

# Projektbericht

## ***Antrags- und Beschlussverwaltungstool***

Göran Heinemann, Alexander Minukow, Mirjam Müller, Xuan Huy Nguyen, Max  
Patecky, Simon Voges

# Inhaltsverzeichnis

1. Projektplanung	1
1.1. Aufgabenstellung und Auftraggeber (MM)	1
1.2. Ausgangssituation in SE II (MM)	1
1.3. Projektorganisation (MM)	1
1.3.1. Rollenverteilung	1
1.3.2. Kommunikation, Dokumentation und eingesetzte Tools	2
1.4. Ablauf einer Iteration (MM)	4
1.5. Techniken und Praktiken (AM, MM)	4
2. Projektdurchführung	6
2.1. SE I - Inception und Elaboration Phase (MM)	6
2.1.1. Inception Phase (12.12.2022 - 21.12.2022)	6
2.1.2. Elaboration Phase (04.01.2023 - 27.01.2023)	6
2.2. Start von SE II und Construction Phase (GH, AM, MM, HN, MP, SV)	7
2.2.1. Iteration 4 (03.04. - 16.04.2023)	7
2.2.2. Iteration 5 (17.04. - 30.04.2023)	8
2.2.3. Iteration 6 (01.05. - 14.05.2023)	9
2.2.4. Iteration 7 (15.05. - 28.05.2023)	9
2.2.5. Iteration 8 (29.05. - 11.06.2023)	11
2.3. Projektabschluss (MM)	11
2.3.1. Iteration 9 (12.06. - 23.06.2023)	11
3. Ergebnisdokumentation	13
3.1. Ergebnisse (MM)	13
3.1.1. Ziel des Projektes	13
3.1.2. Ergebnisse und Bewertung	13
3.1.3. Fazit	14
3.2. Reflexion der Teammitglieder	15
3.2.1. Reflexion von Göran Heinemann	15
3.2.2. Reflexion von Alexander Minukow	16
3.2.3. Reflexion von Mirjam Müller	16
3.2.4. Reflexion von Xuan Huy Nguyen	18
3.2.5. Reflexion von Max Patecky	19
3.2.6. Reflexion von Simon Voges	21

# Kapitel 1. Projektplanung

## 1.1. Aufgabenstellung und Auftraggeber (MM)

Im Rahmen der Module Software Engineering (SE) I und II war unsere Aufgabe als Team I5 ein Antrags- und Beschlussverwaltungstool für den StuRa der HTW Dresden zu entwickeln. Der StuRa ist das zentrale Organ der Studierenden der HTW Dresden und vertritt diese gegenüber der Hochschule. Im Rahmen seiner Tätigkeit werden beim StuRa verschiedene Anträge gestellt und durch Gremien beraten und beschlossen. Für diesen Vorgang war es unsere Aufgabe ein geeignetes Tool zu entwickeln, da dieser Prozess bisher manuell und damit sehr zeitaufwändig ablief. Mehr zur Problemstellung in der Anforderungsspezifikation.

## 1.2. Ausgangssituation in SE II (MM)

Mit Abgabe des Beleges am Ende von SE I endete die Elaboration Phase für unser Projekt. Die Anforderungen in Form mehrere Use Cases waren fertig ausgearbeitet und mit dem Auftraggeber abgestimmt. Der Entwurf für die Architektur des Projektes stand im Groben und bedurfte noch kleinerer Abstimmungen. Auch wurde ein erster Prototyp implementiert und erste Test Cases beschrieben. Zu Beginn des SE II verkleinerte sich das Team, da unser Teamleiter als Wirtschaftsingenieurstudent SE II nicht belegte. Damit wurden Anpassungen in der Teamstruktur notwendig. Mehr dazu unter dem Punkt [Section 1.3.1, "Rollenverteilung"](#). Mit Beginn von SE II konnten wir also mit der Umsetzung der Use Cases beginnen.

## 1.3. Projektorganisation (MM)

### 1.3.1. Rollenverteilung

Zu Beginn von SE I hat sich das Team auf folgende Rollenverteilung geeinigt:

- Göran Heinemann **Programmierung**/Tester
- Alexander Minukow **Analyse**/Entwurf
- Mirjam Müller **Analyse**/Teamleitung
- Xuan Huy Nguyen **Entwurf**/Programmierung
- Max Patecky **Tester**/Technical-Writer
- Marten Petters **Teamleitung**
- Simon Voges **Deployment-Engineer**

Die Rollenverteilung begründet sich hauptsächlich auf die persönlichen Präferenzen der Teammitglieder in Verbindung mit den verschiedenen Vorkenntnissen und Kompetenzen.

Nach dem Ende des 3. Semesters verließ Marten Petters das Team, da im Studiengang Wirtschaftsingenieurwesen nur das Modul Software Engineering I Teil des Studiengangs ist. Dadurch war eine Anpassung der Rollenverteilung notwendig. Im Zuge dessen wurde auch nochmal die generelle Verteilung der Aufgaben geprüft. Neben der Neuorganisation des Projektmanagements wurde

auch die Rolle des Deployment-Engineers neu vergeben, da diese Rolle für ein Teammitglied naheliegender war, weil er selbst im StuRa Administrator ist. So hatten wir in SE II folgende Rollenverteilung:

- Göran Heinemann (GH) **Programmierung/Deployment-Engineer**
- Alexander Minukow (AM) **Analyse/Entwurf**
- Mirjam Müller (MM) **Teamleitung**/Analyse
- Xuan Huy Nguyen (HN) **Entwurf**/Programmierung
- Max Patecky (MP) **Tester**/Technical-Writer
- Simon Voges (SV) **stellv. Teamleitung**/Verantwortlich für das Erstellen von Diagrammen und Wireframes

### 1.3.2. Kommunikation, Dokumentation und eingesetzte Tools

Zur Kommunikation nutzten wir einen WhatsApp Gruppenchat sowie einen Discord Server. Über WhatsApp wurden Termine abgestimmt und kurze Infos mitgeteilt. Auf unserem Discord Server wurden ausführlichere Infos ausgetauscht sowie unsere Meetings abgehalten (sofern sie online stattfanden). Auch unser Coach Manuela Ziesche und unsere Ansprechpartnerin beim StuRa Theresa Minister waren Teil des Discord Servers um sich im Bedarfsfall schnell und unkompliziert treffen zu können. Die Kommunikation mit beiden zwecks Terminabstimmungen etc. lief ebenfalls über WhatsApp, alle anderen Dinge wurde in persönlichen Gesprächen besprochen. Für die Kommunikation mit unserem Coach war die Teamleitung zuständig, die Terminabsprachen mit dem StuRa übernahm eines unserer Teammitglieder aufgrund seiner Tätigkeit im StuRa. Unsere Meetings in SE II fanden im 14-tägigen Rhythmus statt. Zu Beginn des Semesters trafen wir uns immer Dienstags 15 Uhr in der HTW. Im Laufe des Semesters entschieden wir uns die Meetings wieder online zu machen und verlegten dafür den Termin auf 18 Uhr. Die Meetings dokumentierten wir in einem Google Doc und nutzten dazu eine feste Vorlage, durch die auch der Ablauf eines Teammeetings unterstützt wurde. Alles Besprochene sowie die Aufgabenverteilung für die kommende Iteration wurde dort aufgeschrieben.

# 15. Meeting SE II - 18.04.2023

## Themen

- ☒ Auswertung 3. Iteration
  - ☒ Frontend-Überarbeitung
  - ☒ Deployment-Status
  - ☒ Auswertung der Use-Cases
- ☐ Aufgabenverteilung 4. Iteration
  - ☐ Bibliothek für Markdown raussuchen, einlesen und ggf. einbinden
  - ☐ Umstrukturierung des Frontends in Komponenten
  - ☐ vorab Finalisierung des Deployments
  - ☐ UC1 Antrag stellen umsetzen
  - ☐ ggf. Datenbank überarbeiten/anpassen
  - ☐ Risikoliste pflegen

## Auswertung 3. Iteration

### Frontend-Überarbeitung

- gerade wird die gesamte Website komplett geladen (Seite ist überall als Code nochmal mit drin), soll in einzelne Komponenten aufgeteilt werden, um Code zu vereinfachen/strukturieren und die potentielle Leistung zu erhöhen → Garbage
- Trennung der Seiten in einem separaten Ordner
- Überarbeitung aller Templates, damit diese ordentlich geschachtelt werden
- Verwendung eines Frontend-Frameworks (Bootstrap/TailwindCSS)? -> prüfen
- Vereinheitlichung der gesamten Styles und Ordnung dieser

### Deployment Status

- Bereits intern vom StuRa geregelt
  - Nginx-Proxy mittels NixOS-Deployment
  - muss noch mit eingegliedert werden

## Planung 4. Iteration

### Dokumentation

#### Abbildung 1. Beispiel Vorlage Meeting Google Docs

Auch Meetings mit dem StuRa oder unserem Coach wurden hier dokumentiert. So hatten wir ein zentrales Dokument in dem wir sehr gut den gesamten Projektverlauf nachvollziehen konnten. Das war auch sehr hilfreich um z. B. Absprachen mit dem StuRa später noch einmal nachzuvollziehen und in die Umsetzung des Projektes mit einfließen zu lassen. Für unsere Dokumentation und für die Codeverwaltung nutzten wir GitHub. Hierzu haben wir eine Organisation mit allen Teammitgliedern erstellt in der unser Repository angelegt wurde. Für anstehende Aufgaben erstellten wir entsprechende Issues, sowohl für den technischen Teil als auch für die Dokumentation. Außerdem legten wir für jede Phase des Open UP ein Projekt an dem die verschiedenen Issues zugeordnet wurden. Um in unserem Repository gut an verschiedenen Aufgaben arbeiten zu können, erstellten wir außerdem verschiedene Branches z. B. für die Umsetzung eines Use Cases oder das Deploy-

ment.

