

Recherchebericht
Nachrichtenkommunikation für das THW
na17b

Inhaltsverzeichnis

1	Begriffe	2
1.1	RDF	2
1.2	SPARQL	2
1.3	Framework	2
1.4	Modultest	2
1.5	Deployment	2
1.6	Model-View-Controller	2
1.7	Java	3
1.8	Markdown	3
1.9	HTML	3
1.10	CSS	3
1.11	JavaScript	3
1.12	Continuous Integration	3
1.13	git	4
2	Aspekte	4
2.1	Gitlab CI	4
2.2	Slack	5
2.3	Testing Framework	5
2.4	Maven	5
2.5	Jekyll	5
2.6	gitflow	6
2.7	gitlab	6
3	Konzepte	6
3.1	Digitalisierung von Arbeitsabläufen	6
3.2	Multiview	6
3.3	Fachgruppe Führung und Kommunikation	7
3.4	THW Mobile Zentrale	7
3.4.1	Vierfach-Vordruck	7

1 Begriffe

1.1 RDF

kurz für Resource Description Framework, ist ein Datenmodell, das stark von der üblichen Vorstellung von Daten in Tabellen abweicht. Alle Daten werden immer in Triples der Form Subjekt, Prädikat, Objekt dargestellt, z.B. ein Subjekt Leipzig ist durch das Prädikat hatHochschule mit dem Objekt Universität Leipzig verbunden. Durch diese Schreibweise bildet sich bei Verwendung einer großen Menge von Triples eine riesige Baumstruktur auf, in der viele Entitäten durch Relationen miteinander verknüpft sind.

1.2 SPARQL

kurz für Semantic Protocol and RDF Query Language, ist wie SQL für tabellenbasierte Datenbanken, eine Anfragesprache, mit der man gezielt Informationen aus einem RDF Datenmodell extrahieren kann.

1.3 Framework

Ein Framework stellt dem Softwareentwickler einen Rahmen mit festen Regeln und wiederverwendbare Strukturen zur Verfügung. Insbesondere definiert es den Kontrollfluss der Anwendung und die Schnittstellen der Klassen. Im Gegensatz zur Arbeit mit Klassenbibliotheken werden nicht bereits vorhandene Klassen und Funktionen verknüpft und verwendet um den Ansprüchen des Projekts zu genügen, stattdessen werden selbständig Funktionen und Klassen in einem vorgegebenen Rahmen implementiert, welche dann vom Framework genutzt werden um vordefinierte Aufgaben zu erledigen. Woraus sich auch ergibt das Frameworks stark Anwendungsspezifisch sind. [3]

1.4 Modultest

Modultests sind ein Bestandteil des Softwaretests [14] und beschreiben das automatische Testen von Softwaremodulen auf deren korrekte Funktionalität. Hierfür müssen im Vorfeld Testfälle mit möglichst umfassenden Eingabeparametern und den zu erwartenden Ausgabeparametern definiert werden, mit welchen das Modul - bspw. unter Verwendung eines Testing Frameworks - auf Funktionalität geprüft wird. [12]

1.5 Deployment

bezeichnet die Visualisierung und Definierung von Schritten und der Rollenverteilung in einem Prozess. Die aus Deployment resultierenden Flowcharts stellen eine Reihenfolge von Aktionen/Schritten, sowie die Interaktion zwischen verschiedenen Personen und Gruppen dar. Der große Vorteil dieses Modells liegt bei Hervorhebung von Ineffizienz, Dopplung und unnötigen Abläufen. [2]

1.6 Model-View-Controller

MVC ist ein Muster zur Implementierung von User-Interfaces, welches ein Programm in 3 Teile zerlegt. Dies soll die interne Repräsentation von Informationen und deren Verarbeitung von der Art wie diesem dem Nutzer präsentiert werden trennen. Dies erhöht die Wiederverwendbarkeit von Code und erlaubt die parallele Entwicklung der verschiedenen Teile. Das Model ist hierbei die zentrale Komponente, welche das Problem beschreibt. Es managt direkt die Daten und Logik des Programms. Die View-Komponente besteht aus der Darstellung der Daten z.B. durch Tabellen oder Diagramme und ermöglicht es verschiedene Views für verschiedene Nutzer zu definieren. Der Controller akzeptiert User-Input der dann in die entsprechenden Befehle für Model und

