

# Abschlussbericht

## **Team: 1**

Mitglied 1: Fabian Amhof, 11910432

Mitglied 2: Martin Auer, 11910432

Mitglied 3: Simon Brandacher, 11906767

Mitglied 4: Jonas Labermeier, 11917476

Mitglied 5: Thomas Oberroither, 11902834

Mitglied 6: Felix Schuler, 11909204

## **Proseminargruppe: 3**

**Datum: 17.06.2021**

## 1. Analyse des Projektablaufs

Beschreiben Sie zumindest folgende Aspekte:

- Vergleich der geplanten Meilensteine mit dem tatsächlichen Verlauf des Projekts
  - Analyse der geleisteten Stunden (Anteil der jeweiligen Entwurfsaktivitäten an der Gesamtzahl der Stunden)
- 
- Die Meilensteine blieben im Verlauf des Projekts zumeist unverändert. Einige Issues konnten sogar bereits vor der eigentlichen Deadline erledigt werden.
  - Wir haben bei unserem Projekt großen Wert auf eine klare Kommunikation zwischen allen Teammitgliedern gelegt, daher haben wir auch wöchentliche Teambesprechungen durchgeführt, um etwaige Probleme frühzeitig erkennen und diese im Team lösen zu können.
  - Dadurch und den generellen Aufwand des zu implementierenden Systems haben wir eine große Anzahl an Mehrstunden gesammelt. Im Laufe des Projekts haben sich dadurch auch 150h und mehr pro Teammitglied angesammelt.

## 2. Analyse des implementierten Systems

Beschreiben Sie zumindest folgende Aspekte:

- War das initiale Konzept (incl. Use Cases, fachliches Klassendiagramm) während der gesamten Entwicklung stabil? Wenn nein, welche Teile haben sich geändert?
  - War die in der SW-Architektur geplante Komponentenstruktur während der Implementierung stabil? Wenn nein, welche Teile haben sich geändert?
  - Konnte die gesamte geplante Funktionalität des Systems realisiert werden?
  - Welche Erfahrungen konnten Sie mit dem Qualitätsmanagement Ihres Systems sammeln?
- 
- Unsere initialen Dokumente (Konzept, UI-Prototyp, etc.) waren bereits sehr stabil. Im Laufe des Projekts wurden einige wenige Use Cases abgeändert. Auch das fachliche Klassendiagramm wurde auf den neuesten Stand gebracht (Kardinalitäten einzeichnen, etc.).
  - Die Komponentenstruktur blieb während der gesamten Entwicklungsdauer über stabil.
  - Wir konnten alle geplanten Features erfolgreich umsetzen.
  - Um die Qualität der Software zu sichern, haben wir mehrere Maßnahmen gesetzt:
    - Änderungen kommen nur via Merge Requests und Code Review durch eine andere Person in den develop-Branch.
    - Auf jedem Merge Request werden noch vor dem Merge automatisch Checks ausgeführt:
      - Unit- und Integrationstests (alle knapp 300 Tests müssen erfolgreich durchlaufen)
      - Coverage (mindestens 80% Instruction-Coverage)
      - SpotBugs/FindBugs
      - Checkstyle (Coding-Style muss dem "Google Java Style Guide" entsprechen)
    - Vor jedem Release (ca. alle zwei Wochen) wurde die Software manuell getestet.
    - Um die Stabilität der Software nicht zu gefährden, haben wir in den Wochen vor der finalen Abgabe keine größeren Änderungen mehr durchgeführt.

## 3. Ursachenanalyse

- Falls unter den ersten beiden Punkten eklatante Diskrepanzen und Probleme festgestellt wurden, worin könnten die Ursachen gelegen haben?
- Was könnte Ihr Team beim nächsten Mal besser machen?

