

Bachelorthesis

Nachvollziehbarkeit von Nutzerinteraktion und Anwendungsverhalten am Beispiel JavaScript-basierter Webapplikationen

An der Fachhochschule Dortmund
im Fachbereich Informatik
Studiengang Software- und Systemtechnik, Vertiefung Softwaretechnik
erstellte Bachelorthesis
zur Erlangung des akademischen Grades
Bachelor of Science

von
Marvin Kienitz
geb. am 26.04.1996
Matr.-Nr. 7097533

Betreuer:
Prof. Dr. Sven Jörges
Dipl. Inf. Stephan Müller

Dortmund, 14. Oktober 2020

Kurzfassung

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Abstract

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Motivation	1
1.2	Problemstellung	1
1.3	Zielsetzung	2
1.3.1	Abgrenzung	2
1.4	Vorgehensweise	3
1.5	Open Knowledge GmbH	3
2	Ausgangssituation	4
2.1	Browserumgebung	4
2.2	JavaScript-basierte Webapplikationen	4
2.3	Softwarebetrieb	5
2.3.1	Klassisches Vorgehen	5
2.3.2	Agiles Vorgehen	5
2.4	Nachvollziehbarkeit	6
2.4.1	Nachvollziehbarkeit bei Webapplikationen	6
3	Methoden und Praktiken	7
3.1	Fehlerberichte	7
3.2	Logging	7
3.3	Monitoring	7
3.4	Metriken	8
3.5	Tracing	8
4	Beispielhafte Integration	9
4.1	Konzept	9
4.1.1	Architektur	9
4.1.2	Datenverarbeitung	9
4.1.2.1	Erhebung	9
4.1.2.2	Auswertung	9
4.1.2.3	Visualisierung	9
4.2	Implementierung	9
4.2.1	Technologie-Stack	10

Inhaltsverzeichnis

4.3	Demonstration	10
5	Abschluss	11
5.1	Fazit	11
5.2	Ausblick	11
	Eidesstattliche Erklärung	12
	Abkürzungsverzeichnis	13
	Abbildungsverzeichnis	14
	Tabellenverzeichnis	15
	Quellcodeverzeichnis	15
	Literaturverzeichnis	16

1 Einleitung

1.1 Motivation

Entwickler und Betreiber von JavaScript-basierten Webapplikationen werden fast alltäglich damit konfrontiert, dass sie beispielsweise zu Zwecken der Problembehebung nachvollziehen müssen, wie ein Nutzer die Applikation verwendet und wie sich die Applikation dabei verhält. Dies ist in der Umgebung von Browsern deutlich schwieriger, als in anderen Projekten mit Nutzerinteraktion. Denn es gibt von Haus aus keine Möglichkeit über Logs das Verhalten nachzuvollziehen, weil die Logs nur beim Client einsehbar sind. Weiterhin muss man bei einer eigenen Lösung zur Informationsgewinnung mit der JavaScript-Sandbox im Browser umgehen, und CORS-Vorgaben einhalten.

Mit diesen Schwierigkeiten bei Webapplikationen wird sich in dieser Arbeit beschäftigt, um nachhaltig eine Nachvollziehbarkeit zu erreichen.

1.2 Problemstellung

Die besonderen Hürden bei GUIs und speziell bei JavaScript-basierten Webapplikationen, welche die Nachvollziehbarkeit einschränken, gilt es zu bewältigen.

Diese Hürden für Stakeholder umfassen folgendes:

1. Kontextinformationen sind nicht einsehbar und
2. Nutzerverhalten ist nicht bekannt.

Problemberichte sind eine gängige Wahl, um den Stakeholdern eine Verständnishilfe zu bieten (vgl. Abbildung 1.1). Bettenburg *et al* [BJS⁺08] fanden jedoch bei Problemberichten eine Diskrepanz, zwischen dem was die Stakeholder als hilfreich empfanden und dem was Nutzer ihnen als Bericht lieferten.

Dies lässt folgern, dass die Informationserhebung nicht allein von den Nutzern aus geschehen sollte. Es ist ein

Abb. 1.1: Formular aus der Instagram [Inc20] Android App

übergreifendes Konzept anzustreben, welches die Informationslücke ausgleicht und eine Nachvollziehbarkeit für die Stakeholder ermöglicht.

