

---

Alan Page; Ken Johnston; Bj Rollison:

**How We Test Software at Microsoft**

Microsoft Press, 2008



---

A könyv címe magáért beszél: a Microsoft tesztelési módszereit ismerteti meg az olvasóval. Az első része a Microsoft belső felépítését, a tesztelési szerepeket és karrier utakat mutatja be. Rávilágít, hogy hogyan és miért alakult ki, hogy kinek, mit és miért kell tesztelnie a fejlesztés során. Ezután a teszttervezési technikák áttekintése következik, majd a tesztelők által használt eszközöket foglalják össze a szerzők. Zárásként a kialakulóban lévő új tesztelési irányokat összegzi a leírás.

Egy adott cég munkamódszereit bemutató könyv igazi értékét az adja, hogy megosztják a hosszú évek alatt összegyűjtött tapasztalatokat. Nem biztos, hogy tudjuk majd alkalmazni az ő megoldásaikat, de biztosan sokat lehet tanulni a sikereikből és hibáikból. A Microsoft pedig a maga 9000 tesztmérnökével az iparág egyik óriása.

A szerzők a 2007-es állapotot mutatják be, így a tapasztalatok inkább az asztali és kiszolgáló-oldali szoftverekre vonatkoznak, a webes terület kevésbé hangsúlyos. Azzal együtt, hogy pár dolog már nem aktuális a leírásban (például „felhő” helyett még „Software+Services” használatos), a könyv igen hasznos olvasmány. Egyrészt a tesztelési alapokat és technikákat érthetően, jó példákkal szemléltetve mutatja be (ehhez a saját belső oktatási anyagaikat használták fel a szerzők, amelyeket már évek óta csiszoltak). Másrészt, nagyon sok apró részletet és adatot megosztanak a Microsoft belső működéséről: mi a tesztelők és fejlesztők aránya<sup>1</sup>, mekkora kód lefedettséget érnek el általában a specifikáció alapú teszteléssel, milyen eszközöket használnak. Ráadásul a szöveget rengeteg apró, közbeszúrt sztori színesíti, ami vagy egy-egy régi történet tanulságán keresztül szemléltet egy témát, vagy egyszerűen csak bemutatja a tesztelők életének vidámabb perceit. Például megtudhatjuk, hogy lehet akár hibajegyen keresztül intézni a lánykérését („Bug #68648”, és a könyv szerint igent mondott az illető)!

*Első rész: A Microsoftról (Part I: About Microsoft)*

Ez az egyik leghasznosabb rész, annak ellenére, hogy még kevés technológiai részlet van benne. Egy rövid bevezetőben átveszi, hogy milyen részlegek vannak az MS-ben (ezt ne is próbáljuk megérteni, azóta úgymint sokszor átszervezték már), majd rátér arra, hogy a cég kb. 35000 mérnöke (product engineer) milyen területhez tartozhat. Ezek a területek a következők (az angol megnevezés talán pontosabb, ezért ezeket soroljuk fel): Test, Development, Program Management, Operations, Usability and Design, Content, Creative, Research, Localization, Engineering Management. Ezekhez mind különböző karrier utak, oktatások és pozíciók tartoznak, amelyek az évek során folyamatosan alakultak ki. A tesztelők esetén már rögtön a pozíció neve jelzi a cég filozófiáját:

*Software Development Engineer in Test (SDET)*

---

<sup>1</sup> Körülbelül 1:1, ami igen figyelemreméltó!

Megérthetjük ebből az elnevezésből, hogy a tesztelő pozíciókba is szoftvermérnököket vesznek fel, itt is feltétel az alapos programozási és mérnöki ismeret, hasonlóan a fejlesztői területre jelentkezőkhöz. A könyvből megtudjuk, hogy egy SDET munkájához hozzátartozik a teszteszközök fejlesztése, a tesztkörnyezetek és automatikus tesztek készítése, a tesztelési tervek írása és a manuális tesztelés is.

Mióta a Microsoft egyes szoftvereihez a megjelenés utáni tíz évig garantálja a terméktámogatást, nem kérdés, hogy megéri automatizálni a tesztek. Tesztelőnek olyan jelentkezőket keresnek, akiben az informatikai alapképzettségen túl megvan az úgynevezett „Tester DNA” is (többek között: „love of finding out how something works and then how to break it”).

