BACHELORARBEIT

im Studiengang Bachelor Informatik

Software-Test von Web-Applikationen

Ausgeführt von: Bernhard Posselt
Personenkennzeichen: 1010257029

Begutachter: MSc Benedikt Salzbrunn

Wien, 20. Mai 2013

FACHHOCHSCHULE
TECHNIKUM WIEN

Eidesstattliche Erklärung

„Ich erkläre hiermit an Eides statt, dass ich die vorliegende Arbeit selbständig angefertigt habe. Die aus fremden Quellen direkt oder indirekt übernommenen Gedanken sind als solche kenntlich gemacht. Die Arbeit wurde bisher weder in gleicher noch in ähnlicher Form einer anderen Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht. Ich versichere, dass die abgegebene Version jener im Uploadtool entspricht."

Ort, Datum

Unterschrift

Kurzfassung

Die Durchführung und Erstellen von automatisierten Tests für Web-Applikationen unterscheidet sich von klassischen Applikationen. Aufgrund der komplexeren Infrastruktur und Modularisierung werden zusätzliche Testfälle und Strategien benötigt um Web-Applikationen ausreichend abzudecken und eine fortwährende Qualität zu gewährleisten. Diese Arbeit soll Möglichkeiten für den Test von Web-Applikationen aufzeigen.

Schlagwörter: Webapplikationen, automatisierte Tests, Webtest

Abstract

Creation and execution of automatic web application tests is different from tests of classic applications. A more complex infrastructure and modularisation require additional testcases and strategies to guarantee a good enough test coverage which in return ensures constant quality. This thesis highlights various possibilities to test web applications.

Keywords: web applications, automatic tests, webtest

Inhaltsverzeichnis

1 Einführung

1.1 Problemstellung

Durch die zunehmende Verbreitung von Smartphones und anderen mobilen Geräten erfährt der Markt für Software-Applikationen eine zunehmende Fragmentierung [1][S. 1]. Viele dieser Plattformen erfordern das Erlernen von unterschiedlichen Programmiersprachen, Toolkits und Betriebssystemen [2][3]. Will ein/eine Software-EntwicklerIn eine Applikation plattformübergreifend anbieten, erfordert dies daher einen höheren Zeit- und Kostenaufwand.

Da viele dieser Mobilgeräte über einen Web-Browser verfügen, wird das Web als Applikationsplattform immer attraktiver und bestimmte Software-Tools wie PhoneGap [4] erlauben dem/der EntwicklerIn sogar eine plattformübergreifende Mobil-Applikationen auf Basis von Web-Technoligien zu erstellen, welche sich nahtlos in das System integrieren. Durch die Verwendung unterschiedlicher Web-Browser erhöht sich jedoch auch die Fehler-Rate der Applikation.

1.2 Lösungsansatz

1.3 Aufbau

In Kapitel 1 In Kapitel 2 werden die Unterschiede

2 Warum testen

Keine Applikation ist fehlerfrei[5][S. 9]. Diese Fehler führen nicht nur zu unzufriedenen Kunden, sondern auch zu hohen Kosten: „Im Jahr 2000 wurde in den USA ein Schaden durch Softwarefehler in der Auto- und Flugzeugindustrie von 1,8 Milliarden US-Dollar errechnet. Dies entspricht ca. 16 % des Softwareumsatzes.“[6][S. 15]

Die Fehler-Rate wird auf ungefähr 3 pro 1.000 Zeilen geschätzt, was bei einer aufwendigeren Applikation mit 100 Millionen Zeilen Quellcode eine durchschnittliche Anzahl von von 300.000 Fehlern ergibt [7][S. 10].

Je früher diese Fehler entdeckt werden, desto kostengünstiger können diese beseitigt werden [6][S. 17]. Daher ist es wichtig, möglichst früh mit dem Testen zu beginnen und es in den Software-Entwicklungsprozess zu integrieren [6][S. 16].

2.1 Grenzen von Software-Tests

Das Vorhandensein von Tests kann jedoch niemals eine komplett fehlerfreie Software garantieren, nur eine Abwesenheit von bestimmten Fehlern kann garantiert werden, nämlich jenen, die explizit mit Tests abgedeckt werden. [7][S. 12]

Außerdem ist es nicht erstrebenswert, die ganze Applikation mit Tests abzudecken, da dies zu zu hohen Entwicklungskosten führen kann. Idealerweise werden daher nur jene kritischen Fälle mit Tests abgedeckt, deren Fehlerkosten die Kosten für die Erstellung der Tests übertreffen. [7][S. 12]

Die implementierten Testfälle müssen auch regelmäßig gewartet und aktualisiert werden, um ihre Effektivität zu gewährleisten, denn diese nimmt auf Dauer ab. Es ensteht eine sogenannte „Testresistenz“
[7][S. 12], die daraus resultiert, dass die bestehenden Tests nur bekannte Fehlerfälle abdecken und neue mögliche Fehlerfälle nicht berücksichtigen. [7][S. 12-13]

2.2 Vorteile von automatisierten Tests

Da die vorhandenen Tests durch die Einbindung in den Entwicklungsprozess öfters durchgeführt werden müssen, kann sich durch das manuelle Testen der immer gleichen Tests schnell eine gewisse Eintönigkeit einstellen. Das wiederum kann unter anderem dazu führen, dass Tester bestimmte Tests nicht korrekt oder ineffizient ausführen.

