

Softwaretechnik – Planungsdokumente

Das folgende Planungsdokument wurde von allen Teammitgliedern in Zusammenarbeit erstellt. Die Aufgaben wurden von allen Mitgliedern zu gleichen Teilen bearbeitet. Die Text Produktion erfolgt durch Enrico Piscopello.

Das Team:

Lukas Humpe, Mostapha Ahaduch, Max Hammer, Alexander Spindeler, Wenhao Peng, Enrico Piscopello

Inhalt

Zeitplan	2
Risikobewertung	5
Prozessplan	6

Zeitplan

Folgende Meetings sind im Verlauf des kompletten Projekts geplant:

Kennenlernen	Kick-Off	Bi-Daily	Bi-Daily	Bi-Daily
09.11.2020	14.11.2020	16.11.2020	18.11.2020	21.11.2020
Bi-Daily	Bi-Daily	Bi-Daily	Bi-Daily	Bi-Daily
23.11.2020	24.11.2020	26.11.2020	28.11.2020	30.11.2020

In Abbildung 1 sind die vier Meilensteine Vorbereitung/Initialisierung, Portal aus Bewerberansicht, Portal aus der Sicht des Zulassungsausschuss und Orchestrierung/Bug-Fixing und Testing dargestellt. Die User-Stories und Tasks sind den jeweiligen Meilensteinen zugeordnet. Aufgabenzuordnung, Terminierungen, Abhängigkeiten der Meilensteine und aktueller Status sind ebenfalls in Abbildung 1 dargestellt. Tasks mit fehlender Aufgabenzuordnung wurden vom gesamten Team bearbeitet.

