

# Abschlussbericht

## **Team: 1**

Mitglied 1: Clemens Ager, 09817545

Mitglied 2: Bernhard Ertel, 01416637

Mitglied 3: Verena Fritz, 0915606

Mitglied 4: Lorenz Oberhammer, 01016570

Mitglied 5: Michael Sonnerer, 01611745

Mitglied 6: Felix Tschimben, 01605686

Mitglied 7: Claudia Wagner, 08917887

## **Proseminargruppe: 1**

**Datum: 18.06.2021**

## 1. Analyse des Projektablaufs

Der tatsächliche Verlauf des Projekts erfolgte im Großen und Ganzen entsprechend der definierten Meilensteine. In Einzelfällen wurden Issues nachträglich anders priorisiert, d.h. in einen späteren Milestone verschoben. Sämtliche Deadlines zur Abgabe wurden jedenfalls eingehalten.

Die vom Projektteam geleisteten Stunden auf Basis der Zeitaufzeichnungen der einzelnen Mitglieder sowie deren Aufschlüsselung auf verschiedene Aktivitäten sind in der nachfolgenden Tabelle dargestellt. Abweichungen zur ursprünglichen Planung (vgl. Konzeptbeschreibung) können festgestellt werden. Insbesondere wurden mehr Stunden als geplant für die Implementierung aufgewendet. Dabei sollte jedoch berücksichtigt werden, dass die Zuordnung zu den einzelnen Kategorien von Tätigkeiten manchmal nicht ganz eindeutig ist. Zudem erfolgte die Planung noch auf Basis von 6 Teammitgliedern.

<b>Tätigkeit</b>	<b>h</b>	<b>%</b>
<i>LV-Einheit</i>	<i>104.50 h</i>	<i>11.00 %</i>
<i>Softwarekonzept</i>	<i>32.75 h</i>	<i>3.45 %</i>
<i>Systemtest (fremdes System)</i>	<i>30.50 h</i>	<i>3.21 %</i>
<i>Abschlussbericht</i>	<i>29.00 h</i>	<i>3.05 %</i>
<i>Abschlusspräsentation</i>	<i>2.50 h</i>	<i>0.26 %</i>
<i>Einarbeitung, Dokumentation lesen</i>	<i>45.00 h</i>	<i>4.74 %</i>
<i>Software/System Design und Architektur</i>	<i>32.50 h</i>	<i>3.42 %</i>
<i>Implementierung</i>	<i>374.75 h</i>	<i>39.45 %</i>
<i>Tests</i>	<i>48.00 h</i>	<i>5.05 %</i>
<i>Konfiguration und Deployment</i>	<i>17.50 h</i>	<i>1.84 %</i>
<i>Koordination und Projektmanagement</i>	<i>233.00 h</i>	<i>24.53 %</i>
<b>SUMME</b>	<b>950.00 h</b>	<b>100.00 %</b>

## 2. Analyse des implementierten Systems

- Das initiale Konzept war während der Entwicklung größtenteils stabil, es gab Änderungen in der Hinsicht, dass Klassen hinzugekommen sind und die Beziehungen zwischen den Klassen noch spezifiziert worden sind. Die Use Cases haben sich nicht geändert.
- Die Komponentenstruktur konnte so umgesetzt werden wie geplant – analog des 3-Schichten-Models wurden Controller, Services und Repositories aufgebaut.
- Die gesamte geplante Funktionalität des Systems konnte realisiert werden.
- Eine quantitative Qualitätssicherung konnte durch eine hohe Anzahl von Tests und gegenseitigem Checken der Merge Request vor Durchführung erzielt werden. Dadurch konnten Fehler schnell entdeckt werden. Hat sich trotzdem ein Fehler eingeschlichen, konnte dieser mit hotfixes schnell behoben werden.
- Weitere Qualitätssicherung gewährleistete der Abnahmetest durch ein anderes Team. Dadurch konnten letzte Mängel identifiziert und behoben werden.