1.3 Zielsetzung

Ziel dieser Arbeit ist es, eine Möglichkeit zu schaffen, dass die Stakeholder die Interaktionen eines Nutzers und das Verhalten einer Webapplikation nachvollziehen können. Dieses Ziel wird unter dem Begriff "Nachvollziehbarkeit" zusammengefasst.

Folgende Fragen sollen im Zuge der Ausarbeitung beantwortet werden:

1. Was bedeutet Nachvollziehbarkeit?
2. Warum ist Nachvollziehbarkeit wichtig?
3. Wie kann eine gute Nachvollziehbarkeit erreicht werden?
 - a) Was wird hierzu benötigt?
 - b) Wie kann Nutzerverhalten nachvollzogen werden?
 - c) Wie können Fehler nachgestellt werden?
4. Wie ist eine Webapplikation zu erweitern um dies zu erreichen?
(Hierbei sollen Projekte von Open Knowledge untersucht werden.)
 - a) Was für Technologien helfen hierbei?
 - b) Was sind die Auswirkungen für den Nutzer?
 - i. Wird die Leistung der Webapplikation beeinträchtigt?
 - ii. Wie wird mit seinen Daten umgegangen (Stichwort DSGVO)?

1.3.1 Abgrenzung

Bei der Betrachtung von Webapplikationen, sollen nur jene betrachtet werden, die dynamisch mit JavaScript erzeugt werden - auch Single-Page-Applications (SPAs) genannt.

Bei der Implementierung soll ein Proof-of-Concept (PoC) bzw. eine Demonstration der vorgestellten Methoden und Konzepte erstellt werden. Eine allgemeingültige Lösung übersteigt den Rahmen der Bachelorarbeit und ist nicht anzustreben.

Im Zuge der Implementierung der Erweiterung, soll beleuchtet werden, welche Daten erhoben werden und wie mit diesen umgegangen wird. Eine Analyse der Datenverarbeitung in Hinblick auf vollste Konformität mit der DSGVO wird jedoch nicht Bestandteil sein.

1.4 Vorgehensweise

Um das Ziel, die Erstellung eines Proof-of-Concept (PoC), zu erreichen, wird vorerst die Literatur geprüft und die Ausgangssituation erörtert und beschrieben. Die Nachvollziehbarkeit und ihr Nutzen werden vorgestellt.

Durch das gewonnene Verständnis über die Ausgangssituation, werden nun Methoden aufgezeigt, welche eine Nachvollziehbarkeit unterstützen. Methoden wie Fehlerberichte, Logging, Monitoring und ggf. andere sind zu betrachten.

Auf Basis des detaillierten Verständnisses der Problemstellung und der Methoden wird nun die Erstellung eines PoC Fokus sein. Das PoC erfolgt auf Basis bestehender Webapplikation der Open Knowledge GmbH. Sie soll als Ziel haben, die Nachvollziehbarkeit zu erhöhen.

Ist ein PoC nun erstellt, wird analysiert, welchen Einfluss es auf die Nachvollziehbarkeit hat und ob die gewünschten Ziele erreicht wurden.

1.5 Open Knowledge GmbH

Die Bachelorarbeit wird unter anderem im Rahmen einer Werkstudententätigkeit innerhalb der Open Knowledge GmbH erstellt. Der Standortleiter des Standortes Essen, Dipl. Inf. Stephan Müller, übernimmt die Zweitbetreuung.

Die Open Knowledge GmbH ist ein brancheneutrales mittelständisches Dienstleistungsunternehmen mit dem Ziel bei der Analyse, Planung und Durchführung von Softwareprojekten zu unterstützen. Das Unternehmen wurde im Jahr 2000 in Oldenburg, dem Hauptsitz des Unternehmens, gegründet und beschäftigt heute 74 Mitarbeiter. Mitte 2017 wurde der zweite Standort in Essen eröffnet an dem aktuell 13 Mitarbeiter angestellt sind.