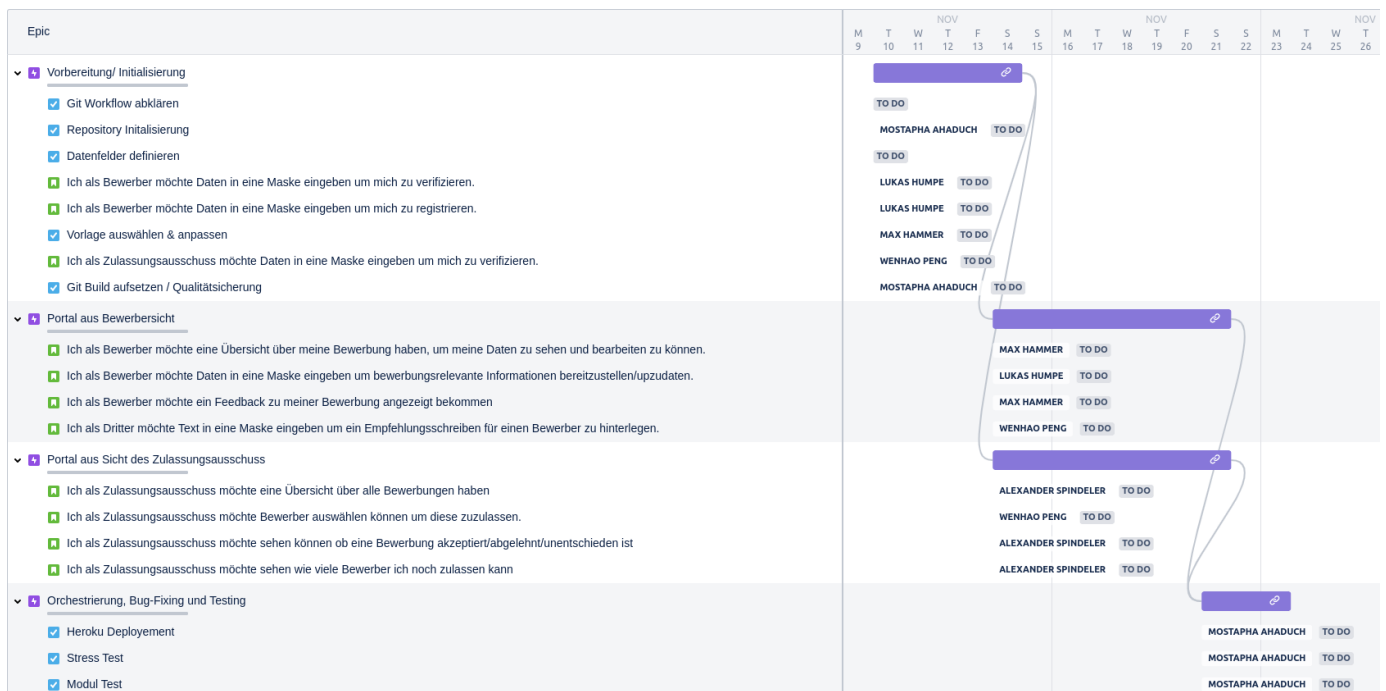


Abbildung 1 Roadmap

Eine detailliertere Ansicht der einzelnen Aufgaben und Meilensteinen ist in Abbildung 2 (siehe Seite 3) ersichtlich. Da alle 2 Tage Meetings stattfinden mit dem Startdatum 14.11.20, ist die Bearbeitung nach Inhalt und Abhängigkeiten der User-Stories und Tasks in vier Sprints unterteilt. Die Ergebnisse der einzelnen Sprints sind ebenfalls definiert und unter dem jeweiligen Sprint kurz erläutert. Der geschätzte Aufwand der einzelnen Aufgaben ist rechts neben dem jeweiligen Task als Zahl zwischen 1 und 4 abgebildet. Eine genaue Erklärung dieser Werte folgt im Prozessplan.

Epics

X

Issues without epic

Vorbereitung/Initialisierung

start date
November 10, 2020

Due date
November 14, 2020

View all details

Portal aus Bewerberbereich

start date
November 14, 2020

Due date
November 21, 2020

View all details

Sprint 1

8 Issues

Build für weitere Programmierung bereitstellen: Repo, Workflow, User Model, Login und Registrierung umsetzen, Design Template ausgeben und anpassen.

SA1-18 Git Workflow definieren

VORBEREITUNG/INITIALISIERUNG

1

100%

🟢

SA1-17 Repository Initialisierung

VORBEREITUNG/INITIALISIERUNG

1

100%

🟢

SA1-34 Datenfelder definieren

VORBEREITUNG/INITIALISIERUNG

1

100%

🟢

SA1-4 Ich als Bewerber möchte Daten in eine Maske eingeben um mich zu verifizieren.

VORBEREITUNG/INITIALISIERUNG

1

100%

🟢

SA1-5 Ich als Bewerber möchte Daten in eine Maske eingeben um mich zu registrieren.

VORBEREITUNG/INITIALISIERUNG

2

100%

🟢

SA1-35 Vorlage auswählen & anpassen

VORBEREITUNG/INITIALISIERUNG

1

100%

🟢

SA1-6 Ich als Zulassungsausschuss möchte Daten in eine Maske eingeben um mich zu verifizieren.

VORBEREITUNG/INITIALISIERUNG

1

100%

🟢

SA1-19 Git Build aufsetzen/Qualitätsicherung

VORBEREITUNG/INITIALISIERUNG

1

100%

🟢

+ Create Issue

Sprint 2

4 Issues

Bewerberdaten-Eingabe schon funktional umgesetzt, Zulassung der Bewerber umgesetzt.

SA1-11 Ich als Bewerber möchte eine Übersicht über meine Bewerbung haben, um meine Daten zu sehen und bearbeiten zu können.

PORTAL AUS BEWERBERSICHT

1

100%

🟢

SA1-10 Ich als Bewerber möchte Daten in eine Maske eingeben um bewerbungselemente Informationen bereitzustellen/updates.

PORTAL AUS BEWERBERSICHT

1

100%

🟢

SA1-12 Ich als Zulassungsausschuss möchte eine Übersicht über alle Bewerbungen haben

PORTAL AUS SICHT DES ZULASSU...

1

100%

🟢

SA1-9 Ich als Zulassungsausschuss möchte Bewerber auswählen können um diese anzulassen.

PORTAL AUS SICHT DES ZULASSU...

1

100%

🟢

+ Create Issue

Sprint 3

5 Issues

Feedback zwischen Bewerber und Zulassungsausschuss umgesetzt, Einbindung der Zulassungsgruppen umgesetzt, Empfehlungsschreiben umgesetzt, Heroku Deployment implementiert.

SA1-13 Ich als Bewerber möchte ein Feedback zu meiner Bewerbung angezeigt bekommen

PORTAL AUS BEWERBERSICHT

2

100%

🟢

SA1-15 Ich als Zulassungsausschuss möchte sehen können ob eine Bewerbung akzeptiert/abgelehnt/unterschieden ist

PORTAL AUS SICHT DES ZULASSU...

1

100%

AS

SA1-43 Ich als Zulassungsausschuss möchte sehen wie viele Bewerber ich noch zulassen kann

PORTAL AUS SICHT DES ZULASSU...