View umgewandelt wird. Ursprünglich wurde MVC vor allem für die Entwicklung von Desktop GUIs benutzt, allerdings hat sich das Muster auch mittlerweile für die Entwicklung von Web- und Mobile- Applikationen bewährt. [13]

1.7 Java

Java ist eine allzweck Programmiersprache, welche Klassen- und Objektorientiert ist und sich vor allem durch ihre Plattformunabhängigkeit auszeichnet. Der Quellcode wird hierfür in bytecode kompiliert, der dann auf jeder Java virtual machine unabhängig von System oder Architektur zum laufen gebracht werden kann. [8]

1.8 Markdown

Markdown ist ein einfacher Syntax, um HTML-Code aus Textdokumenten zu generieren. Die Sprache ist einfach zu lesen, zu schreiben und zu bearbeiten. Verwendet wird Markdown z.B. auf GitHub oder auch für Jekyll.[10]

1.9 HTML

HTML steht für Hypertext Markup Language und ist eine Sprachkonvention, mit der digitale Dokumente für die Interpretation durch Web-Browser semantisch strukturiert werden. HTML wurde von Forschern entwickelt, um Informationen digital austauschen zu können, ohne diese mehrmals umformen und anpassen zu müssen. Ein HTML-Dokument besteht aus Doctype (Dokumenttypdeklaration), Head (ausgeblendete Informationen) und Body (der Inhalt).

1.10 CSS

CSS steht für Cascading Style Sheets ist eine Gestaltungssprache, über die sich die Darstellung von Informationen steuern lässt. Im Gegensatz zu HTML wird sich dabei nicht auf den Inhalt bezogen, sondern allein auf das Aussehen.

1.11 JavaScript

JavaScript ist eine Skriptsprache, die dazu dient, Benutzerinteraktionen mit Webseiten auszuwerten und daraufhin Inhalte anzupassen, nachzuladen oder zu erzeugen. Dadurch sollen die Funktionalitäten von HTML und CSS erweitert werden. JavaScript ist eine dynamisch typisierte, objektorientierte, klassenlose Skriptsprache. Skriptsprache: Eine Programmiersprache, die vor allem für kleine Programme verwendet wird und sich durch Interpreterausführung und oftmals entfallenden Deklarationszwang auszeichnen.

1.12 Continuous Integration

Der Begriff Continuous Integration (CI) beschreibt eine Softwareentwicklungspraxis, in der alle beteiligten Entwickler eines Projektes häufig (ein- bis mehrmals am Tag) ihren aktuellen Arbeitsstand (meistens verwirklicht in einem dezidierten Branch) in den Hauptstand (Masterbranch) integrieren.

Um dieses Ziel zu erreichen werden meist folgende Tools im Entwicklungsprozess verwendet:

- Version Control System (z.B. Git)
- automatisierte Unit-Tests
- CI-Server (z.B. Github CI)

Die direkten Vorteile der CI sind zum einen der minimale Integrationsaufwand sowie die Fähigkeit, jederzeit eine stabile Version veröffentlichen zu können. Der minimale Integrationsaufwand ergibt sich durch die Häufigkeit, mit der die Entwickler ihren Arbeitsstand integrieren. Durch die Bedingung dass vor jeder Integration alle automatisierten Testfälle ohne Fehlermeldungen durchlaufen müssen ergibt sich ein relativ stabiler Hauptstand, der jederzeit als Grundlage für neue Features dienen kann. Bugs die nicht durch Unit-Tests erkannt wurden und erst im Release auffallen, können dank der kompakten Natur kurzer Integrationszyklen schnell erkannt und behoben werden. Kommunikation ist insofern ein wichtiger Aspekt der CI, als dass jederzeit alle Entwickler den Zustand der Software kennen können sollten. Hierbei hilft der Einsatz eines CI-Servers, der beispielsweise jederzeit wenn Code in den Hauptstand integriert werden soll, den kompletten Build Prozess nachvollzieht, alle Tests durchlaufen lässt und entsprechend alle auftauchenden Fehlermeldungen an das Team weitergibt. [1]

1.13 git

Git ist eine freie Software zur verteilten Versionsverwaltung. Die Entwicklung wurde von Linus Torvalds angestoßen. Gegenüber anderen Versionskontrollsystemen unterscheidet es sich allerdings in einigen Aspekten:

Nicht lineare Entwicklung: D.h. jeder Entwickler hat sein eigenes Repository und jedes Feature wird in seinem eigenen Branch entwickelt. "Branching"(Erstellen neuer Entwicklungszweige) und "Merging"(Verschmelzen mind. zweier Entwicklungszweige) sind als integrale Bestandteile der Arbeit mit Git. Des Weiteren werden nativ Tools zur Visualisierung der nicht linearen Geschichte bereit gestellt.

