

# 1. Studienprojekt

## Front- und Backend einer Webanwendung für das Requirements-Engineering

**Studienjahrgang:** 2013  
**Fachbereich:** Duales Studium Wirtschaft - Technik  
**Studiengang:** Informatik  
**Modul:** IT3161 Studienprojekt I  
**Betreuer Hochschule:** Prof. Dr. Dagmar Monett Diaz  
**Anzahl der Wörter:** ?

### Gemeinschaftsprojekt von:

**Name, Vorname:** Kujat, Sven-Erik  
**Matrikelnummer:** 683766  
**Ausbildungsbetrieb:** etomer GmbH

**Name, Vorname:** Hartmann, Marvin  
**Matrikelnummer:** 653861  
**Ausbildungsbetrieb:** MSA - The Safety Company

# Inhaltsverzeichnis

<b>Begriffserklärung</b>	<b>II</b>
<b>1 Einleitung</b>	<b>1</b>
1.1 Allgemeine Einführung . . . . .	1
1.2 Entwicklungsprozedur . . . . .	2
<b>2 Beschreibung der Aufgabenstellung</b>	<b>3</b>
2.1 Aufgabenstellung . . . . .	3
2.2 Rupp-Schablone . . . . .	3
2.3 Erweiterungen . . . . .	4
<b>3 Motivation</b>	<b>5</b>
<b>4 Entwicklung</b>	<b>6</b>
4.1 Konzept . . . . .	7
4.2 Entwicklungsplanung . . . . .	10
4.3 Realisierung . . . . .	12
4.4 Entwicklungsprozess . . . . .	15
4.5 Einbau von Funktionalitäten . . . . .	16
4.6 Review . . . . .	18
<b>5 Webanwendung <i>Red:wire</i></b>	<b>18</b>
5.1 Oberfläche . . . . .	19
5.2 Features . . . . .	24
<b>6 Ausblick und Fazit</b>	<b>26</b>
<b>Abbildungsverzeichnis</b>	<b>III</b>
<b>Literaturverzeichnis</b>	<b>A- 1</b>
<b>Anhang</b>	<b>A- 2</b>
6.1 Grafiken . . . . .	A- 2

# Begriffserklärung

<i><b>Begriffe</b></i>	<i><b>Definition</b></i>
HTML	Struktursprache für Webdokumente
OS	Betriebssystem
XAMPP	Multi OS, Apache-MySQL-Perl-PHP
LAMP	Linux-Apache-MySQL-PHP
MySQL	Relationales Datenbanksystem
PHP	Skriptsprache für Dynamisierung eines Webdokuments
HTTP	Hypertext Transfer Protocol
CSS	Cascading Style Sheets
DOM	Schnittstellenspezifikation für Zugriff auf HTML-Dokumente
JQuery	JavaScript erweiterndes Framework
Bootstrap	CSS erweiterndes Framework

<i><b>Abkürzungen</b></i>	<i><b>Definition</b></i>
z.B.	zum Beispiel
Vgl.	Vergleiche
u. a.	unter anderem
s. a.	siehe auch
s.	siehe
S.	Seite

# 1 Einleitung

## 1.1 Allgemeine Einführung

Bei der Entwicklung eines neuen Produktes, sollte sich der Auftraggeber zunächst überlegen, „Warum“ dieses für Kunden interessant wäre. Daraufgehend ist zu entscheiden, „Wie“ es entwickelt werden soll. Zuletzt muss exakt beschrieben werden, „Was“ für ein Produkt entstehen soll und welche Funktionalitäten es aufzuweisen hat.

Dies ist eine kurz gefasste Beschreibung, aber ein Unternehmer sollte im Optimalfall mittels dieses Paradigmas ein Produkt entwickeln. Letzteres Schema sei angelehnt an den *Golden Circle* des Requirements Engineering<sup>1</sup> von innen nach außen. Fälschlicherweise wird häufig mit dem „Was“ begonnen und am Ende der Entwicklung ein Zusatznutzen gesucht, um Konsumenten das Produkt attraktiv zu gestalten. Dadurch kann es vorkommen, dass teilweise unrealistische Anforderungen an das Produkt übereilt und unprofessionell vertraglich festgehalten werden. Dies sollte nicht geschehen, da gute Anforderungen ein Teil des Garant für einen erfolgreichen Projektabschluss sind. Wenn sie nicht gewissenhaft und strukturiert er- sowie bearbeitet werden, können daraus viele schwerwiegende Probleme und Folgefehler entstehen, welche bis zur Produktverfehlung und / oder Fehlkonstruktion führen können.

Im schlimmsten Fall entsteht solch eine Produktverfehlung. Das Produkt ist ein anderes, als von den Stakeholdern gefordert. Möglicherweise wird das Projekt abgebrochen. Auch kann es vorkommen, dass das Produkt zum Zeitpunkt der Fertigstellung bereits veraltet ist. Dabei kann schon zu Beginn ein Fehler bei der Anforderungsaufnahme genügen, um das gesamte Projekt nachhaltig zu gefährden, oder scheitern zu lassen. Dieser Fehler kann seinen Ursprung allerdings an verschiedenen Stellen haben. Mögliche Verfehlungen in der Aufnahme sind z.B. unvollständige, falsch interpretierte oder unrealistische Anforderungen, sodass die nächsthöhere Prozessinstanz diese nicht exakt deuten, oder durchführen kann.

---

<sup>1</sup>Vgl. Schenkel, Michael (2015) 3. Abschnitt im Hauptframe (siehe Literaturverzeichnis [3]).

## 1.2 Entwicklungsprozedur

Zur Verbesserung der Anforderungsqualität existieren folgende Grundsätze<sup>2</sup>:

Anforderungen müssen...

- ...vollständig sein
- ...korrekt sein
- ...prüfbar sein
- ...eindeutig sein
- ...verstehbar sein
- ...gültig und aktuell sein
- ...realisierbar sein
- ...notwendig sein
- ...verfolgbar sein
- ...bewertet sein

Eine weitere Möglichkeit um den Entwicklungsprozess zu verbessern und Fehlern entgegen zu wirken ist die agile Softwareentwicklung. Bei dieser müssen nicht direkt am Anfang alle Anforderungen in einem Pflichtenheft festgehalten werden, sondern sind dynamisch änderbar in einem Backlog festgehalten. Je nach genutztem System kann beispielsweise monatlich ein Review der bisherigen Arbeit geschehen und anschließend eine neue Liste von Anforderungen für die folgende Entwicklungsphase erstellt werden (Beispiel Scrum<sup>3</sup>). Ein solches System erhöht außerdem die Chance, dass mögliche Fehler frühzeitig erkannt und behoben werden. Diese Methode ist absolut effizient, da die Kosten um einen Fehler zu beheben von der Analyse und Design Phase (ca. 400 €), über die Test- und Abnahme-phase (ca. 4800 €), bis hin zur Veränderung einer bereits in Betrieb befindlichen Software (ca. 9000 €) exponentiell zunehmen. Letzteres kann aus dem Blickwinkel der Stakeholder und Konsumenten eine Abnahme der Produkt- und Firmenakzeptanz („Imageschäden“) bewirken<sup>4</sup> (s. a. Anhang Abbildung 6.1).

Es ist also zu sehen, dass sich die Wahl eines Entwicklungsprozesses in Zusammenhang mit dem Umgang mit den Anforderungen entscheidend auf das Projekt auswirken kann. Um den Umgang mit Anforderungen effizienter zu gestalten, sowie das Risiko eines Projektabbruchs zu minimieren, soll eine Software entwickelt werden, welche sowohl bei der Aufnahme, als auch der Verwaltung von Anforderungen Unterstützung bietet.

---

<sup>2</sup>Vgl. Rupp, Chris & Brenner, Matthias (2011) 2. Abschnitt (siehe Literaturverzeichnis [2]).

<sup>3</sup>Siehe it-agile GmbH, Scrum & The Scrum Guide (2013) (siehe Literaturverzeichnis [9] & [5]).

<sup>4</sup>Vgl. Schenkel, Michael (2014) 3. Abschnitt im Hauptframe (siehe Literaturverzeichnis [4]).

## 2 Beschreibung der Aufgabenstellung

### 2.1 Aufgabenstellung

Die Aufgabenstellung des hier beschriebenen Studienprojektes sieht vor, eine Online-Applikation bzw. Webapplikation zur Anforderungsaufnahme zu entwickeln. Anforderungen sollen mithilfe der „Rupp-Schablone“ von Chris Rupp formuliert werden. Die Idee hinter dem Online-Applikationskonzept ist die einfache Möglichkeit des zentralen Managements der Software und der guten Zugänglichkeit für mehrere Anwender auf die gleiche Datenbasis.

Als Hilfestellung zur Bewältigung dieses Projekts liegt eine Kopie des vorangegangenen Studienprojektes vor. Dieses behandelt die Aufnahme von Anforderungen auf Basis der „Rupp-Schablone“ in einer Offline-Applikation und steht zu Analyse des softwareseitigen Aufnahmeprozesses zur Verfügung.

### 2.2 Rupp-Schablone

Die „Rupp-Schablone“ dient der vereinfachten Aufnahme komplexer Anforderungen. Sie nutzt dazu nur eine Teilmenge der natürlichen Sprache. Genutzt wird die Schnittmenge, welche gleichermaßen Effizienz in der Bearbeitung, sowie Verständlichkeit für den Anwender darstellt.



Abbildung 2.1: Schnittmenge

Der Aufbau der Schablone ist an den deutschen Satzbau angepasst. Dies kann mit einer eingeschränkten Form der Subjekt-Prädikat-Objekt Logik betitelt werden. Das erste Element des Aufbaus ist ein Subjekt. Dieses Subjekt stellt den Satzgegenstand dar. Folgend wird die Wichtigkeit der auszuführenden Tätigkeit auf Basis eines helfenden Prädikats festgelegt. Dieses wird von Rupp als rechtlich bindend bezeichnet (auf Vertragsbasis). Das dritte Element ist ein optionales Dativ-Objekt zur Bestimmung des Empfängers bzw. des Nutznießers der Tätigkeit. Mit dem vierten Element wird zwischen einer selbstständigen oder einer nutzerinitiierten Durchführung unterschieden. An fünfter Stelle steht das vom Subjekt zu behandelnde Objekt und abschließend das eigentlich auszuführende Prädikat. Das Prädikat beschreibt den Umgang bzw. die Beeinflussung des Objekts durch das Subjekt.

