

Projektmanagementplan

Version 1.0.0

Gruppe 5 (Patrick Bucher, Pascal Kiser, Fabian Meyer, Sascha Sägesser)

08.04.2018

Inhaltsverzeichnis

1	Projektorganisation	1
1.1	Organisationsplan	1
1.2	Rollen	2
1.3	Zuständigkeit	2
2	Projektführung	2
2.1	Rahmenplan	2
2.2	Meilensteinplan	3
2.3	Sprintplan	3
2.4	Projektstrukturplan	4
2.5	Risikomanagement	4
2.5.1	Krankheits- / unfallbedingte Ausfälle	4
2.5.2	Ausfall HSLU-Netz	5
2.5.3	Hardwareausfall / Verlust privater Arbeitsgeräte	5
2.5.4	Zeitknappheit	5
2.6	Projektkontrolle	6
2.7	Aufwandschätzung	6
3	Projektunterstützung	6
3.1	Tools für Entwicklung, Test & Abnahme	6
3.2	Konfigurationsmanagement	7
3.3	Releasemanagement	7

1 Projektorganisation

1.1 Organisationsplan

Siehe Abbildung [Organigramm](#).

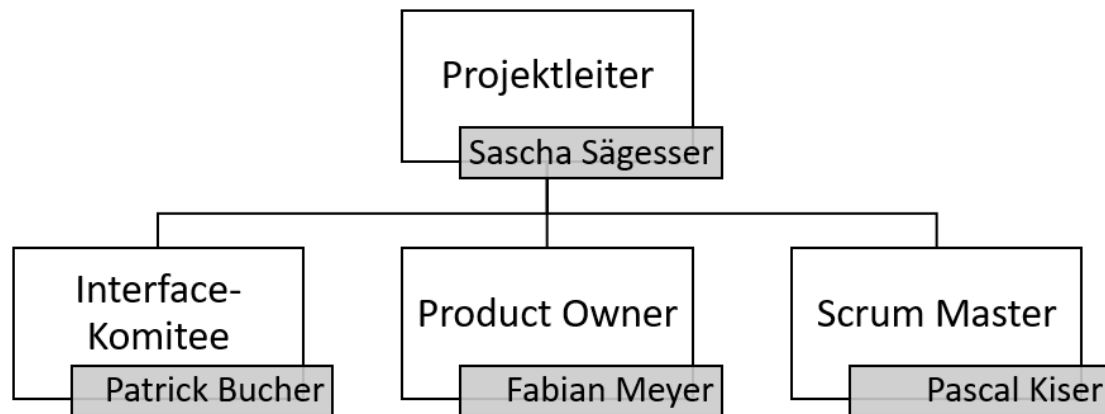


Abbildung 1: Organigramm

1.2 Rollen

- Projektleiter: *Sascha Sägesser*
- Interfacekomitee: *Patrick Bucher*
- Product Owner: *Fabian Meyer*
- Scrum Master: *Pascal Kiser*

1.3 Zuständigkeit

- Dokumentation: *Patrick Bucher*
- Build: *Pascal Kiser*
- Testing: *Fabian Meyer*
- Codereview: *Patrick Bucher*
- Sprintreviews: *Sascha Sägesser*
- Meilensteinberichte: *Fabian Meyer*
- Risikomanagement: *Pascal Kiser*

2 Projektführung

2.1 Rahmenplan

Der Rahmenplan (siehe Abbildung **Rahmenplan**) stellt eine Übersicht auf abstrakter Ebene dar. Die wichtigsten Elemente sind die definierten Meilensteine und Sprints. Dabei ist darauf zu achten, dass die Vorgaben der Aufgabenstellung, wie z.B. die Zwischen- und Schlussabgabe, von Anfang an in das Projekt miteingeplant werden.

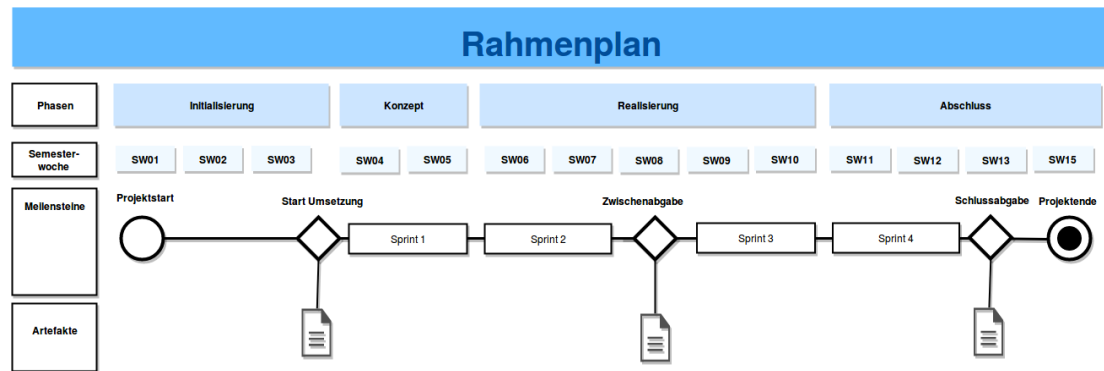


Abbildung 2: Rahmenplan

2.2 Meilensteinplan

Das komplette Projekt ist in drei Meilensteine aufgeteilt. Der erste Meilenstein entspricht dem Start der Projektumsetzung und die beiden letzten Meilensteine ergeben sich aus der Zwischen- und Schlussabgabe. Zu jedem Meilenstein gibt es Deliverables:

