



Hochschule Bremen
Zentrum für Informatik und Medientechnologien
Fakultät für Elektrotechnik und Informatik

Exposee zur Bachelorthesis

im Studiengang Duales Studium Informatik

Thema	Vergleich und Evaluierung verschiedener Frameworks zur Realisierung automatischer Test für angularJS-Webanwendungen
Autor	Nikolas Schreck <nikolasschreck@gmail.com> Matrikel-Nr. 372605
Version vom:	16. März 2017
Erstprüfer	Prof. Dr.-Ing. Heide-Rose Vatterrott
Zweitprüfer	Dipl.-Inf. Jochen Schwitzing

1 Hintergrund

AngularJS ist ein clientseitiges JavaScript-Framework zur Realisierung von Single-Page-Webapplikationen. Es wurde von *Google Inc.* ins Leben und ist seit 2009 ein Open-Source-Projekt. AngularJS ist weitverbreitet und wird Statistiken ¹ zufolge in über 450.000 Webanwendungen, davon bekannten wie *Gmail* ², *PayPal* ³ oder *YouTube for PS3*, eingesetzt.

Im September 2006 wurde Version 2 von Angular veröffentlicht, die mit Version 1 nicht rückwärtskompatibel ist. Zur besseren Unterscheidbarkeit haben sich seitdem folgende Begrifflichkeiten durchgesetzt: *AngularJS* für Version 1, *Angular2* für Version 2, *Angular* als Sammelbegriff für beide Versionen.

Im Commerzbank-Konzern wurde durch das regulatorisch geprägte und dadurch träge IT-Umfeld bei Webanwendungen bisher auf klassische serverseitige Technologien gesetzt. Erst in der jüngeren Vergangenheit wird in vereinzelten Projekten auf AngularJS als technologische Basis gesetzt. Automatisierte Entwicklertests in JavaScript werden hierbei meist stiefmütterlich behandelt; die Anwendungen stattdessen per Hand im Browser getestet.

2 Zielsetzung

Ziel ist es, die Möglichkeiten von automatisierten Tests - Komponenten- sowie End-To-End-Tests - für AngularJS-Webanwendungen zu untersuchen. Es soll aus den verschiedensten existierenden Test-Frameworks und Tools unter Berücksichtigung der Vorgaben und Eigenheiten innerhalb der Commerzbank sowie der Benutzerfreundlichkeit eine Auswahl getroffen werden. Auch soll überprüft werden, ob automatisierte Tests auf die Einhaltung von Programmierrichtlinien möglich und sinnvoll sind.

Aufgrund der bisherigen Verbreitung innerhalb der Commerzbank wird die Auswahl für AngularJS getroffen. Jedoch soll untersucht werden, inwiefern sich durch den Einsatz von Angular2 Verbesserungen im Hinblick auf Testbarkeit ergeben.

3 Vorgehen

In der Thesis sollen in einer Recherche verfügbare Testframeworks und -tools gesucht und diese mittels Programmdokumentationen und weiteren Quellen erkundet werden. Die gefundenen Ergebnisse sollen kategorisiert und anhand von vorher festzulegenden Kriterien in einer Entscheidungsmatrix miteinander verglichen werden. Erwartetes

¹<https://trends.builtwith.com/javascript/Angular-JS>

²www.google.com/gmail

³www.paypal.com

Ergebnis ist eine Auswahl von Software, mittels welcher die Kriterien abgedeckt und somit automatisierte Test bestmöglich realisierbar sind.

Mithilfe der getroffenen Auswahl sollen anschließend für eine exemplarisch gewählte, produktiv genutzte AngularJS-Webanwendung Komponenten- sowie End-To-End-Tests realisiert werden. Dies dient der Validierung der ausgewählten Testframeworks im Hinblick auf Realisierbarkeit, Zusammenspiel und einfache Anwendbarkeit.

Optional soll die Möglichkeit der Einbettung dieser automatisierten Tests in bestehende Continuous-Integration- und Build-Prozesse geprüft werden.

Die Arbeit kann in folgende Teile gegliedert werden:

- Erläuterung des Hintergrunds, Einführung und Erläuterung von Angular sowie automatisierten Tests
- Erstellen einer Liste von verfügbaren Frameworks und Tools für Testautomatisierung in JavaScript
 - Untersuchung der Softwaredokumentationen
 - Untersuchung von Internetquellen, z. B. Erfahrungsberichten, im Hinblick auf die gefundene Software
 - Kategorisierung der gefundenen Software
- Festlegung von Kriterien
- Vergleich und Auswahl der gefundenen Software in einer Entscheidungsmatrix mit den festgelegten Kriterien
- Evaluierung mittels Realisierung von Tests mit der getroffenen Softwareauswahl
- Vergleich von AngularJS und Angular2 im Hinblick auf Testbarkeit