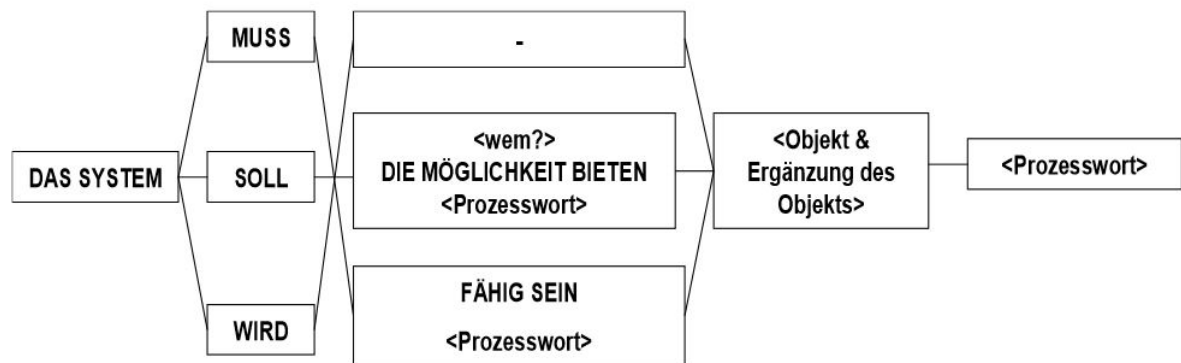


Abbildung 2.2: Schablone

## 2.3 Erweiterungen

Als wünschenswerte Entwicklungen für die Webapplikation seien im Gegensatz zur derzeitigen offline Lösung des vorigen Studienprojekts genannt: ein Organisationssystem zur Zusammenarbeit mehrerer Anwender an einem Anforderungsprojekt, ein Loginsystem zur Authentifizierung der einzelnen Anwender, eine Exportfunktion für die Anforderungen zur Weiterbearbeitung offline und einige Zusatzattribute zu jeder Anforderung (Priorität, Status, Identifikationsnummer, Abhängigkeiten).

### 3 Motivation

Der Leitgedanke, die Formulierung und Verwaltung von Anforderungen zu vereinfachen und effizienter zu gestalten, bietet die Motivation zur Entwicklung einer Software. Diese Software muss zwangsweise einen gleichwertigen, oder höheren qualitativen Output innerhalb einer Planungsphase generieren (in Relation zu Tabellen-, oder Textsoftware), um als effizient bezeichnet zu werden. Wenn Verantwortliche unachtsam oder ungeübt mit Anforderungen agieren, führt dieser suboptimale Umgang zu Qualitätsverlusten. Es erfordert explizit aufgeklärte, oder geschulte Mitarbeiter, um Anforderungen manuell ohne Softwareunterstützung abzulegen. Hierbei wird weder die Formulierung kontrolliert, noch eine konkrete Struktur gewährleistet. Infolgedessen entstehen Redundanzen und Uneindeutigkeit. Dies bedeutet einen zusätzlichen Verwaltungsaufwand, welcher die Kosten und Geschwindigkeit der eigentlichen Produktentwicklung negativ beeinflusst. Um jenen Problemen entgegenzuwirken, wird eine Onlineapplikation konzipiert, welche weniger risikoanfälliges Anforderungsmanagement ermöglicht. Während des Entwicklungsprozesses existiert die Möglichkeit, individuelle Stakeholderwünsche als Zusatzfunktionen zu implementieren, sollten genügend zeitliche Ressourcen verbleiben.

Infolge des Aufbaus als Webanwendung können Anwender jederzeit über einen gängigen Webbrowser auf ihre Anforderungen zugreifen. Aufgrund der steigenden Anzahl von Anwendern, welche über mobile Endgeräte auf Applikationen zugreifen, soll es zudem ermöglicht werden, unterwegs komfortabel über das Endgerät zu arbeiten. Dabei kann die Nutzererfahrung durch ein responsives (dynamisches) Design weiter verbessert werden.

Parallel kann die Software neben dem Einsatz in Projekten als Lernsoftware für die korrekte Anforderungsformulierung fungieren. Als Grundbaustein existiert hierbei eine integrierte Formulierungsschablone nach Chris Rupp. Jene soll dem Anwender die Formulierung durch einen vorgegebenen Satzbau erleichtern.

Außerdem soll es möglich werden, mit mehreren Anwendern zeitgleich an derselben Anforderungsliste zu arbeiten. Durch die resultierende Teamarbeit wird die Effizienz gesteigert und das Arbeitsklima verbessert. Nach Abschluss des Entwicklungsprozesses besteht die Möglichkeit, dieses Projekt unter einer Open Source - Lizenz zu veröffentlichen, um individuelle Weiterarbeit Dritter zu ermöglichen.



## 4 Entwicklung

Die Entwicklung der Webapplikation wird in mehreren Schritten durchgeführt. Vorangeht ist eine ganzheitliche Planung und Strukturierung für den erfolgreichen Abschluss des Projekts unabdinglich. Vorausschauend muss die vorgegebene Zeitspanne zur Erfüllung der Aufgabe eingeteilt und unterteilt werden. Es muss ein Konzept und ein Leitfaden entstehen, auf welchem die Entwicklung basiert und durchgeführt wird. Weiterhin müssen Überlegungen zu möglichen Technologien und deren Vor- und Nachteile getätigt werden. Sobald diese Grundbausteine ausgearbeitet sind, ist es wichtig, eine funktionierende Infrastruktur zur Softwareentwicklung zu entwerfen. Innerhalb dieser muss auch die Lauffähigkeit der zu entwickelnden Applikation gewährleistet sein. Damit kann schon jetzt später möglicherweise auftretenden Fehlern vorgebeugt, bzw. diese beschränkt werden. Nach der darauffolgenden Erstellung der Aufgaben werden diese aufgeteilt, um paralleles Arbeiten effizient zu ermöglichen.

Im Folgenden wird das zu erarbeitende Konzept erläutert. Dazu wird die Funktionsweise einiger exemplarischer Websites evaluiert. Dies dient des besseren Verständnisses für den Aufbau / das Layout und die aktuell gängigen Bedienkonzepte erfolgreicher, bzw. populärer Website-Systeme.

## 4.1 Konzept

Zur Ermittlung eines Konzepts, werden zunächst Arbeitsweisen anderer Internetseiten und Webpräsenzen betrachtet. Dabei ist deutlich zu erkennen, dass das Internet und die Internetseiten zunehmend komplexer werden. Vergleicht man ältere Versionen einer Website, mit deren Version von heute, werden viele Unterschiede ersichtlich, sowohl optisch als auch technisch. Als auffällige Änderungen seien hier beispielsweise das Layout, die Schrift und die Menüstruktur zu nennen. Dies wird deutlich im Vergleich der Abbildungen 4.1 und 4.2.

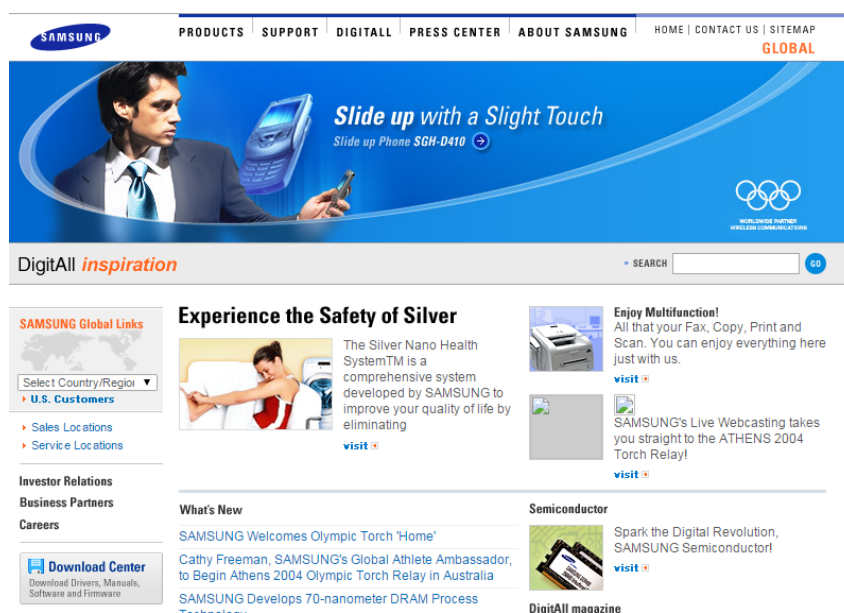


Abbildung 4.1: Internetauftritt Samsung 2004 (Quelle: Screenshot Wayback Machine: „www.samsung.com“)

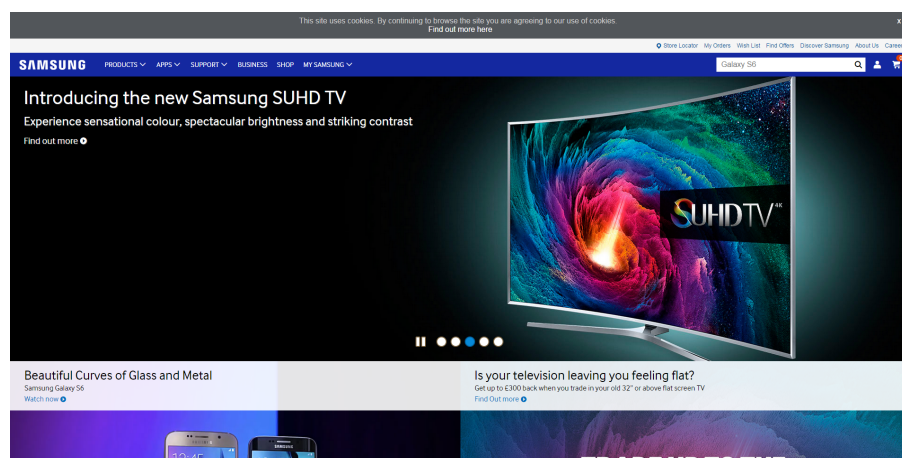


Abbildung 4.2: Internetauftritt Samsung 2015 (Quelle: Screenshot „www.samsung.com“)

Im Bereich der Vernetzung, speziell im Web-Bereich, gibt es stetige Weiterentwicklungen. Folgend an das Moor'sche Gesetz angelehnt existieren immer kürzere Technologie-Lebenszyklen. Dieser Fakt sowie ein verändertes Nutzungsverhalten verringern die Attraktivität statischer Websites. Aktuell spielt die Verbreitung von mobilen Endgeräten und steigende mobile Datennutzung eine größer werdende Rolle.