## 1.4. Ablauf einer Iteration (MM)

Unsere Iterationen in SE II waren immer genau zwei Wochen lang. Zu Beginn jeder Iteration trafen wir uns als Team und werteten zunächst die vergangene Iteration aus und besprachen dabei unsere erreichten Fortschritte, aufgetretene Probleme sowie unerfüllte Aufgaben. Dafür bereitete die Teamleitung aus dem letzten Meeting eine Liste mit den Aufgaben der letzten Iteration vor, die dann schrittweise durchgesprochen wurde. Beendete Issues wurden in GitHub geschlossen. Anschließend starteten wir mit der Planung der nächsten Iteration. Dazu bereitete die Teamleitung ebenfalls Punkte vor die aus Sicht des Projektmanagements für die kommende Iteration wichtig sind. Die anderen Aufgaben wurden während des Meetings von den Verantwortlichen für ihren Aufgabenbereich benannt und anschließend verteilt. Danach haben wir gemeinsam die Aufgaben priorisiert und Aufgaben die nicht sofort erledigt werden müssen als optional gekennzeichnet. Diese Aufgaben wurden dann meist in den Iterationen nur angefangen oder gar nicht bearbeitet und dann in der darauffolgenden Iteration als feste Aufgabe aufgenommen. Das half bei der längerfristigen Planung der Iterationen, da man so schon einen Blick auf den weiteren Projektverlauf und kommende Aufgaben hatte. Dadurch, dass diese Aufgaben optional waren, entstand auch kein Druck für die Teammitglieder die Aufgabe in der Iteration umzusetzen. Am Ende des Meetings übertrugen wir die Aufgaben in unser GitHub Repository und wiesen sie den verantwortlichen Teammitgliedern zu. Während der Iteration gab es kleine Meetings die je nach Bedarf optional angesetzt wurden. Das konnte z. B. die Vorbereitung eines Treffens mit dem StuRa sein oder es trafen sich Teile des Team um gemeinsam am Projekt zu arbeiten.

## 1.5. Techniken und Praktiken (AM, MM)

Im folgenden sind die Techniken und Praktiken unseres Teams aufgelistet, die während der Projektdurchführung von uns angewendet wurden.

Praktik	Umsetzung
Iterative Developement	Das Projekt wird in zweiwöchigen Iterationen organisiert. Dazu trifft sich das Team mindestens einmal in jeder Iteration um die vergangene Iteration auszuwerten und Ziele für die kommende Iteration festzulegen. Mehrere Iterationen werden den Meilensteinen aus dem OpenUp zugeordnet.
Shared Vision	Die Vision mit Zweck und Problembeschreibung wurde in Zusammenarbeit mit dem Auftraggeber erarbeitet und mit diesem abgestimmt. Auch nach der finalen Abstimmung am Ende von SE I wird der Projektfortschritt regelmäßig mit der Vision und dem Auftraggeber abgeglichen und ggf. auf neue oder geänderte Anforderungen angepasst.

Praktik	Umsetzung
Use Case Driven Developement	Dem gesamtem Projekt liegt das ein Use Case Model zugrunde welches in Zusammenarbeit mit dem Auftraggeber entwickelt wurde. Weiterhin wurden system-weite funktionale Anforderungen beschrieben, die nicht als Use Cases ausgedrückt werden können. Auch Anforderungen zur Benutzbarkeit werden beschrieben.
Evolutionary Architecture	Das Architecture Notebook wird schrittweise bearbeitet. Es wird sich hierbei am Projektfortschritt orientiert. Neue Entwicklungen oder Probleme während der Implementierung werden in der Architektur berücksichtigt.
feste Termine für das Teammeeting	In SE I gab keinen festen Termin, wo das Teammeeting stattfand. In SE II wurde zu Beginn ein fester Termin in einem festen Zyklus ausgemacht, wo die Meetings stattfanden. So gab es eine bessere Planbarkeit für alle Teammitglieder und eine bessere Struktur im Projekt.
Meeting Template	Für die Meetings wurde ein festes Template zur Dokumentation genutzt. So entwickelte sich schnell ein fester Ablauf der Meetings, da man sich gut an der (vorbereiteten) Vorlage orientieren konnte so die Punkte effizient abgearbeitet werden konnten.
klare Trennung der Verantwortlichkeiten	Die verteilten Rollen in unserem Projekt arbeiteten sehr stark in ihrem Aufgabenbereich. Die war mehr eine gewachsene Struktur als eine bewusste Entscheidung. So konnte jedes Teammitglied in seiner Funktion arbeiten und den Überblick behalten. Allerdings führte das auch zu einer unterschiedlichen Verteilung der Arbeitslast.
Risikomanagement	Über den Projektverlauf wurde eine Risikoliste geführt und jede Iteration ständig aktualisiert. Damit konnte man Risiken frühzeitig erkennen und entsprechende Gegenmaßnahmen ergreifen.
Fachaustausch	Im Rahmen von SE II fand innerhalb der Praktikumszeiten ein Fachaustausch jeweils zu den Themen Projektmanagement, Architektur und Testen statt. Dadurch musste man als Team selbst die aktuelle Arbeitsweise bzw. den Projektstand reflektieren und bekam andererseits auch Input von anderen Teams und konnte sehen, dass auch andere Teams mit Herausforderungen zu kämpfen hatte.

# Kapitel 2. Projektdurchführung

## 2.1. SE I - Inception und Elaboration Phase (MM)

### 2.1.1. Inception Phase (12.12.2022 - 21.12.2022)

Ziel der Inception Phase und der ersten Iteration war es sich als Team zu finden, Rollen und Verantwortlichkeiten zu verteilen und sich mit der Aufgabenstellung vertraut zu machen. Auch unseren Coach lernten wir kennen. Wir haben den Einsatz verschiedener Tools diskutiert und schließlich die passenden ausgewählt und eingerichtet. Auch fand ein erstes Treffen mit unseren Ansprechpartnern beim StuRa statt um sich gegenseitig kennenzulernen und über die Wünsche und Anforderungen zu sprechen. Da wir nicht die erste Gruppe waren, die an diesem Projekt arbeitete, war es für uns natürlich auch interessant noch ein paar mehr Hintergrundinformationen zu bekommen. Die erste zentrale Frage für uns war, was und wie viel wir von dem Vorgängerprojekt übernommen wollen oder ob wir komplett neu anfangen wollen. Dies wurde uns vom StuRa freigestellt. Wir entschieden schließlich das Front End bzw. das Design des Vorgängerprojektes zu übernehmen und das Back End neu zu entwickeln. Dazu besprachen wir verschiedene Möglichkeiten der Umsetzung und der Machbarkeit dieser. Nach dieser Entscheidung und dem Input durch den StuRa erarbeiteten wir die Vision und erste Ideen für die Use Cases. Die Inception Phase endete mit dem Erreichen des Meilenstein Lifecycle Objective.

