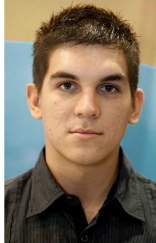




Budapesti Műszaki és Gazdaságtudományi Egyetem
Méréstechnika és Információs Rendszerek Tanszék

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. Ez a megközelítés eltér a hagyományos modellektől – mint például a vízésés modell – abban a tekintetben, hogy egy agilis fejlesztési folyamatban sokkal inkább a rövidtávú célok megvalósítására koncentrálnak, mint a kiszámítható módszertanokban, amelyek hónapokra, akár évekre megtervezett ütemterv szerint haladnak.

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 Kanban táblákon meghatározott limitek miatt nem terhelhető túl egy-egy programozó, ugyanakkor látható, hogy mely munkafázisokban van szükség több emberre, így a csapatok segíteni tudják egymás munkáját a folyamatos előrehaladás érdekében. A folyamatos integráció lényege, hogy elkerüljük a különböző modulok migrációjából adódó többletmunkát. Ha két csapat két különböző feature fejlesztésén dolgozik, majd nem kevés kód átirása után mindketten ki szeretnék adni az elkészült munkát, problémák adódhatnak a kódsorokban előforduló konfliktusok miatt. 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. Ilyen módon biztosítható a különböző részlegek közötti folyamatos kommunikáció, a gyors együttműködés, valamint az egymással szemben támasztott bizalom. 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. Feladataim közé tartozott ezen magas szintű nyelv bővítése és optimalizálása, valamint a jövőben új szoftvertesztelési módszertanokkal kell megismerkednem, és minőségbiztosítási technikák után kutatnom.

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

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