Für Suchanfragen, die von mobilen Endgeräten über die Internet-Suchmaschine „Google“ durchgeführt werden, wurde bereits im Juni 2013 eine Anpassung des Ranking-Algorithmus bekannt gegeben<sup>1</sup>. Am 26.02.2015 kündigte der Konzern an, das „mobile-friendly“ Ranking für Websites mit in die Gesamtbewertung dieser Suchanfragen einfließen zu lassen<sup>2</sup>. Diese Änderung ist planmäßig seit dem 21.04.2015 aktiv<sup>3</sup>. „Google“ unterstützt mittels dieses Schrittes den technischen Wandel, welcher die Denkweise der Webentwickler nachhaltig beeinflusst.

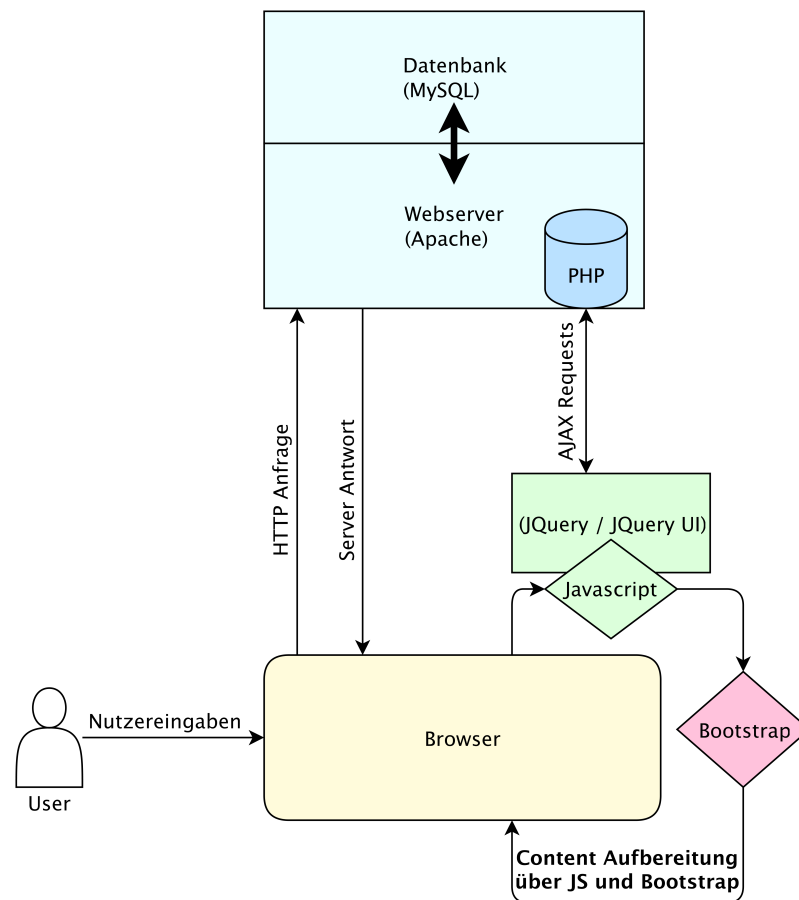
Letzteres sowie das veränderte Konsumverhalten der Technik-Anwender verdeutlichen, dass derzeit eine hohe technische und konsumangepasste Versiertheit der Entwickler nötig ist. Infolgedessen werden bei der Entwicklung der webbasierten Software in diesem Projekt verschiedene Technologien miteinander verbunden. Das spiegelt die tiefergehende Evaluation der Anwenderbedürfnisse wider und fördert sowie verstärkt ein positives Nutzungserlebnis. Eine Auflistung der genutzten Tools, Sprachen sowie Frameworks finden sich unter Punkt 4.3 („Realisierung“). Nach der Überprüfung unterschiedlicher Konzeptionen wird nachfolgendes Konzept festgehalten:

---

<sup>1</sup>Vgl. Google Webmaster Central (11. Juni 2013) (siehe Literaturverzeichnis [8]).

<sup>2</sup>Vgl. Google Webmaster Central (26. Februar 2015) (siehe Literaturverzeichnis [6]).

<sup>3</sup>Vgl. Google Webmaster Central (21. April 2015) (siehe Literaturverzeichnis [7]).

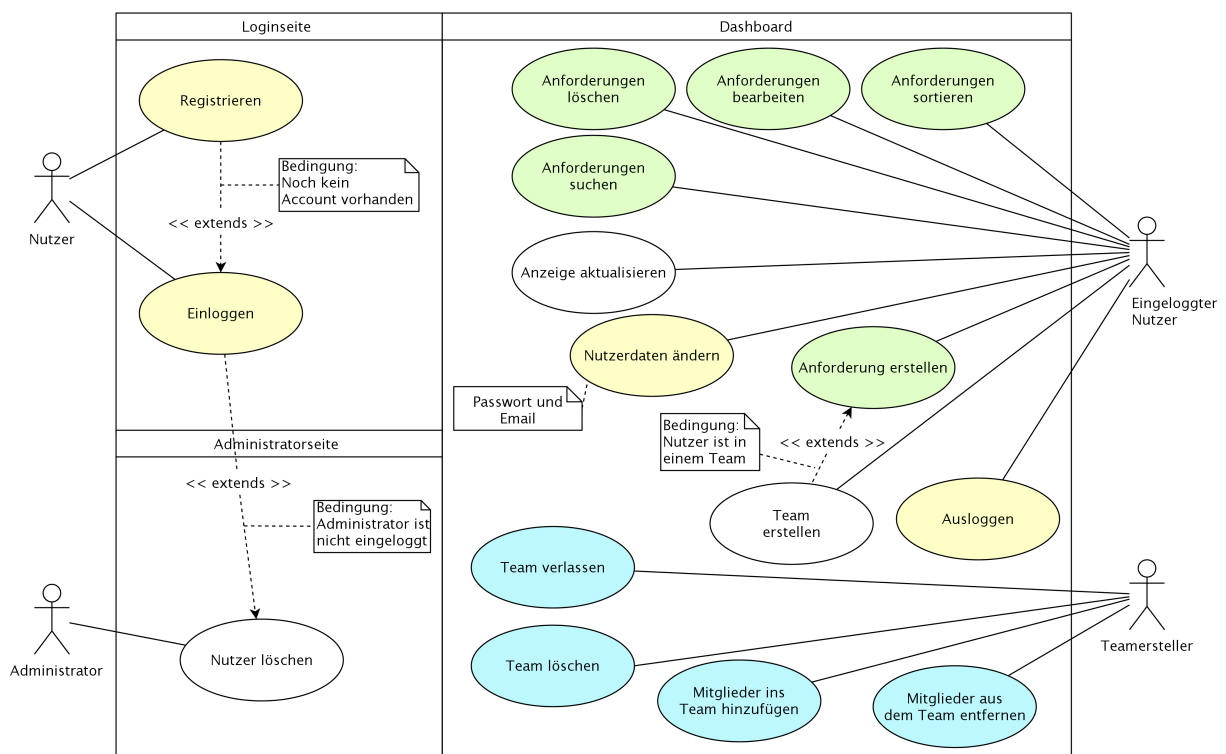


**Abbildung 4.3:** Webanwendung Konzept

Die Anwendereingaben werden vom Browser erfasst. Zunächst schickt dieser einen standard HTTP-Request an den Server. In der Antwort sind anschließend HTML-, CSS- und Scriptdateien enthalten. Diese benutzt der Browser, um den Inhalt aufzubereiten und die Darstellung anzupassen. Somit werden wichtige Funktionen und auch die Darstellung auf mobilen Endgeräten ermöglicht. Weiterhin kann der Funktionsumfang der Website mittels jQuery und Bootstrap nach Belieben erweitert werden. Dem Anwender werden mehrere Funktionalitäten bereitgestellt. Einen Überblick über diese gibt die Abbildung 4.4. Neben dem Registrieren und Einloggen, wird sich der Anwender nach Beendigung seiner Arbeits-Session vom System abmelden können.

Nach der Anmeldung befindet sich der Anwender auf einer Übersichtsseite, genannt „Dashboard“, von welcher aus ihm weitere Interaktionsmöglichkeiten mit dem System bereitgestellt werden. Als Interaktionen bezüglich der Anforderungen existieren das Verfassen sowie Bearbeiten von Anforderungen. Zudem lässt sich die Ansicht individuell an die Anwenderwünsche mittels Zeilensortierungen anpassen. Entsprechende Aktionen sind in Abbildung 4.4 grün markiert.

Alle Anforderungen werden dem Team des Verfassers zugewiesen. Sie sind somit an dieses gebunden und durch alle Mitglieder einsehbar. Alle Mitglieder eines Teams besitzen das Recht, dieses zu bearbeiten. Entsprechende Interaktionen diesbezüglich sind in Abbildung 4.4 blau markiert. Ein Anwender kann zudem seine Nutzerdaten ändern. Dies bezieht sich auf das Passwort und die eingetragene E-Mail Adresse. Diese Interaktionen sind in der Abbildung 4.4 gelb markiert. Um das System zu Warten, besitzt ein Administrator die Möglichkeiten, Nutzer zu entfernen sowie Teams anderen Nutzern zu überschreiben.



**Abbildung 4.4:** Use-Case Diagramm Nutzeraktionen

## 4.2 Entwicklungsplanung

Folgende Meilensteine sind für den Entwicklungszeitraum geplant: das Aufsetzen eines Datenbanksystems sowie aller nötigen Abhängigkeiten. Folgend muss ein funktionsfähiges Authentifizierungssystem für die Anwender implementiert werden. Dies ist notwendig, da die Software direkt auf Basis dieses Multi-User-Systems weiter entwickelt werden soll. Um dieses Grundgerüst zu schaffen und möglichst viele Fehlerquellen schon während der Entwicklungsphase ausfindig zu machen und zu beheben, wird zunächst mit einem Zeitrahmen von etwa sechs Wochen eingeplant. Diese Zeitspanne dient als Orientierungspunkt

und wurde durch etwaige Aufwandsschätzungen der einzelnen Aufgaben festgelegt. Ebenso ist dieser Schritt als Meilenstein anzusehen.

