

Ticketing System for Developing Countries

Projektstrukturplan

Hochschule für Technik Rapperswil

Herbsssemester 2019

10. Januar 2020

Autor: Luca Gubler, Alessandro Bonomo
Betreuer: Prof. Frank Koch
Projektpartner: INS Institute for Networked Solutions
Arbeitsperiode: 16.09.2019 - 10.01.2020
Arbeitsumfang: 360 Stunden, 12 ECTS pro Student

Inhaltsverzeichnis

1 Inhalt	3
1.1 Zweck	3
1.2 Gültigkeitsbereich	3
1.3 Referenzen	3
2 Projektübersicht	4
2.1 Zweck und Ziel	4
2.2 Lieferumfang	4
2.3 Annahmen und Einschränkungen	4
3 Projektorganisation	5
3.1 Organisationsstruktur	5
3.2 Externe Schnittstellen	5
4 Management Abläufe	6
4.1 Zeitaufwand	6
4.2 Zeitplanung	6
4.2.1 Phasen / Sprints	6
4.2.2 Meilensteine	6
4.2.3 Iterationsplanung	6
4.3 Besprechungen	7
5 Risikomanagement	8
5.1 Risiken	8
5.2 Umgang mit Risiken	8
6 Infrastruktur	9
6.1 Dokumentation	9
6.2 Arbeitspakete Verwaltung	9
6.3 Konstruktion	9
6.4 Testing	9
6.5 Infrastruktur	10
7 Qualitäts- und Sicherheitsmassnahmen	11
7.1 Projektmanagement	11
7.1.1 Workflows	11
7.2 Entwicklung	11
7.2.1 Vorgehen	11
7.2.2 Unit Testing	11
7.2.3 Code Reviews	11
7.2.4 Code Style Guides	11

1 Inhalt

1.1 Zweck

Dieses Dokument beschreibt die Planung der Bachelorarbeit, in welchem ein Ticketing System für Entwicklungsländer, welches bereits als frühere Bachelorarbeit realisiert wurde, mit zusätzlicher Funktionalität erweitert wird.

1.2 Gültigkeitsbereich

Dieses Dokument ist während der gesamten Laufzeit der Bachelorarbeit gültig. Die Änderungsgeschichte kann in Github nachverfolgt werden.

1.3 Referenzen

Dieses Dokument wurde mit dem Wissen erstellt, welches in den Modulen Software Engineering 1 & 2, Microsoft Technologies sowie in gewissen Grenzen in Cloud Infrastructure und Cloud Solutions vermittelt wird.

2 Projektübersicht

Es soll das bereits bestehende Ticketing System um zusätzliche Funktionalität erweitert werden. Ziel des Ticketing Systems ist es, Menschen an Orten mit limitiertem Zugang zu modernen Hilfsmitteln zu unterstützen. Dabei können zum Beispiel die Menschen mit Experten einer Hilfsorganisation in Kontakt treten, ohne dass diese physisch vor Ort sein müssen. Somit können sehr viele Menschen mit vergleichsweise niedrigem Zeitaufwand angesprochen werden

2.1 Zweck und Ziel

Das Ticketing System wurde bereits in einer früheren Bachelorarbeit umgesetzt. Ziel dieser Bachelorarbeit ist es, das bereits existierende System um zusätzliche Funktionalität zu erweitern.

Folgende Ziele wurden für diese Arbeit gesetzt:

- Der erfolgreiche Abschluss der Bachelorarbeit und das Erhalten der 12 ECTS.
- Die erlernte Theorie der Module "Software Engineering 1 & 2" in die Praxis umsetzen.
- Den Umgang mit neuen Technologien kennen lernen.

