

Česká zemědělská univerzita v Praze

Provozně ekonomická fakulta

Katedra informačního inženýrství



Teze diplomové práce

Automatizované testování softwaru

Simon Sloup

© 2015 ČZU v Praze

Souhrn

Tato práce se zabývá testováním softwaru jako celku s tím, že důraz je zde kladen i na ne tolik rozšířenou možnost testování – automatizace testování softwaru. Jedná se o disciplínu, která je rozhodně nedílnou součástí vývoje každého softwaru a to již desítky let zpět do minulosti. Nicméně i přes svou historickou hloubku je stále mnohými brána jako druhořadá činnost. To dělá tuto disciplínu o to těžší. Člověk, který v tomto oboru pracuje, musí vše dopodrobna znát a musí umět argumentovat, proč je tato práce v rámci vývoje softwaru zásadní a dosáhnout tak tedy toho, aby ostatní oddělení práci testerů respektovali a brali jako pomocnou ruku, ne jako soupeře. Tato práce si dává za úkol definovat základní charakteristiky testování, tedy popsat testovací proces, rozdělení testů, popis jednotlivých rolí, a ukázat možnosti automatizace.

Klíčová slova: testování softwaru, automatizace, QA, jakost softwaru, Selenium

Diplomová práce je tematicky zaměřena na problematiku tzv. testování softwaru, obecněji zajišťování jakosti softwaru, někdy také nazývané QA (Quality Assurance). Hlavním cílem je popis celého procesu této problematiky při analýze vybraných informačních. V praktické části bylo následně řešeno testování softwaru na konkrétním příkladu a to pomocí automatizace i manuálně. Oba případy byly porovnány. Hlavní cílem této části je tedy zjištění, v jakém případě se vyplatí automatizovat testování a zda může nahradit testování manuální.

Metodika řešené problematiky práce je založena především na analýze a studiu vybraných odborných informačních zdrojů a literatury, zabývajících se testováním a vývojem softwaru. Metodika práce spočívá v sumarizaci, analýze, syntéze informací z odborné literatury. Praktická část je založena na popisu daného podniku a analýze vybraného produktu, na který byl následně implementován manuální i automatizovaný testovací případ. Na základě analýzy informačních zdrojů a řešení praktické části byly formulovány závěry diplomové práce.

Teoretická část této práce měla za úkol definovat jednotlivé náležitosti související s testováním softwaru. Byly tedy popsány jednotlivé typy testů, úrovně testování a

samozřejmě popsán testovací proces. Všechny tyto aspekty se mohou v různých literaturách mírně lišit, je tedy dobré čerpat z podobných zdrojů, aby se nerozházely. Proto zde autor vycházel především z uznávané certifikace ISTQB a jejích učebních osnov, případně z jiných zdrojů, které se také této certifikace drží. Jedná se o materiály, které tvoří zkušenosti odborníci v softwarovém testování. Tato práce může tedy sloužit jako dobrá příprava na případné zkoušky v rámci ISTQB. Dále byly popsány základní metody vývoje softwaru s důrazem na to, jak v nich probíhá testování a čím se liší. Součástí byl také popis rolí v testovacím týmu. Na závěr teoretické části byl popsán nástroj Selenium sloužící pro psaní testovacích automatických skriptů, který byl následně uplatněn v části praktické, kde byl, pomocí tohoto nástroje, automatický test vytvořen.

V praktické části byla potom představena společnost, jejímž produktem se práce zabývala. Daný softwarový produkt byl následně také představen a část, na kterou byl aplikován manuální i automatický test, byla analyzována. S použitím teoretické části, tedy literární rešerše, poté proběhlo psaní jednotlivých testů. Hlavním cílem praktické části bylo nastínit tvorbu automatických testů a následně oba druhy porovnat a ukázat, zda lze manuální testování nahradit automatickým a za jakých okolností se automatizovat vyplatí.

Na první pohled je zřejmé, že psaní automatických testů je mnohonásobně náročnější než tvorba manuálních testovacích případů. Je to způsobeno jednak časovou náročností, ale i náročností technickou. Na testera automatizéra jsou kladeny větší nároky týkající se toho, že musí znát techniky psaní (v našem případě selenium). Navíc musí mít, pokud nepíše pouze automatické testy, také všechny vlastnosti manuálního testera (cit pro detail, atd.).