Kein zentraler Server: Jeder Benutzer hat eine lokale Kopie des gesamten Repository und der Versionsgeschichte. Das bietet zum einen Sicherheit (Datensicherung) und zum anderen ist für die meisten Tätigkeiten so kein Netzwerkzugriff erforderlich. Zwischen den lokalen und entfernten Repositories gibt es zwar keinen technischen Unterschied, aber meist gibt es ein offizielles Repository.

Kryptographische Sicherheit der Projektgeschichte: Die Projektgeschichte wird so gespeichert, dass der Hash-Code einzelner Commits immer auf der ganzen Geschichte basiert. Somit ist nachträgliche Manipulation nicht möglich, ohne den Hash-Code automatisch zu ändern.

Unterstützung vieler Übertragungsprotokolle: Die Unterstützung diverser Übertragungsprotokolle wie HTTP, HTTPS, FTP, SSH und rsync bietet dem Nutzer große Freiheit seine Daten zu senden wie er es für richtig hält.

Säubern des Repositories: Auch wenn man Daten/Aktionen aus den Repositories löscht/zurücknimmt, so bleiben sie doch erstmal erhalten und können zu jedem Zeitpunkt wiederhergestellt werden, was Fehler beim entfernen von Dateien/Aktionen sehr schwer macht. Erst wenn man sich nach diesem Schritt sicher ist kann man die Daten explizit und für immer löschen. [4]

2 Aspekte

2.1 Gitlab CI

Gitlab CI ist ein open-source Continuous Integration Service von Gitlab. Gitlab-CI ist vollständig in Gitlab integriert, was es zu einer naheliegenden CI-Service Wahl für unser Projekt macht. Gitlab-CI unterstützt u.a.:

- Parallele Build-Prozesse
- einfache Konfiguration über .gitlab-ci.yml
- Echtzeit Logging

[7]

2.2 Slack

Slack ist ein webbasierter Instant-Messaging Dienst des Unternehmens Slack Technologies zur Kommunikation innerhalb von Arbeitsgruppen. Slack erlaubt es gemeinsam oder mit Einzelpersonen zu chatten und Dokumente auszutauschen. Die Tatsache dass der Dienst leicht über externe Apps (z.B. Polly, eine Poll-App) an die Anforderung der jeweiligen Gruppe angepasst werden kann macht ihn zu einem mächtigen Kommunikationswerkzeug. Desweiteren ist es über Web-Hooks möglich, dass externe Dienste wie (z.B. Gitlab) Eventnachrichten direkt an die Slack-Gruppe senden können. Das erlaubt es Slack als eine Art zentralen Informations-Hub zu einrichten, in dem sich die Entwickler eine schnelle Übersicht über den aktuellen Stand des Projektes verschaffen können.

2.3 Testing Framework

Ein Testing Framework bezeichnet ein Framework zum automatisierten Softwaretesting. Insbesondere in der agilen Softwareentwicklung mit häufigen Anspruchsänderungen und den einhergehenden Anpassungen einzelner Module, ist eine schnelle, fehlerfreie und automatisierte Funktionalitätsprüfung geeignet. Aber auch um zu überprüfen ob eine Anpassung nötig ist oder das Modul auch den geänderten Ansprüchen standhält kann Softwaretesting verwendet werden. Desweiteren haben somit auch mit dem jeweiligen Modul weniger vertraute Teammitglieder die Möglichkeit schnell verlässliche Antworten auf zahlreiche Fragestellungen zu erhalten. Beispielsweise ist es möglich zu überprüfen ob eine nachträgliche Detailänderung die Funktionalität eines Moduls beeinträchtigt. Aber auch während der Entwicklung erlaubt das Software Testing kontinuierliche Funktionalitätstests und damit eine frühzeitige, unkomplizierte und effiziente Fehlererkennung und Minimierung. Frameworks sind strukturell bestens für die Vereinfachung des Softwaretesting geeignet, weil sie das Definieren der Ein- und Ausgabeparameter in einer bestehenden und durchdachten Struktur erlauben und das eigentliche Testen der Software übernehmen können. Wichtige Parameter für ein solches Testing Framework sind gute Dokumentation, gute Bedienbarkeit insbesondere in Bezug auf die Nutzung mehrerer Parametersätze und präzise Fehlerauswertung. [12] [3]