2.2 Lieferumfang

Folgende Dokumente werden am Ende der Bachelorarbeit abgeliefert:

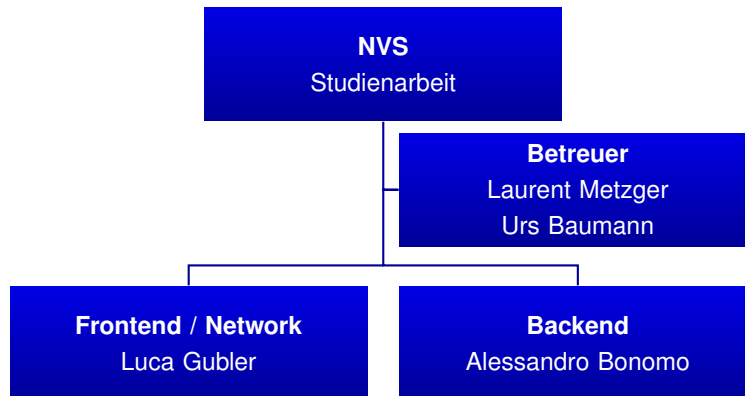
- Abstract
- Aufgabenstellung
- Einverständniserklärung
- Erklärung zur Urheberschaft
- Passwörter
- Persönliche Berichte
- Protokolle
- Source Code

2.3 Annahmen und Einschränkungen

Pro Teammitglied wird mit einem Arbeitsaufwand von 24 Stunden pro Woche gerechnet. Falls jedoch Probleme auftreten oder der Arbeitsaufwand falsch eingeschätzt wurde, kann sich der Arbeitsaufwand auf bis zu 40 Stunden pro Woche erhöhen. Falls es jedoch zu Verzögerungen kommt, wird der Arbeitsumfang in Absprache mit den Betreuern dementsprechend angepasst.

3 Projektorganisation

3.1 Organisationsstruktur



Im Bild wurde die Organisationsstruktur dargestellt. Diese Organigramm wurde jedoch nur zum aufzeigen der groben Zuständigkeiten erstellt. Natürlich wird sich das Team gegenseitig unterstützen, falls Probleme auftreten.

3.2 Externe Schnittstellen

In dieser Studienarbeit übernimmt Professor Frank Koch die Rolle des Betreuers. Stephan Meier, welcher bereits bei der früheren Bachelorarbeit die Rolle als externer Co-Examinators übernahm, wird auch bei dieser Bachelorarbeit diese Aufgabe übernehmen. Zusätzlich wird Professor Laurent Metzger diese Bachelorarbeit als interner Co-Examinator betreuen.

4 Management Abläufe

4.1 Zeitaufwand

Die Studienarbeit begann in der Woche vom 18. Februar 2019 und dauert insgesamt 15 Wochen. Für das Erreichen der 8 ECTS ist geplant, dass jedes Teammitglied 240 Stunden arbeitet. Daraus resultiert eine durchschnittliche Arbeitszeit von 16 Stunden pro Woche.

Projektstart	18.2.2019
Projektdauer	15 Wochen
Arbeitsstunden pro Person	16h pro Woche, Total 240h
Arbeitsstunden Total	480h
Projektende	31.05.2019

Für die Studienarbeit stehen total 480 Stunden zur Verfügung. Mit dem definierten Projektumfang wird diese Arbeitszeit voraussichtlich vollständig ausgenutzt. Sollte der Umfang jedoch früher als erwartet abgeschlossen werden können, kann das Projekt um weitere Funktionalitäten erweitert werden.

4.2 Zeitplanung

4.2.1 Phasen / Sprints

Das gesamte Projekt wird in die vier Phasen Inception, Elaboration, Construction und Transition eingeteilt. Pro Phase gibt es wiederum einzelne Sprints. Die Sprints starten jeden zweiten Donnerstag und beginnen direkt nach dem Meeting mit den Betreuern. Zudem wurden einzelne Meilensteine definiert, welche auf der untenstehenden Tabelle entnommen werden können.

