

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, 12. 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	Instandhaltung und Support	5
2.1	Phase in Softwareprojekten	5
2.2	Nachvollziehbarkeit	5
2.3	Maintenance bei Webapplikationen	6
3	Methoden und Praktiken	7
3.1	Fehlerberichte	7
3.2	Logging	7
3.3	Metriken	7
3.4	Monitoring	7
3.5	Tracing	7
4	Beispielhafte Integration	9
4.1	Konzept	9
4.2	Implementierung	9
4.3	Demonstration	9
5	Abschluss	11
5.1	Fazit	11
5.2	Ausblick	11
	Eidesstattliche Erklärung	12
	Abkürzungsverzeichnis	14

Abbildungsverzeichnis	15
Tabellenverzeichnis	16
Quellcodeverzeichnis	16
Literaturverzeichnis	17

1 Einleitung

1.1 Motivation

In der Welt der Systeme mit Benutzerinteraktionen gibt es stets die Hürde, dass “im Feld“ unvorhergesehene Probleme auftreten. Donald Norman [Nor89] argumentierte bereits 1989, dass bei komplexen Aufgaben und Umgebungen das Unerwartete zu erwarten ist.

Diese Probleme gilt es meist zu beheben, um einen uneingeschränkten Ablauf zu gewährleisten. Bei Webapplikationen, also dem Fokus der Arbeit, können im Regelfall die Nutzer selber die Probleme nicht lösen. Für dessen Behebung werden Informationen benötigt, welche sich auf das System und den Nutzer beziehen, und welche den Verantwortlichen mitzuteilen sind. Diese Verantwortlichen werden in dieser Arbeit mit **Stakeholder** bezeichnet und umfassen Betreiber und Entwickler.

Für eine schnelle und gründliche Behebung benötigen die Stakeholder detaillierte Informationen, was in der Praxis aber selten der Fall ist [RC99]. Fehlende Informationen erschweren die Verständnisk Gewinnung, dadurch dass das Problem nicht oder nicht einfach nachvollziehbar ist.

1.2 Problemstellung

Problemberichte sind eine gängige Wahl, um Nutzern die Möglichkeit zu bieten Probleme zu melden und zu beschreiben (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 übergreifendes Konzept anzustreben, welches die Informationslücke ausgleicht und eine Nachvollziehbarkeit für die Stakeholder ermöglicht.

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

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

Es ist dem Leser zu vermitteln, was die theoretischen Grundlagen sind und wie die der Nachvollziehbarkeit definiert wird. Es gilt zu erörtern, warum die Nachvollziehbarkeit erstrebenswert ist und wie sehr sie bereits Beachtung findet.

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

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

Zum Abschluss ist ein Fazit zu erstellen. Darin enthalten ist eine kritische Eigenreflexion sowie ein Ausblick, welcher beschreibt, wie das Themengebiet und die erstellte Software voranschreiten könnten.

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. [Bec19]

2 Instandhaltung und Support

2.1 Phase in Softwareprojekten

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

Es werden immer bessere Methoden entwickelt, um Probleme in Software - oder auch Bugs - zu verringern. Jedoch erhöht sich zugleich die Komplexität von Software, was zur Ursache hat, dass es mehr Nährboden für Bugs gibt [HL02]. De-facto sind Bugs ein unvermeidbarer Bestandteil einer Software und müssen daher erwartet und gehandhabt werden [Bro95].

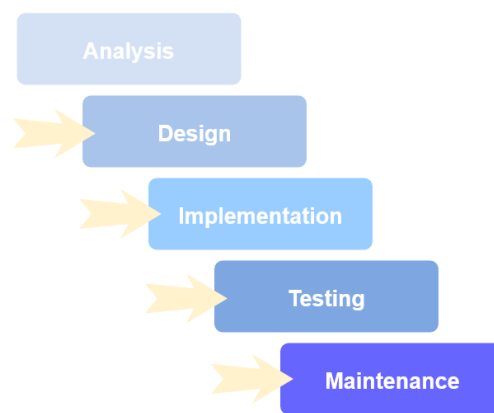


Abb. 2.1: Lebenszyklus einer Software
Quelle: Eigene Darstellung von [Bas12]