2.4 Maven

Maven ist ein Build-Management Tool mit dem das Erstellen und Verwalten von Java Programmen standardisiert werden soll. Hierfür wird beispielsweise definiert wie .java Dateien zu .class Dateien kompiliert und diese dann zu .jar gepackt werden, welche pre-/post-processing Schritte durchgeführt werden oder wie der CLASSPATH geregelt wird. Maven versucht hierbei so viele Schritte wie möglich im Build-Lifecycle, also von der Validierung des Codes bis hin zum Deployment, zu automatisieren. Dabei müssen hauptsächlich nur die Dinge festgehalten oder definiert werden, die vom konventionellen Build Prozess abweichen. Maven versucht beispielsweise auch durch das automatische Downloaden und Updaten von Bibliotheken und Maven-Plugins in sich abgeschlossen zu bleiben, ohne weitere Tools oder Scripts zu benötigen. [11]

2.5 Jekyll

Jekyll ist ein in der Skriptsprache Ruby geschriebenes Programm, welches statische HTML-Webseiten mit Hilfe einfacher Textdateien, die häufig mit Markdown geschrieben sind, generiert.

Das Programm findet vor allem bei Blogs oder kleineren Webseiten seine Anwendung. Die Vorteile liegen in der einfachen Installation und Verwendbarkeit, der Sicherheit und der weiten Verbreitung (unter anderem bei Github-Pages). Es benötigt außerdem keine weiteren Dienste wie bspw. einen Datenbankserver (im Gegensatz zu Alternativen wie WordPress), ist also allein lauffähig. Darüber hinaus sind Webseiten, die mit Jekyll erstellt wurden, leicht zu warten und zu erweitern.[9]

2.6 gitflow

Gitflow ist ein Konzept für einen Arbeitsablauf mit Git. Dabei wird vorallem die Geschichte sehr gut strukturiert und bleibt Übersichtlich auch für Aussenstehende.

Das Konzept basiert auf zwei Hauptverzweigungen, zum einem dem Master-Branch zum anderen dem Develop-Branch, und kleinen Feature- und Release-Branches. Auf den Master-Branch werden dabei nur fertige Releases aus den Release-Branches committed. Der Develop-Branch ist der Zweig für die Entwicklung der einzelnen Releases. In ihn werden die einzelnen Feature-Branches merged. Feature Branches sind dabei kleine Zweige, welche für jedes zu implementierende Feature erstellt werden. Sie werden in den Develop-Branch merged, wenn das Feature fertig implementiert ist. Ein Release-branch wird immer dann aus Develop eröffnet, wenn ein Release ansteht und genug Features im Develop-branch implementiert sind. In ihm werden keine neuen Features mehr hinzugefügt, sondern nur noch Bug-Fixes oder Dokumentation.

Dieses Vorgehen bietet vorallem viel Sicherheit und Struktur. [5]

2.7 gitlab

GitLab ist eine Webanwendung zur Versionsverwaltung für Softwareprojekte auf Git-Basis und bietet sowohl diverse Management und Bug-Tracking-Features, als auch ein System für kontinuierliche Integration. GitLab ist dabei offen für viele verschiedene Arten der Softwareentwicklung (Waterfall, Agile Development,...), da die Management-tools wie z.B. das Issue-Board sehr frei einsetzbar sind. Weitere wichtige Planungs-/Management-Tools sind z.B. Time-Tracking, Milestones, Issues und die Zuweisung zu spezifischen Personen. Auch die Entwicklung an sich gestaltet sich sehr komfortabel, vor allem durch die visuelle Representation von Branches, Merge Requests uvm. Des Weiteren lässt sich sehr klar definieren, wie der Arbeitsablauf aussieht, z.B. Sperrung der Push-Funktion auf den Master-Branch. Somit kann sicher gestellt werden, dass keine Fehler entstehen und sich jeder auf die eigentliche Aufgabe konzentrieren kann, anstatt sich mit dem Management des Projektes zu beschäftigen. [6]