Die Mitarbeiter von Open Knowledge übernehmen in Kundenprojekten Aufgaben bei der Analyse über die Projektziele und der aktuellen Ausgangssituationen, der Konzeption der geplanten Software, sowie der anschließenden Implementierung. Die erstellten Softwarelösungen stellen Individuallösungen dar und werden den Bedürfnissen der einzelnen Kunden entsprechend konzipiert und implementiert. Technisch liegt die Spezialisierung bei der Mobile- und bei der Java Enterprise Entwicklung, bei der stets moderne Technologien und Konzepte verwendet werden. Aufgrund der großen Expertise in den Bereichen Technologien und Konzepte sind sowohl die Geschäftsführer als auch diverse Mitarbeiter der Open Knowledge GmbH als Redner auf Fachmessen wie der Javaland oder Autoren in Fachzeitschriften wie dem Java Magazin vertreten.

2 Ausgangssituation

2.1 Browserumgebung

Hier soll eine Beschreibung der Umgebung erstellt werden, mit Nennung der speziellen Eigenschaften (Keywords: Sandbox, CORS, Logging)

2.2 JavaScript-basierte Webapplikationen

Hier soll beschrieben werden, was JavaScript-basierte Webapplikationen sind.

Diese Ausarbeitung konzentriert sich, wie im Titel beschrieben, auf JavaScript-basierte Webapplikationen. Weiterhin wird sich auf sogenannte Single-Page-Applications (SPAs) konzentriert, welche eine Submenge der JavaScript-basierten Webapplikationen darstellen. Um allen Lesern eine gleiche Grundkenntnis zu ermöglichen, werden diese Konzepte nun kurz vorgestellt.

..Erklärung zu JavaScript-basierten Webapplikationen..

..Erklärung zu SPAs..

Bekannte Frameworks, um SPAs zu Erstellen, sind beispielsweise Angular [Goo20], React [Fac20] oder Vue.js [YM20].

2.3 Softwarebetrieb

Diese Arbeit konzentriert sich auf Software, die sich in der Betriebsphase befindet. Gängige Software-Entwicklungszyklen und ihre Definition dieser Phase werden folgend beschrieben.

2.3.1 Klassisches Vorgehen

In vielen Modellen über den Lebenszyklus einer Software wird die Phase während der Betreuung oftmals “Maintenance” genannt [Gin02] [Bas12], in der Instandhaltung und Support den Alltag bestimmen. Sie ist nach Zelkowitz *et al* [ZSG79] für rund zwei Drittel der Entwicklungskosten verantwortlich, begründet durch exponentielle Steigung [Bec99].

Das Wasserfallmodell [Bas12] sowie das V-Modell XT [BS12] sehen vor, dass in dieser Phase die Software funktionstüchtig gehalten wird und dass die Anforderungen an die Software erfüllt sind. Bei nicht-erfüllten Anforderungen oder Fehlern, werden diese behoben. Jedoch ein kontinuierlicher Verbesserungsprozess ist in dieser Phase nicht vorgesehen.

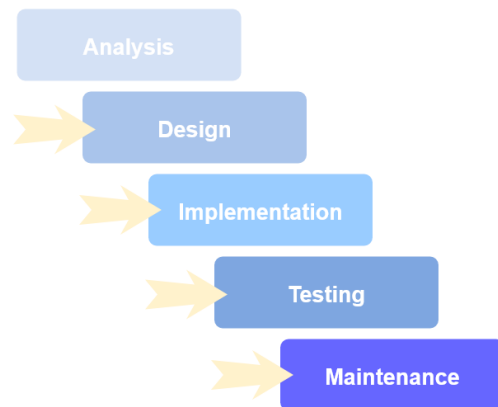


Abb. 2.1: Lebenszyklus einer Software
Quelle: Eigene Darstellung von [Bas12]

2.3.2 Agiles Vorgehen

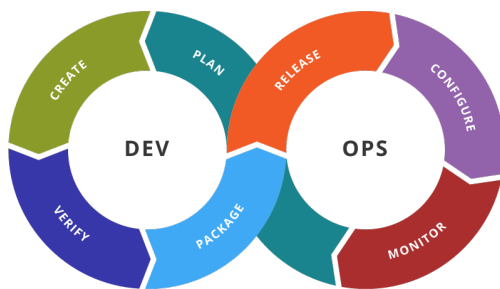


Abb. 2.2: DevOps Toolchain
Quelle: Wikimedia Commons [Kha16]

verfolgt (vgl. Abbildung 2.2).