### 2.1.2. Elaboration Phase (04.01.2023 - 27.01.2023)

In der Elaboration Phase hatten wir zwei Iterationen in der unsere Analyse konkretisiert und die Use Cases detailliert ausgearbeitet und beschrieben wurden. Zu Beginn arbeiteten wir noch an Details der Vision und begannen uns in alle anderen Teile der Dokumentation einzuarbeiten. Wir entschieden für das Glossar die Antragsdaten sehr ausführlich dazustellen. Wir trafen uns zu Beginn erneut mit dem StuRa um den Ablauf der Antragstellung besser zu verstehen und eine bessere Vorstellung zu bekommen. Ein Problem des StuRa war das z. T. in die Tagesordnung Punkte und Informationen aufgenommen werden sollten, die nicht als Anträge gestellt werden. Also mussten wir uns überlegen wie sowohl die Anträge aus dem Tool als auch zusätzliche Informationen in die Tagesordnung aufgenommen werden können. Der StuRa nutze bisher ein Pentapad für die Tagesordnung wo alle Anträge und weiteren Punkte aufgeschrieben wurden und auch während der Sitzung Protokoll geschrieben wurde. Wir diskutierten dazu verschiedene Möglichkeiten wie wir diese Flexibilität weiter gewährleisten konnten und entschieden uns schließlich dazu das Pentapad weiter zu nutzen und über eine API in unser Tool einzubinden. Damit fügte sich das Tool gut in den derzeitigen Ablauf beim StuRa ein und wir erhoffen uns dadurch eine höhere Akzeptanz. Wir erarbeiteten schließlich die Use Cases und formulierten die systemweiten Anforderungen. Gleichzeitig wurden erste Entscheidungen für die Architektur getroffen und ausformuliert.

Die Teammitglieder, die nicht primär mit der Analyse beschäftigt und auch verantwortlich für Implementierung und Test waren, arbeiteten sich in das Framework Django ein und erstellten ein erstes Datenbankschema. In der zweiten Iteration der Elaboration Phase besprachen wir nochmals abschließend alle Use Cases im gesamten Team und diskutierten einzelne Punkte nochmals abschließend wie z. B. die Benennung des Use Cases 5. Für die Use Cases 2 und 5 erstellten wir zusätzlich jeweils ein Wireframe um uns über den Aufbau der Seite klar zu werden und diesen auch mit unserem Auftraggeber anzustimmen. Auch starteten wir mit der Beschreibung erster Test-



fälle. der Test An einem Nachmittag trafen wir uns um gemeinsam das Domänenmodell zu entwickeln. Dazu zeichneten wir das Modell auf einem Whiteboard auf und diskutierten über den Aufbau. Das war sehr hilfreich da so nochmal für alle die Zusammenhänge klar wurden und alle auf dem gleichen Stand waren. In der letzten Vorlesungswoche und auch der letzten Woche der Elaboration Phase fand eine Präsentation der Projekte zum Semesterende statt. Dafür mussten wir auch einen ersten Prototyp implementieren. Bevor wir damit beginnen konnten, musste sich jedoch in das Vorgängerprojekt eingearbeitet werden, da wir das Design der Benutzeroberfläche von diesem übernehmen wollten.

Am selben Tag präsentierten wir den Prototypen sowie unseren Stand zur Analyse am Ende von SE I auch dem StuRa um unsere formulierten Anforderungen nochmals abzugleichen und uns Feedback einzuholen. Die restliche Woche beendeten wir die Dokumentation und schauten insbesondere, dass das Erzeugen der PDF funktionierte und alle Abbildungen und Diagramme richtig eingebunden und angezeigt wurden. Das kostete am Ende doch mehr Zeit als gedacht, aber die Arbeit hat sich gelohnt und wir konnten am Freitag den Beleg abgeben. Damit endete auch die Elaboration Phase mit dem Erreichen des Meilenstein Lifecycle Architecture.

## 2.2. Start von SE II und Construction Phase (GH, AM, MM, HN, MP, SV)

Zu Beginn von SE II stand zunächst die Auswertung des Beleges von SE I mit unserem Coach an. Dazu trafen wir uns in der HTW und gingen den Beleg Schritt für Schritt durch und besprachen die einzelnen Teile. Insgesamt erhielten wir hier gutes Feedback, hatten aber auch einige Punkte, die unsererseits nachgearbeitet werden mussten. Diese übernahmen wir gleich als erste Aufgaben in der kommenden Iteration.

### 2.2.1. Iteration 4 (03.04. - 16.04.2023)

Das Projekt startete zunächst mit dem Finden eines geeigneten Termins für die Meetings. Im Gegensatz zu SE I legten wir einen festen Termin fest, an dem wir uns alle zwei Wochen trafen. Wir einigten uns darauf uns alle zwei Wochen Dienstag 15:00 Uhr in der HTW zu treffen. Als erste Aufgabe mussten wir uns als Team neu organisieren, da ein Mitglied aus dem Team ausschied und wir dadurch die Rollen neu verteilen mussten (siehe [Section 1.3.1, "Rollenverteilung"](#)). Dabei nahmen wir uns die Zeit die Rollen nochmals durchzusprechen und abzugrenzen, um das Risiko zu mindern, dass Verantwortlichkeiten unklar verteilt waren. Aus dem Feedbackgespräch zu SE I entstanden die ersten Aufgaben für die 4. Iteration. Besonders das Projektmanagement hatte hier viel Nachholbedarf, sodass zunächst die Iterationspläne und die Projektstruktur aus SE I nachgearbeitet wurde. Auch besprachen wir die Reihenfolge der Umsetzung der einzelnen Use Cases, um so einen groben Zeitplan für SE II zu haben.

Gleichzeitig startete auch die Implementierung des UC 5 "Antragsverwalter anmelden". Dieser stellte eine wichtige Grundvoraussetzung für unser System dar. Zwar wäre die Nutzung des Tools auch ohne Anmeldung möglich, allerdings weniger sinnvoll da nur ein begrenzter Personenkreis Zugang zu den gestellten Anträgen und der Sitzungsverwaltung erhalten sollte. Für die Umsetzung wurden die mitgelieferten Tools von Django genutzt, wodurch eine unkomplizierte Umsetzung möglich war und der Use Case in dieser Iteration abgeschlossen werden konnte.

Auch startete unser (neuer) Deployment-Engineer mit der Vorbereitung des Deployments. Da er

selbst Administrator im StuRa ist, war in diesem Rahmen wenig (extra) Abstimmung mit dem StuRa nötig, da er der Verantwortliche sowohl bei uns im Team als auch im StuRa selbst ist. Das erleichterte für uns als Team die Arbeit, da so ein geringeres Risiko bestand, dass es hier Fehler in der Kommunikation gab. Zuerst haben wir versucht eine Nix-Flake für das Projekt und all seine Abhängigkeiten zu implementieren. Daran sind wir aber aufgrund der Komplexität der Python-Abhängigkeiten gescheitert und haben uns stattdessen dazu entschieden, eine eigene Dockerfile für die Django Anwendung zu schreiben, die in einer Docker-Compose-Datei zusammen mit einer PostgreSQL Datenbank gestartet wird. Dieses Setup ermöglicht viel Flexibilität zum Testen, da man auch nur die Datenbank in Docker starten kann und die Django Anwendung zum Programmieren und/oder Debuggen mit einem lokal installierten Python ausführen kann.

Wir diskutierten die Möglichkeit unser Repository aufzuteilen und den Code und die Dokumentation zu trennen. Letztendlich entschieden wir uns dagegen, da wir mit der bisherigen Lösung gut zurechtkamen und so alle in demselben Projekt arbeiteten. Die Dokumentation führten wir vorerst in den Dokumenten von SE I fort, wenn wir mehr Informationen zum Belegaufbau in SE II erfahren würden, sollte die Ordnerstruktur entsprechend angepasst werden. Weiterhin legten wir für die Implementierung eine Aufteilung der Verantwortlichkeiten in Frontend und Backend fest um so eine bessere Aufgabenverteilung zu ermöglichen. So übernahm MP die Verantwortung für das Frontend und GH für das Backend.

Das Team entschied sich zunächst kein Treffen mit dem StuRa anzusetzen, da wir seit dem letzten Treffen am Ende von SE I keine weitere Funktionalität vorzuweisen hatten und es unsererseits keine Fragen gab. Gleichzeitig wussten wir durch unser Teammitglied beim StuRa, dass seitens des StuRa keine veränderten Anforderungen gab.