- Wir hätten mehr Zeit in die Planung investieren können, dadurch hätten wir Schnittstellen bereits früher definieren können.
- Die Kommunikation über wöchentliche Teammeetings waren zwar essentiell für unseren Projektfortschritt, jedoch waren diese zeitweise nicht nur produktiv. Dies könnte man eventuell noch verbessern.

## 4. Erfahrungen mit den eingesetzten Werkzeugen

*Beschreiben Sie zumindest folgende Aspekte:*

- *Boten die verwendeten Werkzeuge adäquate Unterstützung bei der Entwicklung?*
  - *Mit welchen Werkzeugen hatten Sie Probleme? Welche haben gut funktioniert?*
- Die von uns verwendeten Werkzeuge boten ausreichende Unterstützung bei der Entwicklung. Einzig und allein Word sehen wir als ungeeignet für manche Dokumente in so einem Projekt an. Die Word-Vorlagen waren hier leider verpflichtend zu verwenden. Aus unserer Sicht wäre das Wiki teilweise besser geeignet gewesen, da es keine Probleme mit Seitenumbrüchen oder unterschiedlicher Darstellung (verschiedene Word-Versionen, LibreOffice, Google Docs) geben kann. Gerade das Testdrehbuch mit den Tabellen war problematisch. Weiters sind Git-Konflikte zwischen Word-Dokumenten nur schwer zu beheben.
  - Wir hatten vorher keine Erfahrung mit GitLab's CI/CD (Pipelines), aber auch das hat gut funktioniert.

## 5. Feedback zur Proseminar-Organisation

- *Fanden Sie die Art der Durchführung des Projekts sinnvoll?*
- *Haben Sie Verbesserungsvorschläge?*

- Das Projekt bot eine gute Möglichkeit, das eigene Wissen bezüglich Software-Architektur, Testen, Implementierung etc. zu erweitern.
- Jedoch waren manche Aspekte am Projekt nicht ideal gelöst:
  - Die Informationen zum Projektablauf sind auf ca. 5–10 Dokumente aufgeteilt. Dies machte es sehr schwer festzustellen, ob man wirklich alle Vorgaben gefunden und entsprechend umgesetzt hat:
    - Aufgabenstellung.pdf
    - 00\_Vorbesprechung.pdf
    - "Workshop Projektmanagement" (W01\_Projektmanagement.pdf) mit weiteren Hinweisen, welche zu beachten waren
    - Anmerkungen in den diversen Templates
    - Zusätzliche Informationen in den PDFs der einzelnen Proseminar-Sessions
  - Auch waren die Einschränkungen bzgl. der verwendeten Technologien nicht förderlich. So ein Softwareprojekt sollte auch die Möglichkeit bieten, sich an neuen Technologien auszuprobieren.
  - Der Abnahmetest war unserer Meinung nach etwas zu früh angesetzt. Die Software, welche wir abnehmen sollten, war zu diesem Zeitpunkt noch nicht wirklich bereit für die Abnahme.
  - Der Gesamtaufwand des Projekts war gemessen an den ECTS des Proseminars zu hoch. Wenn sowohl Implementierung als auch Dokumentation in der Qualität gefordert werden, sollte der Umfang einer der beiden Teile reduziert werden.
  - Aufgrund der COVID-Situation konnten wir uns gerade anfangs nicht treffen um gemeinsam am TimeFlip zu arbeiten. Die Person, die die Hardware hatte, musste neben der Arbeit am Raspberry Pi dann im Endeffekt auch große Teile des

Abnahmetests und des Systemtests übernehmen. Damit auch andere Team-Mitglieder am Spiel arbeiten konnten, war ein Mock/Simulator notwendig.

- Wie in Punkt 4 "Erfahrungen mit den eingesetzten Werkzeugen" ausführlicher beschrieben, hätten wir uns gewünscht, die Word-Vorlagen wären nicht verpflichtend zu verwenden, da sie u. a. Probleme mit Seitenumbrüchen und Versionskontrolle mit sich bringen.