1. **Meilenstein (Start Umsetzung, 12.03.2018)**
 - ProductBacklog
 - Sprintplanung S1/S2
 - Detailplanung S1
 - Meilensteinbericht MS 2
2. **Meilenstein (Zwischenabgabe, 08.04.2018)**
 - Release 1 gemäss Muss-Features deployed und lauffähig
 - Sprintplanung S3/S4
 - Meilensteinbericht MS 3
3. **Meilenstein (Schlussabgabe, 13.05.2018)**
 - Release 2 gemäss Muss-Features deployed und lauffähig
 - Meilensteinbericht MS 4

2.3 Sprintplan

Die Projektdurchführung (MS 2 / 3) ist in vier Sprints aufgeteilt. Für die einzelnen Sprints sind Endtermin und Deliverables wie folgt festgelegt:

1. Sprint (25.03.2018): Sprintreview S1, Sprintplanung S2
2. Sprint (08.04.2018): Sprintreview S2, Sprintplanung S3
3. Sprint (29.04.2018): Sprintreview S3, Sprintplanung S4
4. Sprint (13.05.2018): Sprintreview S4

Jeder Sprint dauert zwei Wochen. Die detaillierte Sprintplanung ist mittels des Online-Tools *ScrumDo* realisiert.

2.4 Projektstrukturplan

Um das Projekt besser planen, durchzuführen und kontrollieren zu können, ist dieses in mehrere Teilprojekte, welche wiederum in Teilaufgaben gegliedert sind, aufgeteilt (siehe Abbildung **Projektstrukturplan**).

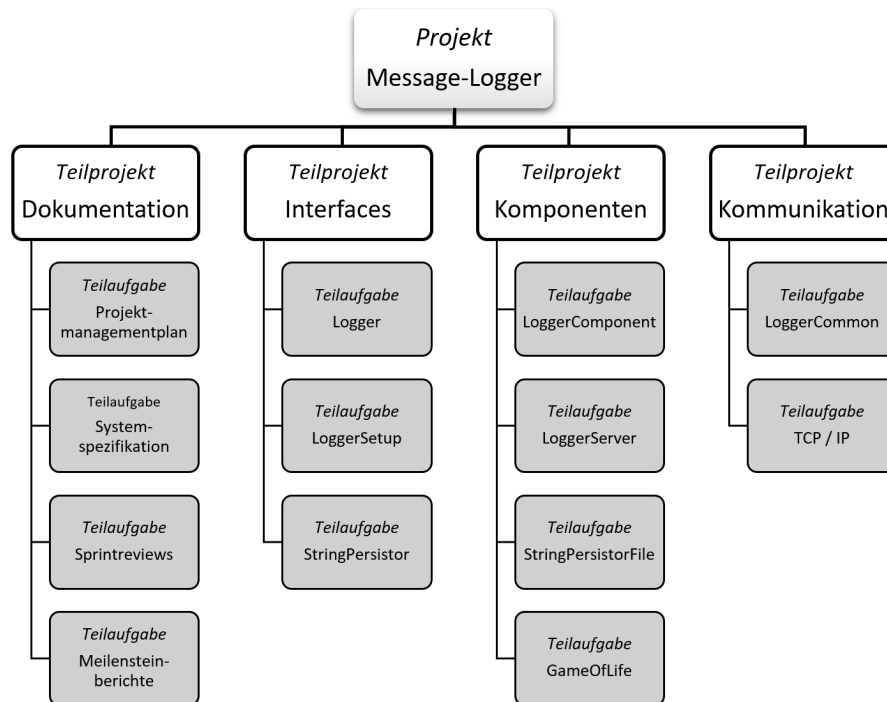


Abbildung 3: Projektstrukturplan

2.5 Risikomanagement

2.5.1 Krankheits- / unfallbedingte Ausfälle

Eine oder mehrere Personen fallen im Verlauf des Projekts krankheitsbedingt oder aufgrund eines Unfalles aus.

Folgende Massnahmen wurden definert, um das Schadensausmass und die Eintreffenswahrscheinlichkeit zu reduzieren:

Reduktion Schadensausmass:

- Wissensaustausch zwischen den Gruppenmitgliedern
- Dokumentation pflegen
- Regelmässiger Austausch im Plenum

Reduktion Eintreffenswahrscheinlichkeit:

- Keine Extremsportarten während den Sprints
- gesunde Ernährung

2.5.2 Ausfall HSLU-Netz

HSLU-Netz, EnterpriseLab oder andere verwendete Infrastruktur fallen aus oder können aus anderen Gründen nicht mehr verwendet werden.