### **2.2.2. Iteration 5 (17.04. - 30.04.2023)**

In dieser Iteration sollte mit der Umsetzung des Use Case 1 begonnen werden. Während der Implementierung des Use Case 5 hatten wir jedoch festgestellt, dass wir vor der Implementierung weiterer Use Cases das Frontend unserer Vorgänger überarbeiten mussten. Es wurde die gesamte Website komplett geladen und sollte deshalb in einzelne Komponenten umstrukturiert werden, um den Code zu vereinfachen und eine bessere Übersichtlichkeit zu gewährleisten. Auch die potenzielle Leistung sollte damit erhöht werden. Dafür sollte eine Trennung der Seiten in einzelne Ordner erfolgen und alle Templates überarbeitet werden. Dafür sollte die Verwendung eines Frontend-Frameworks geprüft werden. Letztendlich entschieden wir uns für Tailwind, da hiermit eine gute Gestaltungsmöglichkeit gab, ohne uns zu sehr einzuschränken und es deshalb als am praktikabelsten erachtet wurde. Die Umstrukturierung konnte in dieser Iteration abgeschlossen werden. Da wir eigentlich in dieser Iteration mit dem Use Case 1 starten wollten, legten wir diesen als optionale Aufgabe fest. Mit der Umsetzung konnte dann tatsächlich nicht begonnen werden, so verschoben wir die Implementierung auf die nächste Iteration.

Während der letzten Iteration kümmerte sich unser Deployment-Engineer um das Deployment beim StuRa. Das sollte auf die internen Regelungen beim StuRa angepasst werden und unser System sollte dort eingegliedert werden. Dafür haben wir einen LXC-Container im ProxMox-Cluster mit Ubuntu erstellt und in diesem Docker installiert, einen DNS-Eintrag auf die uns zugewiesene IP-Adresse gesetzt und getestet, wie Nginx als Webserver verwendet werden muss, um per WSGI (Web Server Gateway Interface) unsere Python Anwendung zu hosten.

Da wir uns entschieden hatten die Nutzung des Pentapads (bzw. Etherpad) in unser Tool einzubin-

den, sollte in dieser Iteration eine passende Bibliothek dafür recherchiert und auf Funktionalität geprüft werden. Als Bibliothek für das Markdown auf Python entschieden wir uns zunächst für Martor. Auch eine Prüfung des Datenbankschemas wurde als Aufgabe aufgenommen, um dieses immer auf dem aktuellen Stand der Entwicklung zu halten.

### **2.2.3. Iteration 6 (01.05. - 14.05.2023)**

Im Teammeeting zu Beginn dieser Iteration entschieden wir uns die kommenden Treffen wieder online stattfinden zu lassen, da das für die meisten Teammitglieder einfacher zu organisieren war. Die Meetings fanden weiterhin dienstags statt, jedoch nun um 18 Uhr auf discord.

Die Teamleitung erarbeitete die finale Struktur für den Beleg in SE II nachdem es dazu konkrete Vorgaben von Prof. Anke gab. Dazu wurde auch die Ordnerstruktur des GitHub Repositorys angepasst und für die erforderlichen Dokumente in SE II ein extra Ordner erstellt. Dort wurden die Dokumente für die Abgabe erstellt und eine erste Strukturierung dieser erarbeitet.

In dieser Iteration sollte nun der Use Case 1 "Antrag stellen" umgesetzt werden. Das ist einer der wichtigsten und zentralsten Use Cases für unser Projekt. Der Standardablauf des Use Cases konnte grundsätzlich fertig implementiert werden und es war nun möglich einen Antrag (der verschiedenen Antragsarten) über das Tool zu stellen und sich diesen in der Datenbank anzeigen zu lassen. Dabei übernahmen wir zu einem großen Teil den Aufbau und die Anordnung des Vorgängerprojektes. Es fehlte nur noch der File Upload, um z. B. eine Kostenaufstellung bei einem Finanzantrag mit einzureichen. Weiter kam die Frage auf, ob es sinnvoll ist das Datum der nächsten Sitzung des ausgewählten Referates in der Antragsmaske mit anzuzeigen. Insbesondere Hinblick auf die Möglichkeit einen Eilantrag stellen zu können wurde dies als sinnvoll betrachtet und sollte so umgesetzt werden. Der alternative Ablauf des Use Cases - Stellen eines Eilantrages - wurde noch nicht umgesetzt.

Durch die Implementierung des Use Cases 1 wurde erneut eine Anpassung des Datenbankschemas notwendig. Es kamen weitere Felder zum Antrag hinzu und andere Felder waren nicht mehr notwendig und wurden deshalb entfernt. Das wurde als Aufgabe für die kommende Iteration aufgenommen.

In dieser Iteration fand der erste Fachaustausch im Rahmen von SE II statt. Dazu erklärte die Teamleitung die Organisation unseres Projektes, wie wir unsere Aufgaben sammeln und verteilen. Auch sprachen wir darüber, was bei uns im Team gut läuft und wo es noch Probleme oder Verbesserungsbedarf gibt. Es war sehr interessant die Vorgehensweisen der anderen Teams zu hören und zu schauen mit welchen Problemen sie zu kämpfen hatten.

Da unser eigentlicher Testverantwortlicher hauptsächlich für die Implementierung zuständig war, haben wir beschlossen, unseren Analysten mit dem Testen zu beauftragen. In der Planungsphase wurden die Prioritäten der zu implementierenden Tests in Absprache mit dem Entwicklungsteam festgelegt. Dabei wurden Initialisierungstests erstellt, insbesondere Unittests für die Datenbankinitialisierung.

### **2.2.4. Iteration 7 (15.05. - 28.05.2023)**

In dieser Iteration planten wir ein Meeting mit dem StuRa, um den Projektfortschritt zu präsentieren und uns Feedback zur Gestaltung und zum Aufbau der Webanwendung einzuholen. Dazu ver-

einbarten wir einen Termin am 23.05.2023. In Vorbereitung dessen sollte das Frontend des ABV-Tools noch weiter angepasst werden. Wir beschlossen dafür nochmals verschiedene Wireframes zu erstellen. Zum Einen für die Implementierung, sodass diese schneller geht und nicht noch darüber nachgedacht werden muss, wo welches Feld hin muss und wie die Aufteilung sein soll. Auch wollten wir die Wireframes bei unserem anstehenden Treffen mit dem StuRa nutzen, um auch über die Teile des Front Ends zu sprechen die noch nicht entsprechend angepasst wurden. So erstellten zwei Teammitglieder Wireframes aus Sicht der Analyse und besprachen sie anschließend mit dem Entwickler.

Weiterhin sollte mit der Implementierung der Logik rund um die Sitzungsverwaltung begonnen werden. Das betraf den Use Case 2 "Tagesordnung erstellen", den Use Case 3 "Sitzung abschließen" und den Use Case 4 "Plenumsitzung vertagen". Zunächst sollte Use Case 2 implementiert werden und für diese Iteration optional der Use Case 4. Die Implementierung des Stellen eines Eilantrages aus Use Case 1 verschoben wir erneut in die nächste Iteration. Bei der Umsetzung der Use Cases 2, 3 und 4 stießen wir auf unvorhergesehene Probleme. Der StuRa verwendete bisher zur Dokumentation seiner Sitzungen Etherpad, was als Pentapad vom Chaos Computer Club Dresden (CCC Dresden) gehostet wurde. Leider akzeptierte Pentapad nicht wie ursprünglich geplant Markdown, und so erfolgte die Darstellung in HTML. Die Alternative wäre die Nutzung eines anderen Markdown-Editors gewesen, wir entschieden uns jedoch gegen einen Wechsel, da der StuRa das Pentapad schon lange nutzte und mit der Verwendung vertraut war. Der Wechsel hätte ein hohes Risiko bedeutet, dass das Tool von den Anwendern nicht oder nur schwer akzeptiert werden würde. Auch eine Verwendung des Pentapads des CCC Dresden war nicht möglich, da wir keinen Zugang zu dem API-Key erhalten konnten. Deshalb mussten wir das Etherpad schließlich selbst hosten und ein identisches Etherpad mit den gleichen Plugins aufsetzen. Das nahm sehr viel Zeit in Anspruch und verschob den Zeitplan nach hinten. Sehr zeitaufwändig war ebenfalls sich in die API-Schnittstelle einzuarbeiten, da die Dokumentation sehr schwer verständlich und veraltet war.

Am 23.05.2023 trafen wir uns mit dem StuRa um den aktuellen Stand zu präsentieren und insbesondere das Front End abzustimmen und kleinere Fragen dazu zu klären. So hatten bisher alle Sitzungen einen festen Zyklus hinterlegt in dem die Sitzungen stattfinden sollen. Im Gespräch ergab sich dann, dass viele Sitzungen (besonders der Referate) doch sehr unregelmäßig stattfinden und es besser wäre, wenn beim Anlegen einer Sitzung das Datum manuell festgelegt werden kann. Wir hatten eine gute Diskussion über kleine Features und Gestaltungsfragen, aber insgesamt entsprach das System den Erwartungen des StuRa und wir konnten guten Gewissens weiter an unserem Projekt arbeiten. Aus dem Wunsch Sitzungen manuell anlegen zu können entstand schließlich der Use Case 7 "Sitzung anlegen".

