt	22
2.3. Tesztkörnyezet.....	22
2.4. Üzleti követelmény (business requirement).....	23
2.5. Felhasználási folyamatleírások (user story).....	23
2.6. Teszt szcenárió (test scenario)	23
2.7. Teszteset (test case)	23
2.8. Tesztlépés (test step)	24
2.9. Review	25
2.10. Teszt státusz	25
2.11. Tesztlépések megfogalmazása.....	27
2.12. Negatív tesztesetek	29
2.13. A tesztelési folyamat főbb állomásai	29
2.13.1. A tesztek megtervezése	29
2.13.2. Teszt tervezés sprintekben.....	30
2.13.3. A tesztelezés	31
2.13.4. Tesztdizájn, előkészítés	32
2.13.5. Tesztvégrehajtás	32
2.13.6. Tesztkiértékelés	32
2.13.7. Tesztelést támogató tevékenységek	33
3. Tesztelési technikák.....	34
3.1. Statikus tesztelési technikák	34
3.2. Dinamikus tesztelési technikák	34

3.2.1.	Fekete doboz tesztelés	34
3.2.2.	Fehér doboz tesztelés	35
3.2.3.	Szürke-doboz tesztelés	35
3.2.4.	Felfedezéssel tesztelés	35
3.3.	Tesztvégrehajtás fajtái	36
3.3.1.	Manuális tesztelés	36
3.3.2.	Automata tesztelés	36
4.	Tesztek osztályozása	38
4.1.	Fejlesztési modellek	38
4.1.1.	A V-modell	38
4.1.2.	Agilis fejlesztés	38
4.1.3.	A SCRUM	39
4.2.	Teszt szintek	40
4.2.1.	Komponensteszt	40
4.2.2.	Integrációs teszt	40
4.2.3.	Rendszerteszt	41
4.2.4.	Elfogadási teszt	41
4.3.	Teszt típusok	42
4.3.1.	Funkcionális teszt	42
4.3.2.	Nem-funkcionális teszt	43
4.3.3.	Regressziós teszt	44
5.	Tesztprojektek	45
5.1.	A tesztprojekt szerepkörei	45
5.1.1.	Tesztmenedzser	45
5.1.2.	Projektkoordinátor	46
5.1.3.	Üzleti elemző	46
5.1.4.	Tesztkoordinátor, tesztvezető	47
5.1.5.	Tesztelő	47
5.2.	A tesztprojekt fázisai	48
5.2.1.	Tervezés, becslés	48
5.2.2.	Projektindító (kick-off) meeting	49
5.2.3.	Teszt dizájn	49
5.2.4.	Tesztvégrehajtás	50
5.2.5.	Ellenőrzés, monitorozás	50
5.2.6.	Jelentéskészítés	50
6.	Tesztkiértékelés	51
6.1.	Hibák dokumentálása	51
6.1.1.	A hiba elnevezése	52
6.1.2.	Hibaleírás	52

6.1.3.	Fontosság (prioritás) és súlyosság	53
6.1.4.	Érvényesség	54
6.1.5.	Hozzárendelés meghatározása	54
6.1.6.	Egyéb tulajdonságok.....	55
6.2.	Teszteredmények kiértékelése.....	55
7.	Tesztelés a gyakorlatban	56
7.1.	A szoftvertesztelés napi gyakorlata	56
7.2.	Hogyan teszteljünk például porszívót?.....	57
7.3.	Gyakorlati példa: webes felhasználói azonosítás tesztelése	59
7.3.1.	A feladat	59
7.3.2.	A tervezés	60
7.3.3.	Jóváhagyás	62
7.3.4.	Kivitelezés	62
7.3.5.	A tesztek „utóélete”	66
7.3.6.	A kreatív tesztesetekről	66
7.4.	Szoftvertesztelés Magyarországon és a nagyvilágban	67
7.5.	Tesztelés különböző platformokon.....	69
7.5.1.	Tesztelés asztali számítógépeken.....	69
7.5.2.	Tesztelés nagygépes környezetben	70
7.5.3.	Tesztelés mobileszközökön.....	70
7.6.	Pár szó az XML formátumról	71
7.7.	Tesztelést támogató szoftverek a gyakorlatban	73
7.7.1.	Tesztelés a Quickteszt segítségével.....	74
7.7.2.	SOAP UI	87
7.7.3.	SOAP UI projekt létrehozása	95
7.7.4.	A projekt finomhangolása.....	97
7.7.5.	Teszthívás létrehozása a projektben	99
7.7.6.	Teszteset létrehozása és végrehajtása	100
7.7.7.	Teszt futtatása	102
7.7.8.	Hívások kézi módosítása	105
7.7.9.	Teszteredmények kiértékelése.....	108
7.7.10.	JIRA	112
7.7.11.	További szoftverek	119
7.8.	Dokumentációs minták.....	119
7.8.1.	Tesztesetek megfogalmazása	119
7.8.2.	Teszteredmények dokumentálása.....	121

