Version 0.1α
letzte Änderung: 5. März 2014

BEUTH HOCHSCHULE
FÜR TECHNIK
BERLIN
University of Applied Sciences

Evaluierung von Techniken zur parallel-synchronen Bedienung einer Web-Applikation auf verschiedenen mobilen Endgeräten

vorgelegt von

Adrian Randhahn

EDV.Nr.:744818

dem Fachbereich VI – Informatik und Medien –
der Beuth Hochschule für Technik Berlin vorgelegte Bachelorarbeit
zur Erlangung des akademischen Grades
Bachelor of Science (B.Sc.)
im Studiengang
Medieninformatik

Tag der Abgabe 5. März 2014

Gutachter

Prof. Knabe	Beuth Hochschule für Technik
Prof. Dr. Wambach	Beuth Hochschule für Technik

[1.5ex]

Erklärung

Ich versichere, dass ich diese Abschlussarbeit ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Quellen und Hilfsmittel benutzt habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich gemacht. Ich erkläre weiterhin, dass die vorliegende Arbeit noch nicht im Rahmen eines anderen Prüfungsverfahrens eingereicht wurde.

Datum Unterschrift

Sperrvermerk

Die vorliegende Arbeit beinhaltet interne und vertrauliche Informationen der Firma New Image Systems GmbH. Die Weitergabe des Inhalts der Arbeit im Gesamten oder in Teilen sowie das Anfertigen von Kopien oder Abschriften - auch in digitaler Form - sind grundsätzlich untersagt. Ausnahmen bedürfen der schriftlichen Genehmigung der Firma New Image Systems GmbH.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

1 Einleitung

In der modernen Webentwicklung durchläuft eine Anwendung verschiedene Etappen eines Entwicklungszykluses. Er beginnt bei einem Auftrag oder einer Idee, darauf folgt dann die Spezifikation einzelner Usecases[1]. Im Anschluss folgt in der Regel die Entwicklung und Implementation[2] der einzelnen Komponenten. Am Ende der jeweiligen Implementationsphase durchläuft das Produkt[3] die Qualitätskontrolle. Sollten in diesem Abschnitt Fehler auftreten wird das Produkt dem Entwickler zur erneuten Bearbeitung vorgelegt. Dieser Vorgang kann sich beliebig oft wiederholen. Bei großen und komplexen

Softwaresystemen ist es trotz zeitgemäßer Implementierung nicht immer Ausgeschlossen, dass Kaskadierungsfehler[4] entstehen. Aus Sicht der Qualitätssicherung ist dies ein lästiges Problem, da diese nach jedem erneuten Modifikationsvorganges eines Softwaresegments einen größeren Segmentblock, wenn nicht sogar das gesamte Softwareystem erneut testen muss. Bei der Entwicklung auf und für mobile Endgeräte[5] kommt noch ein erschwerender

Faktor hinzu, nämlich die diversen, verschiedenen Bildschirmauflösungen. Diese können nicht nur die Darstellung des Inhaltes beeinflussen, sondern auch daraus folgend die Interaktionskonformität beeinflussen.
Im Optimalfall wird die Software erst nach vollständiger Homogenität auf allen unterstützen Geräten freigegeben.

Dieser zyklisch wiederkehrende Prozessablauf ist sehr Zeitintensiv und nimmt linear mit der Anzahl der zu testenden Geräte zu.

Das Ergebnis dieser Forschungsarbeit soll zeigen, wie verschiedene Softwareframeworks die Zeit, die in die Qualitätssicherung investiert wird, beeinflussen können, indem sie die Steuerung diverser Geräte parallel-synchron steuern. Die Evaluierung soll zeigen wo die Vorteile und Nachteile der einzelnen Werkzeuge liegen. Weiterhin soll gezeigt werden ob aktuelle Frameworks erweiterbar sind um Beispielsweise automatisierte Testunits zu implementieren.

Anmerkung

Aus Gründen der besseren Lesbarkeit wird für alle Personen und Funktionsbezeichnungen durchgängig das generische Maskulinum angewendet und bezieht in gleicher Weise Frauen und Männer ein.

[1] Szenario oder auch Anwendungsfall

[2] Einbindung

[3] hier: einzelne Softwarekomponente

[4] Fehler die nicht im eigentlichen Segment auftreten, sondern eine oder mehr Ebenen weiter unten in der Systemhirarchie