Bei agilen Ansätzen wird der Betrieb meist nicht abgegrenzt von der normalen Entwicklung. In dieser Phase werden weiterhin Anforderungen erhoben und diese Stück für Stück umgesetzt [BS12]. Vorteilhaft dabei ist, dass auf neue Wünsche oder Auffälligkeiten sehr einfach reagiert werden kann. Es wird eine kontinuierliche Verbesserung angestrebt.

Um den kontinuierlichen Entwicklungs- und Deploymentprozess reibungslos ablaufen zu lassen, werden Ansätze wie DevOps [Has15]

2.4 Nachvollziehbarkeit

Hier soll die Nachvollziehbarkeit allgemein beschrieben werden und warum sie erstrebenswert ist.

Sie beschäftigt sich mit der Informationserfassung und -aufbereitung, um das Verhalten eines Systems und die Interaktionen der Nutzer für die Stakeholder verständlich zu machen. Sie ist getrennt von der Anstrengung einer Reproduzierbarkeit nach der wissenschaftlichen Methode anzusehen.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Tritt ein Problem bei einem Nutzer auf, aber die Stakeholder erhalten nicht ausreichende Informationen, so kann der Bug ignoriert werden oder in Vergessenheit geraten. Dies geschah im Jahr 2013, als Khalil Shreath eine Sicherheitslücke bei Facebook fand und bei Facebooks Bug-Bounty-Projekt Whitehat meldete [SD13]. Sein Fehlerreport wurde aufgrund mangelnder Informationen abgelehnt:

Unfortunately your report [...] did not have enough technical information for us to take action on it. We cannot respond to reports which do not contain enough detail to allow us to reproduce an issue.

2.4.1 Nachvollziehbarkeit bei Webapplikationen

Hier sollen die Besonderen Hürden bei Webapplikationen hervorgehoben werden (indirekte Kommunikation, keinen Zugriff auf Logs, etc.)

3 Methoden und Praktiken

In diesem Kapitel soll beschrieben werden, wie eine Nachvollziehbarkeit in Webapplikationen erreicht werden kann. Spezielle Methoden und Praktiken sollen vorgestellt und beleuchtet werden.

3.1 Fehlerberichte

Folgende Fragen sollen zur Methode beantwortet werden

1. *Was genau sind Fehlerberichte (=Bug-Reports)*
2. *Welches Verhalten kann hiermit aufgedeckt/nachvollziehbar gemacht werden?*

3.2 Logging

Folgende Fragen sollen zur Methode beantwortet werden

1. *Gibt es Besonderheiten zu Logging in anderen Projekten (Backend vs. Frontend)?*
2. *Wie können Logs an einen auswertenden Stakeholder gelangen??*
3. *Welches Verhalten kann hiermit aufgedeckt/nachvollziehbar gemacht werden?*

3.3 Monitoring

Folgende Fragen sollen zur Methode beantwortet werden

1. *Welche Anwendungseigenschaften sind zu monitoren?*
2. *Welches Verhalten kann hiermit aufgedeckt/nachvollziehbar gemacht werden?*

3.4 Metriken

Folgende Fragen sollen zur Methode beantwortet werden

1. *Welche Metriken können definiert?*
2. *Wie können Metriken definiert werden?*
3. *Welches Verhalten kann hiermit aufgedeckt/nachvollziehbar gemacht werden?*

3.5 Tracing

Folgende Fragen sollen zur Methode beantwortet werden

1. *Welche Nutzerinteraktionen sind zu tracen?*
2. *Welches Verhalten kann hiermit aufgedeckt/nachvollziehbar gemacht werden?*

4 Beispielhafte Integration

4.1 Konzept

4.1.1 Architektur

Hier soll die grobe Architektur geplant werden, welche Komponente es gibt und wie diese kommunizieren sollen.

4.1.2 Datenverarbeitung

4.1.2.1 Erhebung

Wie werden die Daten erhoben (Nennung der verwendeten Methoden!)? Wie gelangen die Daten an eine auswertende Komponente?

4.1.2.2 Auswertung

Wie werden die Daten zusammengefasst und ausgewertet? Wie gelangt das Ergebnis an die darstellende Komponente?

4.1.2.3 Visualisierung

Wie werden den Stakeholdern die Informationen präsentiert?