8. Minőségbiztosítási receptek.....	122
8.1. A tesztelés hiánya.....	122
8.2. A teszt tervezésének elmaradása.....	123
8.3. Tesztelés és fejlesztés különválasztása	123
8.4. Tesztmenedzsment hiánya.....	124
8.5. Tesztcélok hiánya	124
8.6. Dedikált tesztkörnyezet hiánya.....	124
8.7. Tesztadatok hiánya	125
8.8. Dokumentáció hiánya.....	126
8.9. Fejlesztések dokumentálásának hiánya.....	126
8.10. Manuális és funkcionális tesztelés különválasztása	126
8.11. Tesztesetek nem megfelelő dokumentálása.....	127
8.12. Teszteredmények dokumentálásának hiánya	127
8.13. Teszteredmények kiértékelése hiányzik	128
8.14. Feladatok delegálása.....	128
9. Tesztelői kasszótár	129

Előszó

A szerzőről

A szoftvertesztelés a kenyérkereső foglalkozásom, melyet nagyon komolyan veszek. Kis-, közép- és nagyvállalati környezetben is számos projektben vettem részt mint szoftvertesztelő, idehaza és külföldön is. ISTQB oklevéllel rendelkezem, veterán szoftvertesztelőnek számítok.

Egyfajta hackernek tekintem magam, a szó eredeti értelmében: rajongok a számítástechnikáért és nagy örömmel használom a kreativitásomat új dolgok létrehozására.

Egyetemista koromban kifejlesztettem egy vizuális fejlesztőrendszert két programozási nyelvhez, később pedig megalkottam saját térinformatikai rendszeremet, a ZEUSZ-t, amellyel a NASA-ig is eljutottam. Ezen a mai napig folyamatosan dolgozok.

Szívesen készítek ingyenes és nyílt forráskódú alkalmazásokat hagyományos, asztali számítógépes környezetre és mobileszközökre is. Ennek a munkának az egyik gyümölcse az a QuickTest tesztdokumentációs rendszer is, melyet elsősorban szemléltetési célokra fejlesztettem ki és ebben a könyvben is bemutatok.

Az Adobe Flash és az AIR technológia, valamint az Android operációs rendszer lelkes híve vagyok, nagyon kedvelem a C programozási nyelvet is. Ezekről a témákról eddig három könyvem jelent meg a hazai boltokban. Munkáimról bővebben a

<http://feherkrisztian.magix.net/public>

weboldalon is lehet olvasni.

A könyv célja

A könyv célja egy átfogó, de nem áttekinthetetlen kép kialakítása a professzionális szoftvertesztelésről, olyan ismeretanyag átadásával, mely biztos alapot nyújthat a szoftvertesztelésben elhelyezkedő leendő és gyakorló szakembereknek a mindennapi munkájukhoz. Mindezt azonban egyszerűen szeretném elérni: gyakorlatias ember vagyok, a kreatív gondolkodást többre tartom, mint szakkifejezések bemagolását és szajkózását.

A szoftvertesztelőknél óriási felelősség nyugszik és folyamatosan elvárások kereszttüzeiben kell helytállniuk. Mégis, szoftvertesztelés nélkül a legtöbb szoftver el sem jutna a felhasználóig, vagy ha igen, akkor megjelenésük botrányokkal, valamint óriási anyagi és erkölcsi veszteségekkel járna együtt, a rengeteg fel nem tárt programhiba miatt. Szoftvertesztelésre és tesztelőkre ezért igenis szükség van.

Könyvemben ezt a területet szeretném megismertetni a kedves Olvasóval.

