WIRTSCHAFTSINFORMATIK

Studylab

PASCAL MÖDINGER

EIN VERGLEICH MIT DEN TECHNIKEN CUCUMBER UND SPOCK

DAS BEHAVIOUR DRIVEN DEVELOPMENT IN AGILEN PROJEKTEN DER AUTOMOBILBRANCHE

Pascal Mödinger

Das Behaviour Driven Development in agilen Projekten der Automobilbranche

Ein Vergleich mit den Techniken Cucumber und Spock

Bibliografische Information der Deutschen Nationalbibliothek:

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über http://dnb.d-nb.de abrufbar.

Impressum:

Copyright © Studylab 2021

Ein Imprint der GRIN Publishing GmbH, München

Druck und Bindung: Books on Demand GmbH, Norderstedt, Germany

Coverbild: GRIN Publishing GmbH | Freepik.com | Flaticon.com | ei8htz

Abstract

Das Ziel dieser Arbeit ist es herauszufinden, was die Probleme beim Automatisieren von Tests nach den Behaviour Driven Development (BDD) Prinzipien sind und welches Testwerkzeug sich für die Automatisierung am besten eignet. Zudem soll herausgefunden werden, welche Kriterien wichtig für die Auswahl von Testtools sind. Dazu wird folgende Forschungsfrage aufgestellt: „Welches Behaviour Driven Development Framework eignet sich zur Automatisierung von Akzeptanztests am besten?“. Zur Beantwortung der Forschungsfrage werden noch die Fragen, „Welche Probleme gibt es bei dem Einsatz von BDD-Tools in agilen Softwareprojekten?“ und „Welche Kriterien sind bei der Auswahl eines Testframeworks wichtig?“ in dieser Arbeit aufgegriffen und bearbeitet. Zur Beantwortung dieser Fragen wird eine Online-Umfrage durchgeführt, um die Problemstellungen beim Einsatz von BDD-Frameworks herauszufinden. Die BDD-Frameworks werden anhand eines Testszenarios implementiert und mit aufgestellten Kriterien verglichen, die aus den Problemstellungen herausgearbeitet werden. Für den Vergleich werden die Frameworks Cucumber und Spock ausgewählt. Der durchgeführte Vergleich hat ergeben, dass sich Cucumber im Rahmen des aufgestellten Testszenarios besser als Spock für die Testautomatisierung eignet. Die Online-Umfrage hat gezeigt, welche Probleme beim Testen auftreten. Beispielsweise werden Testdaten in den Testfällen eingebaut, wodurch Änderungen der Testdaten sehr viel Aufwand nach sich ziehen. Aber auch eine fehlende oder schlechte Möglichkeit der Darstellung der Testfallergebnisse wurde unter anderem als Problem bei der durchgeführten Umfrage deutlich. Die Gewichtung aufgestellter Kategorien für den Vergleich hat gezeigt, dass eine gute Auswertung der Ergebnisse ausschlaggebend für die Wahl eines Frameworks zur Testautomatisierung ist. Aufgrund der Ergebnisse empfiehlt sich die Wahl eines Frameworks, welches verschiedene und umfassende Möglichkeiten bietet um die Testergebnisse anzuzeigen.

Inhaltsverzeichnis

Abstract ..... **III**

Inhaltsverzeichnis ..... **IV**

1 Einleitung, Motivation und Überblick **1**

2 Problemstellung und Forschungsfrage **3**

3 State of the Art von Softwaretesting **4**

3.1 Testautomatisierung 4

3.2 Test Driven Development (TDD) 8

3.3 Behaviour Driven Development (BDD) 10

3.4 Unterschied zwischen TDD und BDD 12

3.5 Aktuelle Forschungsarbeiten zu dem Thema 13

4 Methodisches Vorgehen **15**

5 Ergebnis **18**

5.1 Vorstellung und Auswahl der BDD-Frameworks 18

5.2 Testszenario 22

5.3 Kriterienkatalog 24

5.4 Ergebnisse der Implementierung mit Cucumber 28

5.5 Ergebnisse der Implementierung mit Spock 39

5.6 Vergleich der Frameworks 45

6 Fazit **50**

6.1 Zusammenfassung 50

6.2 Limitations 50

6.3 Next Steps 51

6.4 Lessions Learned 51

6.5 Contribution to Practice 52

6.6 Contribution to Science 52

Abbildungsverzeichnis **53**

Tabellenverzeichnis **55**

Literaturverzeichnis **56**

Anhang **58**

1 Einleitung, Motivation und Überblick

Behaviour Driven Development (BDD) ist sowohl in der Forschung als auch in der Praxis ein relevantes Thema. Besonders in Projekten, die nach agilen Projektmanagementmethoden vorgehen hat BDD in den vergangenen Jahren immer mehr an Bedeutung gewonnen. Es gibt mittlerweile eine Vielzahl an Testwerkzeugen, die bei der Testautomatisierung von Softwaretests nach den BDD Prinzipien herangezogen werden können. Dabei ist es nicht immer einfach, ein Werkzeug für die speziellen Anforderungen in einem Projekt auszuwählen. Bei der Auswahl eines Testwerkzeuges ist auch darauf zu achten, welche Ebene von Softwaretests automatisiert werden soll. Diese Arbeit soll einen Überblick über gängige Tools zur Testautomatisierung nach den BDD-Prinzipien geben und bei der Auswahl eines Behaviour Driven Development Frameworks helfen. Dazu werden in dieser Arbeit verschiedene BDD-Tools miteinander verglichen.

Aktuell gibt es in dem Projekt die Herausforderung mit der Automatisierung der Softwaretests. In diesem Projekt für die Arbeit wird ein Konfigurator für KFZ-Fahrzeuge entwickelt. Dieser Konfigurator stellt verschieden Schnittstellen zur Verfügung, die beispielsweise den Preis und die Emissionswerte für ein konfiguriertes Fahrzeug errechnen und zurückgeben. Die eigens entwickelten Funktionen des Konfigurators sollen entsprechend mit Akzeptanztests abgedeckt werden. Ein Akzeptanztest ist eine bestimmte Art von Softwaretest und wird zu einem späteren Zeitpunkt in Kapitel 3.1.2 noch genauer vorgestellt. Derzeit werden noch viele Akzeptanztests der Konfigurator-Software manuell ausgeführt. Dabei müssen teilweise die Preise für konfigurierte Fahrzeug noch manuell errechnet, und mit dem richtigen Preis abgeglichen werden. Für den implementierten Konfigurator wurden aber auch schon einige Akzeptanztests mit dem Behaviour Driven Development (BDD) Werkzeug Cucumber automatisiert. Allerdings schlagen die bereits automatisierten Tests fehl, wenn veraltete Testdaten verwendet werden. In dem entwickelten Konfigurator werden sehr häufig neue Fahrzeugkonfigurationen hinzugefügt und veraltete Fahrzeuge entfernt. Diese veralteten Konfigurationen von Fahrzeugen müssen sehr zeitaufwändig manuell entfernt und mit neuen, gültigen Fahrzeugkonfigurationen ersetzt werden.

Zu Beginn dieser Arbeit wird beschrieben, welche Vorteile Testautomatisierung von Softwaretests gerade in Agilen Projekten mit sich bringt. Danach werden in Kapitel 3.1.1 die verschiedenen Testebenen, in die Softwaretests unterschieden werden können, vorgestellt. Da in dieser Arbeit Akzeptanztests im Mittelpunkt der Testautomatisierung stehen, werden diese noch einmal gesondert näher