4.2.2 Meilensteine

Datum	Meilenstein
27.02.2019	M1: Projektplan erstellt
15.03.2019	M2: Domainanalyse und Anforderungsspezifikation erstellt
28.03.2019	M3: End of Elaboration
25.04.2019	M4: Zwischenpräsentation
23.05.2019	M5: End of Construction
31.05.2019	M6: Abgabe der Dokumentation

4.2.3 Iterationsplanung

Zu Beginn jedes Sprints setzt sich das Team zusammen, um den nächsten Sprint zu planen. Dabei wird jeweils besprochen, welche Aufgaben des vergangenen Sprints nicht

vollständig abgeschlossen werden konnten. Die nicht abgeschlossenen Arbeiten werden mit neu definierten Aufgaben in den neuen Sprint übernommen und jeweils zeitlich abgeschätzt und priorisiert. Da sich im Team nur 2 Mitglieder befinden, wird darauf verzichtet, die Arbeitspakete unter den Teammitgliedern untereinander zuzuweisen. Die gesamte Planung und Verwaltung der Aufgaben wird in Jira erledigt.

4.3 Besprechungen

Die Teammitglieder treffen sich mindestens einmal pro Woche zur Besprechung von Fragen, dem klären von Problemen und um das weitere Vorgehen zu planen. Diese Meetings finden jeweils am Donnerstag nach dem Meeting mit den Betreuern statt.

5 Risikomanagement

5.1 Risiken

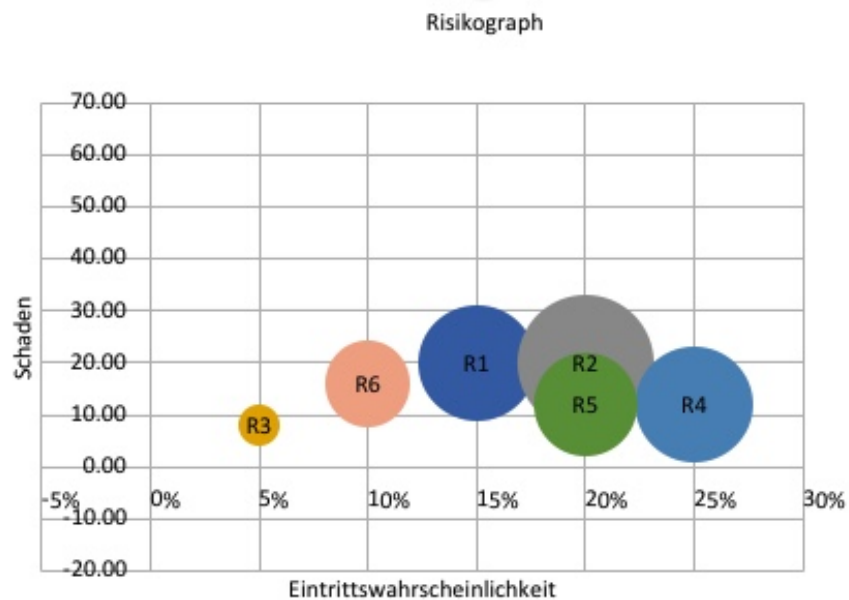


Abbildung 1: Ursprüngliche Risikoanalyse

Die detaillierte Risikoanalyse kann dem Dokument "TechnischeRisiken.pdf" entnommen werden.

5.2 Umgang mit Risiken

Risiken lassen sich in einem grösseren Projekt leider nicht vermeiden. Allfällige Risiken werden jeweils zu Beginn eines Sprints im Team angesprochen und sofern sinnvoll wurden dementsprechend Massnahmen getroffen.

Für die wahrscheinlichen Risiken wurden Massnahmen definiert, um den auftretenden Schaden unter Kontrolle zu bringen. Sollten während des Projektes neue Risiken hervortreten, werden diese in die Risikoanalyse aufgenommen und bewertet.

6 Infrastruktur

Die Infrastruktur setzt sich aus der Hard- und Software zusammen, welche für die Durchführung der Studienarbeit verwendet wird. So kann die Zusammenarbeit im Team erleichtert werden. Zudem kann die Code Qualität besser überprüft und verbessert werden.

Viele Tools sind bereits aus Modulen oder vergangenen Projektarbeiten bekannt, weshalb man sich schnell auf bestimmte Tools einigen konnte. Trotzdem gibt es noch einige Unklarheiten, da einige Tools von anderen abhängen. Erst im späteren Projektverlauf wird sich zeigen, ob diese Tools verwendet werden. Es kann aber auch sein, dass einige Tools, welche später erwähnt werden, gar nicht im Projekt eingesetzt werden.