[5] Smartphones, Tabletts oder Ähnliche

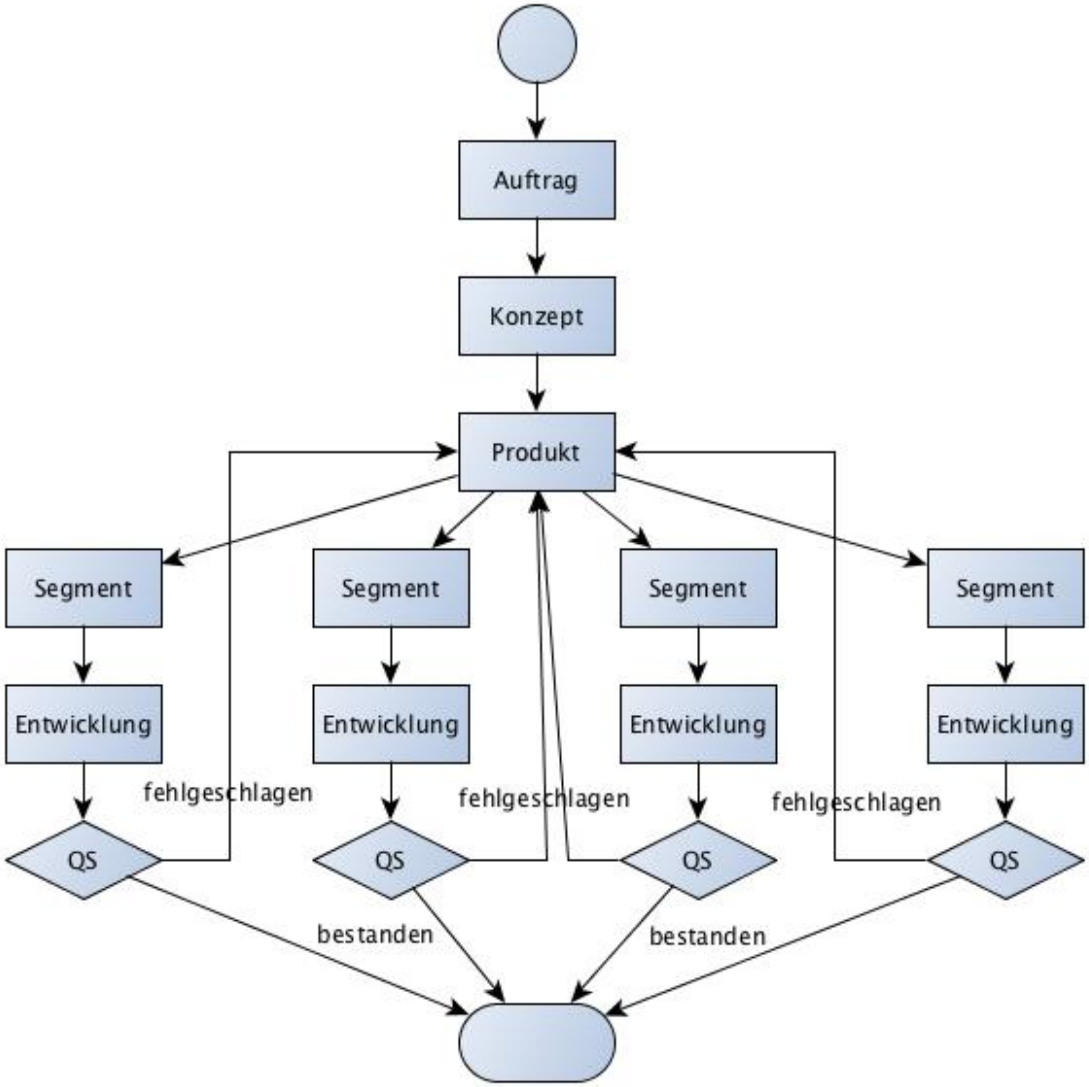

Abbildung 1.1: Vereinfachte Darstellung eines Softwareentwicklungsprozesses

2 Aufgabenstellung

2.1 Problemstellung

Zur Qualitatsprufung wird ein Testszenario erstellt, welches moglichst alle, oder zumindest einen Großteil der Anforderungen erfullt. Dieses Szenario wird nun, von Hand, an jedem vorhandenen Testgerat durchgefuhrt, und dies moglichst immer konstant. Das Ergebnis wird dem Softwareentwickler mitgeteilt, welcher gegebenenfalls die Software anpasst. Dieser Vorgang wiederholt sich solange bis das erwunschte Ergebnis erreicht ist. Bei einer Vielzahl von Testgeraten entsteht das Problem, dass die Testszenarien durch Nachlassigkeit, Unachtsamkeit oder auch Routine nicht immer vollstandig durchlaufen werden, was schlussfolgernd zu verminderter Qualitat fuhrt.

2.2 Annahmen und Einschränkungen

2.3 Zielsetzung

Das Ziel dieser Arbeit ist es, bestehende Frameworks auf verschiedene Faktoren zu Evaluieren.

2.4 Abgrenzungskriterien

3 Grundlagen

3.1 Begriffsklärung

3.1.1 parallel-synchron

3.1.2 Web-Applikation

3.1.3 Endgeräte

3.2 technischer Aufbau

3.3 Komponenten

3.3.1 Raspberry Pi

3.3.2 Hardware

4 Technologien

4.1 Ghostlab

4.2 NodeJS

4.3 Zombie.js

4.4 W3C Touch Events Extensions

4.5 Phantom Limb

4.6 jQuery UI Touch Punch

4.7 jQuery Touchit

4.8 NPM touchit

5 Lösungsansätze

5.1 Adobe Edge Inspect

5.2 Ghostlab

5.3 Remote Preview

5.4 Browser-Sync

5.5 Eigenes Framework

6 Evaluation

6.1 Adobe Edge Inspect

6.1.1 Einführung

6.1.2 Vorteile

6.1.3 Nachteile

6.2 Ghostlab

6.2.1 Einführung

6.2.2 Vorteile

6.2.3 Nachteile

6.3 Remote Preview

6.3.1 Einführung

6.3.2 Vorteile

6.3.3 Nachteile

6.4 Browser-Sync

6.4.1 Einführung

6.4.2 Vorteile

6.4.3 Nachteile

6.5 Eigenes Framework

6.5.1 Einführung

6.5.2 Systementwurf

Ablaufdiagram

Klassendiagramm

6.5.3 Vorteile

6.5.4 Nachteile

7 Helpers

7.1 quote

> Dies ist ein Zitat.

7.2 longquote

> Dies ist ein längeres Zitat.

7.3 fussnote

Dies ist der Text[1]

[1] Und dies ist die Fußnote dazu.

Index