A tesztelői és a fejlesztői karrier egymással párhuzamos előrelépéseket kínál a Microsoftnál. Mindkettőből van Senior, Principal és Partner fokozat, tehát nem az a helyzet, hogy a tesztelőt „előléptetik” fejlesztővé.

Hasonló módon, a menedzser pozíciók sem előrelépések, hanem párhuzamos utak. Pár év után a munkatársaknak megnyílik a lehetőség a menedzseri irányba továbblépni, de lehet akár önálló pozícióban is továbbfejlődni. A *Technical Fellow* pozíció, a legmagasabb egyéni szint például egyenrangú egy *Senior Vice President* pozícióval, azaz egy olyan menedzserével, aki „alá” akár ezer ember tartozik. Egy speciális szerep továbbá a *Test Architect*: olyan kiemelt egyéni tesztelő, aki egy termék egészét érintő speciális problémával foglalkozik. A könyv részletes példákkal ismerteti, hogy melyek az egyes lehetséges pozíciókban dolgozók feladatai és lehetőségei.

Az első rész utolsó fejezete azt mutatja be, hogy hogyan illeszkednek be a tesztelési feladatok a cégnél használt különböző fejlesztési módszerekbe, amelyek tipikusan iteratív vagy agilis megközelítések. Részletezésre kerül, hogy a fejlesztés során az egyes mérföldkövekhez milyen előre definiált tesztelési célokat szoktak megadni (pl. az M2 esetén már 65%-os kódfedést kell elérni és éjszakai stressz tesztek is kell futtatni). Jó ötlet, hogy néha beiktatnak úgynevezett minőségi mérföldköveket (quality milestone), amelyekhez kötődően új funkciót nem kell implementálni, csak a korábban „elhalasztott”, ki nem javított hibákat javítani, vagy az eszközöket és infrastruktúrát kell továbbfejleszteni.

#### *Második rész: A tesztelésről (Part II About Testing)*

A második rész a tesztelési és teszttervezési alapokkal foglalkozik. Teljesen kezdő szintről indulva mutatja be a következő témákat:

- teszttervezési alapok és technikák,
- specifikáció alapú (funkcionális) tesztelés,
- struktúra alapú tesztelés,
- kockázatelemzés a kód komplexitása alapján,
- modell alapú tesztelés.

Az első fejezet tanácsokat ad ahhoz, hogy hogyan készüljünk fel a tesztelésre (stratégia, tesztelhetőség...). Például, fontos tanulság volt a Microsoftnak, hogy ne zárjanak le „won't fix” indokkal működési hibákat csak azért, mert „a felhasználó úgyse tenne ilyet”, hisz ezek potenciális biztonsági réseket jelentenek!

A funkcionális tesztelésről szóló fejezet a szokásos technikákat ismerteti (ekvivalencia osztályok, határérték-analízis, kombinatorikus tesztelés). A fejezet nagyon hasznos része a példa, ezt azóta is gyakran használom az egyetemi gyakorlatokon. Az a feladat, hogy tesztekkel kell tervezni egy olyan

programhoz, ami pusztán csak annyit tesz, hogy egy év, hónap és nap bemeneti mezőkkel megadott dátumhoz megmondja, hogy mi a Gergely-naptár szerinti következő dátum<sup>2</sup>.

Ezután a struktúra alapú tesztelést ismerteti. A Microsoft saját tapasztalata azt mutatja, hogy a specifikáció alapján készített tesztek átlagosan 80%-os utasítás lefedettséget érnek el, ezt a felderítő (exploratory) tesztelés pár százalékkal egészíti ki, majd végül a forráskód tanulmányozásával szokták 90% körüli értékre növelni a tesztek lefedettségét, és itt állnak általában meg. A későbbi részek a blokk, utasítás, döntés, feltétel és út lefedettségét ismertetik példákon keresztül.

A harmadik fejezet azt tekinti át, hogy hogyan tudjuk különböző metrikák segítségével a kód kockázatosságának becslését felhasználni a tesztelésben.

Az utolsó fejezet a modell alapú tesztelés alapjaiba vezeti be az olvasót. Ehhez állapotgépeket, egyszerűbb nyelvtanokat és a hozzájuk tartozó algoritmusokat ismertet röviden a könyv. Később kitér a Microsoft egyik saját eszközének, a Spec Explorer-nek a működésére.