4.2 Implementierung

Auf Basis des Konzeptes soll nun eine Implementierung erfolgen.

4.2.1 Technologie-Stack

4.3 Demonstration

Nachdem nun eine Implementierung steht, soll die Erweiterung auf nicht-technische Weise veranschaulicht werden. Hier soll dargestellt werden, wie die Nachvollziehbarkeit nun verbessert worden ist.

5 Abschluss

5.1 Fazit

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

5.2 Ausblick

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Eidesstattliche Erklärung

Hiermit versichere ich an Eides statt, dass ich die vorliegende Arbeit selbständig und ohne die Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten und nicht veröffentlichten Schriften entnommen wurden, sind als solche kenntlich gemacht. Die Arbeit ist in gleicher oder ähnlicher Form oder auszugsweise im Rahmen einer anderen Prüfung noch nicht vorgelegt worden.

Dortmund, am (Unterschrift)

Abkürzungs- und Erklärungsverzeichnis

DSGVO Datenschutz Grundverordnung

GUI Graphical-User-Interface, Grafische Benutzeroberfläche

PoC Proof-of-Concept

SPA Single Page Application

Stakeholder In dieser Arbeit werden Betreiber und Entwickler einer Webapplikation als Stakeholder bezeichnet

Abbildungsverzeichnis

1.1	Formular aus der Instagram [Inc20] Android App	1
2.1	Lebenszyklus einer Software	5
2.2	DevOps Toolchain	5

Tabellenverzeichnis

Quellcodeverzeichnis

Literaturverzeichnis

- [Bas12] BASSIL, Youssef: A Simulation Model for the Waterfall Software Development Life Cycle. In: *International Journal of Engineering & Technology (iJET)* 2 (2012), Nr. 5
- [Bec99] *Kapitel 5*. In: BECK, Kent: *Extreme programming explained: embrace change*. addison-wesley professional, 1999, S. 21
- [BJS⁺08] BETTENBURG, Nicolas ; JUST, Sascha ; SCHRÖTER, Adrian ; WEISS, Cathrin ; PREMRAJ, Rahul ; ZIMMERMANN, Thomas: What makes a good bug report? In: *Proceedings of the 16th ACM SIGSOFT International Symposium on Foundations of software engineering*, 2008. – ISBN 0–7695–0262–8, S. 308–318
- [BS12] BALAJI, S. ; SUNDARARAJAN MURUGAIYAN, M.: Waterfall vs. V-Model vs. Agile: A comparative study on SDLC. In: *International Journal of Information Technology and Business Management* 2 (2012), Nr. 1, S. 26–30
- [Fac20] FACEBOOK INC.: *React - A JavaScript library for building user interfaces*. <https://reactjs.org>, 2020. – [Online; abgerufen am 12.10.2020]
- [Gin02] GINIGE, Athula: Web Engineering: Managing the Complexity of Web Systems Development. In: *Proceedings of the 14th international conference on Software engineering and knowledge engineering*, 2002, S. 721–729
- [Goo20] GOOGLE LLC: *Angular*. <https://angular.io>, 2020. – [Online; abgerufen am 12.10.2020]
- [Has15] HASSELBRING, Wilhelm: DevOps: Softwarearchitektur an der Schnittstelle zwischen Entwicklung und Betrieb. In: *GI-Fachtagung Architekturen* (2015)
- [Inc20] INC., Facebook: *Instagram App Screenshot*. <https://www.instagram.com/>, 2020
- [Kha16] KHARNAGY: *File:Devops-toolchain.svg*. <https://commons.wikimedia.org/wiki/File:Devops-toolchain.svg>, 2016. – [Online; abgerufen am 13.10.2020]
- [SD13] SHREATEH, Khalil ; DEWEY, Caitlyn: Mark Zuckerberg’s Facebook page was hacked by an unemployed Web developer. In: *The Washington Post* (2013), Aug

- [YM20] YOU, Evan ; MITWIRKENDE individuelle: *Vue.js*. <https://vuejs.org/>, 2020.
– [Online; abgerufen am 12.10.2020]
- [ZSG79] ZELKOWITZ, Marvin V. ; SHAW, Alan C. ; GANNON, John D.: *Principles of Software Engineering and Design*. Prentice Hall Inc., 1979. – ISBN 978-0-13-710202-0