In der gleichen Woche traf sich die Teamleitung mit unserem Coach Manuela Ziesche um den aktuellen Projektstand zu besprechen und Fragen zur Dokumentation zu klären.

In dieser Iteration haben wir mit dem Testen begonnen. Zunächst haben wir das System bei einem Teammitglied auf dem privaten Rechner eingerichtet und überprüft, ob die Einrichtung wie vorgesehen funktioniert. Der Smoke-Test war erfolgreich, und das System konnte fehlerfrei auf dem Rechner eingerichtet werden. Anschließend haben wir einen Unittest für den Login und Logout implementiert und sowohl positive als auch negative Testfälle berücksichtigt.

### 2.2.5. Iteration 8 (29.05. - 11.06.2023)

Zu Beginn der 8. Iteration fand der Fachaustausch zum Thema Architektur statt. Dabei stellte unser Architekt den Aufbau unseres Projektes vor erklärte unser Vorgehen und welche Probleme dabei auftraten. Wir bekamen dabei auch einen guten Einblick in die Entwürfe der anderen Teams und deren individuellen Herausforderungen.

In der folgenden Woche fand der letzte Fachaustausch zum Thema Testen statt. Zwei unserer Teammitglieder erklärten wie das Testen in unserem Projekt organisiert ist und welche Methoden wir dabei einsetzen. Auch hier half es zu sehen, auf welche Konzepte die anderen Team setzten und welche Probleme die anderen Teams bei der Umsetzung hatten. Auch konnten wir noch offene Fragen zum Testen mit Ruben-David Kraus klären, der uns als Ansprechpartner für unsere Projekte zur Verfügung steht.

In dieser Iteration wurde weiter an den Use Cases 2 und 4 gearbeitet. Nach den Problemen mit dem Etherpad in der letzten Iteration musste nun die Vorlage für die Tagesordnung erstellt werden. Die Formatierung, das Übertragen der Daten aus dem ABV-Tool und die richtige Darstellung nahm ebenfalls sehr viel Zeit in Anspruch. Am Ende war jedoch eine Generierung der Tagesordnung mit allen erforderlichen Daten möglich.

Außerdem wurde der neue Use Case 7 "Sitzung anlegen" umgesetzt, welcher sich aus dem Gespräch mit dem StuRa ergeben hatte. Dafür mussten die vorher in einem festen Zyklus stattfindenden Sitzungen so umgewandelt werden, dass der Antragsverwalter das Datum der Sitzung selbst bestimmen kann. Dies konnte problemlos geändert werden. Auch die Logik für das Stellen eines Eilantrages und des File Uploads wurde nun umgesetzt. Damit war nun der Use Case 1 endgültig abgeschlossen.

Für das Testen haben wir außerdem eine GitHub Actions Pipeline eingerichtet, die automatisch vorhandene Tests durchführt, sobald eine Änderung in den Master-Branch gemerged oder committet wird. Zusätzlich haben wir weitere Unittests für das Stellen eines Antrags implementiert. Des Weiteren haben wir begonnen, intensive manuelle End-to-End-Tests für die Standardprozesse durchzuführen.

## 2.3. Projektabschluss (MM)

### 2.3.1. Iteration 9 (12.06. - 23.06.2023)

Die neunte und letzte Iteration sollte gleichzeitig die Transition Phase des Projektes sein. Da die Umsetzung der verbliebenen Use Cases länger dauerte als ursprünglich gedacht, konnte der Projektabschluss nicht so ablaufen wie geplant. Wir wollten jedoch die Use Cases rund um die Sitzungsverwaltung gern noch abschließen, um das Projekt zu einem guten Abschluss zu bringen und dem StuRa kein halbfertiges System übergeben zu müssen. Damit implementierten wir auch in der letzten Iteration noch, und das Ende der Construction Phase und die Transition Phase fielen in den selben Zeitraum. Insbesondere mussten noch der Use Cases 3 "Sitzung abschließen" und der Use Case 6 "Beschlüsse einpflegen" abschließend implementiert werden.

Gleichzeitig haben wir unsere Tests intensiviert und weiter umfangreiche manuelle End-to-End-Tests durchgeführt. Dabei lag unser besonderer Fokus auf Negativtests, bei denen wir gezielt Fehler im System gesucht haben. Dadurch konnten wir einerseits die Fehlertoleranz ermitteln und andererseits kritische Bugs finden und beheben.

Zum Abschluss glichen wir nochmals die Anforderungsspezifikation mit dem tatsächlichen System ab und prüften ob die Abläufe richtig beschrieben waren. Hierbei gab es einige Anpassungen bei den Use Cases 2 und 3, da der Ablauf nun doch anders umgesetzt wurde als ursprünglich geplant. Auch die systemweiten Anforderungen und das Domänenmodell wurde nochmals mit dem Projektstand abgeglichen und kleine Anpassungen vorgenommen. Die Entwickler-, Benutzer und Testdokumentation wurde ebenfalls abschließend bearbeitet und im gesamten Team vorgestellt und besprochen. Als letzte Dokumentation wurde schließlich noch die Anwenderdokumentation erstellt. #Dazu wurden verschiedene Screenshots zu den wichtigsten Funktionalitäten erstellt und daran die wichtigsten Abläufe beschrieben.

Für das finale Deployment haben wir auf dem Cluster im StuRa, nicht wie zuerst angenommen, den bereits für das SE-Projekt im letzten Jahr erstellten Container verwendet, weil dieser mit Ubuntu 20.04 installiert wurde, welches leider nicht die aktuellsten Versionen von Docker und Docker-Compose bietet. Stattdessen haben wir eine Virtuelle Maschine erstellt, da es offiziell nicht empfohlen ist in Lxc-Containern Docker zu betreiben und die neue stabile Version von Debian (Bookworm) neu herausgekommen ist. In der Virtuellen Maschine läuft unsere Anwendung jetzt als Docker-Compose Stack und ist unter der Domain [abv.htw.stura-dresden.de](http://abv.htw.stura-dresden.de) erreichbar. Außerdem haben wir, um eine vertrauliche Übermittlung der Eingaben von unseren Nutzern zu gewährleisten, ein TLS-Zertifikat von LetsEncrypt installiert.

Der Abnahmetermin für das ABV-Tool sollte eigentlich am 21.06.2023 stattfinden. Jedoch konnten hier einige Teammitglieder von vorneherein nicht und als schließlich noch ein weiteres Teammitglied krankheitsbedingt ausfiel, verschoben wir den Termin in Abstimmung mit Prof. Anke auf den 27.06.2023.

# Kapitel 3. Ergebnisdokumentation

## 3.1. Ergebnisse (MM)

### 3.1.1. Ziel des Projektes

Ziel des Projektes war es ein funktionierendes und anwenderfreundliches Tool zur Antrags- und Beschlussverwaltung zu entwickeln und so die Arbeit des StuRa der HTW Dresden zu erleichtern. Dabei war es uns besonders wichtig, das Tool gut in den bestehenden Ablauf zu integrieren.

### 3.1.2. Ergebnisse und Bewertung

#### UC 01 Antrag stellen

Dieses Ziel wurde erreicht. Es ist möglich über das Tool verschiedene Anträge zu stellen sowie sie automatisch der nächsten Sitzung zuzuordnen. Sie werden in der Datenbank gespeichert und sind in einer Gesamtübersicht aller Anträge sowie über die Sitzung einsehbar. Sie bekommen eine eindeutige, fortlaufende Antragsnummer zugewiesen. Der Antragsteller bekommt nach Absenden des Antrages eine Eingangsbestätigung per Mail.

#### Sitzungsverwaltung - UC 02 Tagesordnung erstellen, UC 03 Sitzung abschließen und UC 04 Plenumsitzung vertagen

Nach einigen Hürden konnten wir auch diese Use Cases umsetzen. Hier stießen wir während der Implementierung auf die größten Herausforderungen, da der ursprüngliche Plan nicht so umsetzbar war wie gedacht. Das Suchen nach einer Lösung und die anschließende Umsetzung kostete sehr viel Zeit. Am Ende ist es jedoch gelungen, die Use Cases rund um die Sitzungsverwaltung umzusetzen und das gewünschte Ergebnis zu erzielen. Es ist möglich für eine Sitzung eine Tagesordnung zu generieren und die benötigten Daten aus den Anträgen in das Pentapad zu überführen. Die Sitzung kann nach Abschluss aller Anträge als beschlossen markiert werden. Außerdem kann eine Sitzung vertagt werden, falls diese nicht wie geplant stattfindet. Für die Sitzung kann dann ein neues Datum festgelegt werden und wird dann zusammen mit allen Anträgen auf diesen Tag verschoben.