2

100%

🟢

SA1-36 Heroku Deployment

DECKSTREIFUNG, BUC-FRAME U...

1

100%

🟢

SA1-8 Ich als Dritter möchte Text in eine Maske eingeben um ein Empfehlungsschreiben für einen Bewerber zu hinterlegen.

PORTAL AUS BEWERBERSICHT

1

100%

🟢

+ Create Issue

Sprint 4

2 Issues

Testing implementiert, Bug-Fixes

SA1-45 Stress Test

DECKSTREIFUNG, BUC-FRAME U...

1

100%

🟢

SA1-46 Modell Test

DECKSTREIFUNG, BUC-FRAME U...

2

100%

🟢

+ Create Issue

Sprint 1

9

Start sprint

1

100%

🟢

Sprint 2

9

Start sprint

1

100%

🟢

Sprint 3

10

Start sprint

2

100%

🟢

Sprint 4

3

Start sprint

2

100%

🟢

Abbildung 2 Backlog

Die Abhängigkeit der Aufgaben erfolgt von oben nach unten und ist in Abbildung 3 am Beispiel der User-Story „Ich als Bewerber möchte Daten in eine Maske eingeben, um bewerbungsrelevante Informationen bereitzustellen/upzudaten.“ dargestellt. Eine Task bezieht sich auf das Backend und eine Task auf das View-Layer. Das Frontend ist in einer anderen zugehörigen User-Story geregelt.

Portal aus Bewerbers... / SA1-10

Ich als Bewerber möchte Daten in eine Maske eingeben um bewerbungsrelevante Informationen bereitzustellen/upzudaten.

[📎](#) [👤](#) [🔗](#) [⋮](#)

Zu erledigen ▾

Beschreibung
Beschreibung hinzufügen ...

Untergeordnete Vorgänge ⋮ +
0 % fertig

SA1-27 Model Erstellen	ZU ERLEDIGEN
SA1-28 CreateView erstellen	ZU ERLEDIGEN
SA1-29 UpdateView erstellen	ZU ERLEDIGEN
SA1-37 DeleteView erstellen	ZU ERLEDIGEN

Zugewiesene Person
LH Lukas Humpe

Label
Keine

Sprint
SA1 Sprint 2

Story point estimate
2

Abbildung 3 Subtasks

Risikobewertung

Aufgrund von ungenauen und offen formulierten Projektanforderungen, kann es bei der Planung zu einer erschwerten Aufgabendefinition kommen. Da es keine genauen Vorgaben zu Werkzeugen oder Methoden gibt, hängt die Art und Weise der Umsetzung des Projekts stark von Entscheidungen des Teams ab. Eine Möglichkeit, diesem entgegenzuwirken, ist es Datenstrukturen und Programmaufbau möglichst einfach und modular zu halten. Ebenfalls sollte die Zusammenarbeit der einzelnen Teile über standardisierte und verständliche Schnittstellen erfolgen, so dass sich Änderungen auf möglichst wenig andere Teile auswirken.

Eine falsche Einschätzung des zeitlichen Aufwands, kann ebenfalls zu Problemen führen. Wird in manche Aufgaben zu viel Zeit investiert, kann es dazu führen, dass keine Zeit mehr für die Erfüllung anderer Aufgaben bleibt. Vor allem in einem kurzen Zeitfenster, wie es in diesem Projekt gegeben ist, muss dieses Risiko von Beginn an berücksichtigt werden. Eine Strategie, um dieses Risiko zu minimieren, ist die Aufteilung der Aufgaben in kleine Teile mit einem definierten Zeitraum, um eine möglichst genaue Zeiteinschätzung zu ermöglichen.

Ein weiteres Risiko, welches es zu beachten gilt, ist ein auftretender Fehler und die damit verbundene zeitintensive Fehlersuche beim Zusammenführen der einzelnen Programmteile am Ende des Projekts. Dieses Risiko wird vor allem durch fehlende oder wenig Zusammenarbeit der einzelnen Teammitglieder begünstigt. Eine Möglichkeit, diesem Risiko/Problem vorzubeugen, wäre die kontinuierliche Integration. Einzelne Komponenten werden regelmäßig zusammengeführt, dadurch werden Fehler und Probleme frühzeitig erkannt und es kann direkt auf diese reagiert werden. Ebenfalls erhalten die einzelnen Teammitglieder ein ständiges Feedback über die jeweiligen Programmteile. Dieses Risiko wird ebenfalls durch regelmäßige Meetings und Rücksprachen zwischen den einzelnen Mitgliedern minimiert. Diese Strategie wirkt ebenfalls dem Risiko der falschen Zeiteinschätzung entgegen.