### 3. Ursachenanalyse

- Ursachen für die Erweiterung des Klassendiagramms waren z.B. neue Klassen für Kategorisierung von Attributen (CubeStatus, GameStatus) und das Hinzufügen von Informationen, die am Anfang noch nicht ersichtlich waren, wie z.B. die Klassen um den Zustand des Würfels abzubilden (vgl. HealthStatus).
- Zu den Deadlines gab es eine erhöhte Anzahl an Merge Requests, zusätzlich war der Git-Server zeitweise nicht erreichbar, wodurch kleinere Fehler übersehen wurden.
- Eine Koordinatorenrolle die Software-Entscheidungen trifft hat gefehlt bzw. es war schwer den Überblick über das ganze System zu behalten, da es doch umfangreich war und nicht jeder an allen Teilen mitwirken konnte. So wurden teilweise Designentscheidungen nicht überall konsistent umgesetzt (z.B. Events über Services zu spielen od. direkt im Controller auszulösen), was die Lauffähigkeit aber in keiner Weise beeinträchtigt.
- Es war manchmal schwierig einen gemeinsamen Termin zu finden, da jedes Team-Mitglied einen vollen Terminkalender hatte, auch bedingt dadurch dass sich bei uns viele in einem Arbeitsverhältnis befinden.

### 4. Erfahrungen mit den eingesetzten Werkzeugen

Es wurden folgende Werkzeuge verwendet:

- **JSF / Prime Faces:**  
Wurde verwendet, da die Teammitglieder bereits alle vertraut mit JSF waren und somit konnte die Zeit, die das Einlernen eines neuen Entwicklungswerkzeugs gebraucht hätte, erspart werden. Bei Problemen, die während der Programmierung auftraten, war besonders die Suche nach Fehlerursachen und Behebung sehr mühsam.
- **GIT:**  
Sehr gutes Tool sowohl für Nachverfolgung / Versionierung / Aufteilen v. Implementierungen / Projektplanung (Issues)! Es wurde wie vorgeschrieben mit dem Git Branching-Model gearbeitet.
- **Spring Boot:**  
Guter Schnellstart durch das Skeleton-Beispielprojekt
- **Docker:**  
Eine Basis-Implementierung des Deployments über Docker war relativ einfach, auch dank des angebotenen Workshops. Wenn man darüber hinaus gehen will, dann erweist sich Docker doch als recht komplex und umfangreich.
- **Discord:**  
Für die Abstimmung innerhalb des Teams war Discord sehr hilfreich. Es wurden jeweils eigene Channels für offene Issues und Merge-Request angelegt, um Probleme zu diskutieren und die Team-Mitglieder zum aktuellen Stand zu informieren. So konnte immer der Überblick über die aktuell bearbeiteten Themen behalten werden. Die Team-Meetings wurden regelmäßig im Voice-Channel abgehalten.

- **BBB:**  
Hier wurden die Proseminar-Einheiten abgehalten und Fortschritte im Projekt vorgestellt. Lief immer stabil, für diesen Zweck gut geeignet.
- **365 Office / Google Docs:**  
Wurde für das gemeinsame Bearbeiten von den Word-Vorlagen (Softwarekonzept / Testprotokoll / Abschlussbericht / Abnahmetest / Systemtest) verwendet. Es gab Formatierungsschwierigkeiten und Probleme bei der gemeinsamen Bearbeitung. Akzeptabel aber nicht ideal.
- **DigitalOcean:**  
Als Web-Hosting Service für die Abschlusspräsentation wurde DigitalOcean verwendet. Die Benutzung des Service wurde in der Lehrveranstaltung Rechnernetze und Internettechnik erklärt. Die domain "timeflipgame.online" wurde über Hostinger erworben und durch Let's encrypt! konnten wir ein TLS Zertifikat einholen. Die Basisvariante von DigitalOcean erwies sich auf Grund des geringen Arbeitsspeichers als ungeeignet und es wurde deshalb ein Droplet mit etwas mehr Arbeitsspeicher gewählt.

## 5. Feedback zur Proseminar-Organisation

- Feedback zum ersten Softwarekonzept war hilfreich (detailliert)
- Die Workshops hätten von mehr als 2 Mitglieder pro Team besucht werden können, da diese online stattgefunden haben
- 2-Wochenrythmus war sehr angenehm
- Idee für das Projekt war spannend / Thema interessant
- Zusätzlich / statt den Word-Vorlagen könnten LaTeX-Vorlagen angeboten werden, würde einiges an Formatierungszeit sparen. Auch die gleichzeitige Bearbeitung in der Gruppe und die Änderungsverfolgung wären um einiges einfacher.