### Zweiter Meilenstein

Nachdem der Test des Benutzersystems auf seine Funktionalitäten hin überprüft wurde, erfolgt die Implementierung der Anforderungsschablone sowie dessen notwendige Funktionen. Es muss gewährleistet sein, dass Anforderungen sowohl in die Datenbank geschrieben als auch ausgelesen werden. Dazu ist ein einzigartiger Separator notwendig, damit die Satzglieder der Anforderungen später rekonstruiert werden können. Auch muss es ermöglicht werden, dass Metainformationen wie z.B. die Identifikationsnummer des Erstellers, oder ein Zeitstempel für die Anforderungssortierung gespeichert werden können. Ist dieses Ziel erreicht muss ein geeignetes visuelles Ausgabeformat gewählt und die korrekte Darstellung der Anforderungen überprüft werden. Zusätzlich muss die Übersichtlichkeit sowie Individualisierbarkeit der Anforderungen gewährleistet sein. Um dies zu realisieren, sind Sortierfunktionen erforderlich. Nach Verifizierung der ordnungsgemäßen Funktion soll der zweite Meilenstein nach weiteren sieben bis acht Wochen erreicht sein.

### Dritter Meilenstein

Der dritte Meilenstein besteht aus dem Teamsystem. Jenes soll gleichzeitiges Arbeiten an derselben Anforderungsliste ermöglichen. in Team muss erstellt werden können, damit alle weiteren Implementierungen darauf aufbauen können. Das Verwalten von Teams, insbesondere das Hinzufügen und Löschen anderer Nutzer ist wird im Funktionalitätenumfang der Software enthalten sein, da ein funktionierendes Teamsystem anders nicht möglich ist. Besonders bei diesem Teil des Systems muss kontrolliert werden, dass der Anwender keine Fehler erzeugen kann. Vor allem betrifft dies das Hinzufügen von Nutzern und die Namenswahl der Teams. Um Komplikationen bei den Datenbankabfragen zu vermeiden, sollte sich jeder Nutzer nur in einem Team befinden. Die Teamnamen müssen zur korrekten Identifikation und zur Vermeidung von Redundanzen einzigartig sein.

Weiterhin beinhaltet die Erfüllung des dritten Meilensteins auch die Anpassung der Darstellung, bzw. Optik der Software. Dazu gehören beispielsweise Tooltips, um eine verständlichere Kommunikation und Interaktion zwischen Mensch und Maschine herbeizuführen und zu ermöglichen. Zu diesem Zweck sollen auch Dialoge und Komplexität der Menüführung kontrolliert werden. Zur Überprüfung der Intuitivität kann die Software von Dritten getestet werden. Aufgrund des steigenden Testumfanges, wird der Zeitaufwand hier höher geschätzt, als bei den vorherigen Meilensteinen. Sobald alle Tests diesbezüglich erfolgreich verlaufen sind, wird dieser Meilenstein abgeschlossen sein. Allerdings werden

hier weiterhin Anpassungen und eventuell Fehlerbehebungen notwendig sein, da in diesem zeitlichen Rahmen nicht alle Fälle der Benutzung simuliert werden können. Verbleibende Zeit wird genutzt, um Fehler zu beheben und die Software zu verbessern. Außerdem können weitere Ideen und Features evaluiert und implementiert werden, da die Entwicklung nicht auf einem Wasserfallmodell beruht, sondern an agile Softwareentwicklung angelehnt ist. Dies gibt die Möglichkeit, zusätzliche Anwenderwünsche während jeder Entwicklungsphase berücksichtigen zu können.

## 4.3 Realisierung

### Einleitung

Am Anfang der Realisierung stehen die Planung sowie die Auswahl der Technologien, Programmiersprachen, Tools bzw. Tool-Chain. Diese sollten sorgsam gewählt werden, da sie die gesamte Entwicklungsdauer über genutzt werden. Durch die Analyse der benötigten Architektur und Infrastruktur lassen sich die verfügbaren Tools und Programmiersprachen eingrenzen, welche mit der Umgebung kompatibel sind.

### Vision

Damit das Projekt erfolgreich abgeschlossen werden kann, ist es notwendig, sich zuerst eine Idee bzw. Vision des künftigen Produkts zu erschließen. Darunter fallen insbesondere die Funktionalitäten, das Design und die Bedienung. Zu diesen drei Kategorien werden alle Ideen dokumentiert. Durch Revision der einzelnen Vorschläge resultieren nun folgende Ergebnisse:

- Das Tool muss Anforderungen aus Eingabefeldern auslesen können
- Das Tool muss ein Authentifikationssystem bieten
- Das Tool muss ein Teamsystem bieten
- Das Tool muss ein Securitysystem bieten
- Das Tool muss Anforderungen exportieren können
- Das Tool muss das Bearbeiten vorhandener Anforderungen ermöglichen
- Das Design soll Aktualität widerspiegeln
- Das Design soll Simplizität darstellen
- Das Design soll Offenheit implizieren

- Die Bedienung soll intuitiv sein
- Die Bedienung soll einfach gehalten sein

Durch die exakte Umsetzung dieser Punkte sollte das Projekt erfolgreich abschließen. Durch Recherchen und Evaluationen bestimmter Techniken zum Erreichen positiver Nutzererfahrungen soll ein ansprechendes Design erstellt werden, welches den Anwender durch visuelles Feedback bei der Arbeit unterstützt.

### Architektur

Eine Webapplikation zeichnet sich dadurch aus, dass sie im Netzwerk arbeitet. Das bedeutet, dass sie für alle authentifizierten Anwender in diesem Netzwerk zugänglich sein muss. Aus Sicht des Systems ist der Anwender ein Client, welcher Datenpakete an den Server sendet und abfragt. Basierend auf dieser Tatsache wird abgeleitet, dass es hier um eine Client-Server Architektur handelt. Der Server ist deshalb ein Webserver, der die Dokumente für den Client bereitstellt und für Interaktionen mit dem Anwender bereit steht. Da Daten persistent gespeichert werden müssen, so dass über einen längeren Zeitraum mit ihnen gearbeitet werden kann, wird eine Datenbank benötigt. Diese beiden Komponenten (Webserver, Datenbank) sind für die Datenverarbeitung zuständig, während der Client (Browser) die verarbeiteten Daten empfängt und neue Daten / Anfragen an den Server sendet.

### Sprachen

Die oben beschriebene vorausgesetzte Umgebung lässt sich am einfachsten mittels XAMPP (Windows, OSX), oder LAMP (Linux) realisieren. XAMPP steht für Apache, MySQL, PHP, Perl und LAMP für Linux, Apache, MySQL, PHP. In beiden dieser Pakete sind jeweils schon ein Webserver (Apache) und eine Datenbank (MySQL, relational) enthalten. Damit die Datenbankanbindung gewährleistet ist, bieten diese Pakete PHP-Unterstützung mit an. Nur so lässt sich die Datenbank ansprechen und es können Daten ausgelesen und abgespeichert werden. Das Frontend wird dementsprechend mittels der Struktursprache HTML geschrieben. HTML erlaubt das Einbetten von (u. a.) PHP Code, falls die PHP-Unterstützung des Webserver aktiviert ist. Die Formatierung der HTML-Struktur wird durch CSS realisiert. Zur intuitiveren Bedienung der Webapplikation wird zusätzlich das JavaScript genutzt. Es erlaubt DOM-Manipulationen (Document Object Model) einfach durchzuführen.

### Frameworks



Frameworks stellen schon vorgefertigte Funktionen einer bestimmten Sprache zur Verfügung. In einem größeren Softwareprojekt lässt sich so ohne Mehraufwand, im Vergleich zur reinen Programmiersprache, ein funktionales Code-Grundgerüst bauen. Das Design der Website wird mittels CSS erstellt und setzt auf ein schon strukturiertes HTML-Dokument auf. Derzeit werden allerdings immer mehr Websites über mobile Endgeräte aufgerufen, welche kleinere Displays und/oder Auflösungen als aktuelle Home Computer besitzen. Um das Design trotzdem für alle Geräte zu optimieren, eignet sich CSS nativ nur bedingt. Deshalb wird das CSS-Framework „SASS“ genutzt, welches eine ähnliche Syntax besitzt, jedoch mit weniger Code auskommt. Außerdem wird eine Klassenbibliothek für CSS-Klassen genutzt. Diese heißt „Bootstrap“ und ist in der Webentwicklung weit verbreitet. Bootstrap bietet viele responsive Klasse und Layouts, die z.B. alte nicht-responsive Tabellen-Layouts sehr einfach ersetzen können. Viele Klassen besitzen zusätzliche Effekte bei Hover (Mauszeiger über Element), oder Klickanimationen. Für einige bestimmte Effekte wird weiterhin das von jQuery stammende „jQuery UI“ genutzt, insbesondere für Klickanimationen. JavaScript erlaubt DOM-Manipulationen. Das ermöglicht eine intuitivere Bedienung der grafischen Weboberfläche. Um diese Manipulationen, wie z.B. Fades und Slides einfacher implementieren zu können, bietet das Framework jQuery eine vorgefertigte Funktionsliste, welche nur aufgerufen werden muss.

### Tools

Um eine grundlegende Planung durchzuführen, wird das Projektmanagementtool „Trello“ genutzt. Dieses ermöglicht die Erstellung von Story Cards, teambasiertes Arbeiten an sogenannten „Boards“, Zuweisungen von Aufgaben an Teammitglieder und Sortieren aller Aufgaben. Das Tool ist webbasiert, kostenlos und kann unter [www.trello.com](http://www.trello.com) erreicht werden.

Zur Installation der Softwareumgebung wird das XAMPP-Paket unter Windows installiert. Dieses bietet einen Webserver, eine relationale Datenbank und PHP-Anbindung für den Webserver.