Im Gegensatz zu den bereits genannten entwicklungstechnischen Risiken, gibt es ebenfalls Risiken im Bereich des Produkts zu beachten. Ein solches Risiko, tritt erst beim fertigen Programm auf. Es soll möglich sein, Empfehlungsschreiben in das Zulassungsverfahren mit einzubeziehen. Es besteht die Gefahr, dass ein solches Empfehlungsschreiben vom Bewerber selbst geschrieben und hochgeladen wird. Es gibt keine Möglichkeit dieses Schreiben zu validieren. Eine mögliche Lösungsmöglichkeit hierfür wäre, die verpflichtende Angabe der Kontaktdaten des Empfehlungsschreibenden. Somit wäre eine Rückfrage und die damit verbundene Kontrolle der Echtheit des Schreibens auf Seiten des Zulassungsausschusses möglich.

Prozessplan

Zur Kommunikation innerhalb des Teams werden verschiedene Kanäle zu unterschiedlichen Zwecken genutzt. Organisatorische Angelegenheiten werden per E-Mail oder per WhatsApp-Gruppe kommuniziert. Zusätzlich zum ersten Meeting am 09.11.2020 finden weitere Meetings alle 2 Tage, beginnend am 14.11.2020 via Zoom statt, die zum Pair-Programming genutzt werden. Die Aufgabenverteilung findet nach einer gemeinsamen Analyse der jeweiligen Fähigkeiten und Qualifikationen der einzelnen Teammitglieder statt. Zum Organisations-Planungs- und Aufgabenmanagement wird die Webanwendung Jira Software verwendet, um das gesamte Projekt so transparent wie möglich für alle Teammitglieder zu machen. Es wird die Scrum Methode angewandt. Mithilfe von Jira werden sogenannte Sprints definiert, dies sind definierte Zeiträume, in denen ein bestimmtes Arbeitskontingent erledigt wird. Ebenso wird der Verantwortliche der jeweiligen Aufgabe in den Sprints zugewiesen, sowie der geschätzte Aufwand. Um den Aufwand verständlich zu quantifizieren, werden den Aufgaben Zahlen zwischen 1 und 4 zugewiesen, wobei die Zahl 1 für Aufgaben mit geringer Komplexität und die Zahl 4 für Aufgaben mit hoher Komplexität stehen. Eine grobe Zeiteinschätzung beträgt für die 1 4 Stunden, für die 2 8 Stunden, für die 3 12 Stunden und für die 4 16 Stunden. Die Teammitglieder werden in 2 Teams mit je 2 Spezialisten eingeteilt, wobei sich ein Team mit dem Bewerber befasst und das andere Team mit dem Zulassungsausschuss. Dennoch findet ein stetiger Austausch statt, um alle Mitglieder in alle Bereiche zu integrieren. Es wird parallel gearbeitet, die Orchestrierung aller Sichten findet kontinuierlich statt.

Als Git-Workflow wird GitFlow ausgewählt. Es wird ein Master-Branch, ein Develop-Branch und Feature-Branche verwendet. Es wird ein Branch pro User-Story erstellt und es erfolgen standardisierte Commits zu den einzelnen Tasks. Das Repository wird mit dem Jira Scrum Board verlinkt. Hierbei kommt auch die kontinuierliche Integration zum Einsatz, durch die auftretende Probleme frühzeitig erkannt und beseitigt werden können.

Bei Fertigstellung einer Feature-Branch, wird ein Pull-Request vom jeweiligen Teammitglied erstellt. Anschließend muss der Code von zwei anderen Teammitgliedern geprüft werden, bevor er gemerged wird. Auf diese Weise können mögliche Schwachstellen direkt aufgedeckt und verbessert werden, wodurch die Qualität verbessert wird. Ebenso werden aussagekräftige und verständliche Commit Messages zur besseren Verständlichkeit und Nachvollziehbarkeit getätigt.

Als Standard für den Python-Code wird PEP-8 verwendet. PEP-8 ist ein offizieller Style Guide und erleichtert die Zusammenarbeit im Team, durch einheitliche Vorgaben, welche einen einfach zu lesenden und konsistenten Code ermöglichen. Aus denselben Gründen wird Snake-Case angewendet.