3 Konzepte

3.1 Digitalisierung von Arbeitsabläufen

In einem sich rapide weiterentwickelnden Informations- und Digitalisierungszeitalter möchte jeder innerhalb von Sekundenbruchteilen auf Nachrichten und Daten von überall aus zugreifen können. Der Umstieg von Papier zu Kupferdraht und Glasfaser ermöglicht genau das. Dies ist auch von hoher Relevanz für das technische Hilfswerk, da Informationen wie der Vierfachvordruck nicht manuell von einer Person zur anderen transportiert werden müssen, sondern digital ein Formular ausgefüllt wird, welches dann über das Internet verschickt wird und eine Sekunde später beim Empfänger ausgedruckt werden kann. Dadurch wird nicht nur Schnelligkeit, sondern auch eine gewisse Sicherheit gewährleistet. Papier ist leichter zu verlieren bzw. zu zerstören als digitale Daten. Außerdem sind Entfernungen absolut kein Problem beim digitalen Versenden, der Zeitunterschied von 20km zu 200 km beträgt ca. 1 Sekunde, während beim manuellen Verschicken zwischen 30 Minuten und 3 Stunden unterschieden werden muss.

3.2 Multiview

Unter Multiview versteht man verschiedene auf den Nutzer zugeschnittene Sichten des gleichen Datensatzes. Die virtuelle Representation des beim THW verwendeten Vierfachvordrucks(VV) bietet sich für die Verwendung mehrerer Sichten ideal an. Durch eine individuell Sicht können relevante Informationen schneller erfasst werden, es gibt keinen Informationsüberschuss, aber die gewohnte Formatierung insbesondere Struktur und Farbe können beibehalten werden. So ist es für Sachbearbeiter, Funker und Nachweisausdruck nötig Sicht auf den gesamten Datensatz zu

haben. Der Sichter hingegen erhält nur den zu Prüfenden Inhalt. Der Ausdruck am Lageplan dient wiederum zum Erfassen der Gesamtsituation und benötigt keine Information über die Art der Übermittlung.

3.3 Fachgruppe Führung und Kommunikation

Die Fachgruppe Führung und Kommunikation (kurz: FGr FK) des THW dient der Koordinierung mehrerer Einheiten (ab 2-3 technische Züge) bei größeren Schadensereignissen. Sie wird eingesetzt, wenn es sinnvoll ist, einen Einsatzabschnitt ganz an das THW zu übertragen. Die Fachgruppe besteht aus Führungstrupp, Fernmeldetrupp und Führungs- und Kommunikationstrupp und wird in einer mobilen Einsatzzentrale aus THW-Fahrzeugen untergebracht. Für sie sind 5 Führer (darunter ein Gruppenleiter), 5 Unterführer und 8 Helfer vorgesehen. Der 4-Fach-Vordruck wird zur internen Kommunikation und Protokollierung verwendet.

3.4 THW Mobile Zentrale

Die Mobile Zentrale ist ein durch ausklappen erweiterbares Fahrzeug mit dem das THW sich an Einsatzorten temporär stationieren kann. In der Zentrale sind sich Fernmeldezentrale, Lagekarte und andere grundlegende Arbeitsplätze enthalten. Dadurch kann die Kommunikation und Organisation innerhalb des Einsatzortes und "nach außen" zu anderen Zentralen gewährleistet werden.