Das eigentliche Programmieren erfolgt in „Notepad++“. Ein Texteditor mit Syntax-Highlighting für viele verschiedene Sprachen. Genau diese Funktionalität macht es zu einem einfach nutzbaren und konfigurierbaren Entwicklungstool. Alle in dem Projekt genutzten Sprachen werden vom Highlighting unterstützt. Zum automatisierten Kompilieren von SASS „.scss“ Dateien in „.css“ Dateien wird das Command-Line Tool „Compass“ genutzt. Dieses besitzt eine eigene Ruby Command-Line, mit welcher die Kompilierung getriggert wird. Es kann außerdem einen Verzeichnispfad beobachten und kompiliert alle

dort befindlichen Dateien, sobald eine Änderung der Checksumme eben dieser wahrgenommen wird.

Für Debug-Zwecke dient die Konsole im Firefox-Browser. Über diese können beispielsweise Variablenwerte ausgelesen, Funktionsaufrufe durchgeführt und Elemente untersucht werden.

## 4.4 Entwicklungsprozess

### Datenbank

Um die Daten der Applikation zu speichern, wird eine Datenbank aufgesetzt. Diese enthält drei Tabellen. Eine Tabelle für die Nutzerinformationen, eine für die Anforderungsinformationen und eine für die Teaminformationen. Der Zusammenhang dieser besteht mittels Fremdschlüsselvergabe an jeweilige Attribute. Zur eindeutigen Identifikation eines Datensatzes werden Identifikationsnummern vergeben.

### Grundstruktur

Nach allen Vorbereitungen werden die manifestierten Ideen umgesetzt. Als Hauptelement dient ein HTML - Template. Das Template ist eigens für diese Applikation erstellt worden und umfasst die Seitenaufteilung in mehrere Bereiche (Layout). Alle Bereiche sind gekapselte Objekte und nicht voneinander abhängig, um eine optimale Modularität zu gewährleisten. Das Template besteht aus Kopf- sowie Fußbereich, aus Inhaltsbereich und aus links positionierter Menüleiste, ist zunächst rein statisch und dient zur Demonstration der Optik.

Für das Authentifizierungssystem wird ein weiteres HTML - Template erstellt, welches als Login-Seite genutzt wird. Auch dieses ist statisch und wird nach und nach im Funktionsumfang erweitert.

Für die Administrationsseite wird eine abgewandelte Form des Applikations-Haupttemplates verwendet, welches nur die notwendigen Funktionen zur Administration enthält, jedoch das gleiche Layout besitzt.

Mittels CSS werden die Seiten formatiert. Die Admin- und Hauptseite greifen auf die selbe CSS-Datei zu, um Einheitlichkeit zu behalten. Für die Loginseite wird eine eigene CSS-Datei erstellt.

## 4.5 Einbau von Funktionalitäten

### Loginseite

In die nun fertigen HTML-Seiten werden die Funktionalitäten einzeln eingebaut. Zuerst werden die Authentifizierungsfunktionen implementiert. Dazu zählen:

- Einlesen der Nutzerdaten (Name, Passwort)
- Abgleichen der Nutzerdaten mit den in der Datenbank hinterlegten
- Weiterleitung an die Hauptseite
- Fehlermeldung bei Falscheingabe

Zum Testen des korrekten Logins wird ein Datensatz manuell in die Datenbank eingetragen. Mit diesen eingetragenen Daten wird sich eingeloggt. Um jedem die Möglichkeit zu bieten, das Tool zu nutzen, wird außerdem eine Registrierungsmöglichkeit geboten. Diese erweitert den Funktionsumfang der Seite um:

- Einlesen neuer Daten
- Speicherung der Daten in der Datenbank

Diese Funktionen werden mit dem Registrieren eines neuen Nutzers und dem anschließenden Login mit dessen Daten getestet.

### Hauptseite

Die Applikations-Hauptseite muss einen sehr großen Umfang an Funktionalitäten bieten. Grundlegende Funktionen sind hier:

- Aufnahme von Anforderungen
- Darstellung von Anforderungen
- Bearbeiten / Löschen von Anforderungen
- Erstellen von Teams
- Beitreten in Teams / Austreten aus Teams
- Ändern der Profildaten
- Logout aus der Applikation

Zum Testen dieser Funktionen werden Anforderungen erstellt und ausgelesen. Sobald dieser Test erfolgreich ist, wird das Bearbeiten sowie Löschen der Funktionen getestet. Es wird ein Team erstellt, Nutzer diesem Team zugewiesen und das Team gelöscht. Weiterhin werden die Profildaten neu eingegeben und es wird sich aus der Applikation ausgeloggt.

### Adminseite

Die Adminseite ist nur dem Administrator zugänglich und bietet einen eingeschränkten Funktionsumfang. Es lassen sich lediglich Nutzer und/oder Teams löschen. Die Nutzerdaten des Admins können außerdem geändert werden.

Zu Testen ist dies, indem ein bestehender Nutzer gelöscht, sowie dessen Team an einen anderen Nutzer übertragen wird.

### Tests

Die im letzten Abschnitt definierten Testfälle werden nun durchgeführt. Währenddessen fällt auf, dass die ersten Testdurchführungen oft misslingen, da kleine Fehler übersehen wurden. Diese können jedoch einfach mittels der browserinternen Debugkonsole ausfindig gemacht und korrigiert werden. Schwerwiegendere Fehler, welche während der Testphasen aufgefallen sind:

- Die Möglichkeit von HTML Injections: in die Felder für die Anforderungsaufnahme können kurze JavaScript Codesequenzen eingefügt werden. Werden die Anforderungen aus der Datenbank geladen, wird der Code in die HTML-Struktur eingebettet und nach dem Laden der Seite ausgeführt.
- Beim Bearbeiten von Anforderungen werden die Anforderungen gelöscht. Das Bearbeiten besteht aus dem Löschen und darauffolgend dem neuen Einfügen einer Anforderung. Fehlerhaft war hier das erneute Einfügen einer Anforderung.
- Jedem Teammitglied ist es möglich, ein Team zu löschen, unabhängig davon, ob er der Ersteller ist, oder nicht.
- Der Administrator kann keine Nutzer löschen, da in der darunterliegenden Funktion keine Abhängigkeiten der Datenbank berücksichtigt werden.

Diese Fehler sollen einen kurzen Eindruck schaffen, wie solche Tests ablaufen und was bei einem Projekt fehlerhaft sein kann. Sie sind alle beseitigt worden, sodass das Tool mit vollem Funktionsumfang arbeiten kann.

## 4.6 Review

Rückblickend erwies sich die Entwicklung der Webapplikation nach einer umfangreichen Planungsphase über schätzungsweise sechs Wochen relativ unkompliziert. Da die Deadline für die Projektabgabe im August 2015 ist, konnte die Zeit gut aufgeteilt werden. So gab es über den gesamten Prozess keinen erhöhten Zeitdruck und alles geplante ließ sich problemlos implementieren.

Zum Ende der Entwicklungsphase gab es noch genügend Ressourcen, um zusätzliche Features zu implementieren (s. Abschnitt 5.2 „Features“). Diese Erweiterungen setzen auf die schon existierenden Funktionalitäten auf und erleichtern die Arbeit mit ihnen.

## 5 Webanwendung *Red:wire*

Die Webapplikation Red:Wire hat ihren Namen aus der Wichtigkeit der Planungsphase für ein erfolgreiches Projekt. Als zu deutsch „Roter Faden“ ziehen sich die Auswirkungen der Planung durch den ganzen Projektzeitraum. Des Weiteren besteht in diesem Wortspiel auch eine Ähnlichkeit zu dem englischen Verb „Require“ (dt. „Anfordern / Voraussetzen“), welches das Thema dieser Anwendung ist: Anforderungen aufnehmen. Aus diesem Grund bekommt das Tool eben diesen Namen.

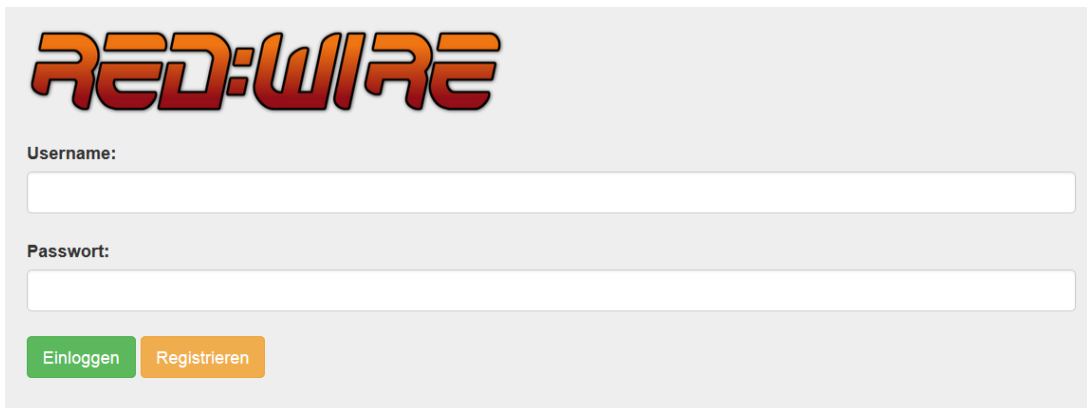
Die Voraussetzungen an den Server sind, wie Oben erklärt, zusammengefasst ein Webserver mit einer MySQL-Datenbank und PHP-Anbindung. Eine genaue Anleitung zur Installation der Webapplikation finden Sie im Anhang.

Die Voraussetzungen für den Client (Browser) sind aktiviertes JavaScript, da das Tool ohne die Möglichkeit der JavaScript-Ausführung nicht genutzt werden kann. Es kann sonst jeder aktuellere Browser (Internet Explorer 11+, Firefox, Chrome) zum Aufrufen der Applikation genutzt werden.

Die Gestaltung sowie die Funktionalitäten werden im nächsten Abschnitt genauer erklärt. Die exakte Funktionsweise aller Komponenten finden Sie in der Bedienungsanleitung zu der Webapplikation im Anhang.