A könyv felépítése

A könyv első része a szoftvertesztelés általános bemutatásáról szól, megteremtve ezzel az alapot az alapvető szakmai fogalmak bemutatásáig.

Ezt követi a könyv döntő részét képező törzs ismeretanyag, mely részleteiben tárgyalja a szoftvertesztelést és annak helyét a fejlesztési folyamatokban.

Végezetül pedig jó adag gyakorlati ismerettel végezhetjük fel magunkat, melynek során valódi, a tesztelést támogató alkalmazást ismerhetünk meg.

A szoftverteszteléssel most ismerkedő szakembereknek és laikusoknak kimondottan hasznos lehet ez a könyv, de fejlesztők és cégvezetők számára is tartogat hasznos információkat, melyek segítségével bevezethetik, illetve hatékonyabbá tehetik a szoftvertesztelést munkájuk során.

Kellemes és hasznos időtöltést kívánok a könyv olvasásához!

Fehér Krisztián

1. Gondolatok a szoftvertesztelésről

A professzionális szoftvertesztelés hazánkban még fiatal szakterületnek számít.

Van, aki adottságai révén lesz jó szoftvertesztelő, van, aki kemény tanulás árán válik azzá.

De miért van szükség szoftvertesztelésre? Általában ez az első kérdés, amivel szembesül az ember, amikor ezzel a témával találkozik. Mi fán terem a szoftvertesztelés? Kik a szoftvertesztelők? Hogyan dolgoznak?

A könyv első fejezetében a fenti kérdésekre keressük a választ, megtéve az első lépést a szakterület megismerése felé.

1.1. Mi a szoftvertesztelés?

Ha egy mondatban kellene megválaszolnunk a fenti kérdést, ilyen választ adhatnánk rá: **a szoftvertesztelés a szoftverfejlesztési folyamatot tervezetten kísérő, azzal szorosan összefüggő, ám annak önálló része, melynek célja a tesztelt szoftver minőségének feltárása.**

Itt gyorsan álljunk is meg egy szóra. Mit értünk **minőség** alatt? A minőség a szoftverek esetében olyan tulajdonságok összegző leírása, ill. megnevezése, melyek a szoftvert jellemzik.

Fejlesztési projekteknél minőségbiztosítás alatt olyan tevékenységeket értünk, melyek a fejlesztett szoftver tervezett minőségét igyekeznek biztosítani.

A szoftvertesztelés a **minőségbiztosítási** tevékenységek körébe tartozik.

A szoftverek tesztelését megtervező, végrehajtó és az eredményeket kiértékelő szakembert **szoftvertesztelőnek**, **tesztmérnöknek** nevezik (ez angolul tester, ill. test engineer).

1.2. Miért van egyáltalán szükség szoftvertesztelésre?

A fenti kérdést takarékosági szempontokra hivatkozva sajnos fejlesztőcégek, sőt nagyvállalatok is komolyan fel szokták tenni.

A szoftverek szervezett és irányított, képzett szakemberekkel történő tesztelése ugyanis pénzbe kerül, mely a fejlesztési projekt költségének jelentős hányadát teheti ki, ezért a leggyakoribb ellenérv a teszteléssel szemben a tesztelés anyagi vonzata szokott lenni.

Nos, hogy valóban drága-e a szoftvertesztelés és valóban megéri-e, az viszonylagos.

Képzeljünk el egy nagyon is életszerű példát! Példánkban egy neves bank meglévő banki rendszerét egy teljesen újra kívánja lecserélni. A projekt tervezett ideje 6 hónap.

Mivel az új rendszer fejlesztését is házon belül, ugyanazok a fejlesztők végzik, akik a régi rendszert is készítették, a menedzsment úgy dönt, hogy nem alkalmaznak külön tesztelőcsapatot.

Az érvek a tesztelés **ellen**, a menedzsment logikája szerint:

- A jelenlegi fejlesztők készítették a korábbi rendszert, így ők ismerik azt a legjobban. Külön tesztelők bevonása felesleges időhúzással és kiadásokkal járna.
- A fejlesztők maguk is végeznek tesztek, így mi szükség van külön tesztelőkre?
- A tesztelők úgyis csak nagyon egyszerű dolgokat tudnak ellenőrizni, így emiatt is felesleges külön tesztelőcsapatot fenntartani.

Elkezdődik a fejlesztés.