#### UC 05 Antragsverwalter anmelden

Diesen Use Case haben wir als erstes umgesetzt. Durch die Nutzung von mitgelieferten Tools von Django war eine unkomplizierte Umsetzung möglich. Durch die Anmeldung wird das Tool in einen öffentlichen und einen internen Bereich unterteilt. Nur berechtigte Mitglieder des StuRa haben Zugang zur Antrags- und Sitzungsverwaltung.

#### UC 06 Beschlüsse einpflegen

Dieser Use Case wurde gemeinsam mit Use Case 3 implementiert, da abgeschlossene Anträge Voraussetzung für den Abschluss der Sitzung sind. In einem extra Fenster kann der Beschluss mit Beschlusstext und Abstimmungsergebnis erfasst und so der Antrag abgeschlossen werden. Er wird in der Übersicht anschließend als "abgeschlossen" markiert.

#### UC 07 Sitzung anlegen

Dieser Use Case entstand erst im weiteren Projektverlauf durch die Abstimmung mit dem Auftraggeber. Er konnte jedoch schnell und unkompliziert umgesetzt werden, so dass Sitzungen je nach Wunsch seitens des StuRa angelegt werden können. Dabei wird für ein ausgewähltes Referat ein Sitzungsdatum festgelegt. Falls an diesem Tag bereits eine Sitzung des Referates stattfindet wird der Nutzer mit einer entsprechenden Fehlermeldung gewarnt. So werden redundante Sitzungen vermieden.

### **Umsetzung weiterer Anforderungen**

Auch die systemweiten Anforderungen konnten gut umgesetzt werden. Mit der Antragsstellung kann sofort nach Aufruf der Seite und einem weiteren Klick begonnen werden. Überall werden Hilfstexte und Warnungen angezeigt, die es sowohl Antragsteller als auch Antragsverwalter erleichtern sollen. Auch die persistente und verlustlose Speicherung aller Daten konnte sichergestellt werden. Nur ein berechtigter Personenkreis hat Zugang zur Antrags- und Sitzungsverwaltung, was eine der wichtigsten Anforderungen für die Arbeitsweise des ABV-Tools ist.

### **3.1.3. Fazit**

Insgesamt werden wir dem StuRa der HTW Dresden ein funktionierendes System übergeben können. Wir haben hierbei versucht alle Wünsche und Anforderungen des StuRa so umzusetzen, dass das ABV-Tool akzeptiert und auch wirklich genutzt wird. Das erhoffen wir uns vor allem durch die flexibel gestaltete Tagesordnung, die sich sehr gut in den bisherigen Ablauf des StuRa integriert und die Nutzung des Pentapads bzw. Etherpads weiterhin möglich macht.



## 3.2. Reflexion der Teammitglieder

### 3.2.1. Reflexion von Göran Heinemann

Dieses Software-Engineering Projekt war für mich besonders wichtig, um die Zusammenarbeit in einem „größeren“ Team zu probieren und erlernen, da ich bisher immer nur alleine, bzw. maximal zu zweit an Software Projekten gearbeitet habe. Außerdem ist mir bewusst geworden, wie wichtig gute Dokumentation ist, gerade wenn die Anzahl der Teammitglieder so groß ist, dass man nicht mehr alle Änderungen von jedem Einzelnen nachvollziehen kann. Des weiteren konnte ich einige Einblicke in die Zusammenarbeit mit den verschiedenen Stakeholdern gewinnen, was ich bei persönlichen Projekten bisher noch nie so beachten musste. Dazu kam selbstverständlich die Anwendung der Methoden, die wir im Laufe der Lehrveranstaltung erlernt haben. Diese schienen mir am Anfang etwas „übertrieben“, aber ich habe daraus besonders mit Blick auf mein zukünftiges Berufsleben wichtige Vorgehensweisen kennengelernt, die ja besonders bei Firmen mit eventuell wechselnden Angestellten wichtig werden.

Auch aus technischer Sicht habe ich einiges lernen können. Da ich im ersten Semester den Tätigkeitsschwerpunkt Softwareentwickler hatte, musste ich mich in die Programmierung mit Python3 neu einarbeiten. Zuvor hatte ich nur mit Python 2.7 zu tun und ich muss sagen, dass die Sprache sehr gut geeignet ist, komplexe Geschäftsprozesse schnell und eindeutig abzubilden. Nachdem das Projekt im zweiten Semester durch mehr Features komplexer wurde und immer mehr Abhängigkeiten bekam ist mir allerdings aufgefallen, was für eine Menge an „Technical Debt“ man sich schnell einfangen kann, wenn man in der Paketverwaltung nicht ganz genau aufpasst. Des weiteren ist mir besonders in der Rolle des Deployment-Engineers und als StuRa-Admin aufgefallen, dass das Python-„Ökosystem“ meiner Meinung nach nicht sehr gut für reproduzierbare Entwicklungsumgebungen und natürlich Deployments geeignet ist. Diese Ansicht habe ich vermutlich aber nur, weil wir im StuRa derzeit versuchen so viel wie möglich mit Nix, bzw. NixOS zu paketieren und zu Hosten. Leider lässt sich unsere Django Anwendung auf Grund von Inkonsistenzen in der Auslieferung der Pakete von PyPi.org nicht mit Nix ausliefern, sodass die Anwendung leider nur über Umwege in unsere Zentral verwaltete Infrastruktur integriert werden kann. Diese letzten Punkte sollen aber nicht nur wie eine „Rant“ über Python klingen, sondern eher meine Erkenntnisse für etwaige zukünftige Anforderungen an Projekte, die wir als Bereich Administration Rechentechnik an zukünftige Anwendungen stellen, zusammenfassen. Die Kompromisslösung, das Projekt-Deployment in einer Docker-Compose Datei zu beschreiben, welche sowohl auf den lokalen Entwicklermaschinen, als auch auf dem Server im StuRa verwendet werden kann ist daher meiner Meinung nach gut geeignet, da einmal gebaut Docker-Images in der Regel nicht geändert werden und auch viel Spielraum bei der Wahl des darunterliegenden Systems lassen. Das erstellen der Docker Infrastruktur hat mich außerdem dazu bewegt mich tiefer damit auseinanderzusetzen, wie man eigene Container erstellt und gewährleistet, dass diese reproduzierbar gebaut und ausgeliefert werden können.

Abschließend kann ich sagen, dass ich eine Reihe neuer Methoden und Technologien kennengelernt habe, die mir bereits jetzt, im Studium, im Rahmen meiner Tätigkeit im StuRa und besonders später, im Rahmen einer Tätigkeit als Softwareentwickler, oft von Nutzen sein werden.

### 3.2.2. Reflexion von Alexander Minukow

Rückblickend war das Projekt ein wertvoller Einblick in das Software-Engineering, bei dem ich gezwungen wurde, mich mit allen Bereichen eines Softwareprojekts auseinanderzusetzen. Dabei traten natürlich viele Probleme und Herausforderungen auf, aber die Bewältigung dieser Schwierigkeiten war der beste Weg, aus Fehlern und Fehleinschätzungen zu lernen. Das Projekt ermöglichte mir umfangreiche Erfahrungen in Bereichen wie Analyse, Entwurf, Entwicklung und Testing zu sammeln. Doch eines hatten sie alle gemeinsam: Softskills. Denn um ein umfangreiches Softwareprojekt erfolgreich umzusetzen, reicht es nicht aus, nur in einem Bereich gut zu sein. Die Art und Weise, wie man kommuniziert, plant und zusammenarbeitet, prägt den Erfolg des Projektes maßgeblich. Dies betrifft nicht nur den Umgang innerhalb des Teams, sondern auch den Umgang mit dem Kunden. Die Fähigkeit, eine angemessene Beziehung zum Kunden aufzubauen, ist der Grundbaustein jedes Softwareprojekts und genauso wichtig wie die strukturelle Organisation des Projektablaufs und die regelmäßige Absprache innerhalb des Teams. Darüber hinaus habe ich erkannt, dass es von Vorteil ist, eng im Team zusammenzuarbeiten, um Wissen zu teilen und die Stärken der Teammitglieder optimal zu nutzen.

Die Auseinandersetzung mit dem Thema DevOps mithilfe von GitHub war ebenfalls eine wichtige und wertvolle Erfahrung. Durch das Projekt konnte man grundlegende Konzepte wie den Umgang mit Issues und das Merging erlernen, aber auch fortgeschrittene Funktionen wie Actions/Pipelines von GitHub nutzen.