6.1 Dokumentation

Für das Erstellen der Dokumentation wird \LaTeX verwendet. Dieses Tool erforderte zwar etwas Zeit, um sich mit diesem Tool vertraut zu machen. Das Team ist jedoch überzeugt, dass man mit \LaTeX weniger Probleme hat, wenn man bereits eine gute Vorlage hat.

Die einzelnen Dokumentationen werden jeweils lokal und auf Github gespeichert. Da die Dokumentation auf Github gespeichert ist, kann auch zu jeder Zeit nachverfolgt werden, wann etwas geändert wurde.

6.2 Arbeitspakete Verwaltung

Für das Verwalten der Arbeitspakete entschied sich das Team Jira zu verwenden. Für das Engineering Projekt wurde ebenfalls Jira verwendet und es machte einen guten Eindruck. So ist es mit Jira sehr einfach, neue Arbeitspakete zu verfassen und die Arbeitszeit abzuschätzen. Zudem ist das Reporting mit Jira sehr gut und man sehr schnell Burndown Charts erstellen.

6.3 Konstruktion

Als Versionsverwaltung wird GIT verwendet, da es zum de facto Standard in der Software Entwicklung gehört. Der grosse Vorteil ist, dass man nicht ständig mit dem Server verbunden sein muss. Zudem konnte jedes Mitglied bereits Erfahrungen mit Github sammeln, was den Einstieg erleichtert.

Als Entwicklungsumgebung wird PyCharm von JetBrains verwendet, da es eine sehr gute IDE für Python ist.

6.4 Testing

Sämtlicher Code muss getestet werden, weshalb geeignete Test Frameworks im Frontend sowohl als auch im Backend zum Einsatz kommen. Diese Frameworks müssen so-

wohl Unit Tests wie auch Mocken können. Mit diesen Vorganen können anschliessend de Micro Tests geschrieben werden.

6.5 Infrastruktur

Das Front- und Backend wird auf Servern des INS laufen. So können die Teilnehmer mit ihrem Webbrowser aber nur auf das Frontend zugreifen, wenn sie sich im selben Netzwerk wie die Server befinden.

Für das Projekt selber wird eine 3-Tier-Architektur angestrebt. Als Datenbank wird MySQL verwendet.

7 Qualitäts- und Sicherheitsmassnahmen

Um sicherzustellen, dass die Qualität der Arbeit immer gewährleistet ist, wurden folgende Massnahmen getroffen:

Massnahmen	Zeitraum	Ziel
Meeting	Wöchentlich	Hier wird der aktuelle Stand des Projektes besprochen und neue Entscheidungen werden gefällt
Besprechung der Meilensteine	6 Mal während der gesamten Projektdauer	Dient zur Kontrolle, ob das Team auf dem richtigen Weg ist
Code Reviews	Fortlaufend während der gesamten Projektdauer	Dient zur Erhöhung der Qualität des Codes
Verwendung aktueller Code Versionen	Fortlaufend während der gesamten Projektdauer	Bei den aktuellen Versionen wurden Bugs behoben, welche nicht mehr auftreten sollten

7.1 Projektmanagement

Als Projektmanagementsystem wird Jira verwendet. Dieses Tool erlaubt es, einzelne Arbeitspakete zu erstellen. So hat jedes Teammitglied jederzeit den Überblick über den aktuellen Stand des Projektes. Zudem ist ein Zeit-Management-Tool integriert. Somit kann während dem gesamten Projektverlauf der Zeitaufwand im Auge behalten werden.

7.1.1 Workflows

7.2 Entwicklung

7.2.1 Vorgehen

7.2.2 Unit Testing

7.2.3 Code Reviews

7.2.4 Code Style Guides

Abbildungsverzeichnis

1	Ursprüngliche Risikoanalyse	8
---	---------------------------------------	---