## 5.1 Oberfläche

Die Startseite, bzw. Loginseite der Webanwendung besteht aus lediglich vier Komponenten. Darunter finden sich die Eingabefelder, die Buttons zur Formularbestätigung, die Kopfzeile mit Softwarelogo sowie dem Eingabeformular selbst (Abbildung 5.1).). Diese Seite ist so aufgebaut, dass sich die Eingabefelder unter dem Logo befinden. Die Buttons befinden sich unter den Eingabefeldern und das gesamte Interaktionsfeld wird von dem Formularrand umschlossen. Die Eingabefelder sind für den Nutzernamen des Anwenders sowie dessen Passwort bestimmt. Die Buttons haben unterschiedliche Funktionen. Der erste validiert die Eingabe der eingegebenen Nutzerdaten und leitet den Anwender auf das Dashboard weiter. Der zweite Button dient der Registrierung neuer Nutzer. Durch diese Aufteilung funktioniert der Login schnell und intuitiv, da dies ein verbreitetes visuelles Konzept in der Webentwicklung darstellt. Nach der erfolgreichen Authentifizierung befindet sich der Anwender auf dem Dashboard. Dieses ist die Hauptseite der Anwendung.

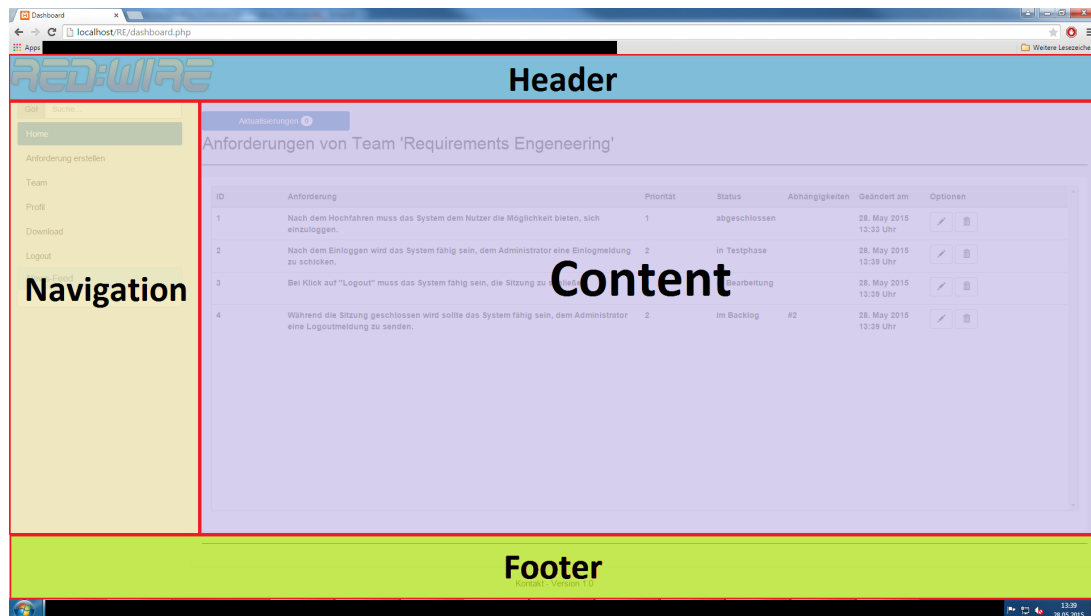


**Abbildung 5.1:** Screenshot der Loginseite

### Dashboard

Die Struktur der Hauptseite setzt sich aus vier Bereichen zusammen (Abbildung 5.2). Dieses Layout bietet ausreichend Platz und alle Funktionen sowie Komponenten sind übersichtlich angeordnet. Der Das Logo der Software befindet sich im oberen Bereich der Seite in der Kopfzeile. Im unteren Bereich, der Fußzeile, befinden sich Links zu aktuellen Versionsinformationen und Kontaktinformationen. Die Fußzeile ist dynamisch und somit auf Wunsch einfach erweiterbar und individualisierbar. Im mittleren Seitenbereich befindet sich der für den Anwender relevante Inhalt. Dies bezieht sich auf die seinem Team zugeordneten, ausgelesenen Anforderungen. In dem linken Seitenviertel ist die Menüleiste positioniert. Diese besteht aus Suchfeld im oberen Bereich, sowie Funktionsbuttons,

welche den Anwender mit dem System interagieren lassen. Der untere Bereich der Leiste beinhaltet einen „News-Feed“. Dies ist ein Panel, welches aktuelle, für den Nutzer relevante, Informationen ausgibt. Das Menü ist dauerhaft sichtbar und nicht ausblendbar. Dies soll die strukturelle Orientierung des Anwenders gewährleisten. Das News-Feed Panel ist nach unten erweiterbar und kollidiert mit keinem der darunter liegenden Objekte.

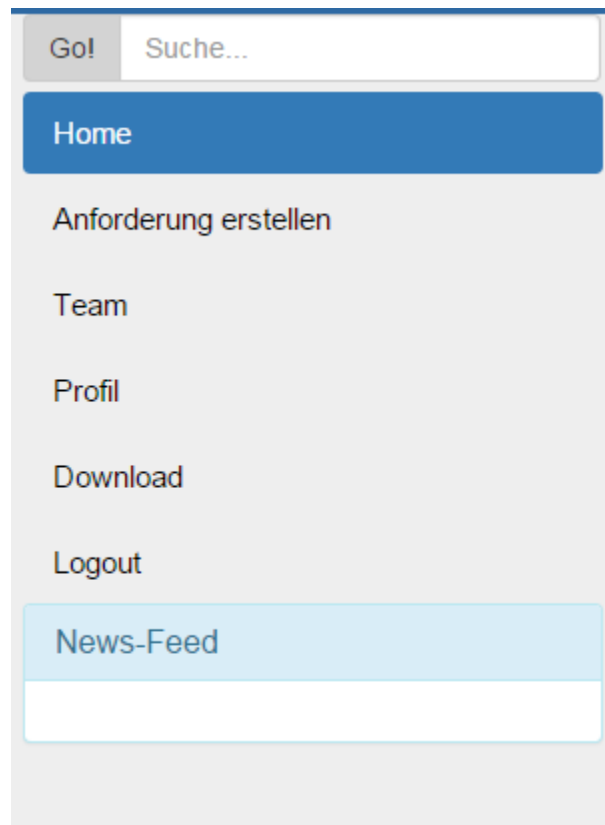


**Abbildung 5.2:** Layout der Website Aufteilung

### Menü und Struktur

Da die Webanwendung die Arbeit erleichtern und effizienter gestalten soll, steht bei dem Design der Benutzeroberfläche besonders die Übersichtlichkeit im Vordergrund. Diese wird von der Menüstruktur, den einzelnen funktionalen Bereichen sowie der Positionierung von allein stehenden Elementen beeinflusst. Dafür ist es wichtig, dass Zusammenhänge der Menüpunkte und die aktuelle Position in den Ebenen der Website immer erkennbar sind. Außerdem sollten die Bezeichnungen der Menüpunkte selbsterklärend, bzw. eindeutig zuzuordnen sein. Zu diesem Zweck sind die Bezeichnungen der Menüpunkte möglichst kurz gehalten (Abbildung 5.3).

Die Hauptseite und gleichzeitig Startseite ist mittels „Home“ betitelt. Darunter die Hauptfunktion des Systems: „Anforderung erstellen“. Dem folgend befindet sich die zwei Verwaltungsmenüs „Team“ und „Profil“. Jene ermöglichen eine Verwaltung der Teams sowie die Bearbeitung der eigenen Nutzerinformationen. Folgend existiert ein Button zum Download, d.h. Export der Anforderungen, gefolgt von dem letzten Menüpunkt „Logout“. Mit diesem wird die aktuelle Sitzung beendet und zur Loginseite zurückgeleitet. Dabei werden



**Abbildung 5.3:** Screenshot der Navigationsliste

die Metadaten der Sitzung aus dem lokalen Speicher entfernt, um Missbrauch zu vermeiden. Der aktuelle Bereich ist zu jeder Zeit der im Menü blau markierte Eintrag. Dadurch ist stets zu erkennen, auf welcher Ebene der Website sich der Anwender gerade befindet.

Direkt über dem Menü befindet sich ein Suchfeld. Dort kann der Anwender zu jeder Zeit Anforderungen suchen und diese auf der Hauptseite im Inhaltsbereich darstellen lassen. Unter dem Menü befindet sich ein passiver News-Feed. Mit diesem kann nicht interagiert werden, da der Inhalt automatisch aktualisiert wird und von den Interaktionen anderer Teammitglieder abhängt.

### Inhaltsbereich

Der Inhaltsbereich verändert sich je nach Nutzeraktionen und wird dynamisch nachgeladen. Dynamisches Laden verhindert, dass jedes Mal die gesamte Seite neu übertragen werden muss und schon die Bandbreite sowie Ladezeiten, was die Effizienz der Software steigert. An oberster Stelle des Bereichs befindet sich ein blaues Nachrichtenfeld. In diesem wird die Anzahl gefundener Änderungen durch andere Teammitglieder als Zahl angezeigt. Sollte ein Teammitglied eine Aktion durchführen, bei welcher der Anwender eine Nachricht im News-Feed Panel angezeigt bekommt, erhöht sich diese Zahl um die



Anzahl der im Panel befindlichen Nachrichten. Bei mindestens einer noch nicht abgerufenen Aktion, ändert sich die Farbe der dargestellten Zahl in Rot. Ein Klick auf dieses Feld aktualisiert den Inhalt und setzt die Zahl auf null zurück, wodurch diese wieder weiß angezeigt wird.

Darunter ist als Überschrift der Name des aktuellen Teams zu sehen. Anschließend sind im Inhaltsbereich alle Anforderungen, zur besseren Übersicht tabellarisch, dargestellt. Die Spaltennamen hierbei lauten in Reihenfolge: „ID“, „Anforderung“, „Priorität“, „Status“, „Abhängigkeiten“, „Geändert am“, „Optionen“. In jeder Spalte werden die entsprechenden Daten zu jeder Anforderung dargestellt. Ausnahme ist die Spalte „Optionen“. In dieser sind permanent zwei dargestellte Buttons zum Bearbeiten und Löschen der entsprechenden Anforderung in dieser Zeile. Sobald diese Tabelle die Grenzen des Inhaltsbereiches erreicht, bekommt die Tabelle eine zusätzliche Eigenschaft, um die Seitenstruktur nicht zu verändern. Diese ist die Möglichkeit, die Tabelle vertikal zu scrollen. Damit wird präventiv die Verschiebung der Fußleiste unterbunden.