Die Eintreffenswahrscheinlichkeit kann von den Gruppenmitgliedern mit beeinflusst werden. Folgende Massnahmen wurden definert, um das Schadensausmass reduzieren:

Reduktion Schadensausmass:

- Dokumentation und Sourcecode wird lokal von den einzelnen Gruppenmitgliedern gesichert
- Ausweichmöglichkeiten definiert (beispielsweise GitHub bei Ausfall von EnterpriseLab/Gitlab)
- Die Dokumentation des Projekts wird unabhängig vom Sourcecode verwaltet

2.5.3 Hardwareausfall / Verlust privater Arbeitsgeräte

Geleistete Arbeit geht verloren durch Ausfall oder den Verlust verwendeter Arbeitsgeräte.

Folgende Massnahmen wurden definert, um das Schadensausmass und die Eintreffenswahrscheinlichkeit zu reduzieren:

Reduktion Schadensausmass:

- Versionsverwaltung (Git)
- Instruktion Gruppenmitglieder: regelmässiges Comitten/Pushen

Reduktion Eintreffenswahrscheinlichkeit:

- Eigenverantwortung Gruppenmitglieder
- Keine lokalen Dateien; alle relevanten Dateien zum Projekt werden extern gehostet (Gitlab, Github etc).

2.5.4 Zeitknappheit

Zeitknappheit aufgrund schlechter oder fehlender Planung.

Folgende Massnahmen wurden definert, um das Schadensausmass und die Eintreffenswahrscheinlichkeit zu reduzieren:

Reduktion Schadensausmass:

- Priorisierung von Features
- Minimalanforderungen definieren

Reduktion Eintreffenswahrscheinlichkeit:

- Angemessene Zeitplanung
- Regelmässige Kontrolle, klar definierte Verantwortlichkeiten

2.6 Projektkontrolle

Für die Projektkontrolle sind folgende Tools und Methoden im Einsatz:

- ScrumDo
- ProductBacklog
- SprintBacklog
- Backlog-Grooming
- Definierung von Definition of Done der Tasks
- Sprintreviews
- Meilensteinberichte

2.7 Aufwandschätzung

Es wird davon ausgegangen, dass pro Person und Woche, zusätzlich zu den Vorlesungen, 5 Stunden Aufwand hinzukommen. Insgesamt entsteht so ein geschätzter Aufwand von 40 Stunden pro Sprint, respektive 80 Stunden jeweils für die Zwischen- und Schlussabgabe.

Der Aufwand wird weiterhin in folgende Kategorien unterteilt:

- Dokumentation
- Entwicklung
- Testing
- Reviewing

Für die Zwischenabgabe besteht keine detaillierte Aufwandschätzung. Für die Schlussabgabe wird jedoch eine erstellt.

3 Projektunterstützung

3.1 Tools für Entwicklung, Test & Abnahme

Für das **Projektmanagement** sind folgende Tools im Einsatz:

Bereich	Tool
Vorgehensmodell: Scrum	ScrumDo
Datenaustausch	git, GitHub
Dokumentation	Markdown, Pandoc, Microsoft Office 2016, draw.io, plantuml

Für die **Entwicklung** sind folgende Tools im Einsatz:

Bereich	Tool
Entwicklungsumgebung	ItelliJ IDEA 2017
Programmiersprache	Java 1.8
Versionskontrolle	git, GitLab
Testing	JUnit, Integrationstests, Systemtests
Build	Maven, make
Continuous Integration	Jenkins

3.2 Konfigurationsmanagement

Das Konfigurationsmanagement soll die Einhaltung von Regeln für einen organisatorischen und verhaltensmässigen Lebenslauf eines Produkts und seiner Configuration Items (Konfigurationseinheiten) gewährleisten.

Ein Configuration Item ist eine beliebige Kombination aus Hardware, Software oder Dienstleistung. In diesem Projekt sind diese die Dokumentationen, Komponenten und Interfaces. Im Kapitel "Release-management" sind diese ausführlich aufgelistet.

3.3 Releasemanagement

Das Releasemanagement befasst sich mit der Planung und Durchführung der Veröffentlichung. Dieses Projekt beinhaltet zwei Releases:

- Release 1: Zwischenabgabe (SW08)
- Release 2: Schlussgabe (SW13)

Configuration Item	Release 1	Release 2
Projektmanagementplan	1.0.0	
Systemspezifikation	1.0.0	
TCP-Schnittstelle	1.0.0	
Testplan	1.0.0	
Scrum-Dokumentation	1.0.0	
Logger	1.0.0-SNAPSHOT	
LoggerSetup	1.0.0-SNAPSHOT	
LoggerCommon	1.0.0-SNAPSHOT	
LoggerComponent	1.0.0-SNAPSHOT	
LoggerServer	1.0.0-SNAPSHOT	
StringPersistor	4.0.1	
StringPersistorFile	1.0.0-SNAPSHOT	
GameOfLife	1.0.0-SNAPSHOT	