Des Weiteren wurde mir vermittelt, wie wichtig es ist, gemeinsam mit dem Team Modelle zu erstellen und diese durchzusprechen. Regelmäßige konstruktive Diskussionen sind entscheidend, um wichtige Entscheidungen kritisch zu hinterfragen und Denkfehler eines Einzelnen zu vermeiden.

Die Entwicklung des Antrags- und Beschlussverwaltungstools war sehr umfangreich, und wir mussten uns intensiv mit dem Workflow des StuRa auseinandersetzen. Dadurch erhielt ich einen wertvollen Einblick in das Thema Analyse. Zudem mussten wir uns mit Programmiersprachen, Frameworks und anderen Technologien auseinandersetzen, mit denen wir zuvor keine Erfahrung hatten. Dadurch konnte ich wertvolle Fähigkeiten im Bereich Testing und Analyse erlernen.

Für meine zukünftigen Softwareprojekte habe ich mir vorgenommen, die Aufgaben strukturierter zu planen und abzuarbeiten. Außerdem werde ich verstärkt Absprachen mit meinen Teammitgliedern treffen, um gemeinsame Vorgehensweisen zu besprechen.

Abschließend kann ich sagen, dass das Projekt ein Erfolg war und wir als Team stolz auf unsere umgesetzte Webseite sein können.

### 3.2.3. Reflexion von Mirjam Müller

Für mich war das ABV-Tool das erste Projekt, wo über einen längeren Zeitraum systematisch eine Software entwickelt werden sollte. Ich übernahm in diesem Zeitraum zwei Rollen. In SE I war ich Teil des Analyseteams und in SE II übernahm ich die Rolle der Teamleitung, da unser bisheriger Projektleiter aus dem Team ausschied. In beiden Rollen konnte ich eine gute Einsicht in die damit verbundenen Aufgaben erlangen und mir hierfür Wissen und Methoden aneignen. In der Analyse konnte ich lernen wie man aus einer Problemstellung und einem bestehenden Ablauf bei einem Auftraggeber Use Cases ableitet und Anforderungen formuliert. Das Denken in Use Cases und das Ausformulieren dieser musste ich erst lernen, bedeutete für mich jedoch eine große Bereicherung.

Im Projektmanagement konnte ich lernen, Aufgaben herauszuarbeiten, zu priorisieren und ihren Aufwand einzuschätzen. Natürlich erhielt ich auch einen guten Einblick in die anderen Rollen, auch wenn ich hier nicht aktiv beteiligt war. Auch die Arbeit im Team mit Zuhilfenahme von Tools wie GitHub und Asciidoc waren mir zu Beginn von SE I neu. Hier habe ich viel dazu lernen können und die Möglichkeiten die insbesondere GitHub bietet werde ich auch weiterhin nutzen.

Als Team fiel es uns nicht so leicht die agile Entwicklung richtig umzusetzen. Besonders in SE I haben wir die Arbeitsweise weniger umgesetzt, sondern mehr nebeneinanderher und ohne wirklichen Plan gearbeitet. In SE II war es das Ziel als Team besser zusammenzuarbeiten und so effektiver zu sein. Das ist uns auch größtenteils gelungen. Das Projekt hatte mehr Struktur und die regelmäßigen Meetings halfen einen festen Ablauf zu integrieren. Rückblickend wäre es vielleicht besser gewesen die Meetings einmal pro Woche, statt alle zwei Wochen stattfinden zu lassen, da man so den Fortschritt besser überwachen und Aufgaben kleinteiliger verteilen kann. Trotzdem blieb eine starke Abgrenzung zwischen den Rollen, was zu einer unterschiedlichen Arbeitsbelastung führte. Auch die unterschiedlichen Wissensstände und Fähigkeiten, die jedes Teammitglied mitbrachte und die persönliche Motivation des Einzelnen führten zu der unterschiedlichen Arbeitsbelastung. Unsere Arbeitsweise war sicher nicht perfekt, aber aus den Dingen, die nicht so gut liefen, kann man gut lernen und sie bei anderen Projekten besser machen.

Am Ende können wir trotz einiger Schwierigkeiten dem StuRa ein gut funktionierendes System übergeben und stolz auf das Gesamtergebnis sein. Das ist nicht zuletzt auch dem Einsatz einiger weniger Teammitglieder zu verdanken, die in das Projekt sehr viel Zeit investiert haben. Ich konnte in diesem Projekt viel dazulernen, musste mich aber auch einigen Herausforderungen stellen. Generell habe ich einen guten Überblick über die Vielseitigkeit des Software Engineerings erhalten und viele Erfahrungen gemacht die man für zukünftige Projekte nutzen kann.

### 3.2.4. Reflexion von Xuan Huy Nguyen

Diese Reflexion gilt der Gruppenarbeit im zweisemestrigen Modul Software Engineering. Wir haben hierbei das Projekt I5 Antrags- und Beschlussverwaltungstool für den StuRa bearbeitet. Die Aufgabe war es ein Tool für den StuRa der HTW Dresden zu entwickeln, welches die Möglichkeit bietet digital Anträge zu stellen, diese in der gewünschten Reihenfolge eine Tagesordnung einzufügen, sie dann zu beschließen und schließlich um sie zu archivieren. Damit sollte die nervenaufreibende und zeitfressende Papierarbeit von den jeweiligen Sitzungen eingeschränkt werden und der Workflow verbessert werden. Ich habe im Rahmen dieses Projektes die Rolle im Entwurf eingenommen.

Am Anfang hatte ich Vorkenntnisse in der Programmierung von C, C++, Python und einigen anderen Sprachen, allerdings waren diese nur grundlegend vorhanden und ich hatte kaum Erfahrungen, was die Entwicklung von größeren Anwendungen angeht und vor allem nicht in einer Gruppe. Das war der Hauptgrund, warum ich sehr aufgeregt auf diese Arbeit war. Es gab mir die Möglichkeiten meine aktuellen Kenntnisse aktiv mit anderen zu vergleichen, teilen und fortlaufend zu verbessern. Zumindest waren dies meine Erwartungen.

Zu Beginn hatten wir einen recht holprigen Start, da wir Probleme mit Terminfindungen in der Gruppe hatten und so auch ein Workflow nicht wirklich existierte. Allerdings lernten wir uns mit der Zeit kennen und haben eine guten Arbeitsrhythmus gefunden. Generell bin ich sehr froh über unsere Gruppendynamik. Durch ein ähnliches Denken und einer guten Kritikfähigkeit von jedem Teammitglied konnten wir Entscheidungen zu Design, grundlegende Architektur und den Programmierprinzipien sehr schnell fällen. Ich habe sehr viel über webbasiertes Programmieren und die Arbeit in GitHub von Max gelernt. Dafür möchte ich ihm hier auch danken!

Dennoch sind nicht nur gute Dinge vorgefallen. Der StuRa als Auftraggeber und Stakeholder schien aus meiner Sicht uninteressiert am Auftrag, da jedes Meeting und jede Anfrage von uns ausging. Es ist mir auch bewusst, dass im StuRa auch nur andere Studierende tätig sind. Jedoch bin ich der Meinung, dass man als Auftraggeber die Verantwortung trägt einen Fortschritt durch regelmäßiges Nachfragen vom Stand und Ausüben von etwas Druck zu erzielen. Was auch nicht gut lief war, dass gerade in der Motivation es mir zum Anfang von SE II fehlte. Dies führte zu einem sehr geringen Fortschritt des gesamten Projektes zu der Zeit. Doch ich konnte mich mithilfe der Gruppe fangen und wir können ein Produkt präsentieren, auf das wir stolz sind. Besonders stolz bin ich allerdings nur auf eine kleine Sache. Das grundlegende Datenbankschema, welches wir uns zu Anfang überlegten, ist fast eins zu eins auch so in die finale Version umgesetzt worden. Natürlich sind deutlich mehr Spalten pro Tabelle hinzugekommen, aber das war uns von Anfang an klar.

Der große Dokumentationsaufwand schien mir teilweise etwas überflüssig. Dennoch bin ich der festen Überzeugung, dass ich nur daraus gelernt habe. Insgesamt war die Projektarbeit ein voller Erfolg. Sie war eine gute Mischung aus Spaß und Freundschaft, Disziplin und Einzelarbeit, Austausch und Auffassung, Lernen und Verstehen und Frustration und Erfolg. Die verbesserten Softskills, Kenntnisse und erlernten Methoden werden mir weiterhin nützlich bleiben.