Dieses Problem kann durch eine Automatisierung des Testprozesses gelöst werden. Die Automatisierung erlaubt zudem bei zukünftigen Testdurchläufen eine Reduzierung auf ein Minimum des ursprünglichen Zeitaufwandes. Daher sollte versucht werden, möglichst viele dieser Tests zu automatisieren. [8][S. 22-23]

2.3 Testen im Projektmanagement

Tests können jedoch nicht nur dazu verwendet werden, um Fehler in der Applikation zu finden, sondern auch um einen Überblick über den derzeitigen Stand der Implementation zu gewinnen: Sie geben dem/der EntwicklerIn und ProjektmanagerIn ein direktes Feedback über bereits korrekt implementierte Teile der Software-Spezifikation. Auch Milestones können durch Tests definiert werden. [8][S. 2]

3 Unterschiede zu klassischer Software

Im Gegensatz zu klassischen Desktop- oder Mobil-Applikationen bestehen Web-Applikationen wegen ihrer Client-Server-Architektur immer aus mehreren Modulen, die meist über ein Netzwerk miteinander verbunden sind, beispielsweise:

- Datenbank-Server
- Web-Server
- Applikations-Server
- Authentifizierungs-Server
- Web-Browser

Durch diesen modularen Aufbau ist es besonders schwer einen Fehler zu lokalisieren. Ein Fehler kann z.B. durch einen Fehler im Quellcode des Applikations-Servers oder durch ein Netzwerkproblem entstanden sein. [9][Foreword]

Außerdem gibt es eine größere Vielfalt an Plattformen, auf welchen Web-Applikationen ausgeführt wird: Auf der Serverseite sind diese Plattformen noch vom/von der BetreiberIn festlegbar, sprich welches Betriebssystem und welche Datenbank eingesetzt wird, auf der Clientseite ist dies aber schon nicht mehr möglich. Die BesucherInnen der Webseite verwenden verschiedene Web-Browser auf verschiedenen Betriebssystemen, welche beide in unterschiedlichen Versionen vorliegen können. Auch können unterschiedliche Plugins und Fonts in unterschiedlichen Versionen installiert sein. [9][Foreword]

Zudem verlagert sich auch immer mehr Applikations-Logik von der Server- auf die Clientseite[9][S. 13]. Durch das Verwenden von *Events*, welche unter anderem durch BenutzerInnen-Eingaben ausgelöst werden können, stellt clientseitiger Code eine größere Herausforderung für den/die TesterIn dar als Serverseitiger: Events können in unterschiedlicher Reihenfolge und Kombination auftreten, manche Aktionen lösen sogar mehrere Events aus. [9][S. 18]

Eine weitere Herausforderung stellt das Session Modell von Web-Applikationen dar: viele Web-Applikationen verwenden nur eine Session pro NutzerIn, erlauben aber mehrere gleichzeitige Logins. Werden mehrere Instanzen der Applikation gestartet - z.B. loggt sich der/die NutzerIn auf dem Mobiltelefon und dem Laptop auf der Webseite ein - kann dies zu Synchronisationsproblemen zwischen den einzelnen Instanzen führen: Wird in einer Instanz ein Eintrag gelöscht, kann dieser durch eine fehlerhafte Synchronisation in einer andere Instanz immer noch existieren und in weiterer Folge zu Fehlern führen. [9][S. 20]

Diese Vielfalt an verschiedenen möglichen Konfigurationen und Herausforderungen erfordert eine neue Herangehensweise an das Thema Software-Test: Die bestehenden Techniken sind „zwar auch notwendig, aber nicht ausreichend, um die Qualität der Applikation sicherzustellen“ [10][S. 18]

Literaturverzeichnis

[1] F. L. und Weiguo Ye, "Operating system battle in the ecosystem of smartphone industry," 2009.

[2] "Android Developer Website," Website, Google, 2013, http://developer.android.com/ [Zugang am 20.05.2013].

[3] "iOS Dev Center," Website, Apple, 2013, https://developer.apple.com/devcenter/ios/index.action [Zugang am 20.05.2013].

[4] "PhoneGap Website," Website, Adobe, 2013, http://phonegap.com/ [Zugang am 20.05.2013].

[5] M. Schmidt, "An Empirical Investigation of Human-Based and Tool-Supported Testing Strategies," Master's thesis, Technische Universität Wien, 2011.

[6] F. Richter, "Betriebliche Einführung von automatischen Softwaretests unter Berücksichtigung ökonomischer und technischer Rahmenbedingungen," Master's thesis, Technische Universität Wien, 2007.

[7] M. Oberreiter, "Evaluierung, Konzeption und Härtung eines Testprozesses für automatisierte Regressionstests in einem mittelgroßen Entwicklungsszenario," Master's thesis, Technische Universität Wien, 2011.

[8] A. Waser, "Test Automation A Case Study," Master's thesis, Technische Universität Wien, 2002.

[9] H. Q. Nguyen, *Testing Applications on the Web*. Wiley Computer Publishing, 2001.

[10] S. Avci, "Evaluieren von automatisierten Tests bei Web-Applikationen," Master's thesis, Technische Universität Wien, 2010.

Abbildungsverzeichnis

Abkürzungsverzeichnis