### *Harmadik rész: Tesztelési eszközök és tesztrendszerek (Part III Test Tools and Systems)*

A harmadik nagy rész a Microsoftnál használt eszközöket mutatja be. Mivel sok témát tartalmaz, nagyon részletesen nem tud mindegyikkel foglalkozni, de azért kellő áttekintést nyújt.

- *Hiba- és tesztelés-kezelő eszközök:* a fejezet részletesen ismerteti a hiba lehetséges életútját és egy hibajegy fontos tulajdonságait. Sok tanácsot ad, hogy mi az ismérve egy jó hibajelentésnek vagy egy tesztelésnek, és, hogy mit érdemes mérni ezekkel kapcsolatban. Fontos javaslat, hogy ezeket a metrikákat ne használjuk a tesztelők teljesítményének mérésére. Ugyanis, ha például azt várjuk el, hogy mindenki legalább 10 hibát találjon hetente, akkor, ha a tesztelő többet találna, azokat „elrakhatja” a következő hétre.
- *Tesztautomatizálás:* a fejezet eleje kitér arra, hogy mikor éri meg automatizálni a teszteket, és mikor nem. Az automatizált tesztek esetén a következő részfeladatokat javasolja: SEARCH – Setup, Execution, Analysis, Reporting, Cleanup, Help. A leírás jól összeszedett, a részfeladatok ismertetésénél sok, elsősre nehezen automatizálhatónak tűnő példát mutat.
- *Nem-funkcionális tesztelés:* a szöveg sorra veszi a teljesítmény, stressz, kompatibilitás, hozzáférhetőség és biztonság tesztelésének alapvető kérdéseit és a használt megoldásokat. A leírásokon túl mindegyikhez tartozik egy-egy történet is, például, hogy a Microsoftnál a stressz tesztekhez az alkalmazottak gépeit is használják éjszakánként.
- *Egyéb eszközök:* itt kerülnek elő olyan fontos témák, mint a verziókezelők, build rendszerek és statikus elemzők. Bemutatja, hogy hogyan néz ki egy tipikus build folyamat, mik a jellemzői a jó smoke tesztnek, vagy hogyan érdemes elkezdni használni egy statikus elemzőt. Végezetül egy jó tanács: kezeljük a teszt kódot is ugyanúgy, mint az éles kódot (azaz a tesztelés és tesztelési eszköz kódjával szemben az éles kóddal azonos minőségi elvárások legyenek).
- *Ügyfél-visszajelző rendszerek:* a fejezet a telemetriai rendszerekből szerezhető adatok felhasználását mutatja be (pl. Customer Experience Improvement, Windows Error Reporting). Fontos tanulság volt a Microsoft számára, hogy sok, felhasználók által jelzett hibát olyan komponensben találtak meg, amelyhez voltak nagy lefedettséget elérő tesztek – de a felhasználók máshogyan használták a szoftvert, mint ahogyan azt a tesztelők elképzelték.
- *Software Plus Services:* zárásként tanácsokat olvashatunk webes szolgáltatások teszteléséhez.

---

<sup>2</sup> A feladat meglepően bonyolult, pl. ki az, aki tudja, hogy miért érdemes vizsgálni 1752. szeptember 3-át?

*Negyedik rész: A jövőről (Part IV About the Future)*

Az utolsó rész a fontosabb új tesztelési irányokat tekinti át (mivel a könyv 2008-as, így ezek egy része már a jelen, vagy akár múlttá is vált már): automatikus hibaelemzés, virtualizáció, kód átvizsgálása.

Végezetül a könyv bemutatja az Engineering Excellence csoport tesztlői részét, amelynek az a feladata, hogy megtalálja és megossza a bevált módszereket a cégen belül. A munkájuk sokrétű: belső hírlevelek szerkesztése, előadások tartása, a megfelelő emberek összehozása (Test Leadership Team) vagy akár olyan programok szervezése, mint a „Have a lunch with a Test Architect” (a cég bármelyik mérnöke foglalhat egy időpont egy tapasztalt TA-val). A könyvet azzal a gondolattal zárják, hogy a minőség nem a tesztlők feladata, hanem minden résztvevő felelőssége, de ehhez megfelelő „kulturális” változások is kellenek egy cégben.

dr. Micskei Zoltán

BME MIT