### Anforderung erstellen

Befindet sich der Anwender im Menüpunkt „Anforderung erstellen“, wird im Inhaltsbereich die Anforderungsschablone angezeigt. Entsprechend der Anforderungsschablone nach Rupp helfen die Art und Beschreibung der Eingabefelder dabei, das Muster einzuhalten. Zusätzlich bietet sich die Möglichkeit, manuell eine eigene Projektbezogene Identifikationsnummer der Anforderung festzulegen (natürliche Zahlen), sowie optional Abhängigkeiten anzugeben. Die Identifikationsnummer akzeptiert auch Kommazahlen, welche in englischer Trennweise mittels einem Punkt separiert werden. Diese werden dann gerundet. So werden weniger schwerwiegende Fehleingaben des Anwenders verbessert. Sollte bei der Priorität oder dem Status nichts ausgewählt werden, wird der gesetzte Standardwert der Schablone übernommen. Dieser ist im Feld Priorität „0“ und im Status „im Backlog“. Dies dient dazu, eine gewisse Hochwertigkeit der Anforderungen sicherzustellen.

### Designtechnische Features

Die Verwaltungsoptionen „Team“ und „Profil“ des Menüs werden separat in einem modalen Dialogfenster geöffnet. Dadurch wird eine Manipulation der Anforderungen, während das Team gewechselt wird, oder der Anwender seine Nutzerdaten ändert, verhindert. Die Option „Download“ öffnet ein kleineres Fenster, mit zwei Buttons darin. Jene werden genutzt, um zwischen einer .csv-Datei und einer .xls-Datei zu wählen. Mittels eines Klicks auf den entsprechenden Button, wird der Downloadprozess gestartet.

Die Webanwendung ist so konzipiert, dass sie alle Funktionen möglichst selbsterklärend darstellt. Das bedeutet, dass zu den meisten Funktionen Tooltips bei Mouseover erscheinen und der Anwender dadurch mehr Informationen erhalten kann. Diese Tooltips lassen sich dennoch mit dem Entfernen des Hakens im Kästchen „Tooltips“ deaktivieren. Außerdem werden verbreitete und verständliche Symbole für die Icons, wie etwa ein Mülleimer zum Löschen, verwendet. Versucht der Anwender eine ungültige Aktion durchzuführen, so wird er durch eine Fehlermeldung darauf hingewiesen. Diese Fehlermeldungen weisen ihn auf die Fehlerursache hin und geben Anweisungen, welche Schritte auszuführen sind. Dies ist beispielsweise der Fall, falls sich der Anwender bei dem Erstellen einer Anforderung nicht in einem Team befindet (Abbildung 5.4).

**Anforderungen**

Fehler: Sie müssen erst Mitglied eines Teams sein, um Anforderungen eintragen zu können!

Hallo testUser1, tragen Sie eine neue Anforderung ein:

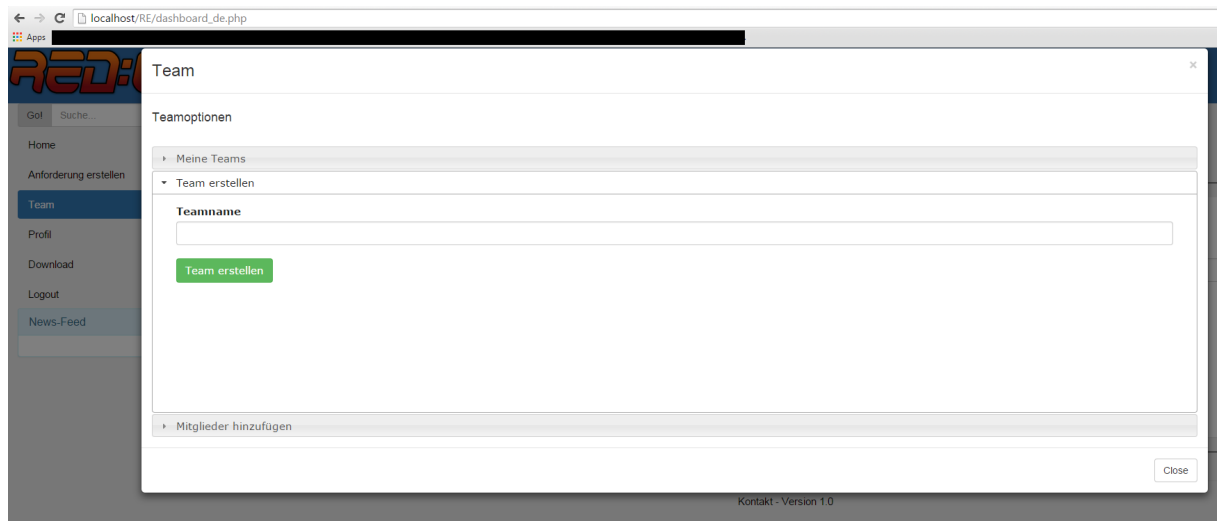
Nach dem Systemstart muss das System sich einzuloggen 2

Priorität: 1 Status: in Bearbeitung

Bestätigen

**Abbildung 5.4:** Screenshot einer Fehlermeldung

Zur Teamverwaltung gibt es den Menüpunkt mit der Bezeichnung: „Team“. Wie auch beim Punkt „Profil“ öffnet sich ein Dialogfenster. In diesem sind drei Unterpunkte zur Teamverwaltung in einem sogenannten „Akkordiondesign“ so angeordnet, dass jeweils nur maximal ein Punkt eingeblendet ist (Abbildung 5.5). Dadurch wird ein dynamisches Nachladen des Inhalts der einzelnen Menüpunkte bei einem Klick auf diese ermöglicht. Die Eingabe- und Auswahlfelder sowie die Buttons sind im gleichen Design wie der Punkt „Anforderung erstellen“ gehalten. Dies soll ein einheitliches Design repräsentieren („Corporate Design“). Gleiches gilt für das Dialogfenster des Punktes „Profil“. Hier wird Aufgrund des zur Verfügung stehenden Platzes auf ein Akkordiondesign verzichtet.



**Abbildung 5.5:** Screenshot des Team-Dialogfensters

In der Fußzeile der Website befindet sich ein Link „Kontakt“, welcher das Impressum darstellt, sowie ein Link „Version“ zum Versionsverlauf der Webanwendung. Das Anklicken dieser öffnet jeweils ein modales Dialogfenster. Hier ist es möglich Kontaktdaten der Entwickler zu betrachten und sich im Versionsverlauf die Neuerungen der Webanwendung anzuschauen. Da diese Informationen nicht zum Arbeiten benötigt werden, befinden sie sich unten in der Fußzeile.

## 5.2 Features

Die Hauptfunktion der Software ist es, Anforderungen in Form der Rupp-Schablone aufzunehmen und persistent zu speichern. Doch Red:Wire bietet umfangreiche zusätzliche Features, um das Arbeiten sehr viel leichter zu gestalten.

Red:Wire baut auf ein Team-System auf. Das bedeutet, dass alle durchgeführten Aktionen in einem Teamkontext geschehen. Erstellt ein Anwender eine Anforderung, werden alle anderen Mitglieder seines Teams benachrichtigt, sollten sie zeitgleich auch mit dem Tool arbeiten.

Jeder Nutzer kann beliebig viele Teams erstellen und mit zwei Mausklicks zwischen ihnen wechseln. Jedes Team kann beliebig viele Mitglieder haben. Jeder Nutzer kann sich jedoch zeitgleich nur in einem Team befinden. Jedes Teammitglied hat vollen Zugriff auf die erstellten Anforderungen aller anderen Mitglieder.

Zur Sicherheit der in der Anwendung behandelten Informationen und zur Identifikation der Anwender zur Teambildung/-arbeit wird ein Authentifikationssystem basierend auf Nutzernamen und Passwort genutzt. Jeder registrierte Anwender kann sich so einfach in die Anwendung einloggen. Nicht registrierte potenzielle Anwender können sich einen Account über die Registrierungsfunktion anlegen und sofort anmelden.

Dem einzelnen Anwender besteht die Möglichkeit, seine Profildaten (Passwort, E-Mail) zu ändern. Außerdem kann die Sprache mit einem Klick gewechselt werden. Derzeit stehen dafür die Sprachen Deutsch und Englisch zur Verfügung.

Um die Arbeit auch ohne Red:Wire weiterführen zu können, gibt es die Möglichkeit des Anforderungsdownloads. Der Download wird als .csv Datei mit Tabulatorstopp sowie als .xls Microsoft Excel Datei angeboten.

Damit die Darstellung der Anforderungen auf dem Dashboard immer den Anwenderwünschen gerecht wird, gibt es Sortierfunktionen für bestimmte Attribute. Diese können mit einem Klick ausgewählt werden. Die folgende Darstellung der Anforderungen wird auf den Dateidownload übertragen.

Falls nur bestimmte Anforderungen gelistet werden sollen, bietet Red:Wire ein Suchfeld in der Menüleiste. In dieses können beliebige Ausdrücke eingegeben werden und Red:Wire durchsucht die Datenbank nach Anforderungen mit diesen Ausdrücken. Die folgende Darstellung der Anforderungen wird auf den Dateidownload übertragen.

Um immer auf dem neuesten Stand zu bleiben, existiert im unteren Bereich der Menüleiste ein News-Feed Panel. Dieser wird alle zwei Sekunden aktualisiert und bietet Informationen rund um alle Veränderungen Ihrer Anforderungen. Daran inbegriffen sind Status zu gelöschten, hinzugefügten und geänderten Anforderungen.

Ein Administrator ist zudem in der Lage, Anwenderkontos zu löschen, bzw. deren Teams anderen Nutzern zu überschreiben. Er sieht weiterhin alle angegebenen Profilinformativen der Nutzer, ausgeschlossen des Passwortes. Sollte ein Anwender sein Passwort vergessen, muss der Administrator sein Konto löschen. Der Anwender legt sich ein neues Konto an und loggt sich erneut ein. Durch das Überschreiben des Nutzerteams an einen anderen Nutzer bleiben auch die dort gespeicherten Anforderungen erhalten.