Az első hónap végén kiderül, hogy a fejlesztők nem rendelkeztek elegendő információkkal az új szoftvert illetően és félreértelmeztek bizonyos üzleti követelményeket.

A menedzsment kapkod, igyekeznek elbagatellizálni a hibát. Hosszas és fáradságos megbeszélések árán sikerül tisztázni a felmerült kérdéseket.

Közben a fejlesztők is idegesek, mert emiatt valószínűleg nem fogják tudni tartani a határidőt és mindenki szeme előtt a túlórák réme lebeg.

A folyamatos csúszások közepette a projekt félidőhöz érkezik. Az üzleti elemzők a szoftver első bemutatásakor olyan hiányosságokat fednek fel, melyekre addig senki sem gondolt.

A menedzsment tombol, mert már egyértelmű, hogy 6 hónap kevés lesz befejezni az új szoftver fejlesztését.

Néhány vezetőt lecserélnek. Újabb egyeztető megbeszélések és áttekintések sora veszi kezdetét.

A fejlesztők elkötelezettségének és emberfeletti túlórázásának köszönhetően 3 hónapos csúszással ugyan, de elérkezik az (első) elfogadási teszt ideje.

A menedzsmen és a vezető fejlesztők elégedetten szemlélik az új szoftver általuk késznek hitt változatát. Azaz csak szemlélnék, mert pár perc után a rendszer összeomlik, a rendszer komponensei megszűnnek kommunikálni egymással.

Újabb megbeszélések veszik kezdetüket, melyek egy válságtanácskozásban csúcsosodnak ki. Megszületik a döntés: mivel már csillagászati összegeket öltek a fejlesztésbe, ezért ha törik, ha szakad, be kell fejezni azt, kerül, amibe kerül.

Újabb 3 hónap kemény munka, folyamatos túlórázás árán a fejlesztők befejezik a munkát.

Az eredetileg hat hónaposra tervezett projekt több mint egy év után véget ér, a szoftvert bevezetik és néhány katasztrofális leállástól eltekintve az éles rendszer – döcögve ugyan, de - megkezd működését.

Mi lett volna, ha tesztelőket is alkalmaznak?

Ehhez nézzük meg tételesen a projekt elején a tesztelés ellen felhozott érveket:

1. A jelenlegi fejlesztők készítették a korábbi rendszert, így ők ismerik azt a legjobban. Külön tesztelők bevonása felesleges időhúzással és kiadásokkal járna.

A fejlesztők a projekt elején hiányos ismeretekkel rendelkeztek az új rendszer működését illetően, de ez csak hónapokkal később derült ki. A fejlesztés kezdeti fázisai ugyanis nem érintették az ide vonatkozó kérdéseket, ezért azok kidolgozását is későbbre időzítették.

Ha tesztelőket alkalmaznak, akkor a tesztelés vezetője legkésőbb már az első hét végén félreveri a vészhangokat, pontosabb rendszerterveket, leírásokat követelve.

2. A fejlesztők maguk is végeznek tesztek, így mi szükség van külön tesztelőkre?

Kiderült, hogy a fejlesztők sok eshetőségre nem is gondoltak. *„Hiszen a mi gépeinken, így és így végrehajtván a folyamatot, minden működött...”*

Mivel a tesztelők az üzemeltetők és nem utolsó sorban a leendő felhasználók szemszögéből is alaposan meg szokták vizsgálni a szoftvert, idejekorán olyan hibákat is felfedtek volna, melyek később kevesebb, vagy semmilyen többletmódosítást nem igényeltek volna a szoftveren.

Természetesen a programozók által elvégzett tesztelés is fontos (ezek elsősorban az ún. egység (unit) tesztek, melyek nincsenek könyvünk fókuszában), sőt elengedhetetlen része a fejlesztésnek és tesztelésnek. Viszont arányait tekintve csak egy apró részét képezi a teszteknek. Pusztán fejlesztői tesztek általában nem elegendőek a kívánt minőség elérésére.

3. A tesztelők úgyis csak nagyon egyszerű dolgokat tudnak ellenőrizni, így emiatt is felesleges külön tesztelőcsapatot fenntartani.

Felkészült tesztelők bizony benéztek volna a „motorháztető” alá is.

A fájdalmas tanulság az, hogy a jól szervezett tesztelés nemcsak, hogy segítette volna a határidők és a költ-