Wenn nun ein Bug auffällt, sei es durch einen Nutzer oder auch zufällig einem Stakeholder, muss entschieden werden, ob dieser zu beheben ist. Wenn eine Behebung angestrebt wird, benötigt der Stakeholder meistens Rahmeninformationen [BJS⁺08] um den Bug ggf. zu reproduzieren und die Situation nachzuvollziehen. Desto mehr Verständnis der Stakeholder über das Problem erhält, desto schneller und präziser kann er die Ursache aufdecken. Die Ermöglichung der schnellen Verständnis über ein Problem, wird in dieser Arbeit **Nachvollziehbarkeit** genannt.

2.2 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 Shreatch 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.

2.3 Maintenance bei Webapplikationen

Hier sollen die Besonderen Hürden bei Webapplikationen hervorgehoben werden (ungeschulte Nutzer, indirekte Kommunikation, etc.)

Webapplikationen haben in der Phase der Maintenance ein höheres Potenzial Hindernissen zu begegnen. Einerseits ist JavaScript dafür bekannt, dass fehleranfälliger Quellcode häufiger auftritt [Swi17] [GVS⁺19]. Andererseits ist oftmals eine große und breit verteilte Nutzerschaft die Zielgruppe. Dies bedeutet potenziell viele unterschiedliche Umgebungen, stark abweichende Verständnisse einer Webapplikation und indirekte Kommunikationswege zu den Stakeholdern.

3 Methoden und Praktiken

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

3.1 Fehlerberichte

3.2 Logging

3.3 Metriken

3.4 Monitoring

3.5 Tracing

4 Beispielhafte Integration

4.1 Konzept

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

4.2 Implementierung

Auf Basis des Konzeptes soll nun eine Implementierung erfolgen.

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 so zusammengefasst

Abbildungsverzeichnis

1.1	Formular aus der Instagram [Inc20] Android App	1
2.1	Lebenszyklus einer Software	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
- [Bec19] BECKER, Vincent: *Vorstellung und Demonstration des Frameworks zur Überprüfung der Akzeptanzkriterien von User Stories einer RESTful API*. 2019
- [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
- [Bro95] *Kapitel 2*. In: BROOKS, Frederick P. J.: *The mythical man-month: essays on software engineering*. Pearson Education, 1995
- [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
- [GVS⁺19] GYIMESI, Péter ; VANCICS, Béla ; STOCCO, Andrea ; MAZINANIAN, Davood ; BESZÉDES, Árpád ; FERENC, Rudolf ; MESBAH, Ali: BugsJS: a benchmark of JavaScript bugs. In: *2019 12th IEEE Conference on Software Testing, Validation and Verification (ICST)* IEEE, 2019, S. 90–101
- [HL02] HANGAL, Sudheendra. ; LAM, Monica S.: Tracking down software bugs using automatic anomaly detection. In: *Proceedings of the 24th International Conference on Software Engineering. ICSE 2002* IEEE, 2002, S. 291–301
- [Inc20] INC., Facebook: *Instagram App Screenshot*. <https://www.instagram.com/>, 2020
- [Nor89] NORMAN, Donald A.: The “Problem“ of Automation: Inappropriate Feedback and Interaction, Not “Overautomation“. In: *ICS Report 8904* (1989)

- [RC99] RENAUD, K. ; COOPER, R.: An error reporting and feedback component for component-based transaction processing systems. In: *Proceedings User Interfaces to Data Intensive Systems*, 1999. – ISBN 0–7695–0262–8, S. 141–150
- [SD13] SHREATEH, Khalil ; DEWEY, Caitlyn: Mark Zuckerberg’s Facebook page was hacked by an unemployed Web developer. In: *The Washington Post* (2013), Aug
- [Swi17] SWIGGETT, Jack: Beyond JavaScript: Building Stable Web Applications. (2017)
- [ZSG79] In: ZELKOWITZ, Marvin V. ; SHAW, Alan C. ; GANNON, John D.: *Principles of Software Engineering and Design*. Prentice Hall Professional Technical Reference, 1979. – ISBN 978–0–13–710202–0