### 3.2.5. Reflexion von Max Patecky

Als Team sollten wir im Rahmen des Moduls Software Engineering I und II eine Webanwendung für den Student:innen-Rat der HTW Dresden programmieren. Unsere primäre Aufgabe war dabei, die Workflows des StuRa zu vereinfachen und so die Abläufe einheitlich und fehlerfrei zu optimieren. Zu Beginn des Semesters haben wir – nachdem unser voriger Teamleiter Marten die Gruppe verlassen hat – die Rollen neu zugeteilt. So wurde ich letztlich Tester & Technical-Writer. Im Verlauf des 2. Semesters haben sich diese jedoch zwangsweise geändert.

So begann das 2. Semester damit, dass wir uns grundlegend neu organisieren mussten. Wir haben versucht, uns an dem Prozess des OpenUP zu orientieren und auch entsprechend nach Iterationen in verschiedenen Phasen zu arbeiten. Dies lief auch die ersten drei Iterationen gut, verlief sich jedoch immer mehr gegen Ende des Semesters. Meiner Auffassung nach lag dies daran, dass wir keine festen Aufgaben vergeben und uns selbst nicht wirklich Druck gemacht haben. Ebenfalls hatten wir so einige Probleme mit unserem Auftraggeber – dem StuRa – worauf ich später noch eingehen werde. Daher ging die Entwicklung nur langsam und schwer voran, woran sich jedoch etwas ändern musste.

Dadurch habe ich mich im Laufe des Semesters zum alleinigen Entwickler der Software ernannt, damit das Projekt weiter voranschreitet. Einerseits konnte ich so die vielen Funktionen und Möglichkeiten mit Django erkunden, andererseits hätte ich mir gerne jemanden gewünscht, der mit mir die Anwendung entwickelt. Wie man eine Anwendung allein programmiert, war mir vorher schon bekannt; eine Teamaufgabe wie diese war mir jedoch komplett neu.

Trotz dieser Umstände konnte ich mir gerade in Bezug auf die Dokumentation und den Entwurf der Software so einiges aneignen. Welche Funktionen soll die Software haben? Welche Architektur soll ihr zugrunde liegen? Wer sind die Benutzer und Akteure in der Software? Was für Rahmenbedingungen gibt es? Bei all diesen Fragen habe ich mir vorher ständig den Kopf zerbrochen und bin meist schon am Entwurf gescheitert.

Doch in Zusammenarbeit mit meinem Team konnten wir all diese Fragen aus den unterschiedlichsten Perspektiven betrachten und über verschiedene Möglichkeiten diskutieren. Diese Erkenntnis hat mich sowohl als Student als auch in meiner späteren Berufung als Softwareentwickler deutlich vorangebracht.

Neben dem Entwurf hat mich das Projekt auch sehr viel als Entwickler weitergebracht. Ich hatte schon diverse Vorkenntnisse mit Webentwicklung gerade im Bereich Frontend und konnte diese durch die von mir vorher nicht-genutzte Model-View-Controller-Architektur weiter ausbauen. Ebenfalls war es einfach super, mit Django als Backend sowie dem mitgelieferten ORM zu arbeiten. Gerade deshalb werde ich Django auch in meinen weiteren Projekten mindestens als Backend verwenden.

Auch wurden mir im Bereich Deployment einige neue Türen geöffnet. So habe ich Docker sowie docker-compose vorher nur selten genutzt; jetzt wird es aus vielerlei Gründen mein Go-To für Virtualisierung und Deployment. Die Einblicke in NixOS und die Infrastruktur des StuRa waren auch sehr interessant und ich konnte einiges für meine zukünftigen Projekte übernehmen.

Nun noch einmal zu unserem Auftraggeber: Ein weiterer Aspekt, weshalb wir nur schleppend vorankamen, war das Desinteresse unseres Auftraggebers für das Projekt. Anfangs war er hilfsbereit

und hat uns viel bei der Findung der korrekten Use-Cases sowie anderer Anforderungen geholfen. Im 2. Semester jedoch schien das Interesse nahezu gänzlich verloren. So mussten wir stetig lang auf Antworten auf unsere Fragen warten, teilweise kam gar keine Rückmeldung und Begeisterung war seitens des StuRa außerhalb der Meetings auch nicht spürbar.

Wir haben uns gefragt, woran die beiden vorigen Projekte für das ABV-Tool sowie für die Mitgliederverwaltung gescheitert sind und wir sind uns einig: der StuRa ist das Problem. Das Desinteresse sowie das wohl stetig scheiternde Deployment und die Verweigerung, die Anwendungen, in welche die Teams sehr viel Herzblut gesteckt haben, tatsächlich zu nutzen, finde ich unverschämt. Nicht nur der StuRa sondern allgemein die Auftraggeber sollten auch nur als solche agieren, wenn auch ein tatsächliches Interesse besteht. Sollten Projekte mehrfach wie beim StuRa hintereinander „scheitern“, sollte sich der Auftraggeber auch dementsprechend rechtfertigen. Unsere Übergabe steht noch an: Ich bin gespannt.

Zusammenfassend will ich noch folgendes sagen: Mir hat das Projekt trotz der teilweise schwierigen Umstände verdammt viel Spaß gemacht. Ich konnte viele neue Facetten der Softwareentwicklung aufdecken und bin auch oft an meine Grenzen gestoßen. Gerade in diesen Momenten fand ich es echt super, wie meine Teammitglieder stets versucht haben, mich in vielerlei Hinsicht zu unterstützen. Ich denke, ich habe mich sowohl fachlich als auch in Teamarbeit deutlich weiterentwickelt und kann nur sagen: Gerne wieder!

### 3.2.6. Reflexion von Simon Voges

Während der Arbeit an dem ABV-Tool Projekts in Software Engineering II erhöhte sich mein Wissen in vielen Bereichen. Das Ziel dieses Semesters war das Antrags- und Beschlussverwaltungstool des StuRas zu entwickeln. Mein Aufgabenbereich in diesem Projekt war die Teamleitung und, wegen eines unglücklich abgeschlossenen Jahresabos, das Erstellen von vielen der Diagrammen und Wireframes. Obwohl ich aufgrund einer früheren Tätigkeit bereits Erfahrung mit der Arbeit im Team habe, habe ich viel darüber gelernt, wie man effektiv im Team zusammenarbeitet und wie wichtig Kommunikation für den Erfolg eines Gruppenprojekts ist.

Ich hatte bereits einige Erfahrung mit den Programmiersprachen Python und C++, allerdings nicht im Rahmen eines großen Gruppenprojekts. Deshalb fand ich es spannend, mit anderen Studenten an einer komplexen Aufgabe zu arbeiten.

Zu Beginn des zweiten Semesters hatte unser Team einige Probleme. Es war schwer für mich, zu dem Arbeitsfluss zurückzukehren, den ich im ersten Semester aufgebaut hatte. Nach einiger Zeit konnte ich jedoch mehr Motivation sammeln, und vor allem die ersten Treffen mit dem Team brachten mich wieder in den Arbeitsfluss, den ich am Ende der SE I hatte. Im Allgemeinen würde ich die Atmosphäre im Team als sehr positiv beschreiben. Ich konnte viel von meinen Teamkollegen lernen und durch den gegenseitigen Austausch von Ideen konnten wir viele Probleme und Fragen zu Layout und Design gemeinsam lösen. Aufgrund der unterschiedlichen Kenntnisse und Fähigkeiten innerhalb eines Teams war die Arbeitsbelastung manchmal ungleich verteilt, was im Nachhinein betrachtet, anders hätte gehandhabt werden sollen.

Durch die Erstellung verschiedener Diagramme, wie z.B. des Domänenmodells, konnte ich besser verstehen, wie wichtig Architekturentscheidungen in einem Softwareprojekt sind und wie hilfreich Modelle bei der Implementierung des Projekts sein können.

Ich habe die Arbeit und die Kommunikation mit dem StuRa größtenteils als effektiv empfunden. Ich denke, wir hätten zu Beginn der SE II mehr Treffen haben können, um zu sehen, ob sich die Anforderungen geändert haben. Die Sitzungen, die wir hatten, waren jedoch sehr produktiv und es mussten nicht viele Änderungen vorgenommen werden.

Für kommende Projekte nehme ich es mir vor strukturierter zu Arbeiten und mehr auf Zeitaufwand und Prioritäten setzen zu achten.

Ich habe dank dieses Projekts eine Menge gelernt, und es war insgesamt eine sehr angenehme Erfahrung. Manchmal dachte ich, dass die geforderte Dokumentation ein wenig exzessiv war, jedoch merkte ich schnell, wie hilfreich sie für mein Verständnis war. Auch wenn wir einige Probleme hatten, bin ich dennoch sehr glücklich und stolz auf das fertige Projekt und unsere Arbeit als Team und hoffe, dass es den Anforderungen des StuRa entspricht und ihm bei der Verwaltung und Reduzierung des Papierkrams in den Sitzungen helfen wird.