



Budapesti Műszaki és Gazdaságtudományi Egyetem
Méréstechnika és Információs Rendszerek Tanszék

Tesztvezérelt termékfejlesztés, agilis módszertanok



Kamarás Károly (DPGED1), I. évf. (MSc) mérnök inf. szakos hallgató

Konzulensek:

dr. Ráth István, MIT

Török Gábor, Prezi.com

Szolgáltatásbiztos rendszertervezés szakirány

Önálló laboratórium 1. összefoglaló

2011/12. II. félév

A félév folyamán az önálló laboratóriumi munkámat a Prezi.com Kft.-nél [1] végeztem minőségbiztosítási mérnök (QA engineer) pozícióban. Feladataim közé tartoztak a termék működését ellenőrző funkcionális tesztek készítése, meglévő tesztek javítása, valamint a tesztkörnyezet optimalizálása és fejlesztése. Munkám során megismerkedtem az agilis szoftverfejlesztés alapelveivel, melyek alkalmazásával hatékonyan követhetők a szoftver specifikációjában bekövetkezett kisebb-nagyobb változások. Mindez a gyakorlatban Kanban rendszer és folyamatos integráció (continuous integration) alkalmazásával valósul meg. Ezzel lehetővé válik, hogy a feladatok megfelelően legyenek felosztva a csapatok között, valamint az egyes csapatokon belül. A folyamatos integráció alkalmazásával minél kisebb változást minél gyakrabban kell commitolniuk a fejlesztőknek, ezáltal megszűnik a nagy modulok integrációjából adódó overhead, és a hibák lekövetése is egyszerűsödik.

A QA (quality assurance) csapat egyik fő célja egy olyan tesztrendszer kifejlesztése, amely átfogó képet alkot a szoftver valamint a szolgáltatás működéséről a szoftverfejlesztőknek, termékmenedzsereknek, deignereknek és a supportnak egyaránt. Jól megírt, megfelelő mennyiségű teszttel a regressziós hibák száma minimalizálható, és a programtól elvárt viselkedés is könnyen leírható. Optimális esetben minden új szolgáltatás bevezetése előtt elkészülnek azok a tesztek, amelyek a feature működését fogják ellenőrizni, ezáltal maga a teszt is bizonyos szempontból specifikációként szolgálhat a fejlesztők számára. Fontos azonban a lehető legtöbb már létező feature tesztelése is, ezekkel szűrhetőek ki a regressziós hibák.

Az általunk elkészített tesztek automatizált környezetben futnak, így a kódban minden változás után azonnal jelezhetik a fejlesztők felé az esetleges hibákat. A funkcionális tesztek írására a Twist agilis szoftvertesztelő alkalmazást [2] használjuk, amellyel magas szintű nyelven fogalmazhatóak meg a teszt lépései. A Twistben megírt tesztek angol nyelvű mondatok formájában írjuk, melyekhez létezik egy-egy generált Java kódú függvényhívás. A magas szintű utasításban a változó paramétereket átadjuk a Java függvénynek, amelyet aztán keretrendszerrel függően használ tovább a Twist. Két típusú tesztkörnyezetet használunk: az egyikkel a flash action scriptben megírt egér és billentyűzet-szimulátor eseményeit szimuláljuk, a másik pedig a Java AWT Robot osztályát használja fel input események generálására. Ezek az események a programban különböző funkciókat érnek el, amelyek lefutását verifikációs lépésekkel ellenőrizzük, melyeket szintén angol nyelvű mondatok formájában fogalmazunk meg. Egy konkrét példa: ha rákattintunk a szögletes frame ikonjára, majd az egérrel (x1,y1) pontból (x2,y2) pontba drag-gelünk, akkor valóban létrejön egy frame. Sok esetben aszinkron események generálódnak, ezért az elvárt eredmények ellenőrzésére szükség volt egy polling technikájú lekérdezés implementációjára, amelyben egy adott maximális ideig kérdeztjük a szoftvert, hogy elért-e a kívánt állapotba. Ezzel megszüntettük a tesztekben feleslegesen elhelyezett várakozási lépéseket, amelyek legtöbbször tovább vártak, mint az esemény tényleges lefutása, így a tesztek lefutási ideje csökkent.

A következőkben ennek a keretrendszernek a javítása lesz a cél úgy, hogy adott esetben az a dolgozó is képes legyen értelmezni, akár megírni egy tesztet, aki nem látott még programkódot, annak érdekében, hogy a programtól elvárt működést ellenőrizni tudja. Továbbá lehetővé kell tenni a különböző platformok (flash, iPad) tesztelését úgy, hogy minél jobban fedje egymást a két rendszer.

[1] <http://prezi.com/>

[2] <http://www.thoughtworks-studios.com/twist-agile-testing>