3.4.1 Vierfach-Vordruck

Der Vierfach-Vordruck ist ein Dokument, was essentiell für die Kommunikation zweier räumlich getrennter Einrichtungen des THWs ist. Er besteht aus dem selben Dokument in vier Farben, die während des Workflows an unterschiedliche Orte und somit Mitarbeiter kommen, um dort zu einem Überblick der Gesamtsituation zu verhelfen. Auf dem Dokument werden verschiedene Daten notiert, die die Progression des Einsatzes möglichst zuverlässig gewährleisten sollen. Der mit dem Vierfach-Druck verbundene Workflow lässt sich in zwei phasenähnliche Abläufe innerhalb je einem Einheitsstandortes untergliedern - den Eingang und den Ausgang. Der wesentliche Ablauf ist gleich, jedoch unterscheiden sie sich in geringen Merkmalen. So wird bereits im Kopf des Dokuments vermerkt, ob es sich um eine eingegangene oder ausgegangene Nachricht handelt und über welchen Kommunikationsweg die Nachricht geliefert wurde beziehungsweise wurde. Für die Organisation wird eben da auch vermerkt, wer der "Konterpart" der Nachricht ist, bezeichnet als "Rufname der Gegenstelle". Handelt es sich um den Ursprung der Nachricht, spricht den Ort des Ausgangs so wird nach vollständigen Erfassen der Nachricht Blätter Zwei bis Vier an den Sichter zur Qualitätskontrolle weitergeleitet, während Blatt Eins (Blau) bei dem Verfasser verbleibt. Der Sichter kontrolliert das erfasste Dokument auf Lesbarkeit und Verständlichkeit. Wird die Nachricht als unzureichend eingestuft, so muss der Verfasser seine Nachricht erneut formulieren. Sobald der Sichter sein okay gibt, werden die verbleibenden Blätter an die Fernmeldezentrale (FMZ) weitergeleitet, wo die Nachricht an den Empfänger vermittelt wird. Dies geschieht durch einen Unterführer des Führungs- und Kommunikationstrupps, den Leiter des Fernmeldebetriebs (LdF). Nach erfolgreichem Versenden der Nachricht wird Blatt Vier (Gelb) in das Technische Betriebsbuch abgeheftet. Im technischen Betriebsbuch werden chronologisch jegliche Nachweise abgeheftet, sodass im Fall von Komplikationen oder Fehlern analysiert werden kann, wo der Fehler unterlaufen ist. Während Blatt Drei (Rot) zu Sachgebiet 2 Lage weitergeleitet wird, um dort an die Lagekarte geheftet zu werden, wird Blatt Zwei (Grün) an den Ltr.Stab geleitet. Die Lagekarte dient dazu allgemein den Überblick des Einsatzes zu behalten um jegliches weiteres Verfahren im Einsatz zu planen. Der Ltr.Stab hat hingegen Aufgaben wie beispielsweise das Einsatztagebuch zu führen. Wie bereits erwähnt, unterscheidet sich der Vorgang des Eingangs der Nachricht nur geringfügig zu dem Ausgang. Zuerst wird die Nachricht in der FMZ des Zielortes

vom LdF aufgenommen. Die abgefasste Nachricht wird auch hier auf die Qualität durch einen Sichter geprüft. Wird das Dokument als akzeptabel eingestuft, so wird Blatt Vier (Gelb) in dem Technischen Betriebsbuch vor Ort abgeheftet. Ebenfalls wird auch hier Blatt Drei (Rot) an die sich dort befindende Lagekarte geheftet. Was eindeutig unterschiedlich ist, ist das Blatt Eins (Blau) an den Ltr. Stab weitergeleitet wird und Blatt 2 (Grün) an einen Bearbeiter. Über den Bearbeiter wird dann die eingegangene Nachricht ausgewertet.

Literatur

- [1] Continuous integration. <https://www.martinfowler.com/articles/continuousIntegration.html>. Besucht 16.11.2017.
- [2] Deployment. https://en.wikipedia.org/wiki/Deployment_flowchart. Besucht 21.11.2017.
- [3] Framework. <https://de.wikipedia.org/wiki/Framework>. Besucht 19.11.2017.
- [4] Git. <https://de.wikipedia.org/wiki/Git>. Besucht 24.11.2017.
- [5] Gitflow. <https://www.atlassian.com/git/tutorials/comparing-workflows/gitflow-workflow>. Besucht 24.11.2017.
- [6] Gitlab. <https://de.wikipedia.org/wiki/GitLab>. Besucht 24.11.2017.
- [7] Gitlab ci. <https://about.gitlab.com/features/gitlab-ci-cd/>. Besucht 16.11.2017.
- [8] Java. [https://en.wikipedia.org/wiki/Java_\(programming_language\)](https://en.wikipedia.org/wiki/Java_(programming_language)). Besucht 21.11.2017.
- [9] Jekyll. <https://jekyllrb.com/>. Besucht 18.11.2017.
- [10] Markdown. markdown.de. Besucht 18.11.2017.
- [11] Maven. https://en.wikipedia.org/wiki/Apache_Maven. Besucht 21.11.2017.
- [12] Modultest. <https://de.wikipedia.org/wiki/Modultest>. Besucht 18.11.2017.
- [13] Mvc. <https://en.wikipedia.org/wiki/Model-view-controller>. Besucht 21.11.2017.
- [14] Softwaretest. <https://de.wikipedia.org/wiki/Softwaretest>. Besucht 18.11.2017.