## 6 Ausblick und Fazit

Alle geplanten Funktionalitäten sowie Features sind implementiert worden. Zusätzlich flossen weitere Wünsche und während der Entwicklung entstandene Ideen direkt in die Entwicklung mit ein. Die Übertragung der Offlinesoftware in eine Webanwendung ist somit erfolgreich verlaufen. Die Unterschiede dieser beiden Tools sind dadurch ersichtlich, dass Red:Wire eine online Applikation ist. Red:Wire bietet daher viel mehr Möglichkeiten, als letzteres offline Tool. Viele implementierte Features ließen sich erst dadurch umsetzen. Durch umfangreiches Testen konnte eine Vielzahl an Fehlern bereits während der Implementierung behoben werden. Die Lauffähigkeit der Software wird somit gewährleistet. Getestete Funktionen und Features sind:

- Die Anbindung des Servers an die Datenbank.
- Sowohl Kommunikation als auch Datenaustausch mit der Datenbank.
- Das Teamsystem mit mehreren unterschiedlichen Nutzern.
- Die Funktionalität des News-Feeds bei gleichzeitiger Arbeit im selben Browser, sowie Cross-Browser (Firefox, Chrome).
- Die Performance der Website bei > 1000 Anforderungen.
- Ändern der Profildaten
- Ausloggen aus der Applikation

Ein Problem bei der Entwicklung war die Umsetzung des Testens. Selbstverständlich wurden alle Tests nach besten Möglichkeiten durchgeführt. Allerdings fehlten Ressourcen zu Tests in großem Umfang, sodass nicht alle Testfälle simuliert werden konnten. Zur Simulation wurden mehrere Account auf einem Rechner angelegt und simultan betrieben. Ebenso wurde die mobile Variante der Website im Device-mode des Browsers emuliert.

Für die Entwicklung werden aktuelle Funktionen, Designs und Frameworks genutzt. Aufgrund dessen sollte ein moderner Browser genutzt werden. Da sowohl die Entwicklung, als auch Tests unter Chrome und Firefox stattgefunden haben, kann die Funktionsfähigkeit unter folgenden Browsern garantiert werden:

- Google Chrome, Version 43.0.2357.124 (Stand 22.06.2015)
- Mozilla Firefox, Version 38.0.5 (Stand 22.06.2015)

Dadurch, dass Struktur-, Style- sowie Scriptdateien separat abgelegt sind, ist es möglich diese einzeln zu bearbeiten. Somit wird die Möglichkeit, Erweiterungen zu implementieren, vereinfacht. Für die Zukunft bieten sich mehrere Erweiterungsideen.

Eine Idee ist, eine Kategorisierung der Anforderungen zu entwickeln. Das bedeutet, eine Zuordnung und Darstellung der Tabelle mit selbst erstellten Unterkategorien. Beispielsweise seien dafür genannt: „Hardware“ und „Software“. Das erhöht bei einer größeren Anzahl von Anforderungen die Übersichtlichkeit und steigert **unseres Erachtens** nach die Motivation der Anwender durch höhere Individualisierbarkeit der Software. Dies ist besonders dann der Fall, wenn die vorhandenen Sortierungs- und Suchmöglichkeiten dem Anwender nicht mehr genügen.

Eine weitere Erweiterungsidee ist, dass der Anwender selbsterstellte Satzbau-Templates nutzen kann. Dies erlaubt dem Anwender sowie Teamleiter, eigene Ansprüche, oder Strukturen beizubehalten, oder durchzusetzen. Letzteres könnte mittels manuell erstellter XML-Templates implementiert werden. Ebenso würde dies, aufgrund der änderbaren Grammatik, eine internationale Nutzung ermöglichen.

Zum jetzigen Zeitpunkt bietet die Website bereits eine Übersetzung auf Englisch an, welche eine Nutzung internationaler Anwender jedoch aufgrund der, an die deutsche Grammatik angepassten, Schablone beschränkt. Die Anforderungsformulierung ist somit für den deutschsprachigen Gebrauch optimiert. Die Software ist zum jetzigen Stand (30. Juni 2015) voll funktionsfähig und kann sowohl privat, in der Hochschule als auch kommerziell genutzt werden.

# Abbildungsverzeichnis

2.1	Schnittmenge . . . . .	3
2.2	Schablone . . . . .	4
4.1	Internetauftritt Samsung 2004 . . . . .	7
4.2	Internetauftritt Samsung 2015 . . . . .	7
4.3	Webanwendung Konzept . . . . .	9
4.4	Use-Case Diagramm Nutzeraktionen . . . . .	10
5.1	Screenshot der Loginseite . . . . .	19
5.2	Layout der Website Aufteilung . . . . .	20
5.3	Screenshot der Navigationsliste . . . . .	21
5.4	Screenshot einer Fehlermeldung . . . . .	23
5.5	Screenshot des Team-Dialogfensters . . . . .	24
6.1	Kostenfaktoren für Fehlerbehebungen . . . . .	A- 2

# Literaturverzeichnis

- [1] Authornamen: *Titel(, Seite)*. URL: <http://www.google.com>.
- [2] Chris Rupp, Matthias Brenner: *Nur qualitativ hochwertige Anforderungen erzeugen hochwertige Produkte*. URL: [http://www.sophist.de/index.php?id=180&tx\\_ttnews%5Byear%5D=2011&tx\\_ttnews%5Bmonth%5D=11&tx\\_ttnews%5Bday%5D=09&tx\\_ttnews%5Btt\\_news%5D=514](http://www.sophist.de/index.php?id=180&tx_ttnews%5Byear%5D=2011&tx_ttnews%5Bmonth%5D=11&tx_ttnews%5Bday%5D=09&tx_ttnews%5Btt_news%5D=514) [siehe S. 2].
- [3] Michael Schenkel: *Der Golden Circle im Requirements Engineering*. URL: <http://www.microtool.de/requirements/golden-circle-im-requirements-engineering/> [siehe S. 1].
- [4] Michael Schenkel: *Warum der Umgang mit Anforderungen so wichtig ist*. URL: <http://www.microtool.de/requirements/warum-der-umgang-mit-anforderungen-wichtig-ist/> [siehe S. 2, A- 2].
- [5] ScrumGuides: *The Scrum Guide*. URL: <http://www.scrumguides.org/scrum-guide.html> [siehe S. 2].
- [6] Takaki Makino, Chaesang Jung, Doantam Phan: *Finding more mobile-friendly search results*. URL: <http://googlewebmastercentral.blogspot.de/2015/02/finding-more-mobile-friendly-search.html> [siehe S. 8].
- [7] Takaki Makino, Doantam Phan: *Rolling out the mobile-friendly update*. URL: <http://googlewebmastercentral.blogspot.de/2015/04/rolling-out-mobile-friendly-update.html> [siehe S. 8].
- [8] Yoshiaki Kato, Pierre Far: *Changes in rankings of smartphone search results*. URL: [http://googlewebmastercentral.blogspot.de/2013/06/changes-in-rankings-of-smartphone\\_11.html](http://googlewebmastercentral.blogspot.de/2013/06/changes-in-rankings-of-smartphone_11.html) [siehe S. 8].
- [9] it-agile GmbH: *Scrum*. URL: <http://www.it-agile.de/wissen/methoden/scrum/> [siehe S. 2].



# Anhang

## Installationsanleitung

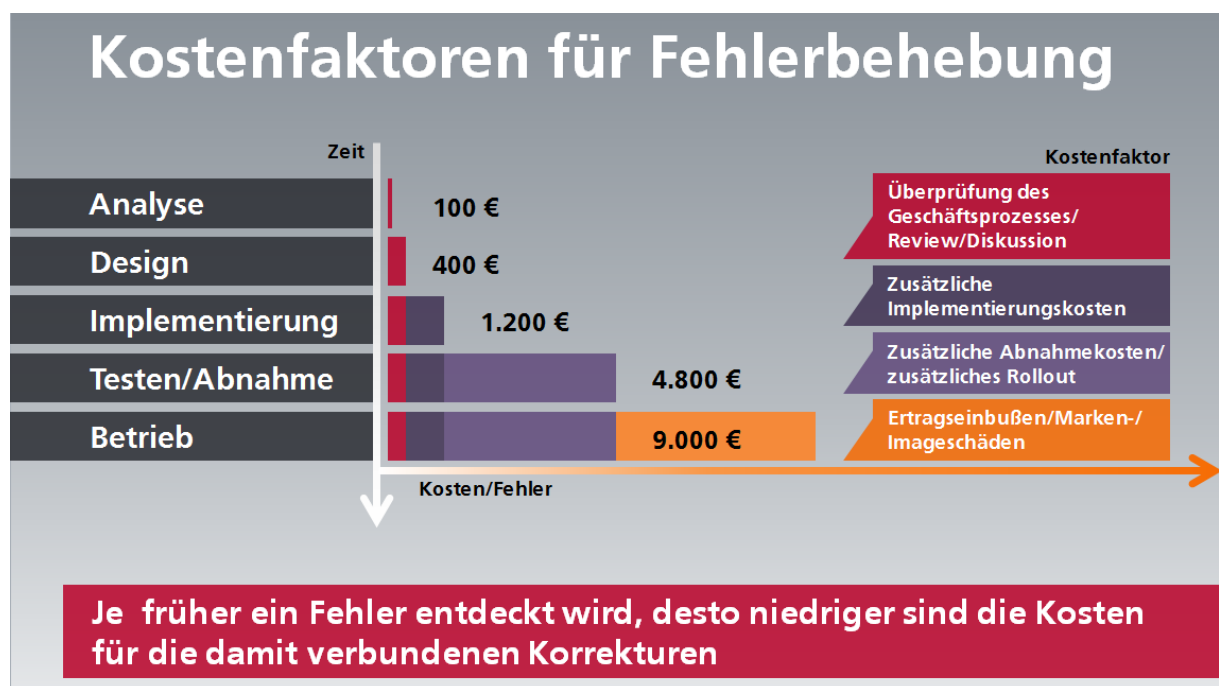
Siehe CD: Unterordner „Installationsanleitung“

## Nutzerhandbuch

Siehe CD: Unterordner „Nutzerhandbuch“

Google Drive: <https://goo.gl/pBD6Wi>

## 6.1 Grafiken



**Abbildung 6.1:** Grafik Kostenfaktoren für Fehlerbehebungen:

(Quelle: Schenkel, Michael; „Warum der Umgang mit Anforderungen so wichtig ist“ (Literaturverzeichnis [4]))