Tvorba manuálních testů je vcelku jednoduchá. Krok po kroku se popíše, jak se má daná část aplikace otestovat. Zde potom samozřejmě záleží na zkušenostech jednotlivých testerů a jejich znalostech dané aplikace. Pokud se jedná o jednu aplikaci, která je vyvíjena delší dobu a tester pracuje kontinuálně pouze na ní, jeho zkušenosti s aplikací jsou obrovské a testy nemusí být popsány vůbec podrobně. Pokud jsou ovšem jeho zkušenosti malé a aplikaci nezná, musí být test popsán dopodrobna.

Automatický test je v každém případě potřebné psát podrobně, aby vykonal sled kroků, který je určen. Tento test neudělá nic navíc, oproti manuálnímu testování, které je kontrolováno lidským okem a tester si může všimnout i věcí mimo testovací krok. Automatické testy navíc

mohou mít problém s identifikací designových chyb. Čas strávený psaním automatického testu je delší; manuální testy zaberou mnohem méně času. Další nákladnou položkou je údržba automatizačního frameworku, o kterou se musí starat speciální pracovníci se znalostmi programování.

Oba přístupy mají společné to, že je potřeba je udržovat. Při změně zadání je vždy nutné aktualizovat jak testy automatické, tak testy manuální. U automatických to však vyžaduje údržbu vyšší. V případně webové aplikace, kdy se při přidání nové funkcionality změní některá část kódu, se může změnit XPath k některému elementu a test začne selhávat a je potřeba ho upravit.

Co tedy z hodnocení obou typů přístupů testování vychází je to, že tvorba manuálních testů je jednodušší, rychlejší a ve výsledku i levnější. Automatické testy jsou dražší a náročnější. Je potřeba více kvalifikovaný personál a větší údržba nejen samotných testů, ale i frameworku. Nicméně síla automatických testů je v tom, že se dají jednoduše a téměř zdarma spouštět. K provádění manuálních testů je potřeba personál, který znamená náklady navíc. Automatické testy mohou sloužit jako pravidelné smoke testy, které probíhají například každou hodinu a hodnotí, zda je aplikace životaschopná a splňuje alespoň základní požadavky. Případně mohou sloužit k regresním testům, které se pravidelně opakují a kontrolují aplikaci před vypuštěním nové verze do produkce. Po dlouhém psaní automatického testu tak není spuštěn jen jednou a poté vyhozen.

Rozdíl v ceně obou přístupů je v tom, že manuální testování má po celou dobu zhruba konstantní náklady. U automatického testování jsou počáteční náklady na automatizaci obrovské, ale mezní náklady minimální. To znamená, že průměrné náklady v čase klesají a vyplatí se tedy v případě, že se testy mnohokrát opakují.

Z analýzy této práce vyplývá, že automatické testy mohou testy manuální nahradit v případech smoke testů a regresních testů, tedy testů, které se neustále opakují a kontrolují především funkcionality a životaschopnost aplikace. Závěrem lze říci, že se tedy nedají plně nahrazovat a spíše se tyto přístupy doplňují.

Vybrané citované zdroje

BECK, Kent a spol., agilemanifesto.org [ONLINE], [cit. 2015-02-17], 2001. *Manifesto for Agile Software Development*. Dostupné z WWW: <http://agilemanifesto.org/>

BURNS, David. *Selenium 2 Testing Tools: Beginner's Guide*. 2. vydání. Birmingham: Packt Publishing, 2012. ISBN 1849518300

COPELAND, Lee. *A Practitioner's Guide to Software Test Design*. 1. vydání. Londýn: Artech House, 2003. ISBN 1-58053-791-X

CRAIG, Rick D. a Stefan P. Jaskiel. *Systematic Software Testing*. 1. vydání. Londýn: Artech House, 2002. ISBN 9781580535083

KADLEC, Václav. *Agilní programování: metodiky efektivního vývoje softwaru*. 1. vydání. Brno: Computer Press, 2004. ISBN 80-251-0342-0

MCWHERTER, Jeff a Ben Hall. *Testing ASP.NET Web Applications*. 1. vydání. Indianapolis: Wiley Publishing, 2010. ISBN 978-0-470-49664-0

PAGE, Alan a spol. *Jak testuje software Microsoft*. 1. vydání. Brno: Computer Press, 2014. ISBN 9788025128695

PATTON, Ron. *Testování softwaru*. 1. vydání. Brno: Computer Press, 2002. ISBN 80-7226-636-5

ROUDENSKÝ, Petr a Anna Havlíčková. *Řízení kvality softwaru*. 1. vydání. Brno: Computer Press, 2013. ISBN 9788025138168