

WEB APPLICATIONS QUALITY ASSURANCE USING AUTOMATED TESTING TOOLS

Radim Reš

Master Degree Programme (2), FIT BUT

E-mail: xresra00@stud.fit.vutbr.cz

Supervised by: Jitka Kreslíková

E-mail: kreslika@fit.vutbr.cz

Abstract: The subject of this paper is web applications quality assurance using automated testing tools. The main goal of this paper is design solution for automated regression testing of web application.

Keywords: software testing, automated testing tools, Selenium

1 ÚVOD

Tato práce se zabývá zajištěním kvality webových aplikací pomocí automatického testování. Testování spadá jako nový obor do softwarového inženýrství, kde se úzce prolíná s vývojem software s cílem zajistit kvalitu výsledného produktu. V oblasti agilního vývoje software tester dostává dokumentace, které popisují určité uživatelské příběhy ve formě uživatelských požadavků na vyvíjený systém. Testeři požadavky analyzují a píší testy ve formě testových případů. Podle těchto testových případů probíhají regresní testy. Automatizace tohoto procesu je předmětem této práce [1].

2 PRINCIP TESTOVÁNÍ

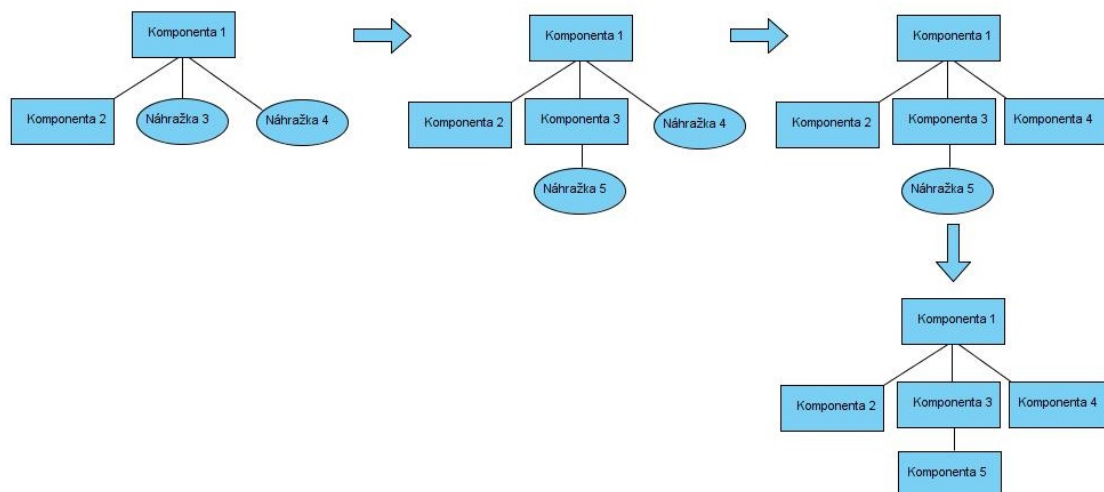
Testem rozumíme sadu kroků nebo-li také akcí, kterými porovnáváme očekávané výsledky s výstupy analyzovaného systému. Takové kroky realizují testový případ. Testovými případy analyzujeme případ použití. Pokud výstup analyzovaného systému neodpovídá očekávanému výsledku, pak případ použití obsahuje chybu, která je hlášena a testerem dále sledována její oprava [2].

2.1 PŘÍRŮSTEK V SYSTÉMU

Hlavní událostí zahájení testování je integrace nového přírůstku do systému obsahujícího implementaci komponent případu použití. Systém analyzujeme testem buď shora dolů nebo zdola nahorů dle pořadí implementace jednotlivých komponent. Testování metodou shora dolů přidáváme komponenty systému od kořene hierarchie směrem k listům, což platí analogicky i pro testování metodou zdola nahoru. Aby test mohl proběhnout, musíme nahradit chybějící komponenty náhražkami, což popisuje obrázek 1 [3].

2.2 AKCEPTAČNÍ TESTOVÁNÍ

V případě této práce analyzujeme systém na úrovni uživatelského rozhraní, které verifikujeme akceptačními testy. Takové testování obnáší provádění akcí nad uživatelským rozhraním analyzovaného systému, kde akcemi rozumíme klikni, počkej nebo ověř. Takové akce lze automatizovat.



Obrázek 1: Integrovní testování metodou shora dolů

3 NÁVRH ŘEŠENÍ AUTOMATICKÉHO TESTOVÁNÍ

Automatické testování zajišťují testovací rámce, které interpretují skript popsaný v programovacím jazyce na akce prováděné nad webovým prohlížečem. Pokud některá akce nemůže být provedena nebo nevrátí očekávaný výstup, automatický test zaznamená chybu.

3.1 SELENIUM

K hledání vzorových řešení automatického testování je použit rámec Selenium, který obsahuje základní operace, které jsou prostřednictvím skriptu interpretovány a vizualizovány prostřednictvím instance webového prohlížeče. Tyto operace použijeme k vytvoření vzorových řešení pro otestování jednotlivých akcí nad uživatelským rozhraním webové aplikace. Selenium komunikuje s instancí webového prohlížeče prostřednictvím ovladače. Takové ovladače jsou součástí nástroje Selenium. Nutno podotknout, že momentálně se Selenium omezuje pouze na ovladače pro běžné webové prohlížeče, což však pro naše účely automatického testování webových aplikací zcela dostačuje[4].

3.2 POKRAČOVÁNÍ PRÁCE

Předmětem této práce bude rozšíření rámce Selenium o funkcionalitu pokrývající webové uživatelské rozhraní testem. K zaměření elementu webové stránky lze využít selektory typu:

- ID - na stránce je tento atribut jedinečným, nemusí však být u elementu použitým.
- Name - většinou jde o atribut použitý v rámci formulářových elementů.
- CSS - elementy mohou vlastnit složený styl, který je definován pomocí tříd. Kombinace tříd může i nemusí být pro element jedinečnou.
- XPATH - elementy lze vždy adresovat unikátním klíčem popsaným jazykem vyjadřujícího strukturu XML dokumentu. Při změně struktury webové stránky to však obnáší vyšší režii na udržovatelnost skriptu automatického testu.

Navržená funkce může jednoduše pracovat s takovými selektory a vrátit identifikovaný element. K pokrytí webového uživatelského rozhraní testem bude nezbytné implementovat funkce v následujícím výčtu, které mohou potřebovat takto zaměřené elementy použít:

- Otevření stránky - funkce určující výchozí webovou stránku k testování.
- Kliknutí - funkce pokrývající pohyb po struktuře webové aplikace.
- Kontrola obsahu stránky - funkce ověřující pravdu / nepravdu tvrzení, kterými popisujeme očekávaný obsah webové stránky.
- Aktivace události elementu - funkce vyvolávající libovolnou JavaScript událost.
- Získání hodnoty elementu - funkce extrahující hodnotu elementu nebo jeho atributu.
- Kontrola cookies - funkce extrahující hodnotu uloženou v cookies pro analyzovanou webovou stránku.
- Práce s formulářovými prvky a odeslání formuláře - funkce vkládající text do textového pole nebo vybírající položky z výběru.
- Připojení souboru - funkce připojující zvolený soubor k elementu.
- Drag & Drop - funkce přiřazující element z jednoho elementu do druhého.
- Volání JavaScript a AJAX funkcí - funkce interpretující Javascript volání schopná počkat na výsledek.
- Pauza - funkce zajišťující čekání na splnění podmínky.

4 ZÁVĚR

S rostoucími nároky na kvalitu webových aplikací přibývají i požadavky na zajištění jejich kvality. Touto činností se zabývají testovací týmy, které musí opakovaně testovat krok za krokem stále to stejné, k čemuž vždy přibývá s každou novou verzí software nový přírůstek v systému k testování. Tímto jsou na testovací tým vyvíjeny stále větší nároky. Testeři jsou však jenom lidé a nejsou neomylní, čímž s přibývajícím požadavky klesá postupně i úroveň zajištění kvality testovacím týmem. Tento problém může vyřešit automatizace testovacích činností, která posouvá testery o úroveň výš a to z procesu blízcího se uživatelským činností k procesu analýzy a psaní skriptů pro automatické testy, které mohou pomoci usnadnit činnost testovacího týmu.

PODĚKOVÁNÍ

Tento příspěvek vznikl za podpory grantu VUT FIT, FIT-S-14-2299, "Výzkum pokročilých metod ICT a jejich aplikace".

REFERENCE

- [1] Ilene Burnstein. *Practical Software Testing*. Springer-Verlags, 2002. ISBN 0-387-95131-8.
- [2] Peter Farrell-Vinay. *Manage Software Testing*. Aurbach Publications, 2008. ISBN 978-0-8493-9383-9.
- [3] Jaroslav Zendulka a kolektiv. Analýza a návrh informačních systémů ais - studijní opora [online]. <https://wis.fit.vutbr.cz/FIT/st/course-files-st.php/course/AIS-IT/texts/AIS.pdf>, 2006 [cit. 2014-01-01].
- [4] Kolektiv autorů. What is selenium [online]. <http://docs.seleniumhq.org/>, [cit. 2014-01-01].