

Abschlussbericht

Team: Coffee @Beans/ Nr. 4

Mitglied 1: Natalie Höpperger, 01516180

Mitglied 2: Paul Bernhard Kirchebner, 11829301

Mitglied 3: David Josef Arnold, 11719198

Mitglied 4: Christoph Wittauer, 11723544

Mitglied 5: Christoph Kugler, 01316789

Proseminargruppe: 2

Datum: 19.06.2020

1. Analyse des Projektablaufs

Prinzipiell konnten die Meilensteine eingehalten und im geplanten Zeitraum abgeschlossen werden. Allerdings lässt sich sagen, dass manche Bereiche des Projekts (vor allem in Bezug auf die Implementierung) mehr Zeit in Anspruch nahmen als andere. Deshalb lagen wir zunächst auch gut in der Zeit, hatten aber zwischendurch enormen Zeitdruck, um die vorgesehenen Meilensteine zeitgerecht abzuschließen.

Trotzdem lässt sich sagen, dass wir gegen Ende die Zeit aufholen und das Projekt dann recht stressfrei abschließen konnten. Dadurch, dass wir bereits vor dem Abnahmetest eine Testabdeckung von über 70% hatten, mussten wir auch nach Erhalt des Feedbacks keine großen Mängel mehr beheben.

Der Großteil der Zeit wurde natürlich in die Implementierung des Projekts investiert. Einen weiteren großen Teil hat das Verfassen der verschiedenen Berichte und die Teilnahme am Proseminar eingenommen. Auch in Teammeetings und Besprechungen haben wir viel Zeit investiert.

Bei der Implementierung lässt sich grob in den Entwurf beziehungsweise die vorangehende Konzeption des Systems, in die tatsächliche Implementierung, in das Bugfixing und Nachbessern sowie in Tests unterscheiden.

Die Konzeption wurde hauptsächlich zu Beginn erledigt, allerdings gab es zwischendurch auch Phasen, in denen in gemeinsamen Besprechungen bestimmte Teile des Systems und vor allem deren Umsetzung geplant wurden.

Auf solche Phasen folgten dann immer wieder Implementierungsphasen, in denen jeder für sich unabhängige Teile des Systems implementiert hat. Anschließend wurden oft direkt gemeinsam Probleme behoben und wir haben uns die implementierten Funktionen gegenseitig vorgestellt und direkt Feedback ausgetauscht, das dann wiederum bei Nachbesserungsarbeiten beachtet wurde. Zwischendurch wurde außerdem auch immer wieder Bugfixing betrieben, bevor neue Teile des Systems implementiert wurden.

Nachdem alle Funktionen eingebettet wurden, haben wir damit begonnen die dazugehörigen JUnit-Tests zu schreiben. Auch hier gingen wir wieder nach dem Prinzip vor, dass jeder für sich die von einem selbst erstellten Methoden und Klassen getestet hat.

Prinzipiell lässt sich sagen, dass die Arbeit im Team gut funktioniert hat und die Arbeitsteilung fair und gerecht war. Gerade am Anfang wäre es von Vorteil gewesen, wenn wir uns öfter getroffen hätten. Nachdem wir diesen Aspekt dann aber auch in der Gruppe besprochen haben, haben wir uns darauf geeinigt, uns mindestens einmal pro Woche zu treffen.

2. Analyse des implementierten Systems

Recht schnell haben wir uns dazu entschieden bei den Use Cases zwischen verschiedenen Arten von Anwendern zu unterscheiden. Im initialen Konzept hatten wir nämlich nur zwei Anwender, einen "User" sowie einen "Admin". Während der Implementierung wurde uns schnell klar, dass ein einziger Anwender (der "User") nicht reicht, weshalb wir die Use Cases des "Users" auf drei verschiedene Anwender, die den Rollen des Systems entsprechen, aufgeteilt haben. Dadurch erhielten wir den "Mitarbeiter", den "Teamleiter" sowie den "Abteilungsleiter". Je nach Anwenderrolle hat der User eine unterschiedliche Menüansicht sowie individualisierte Menüpunkte und kann dadurch auch unterschiedliche Funktionen des Systems nutzen. Gerade diese Unterscheidung macht die Nutzung unseres Systems intuitiv und einfach für die verschiedenen Anwender. Auch der Anwender "Admin" blieb bestehen und erhielt individualisierte Funktionen.

Auch das Klassendiagramm hat sich in dem Sinne geändert, als dass mehr Klassen benötigt als am Anfang festgelegt wurden. Zudem haben wir am Anfang vergessen, die Beziehungen zwischen den Klassen im Diagramm festzuhalten. Dies wurde von uns nachträglich noch ausgebessert.

Die SW-Architektur hat sich im Laufe des Projekts weiter verfeinert, da zusätzlich zur Bausteinsicht Darstellungen der Verteilungs- und Laufzeitsicht erstellt wurden, die so implementiert wurden und nachträglich in das Konzept eingefügt worden sind.

Es konnten alle Funktionalitäten realisiert werden außer die Speicherung eines User-Profilbilds. Dies wurde aufgrund von fehlender Zeit nicht mehr umgesetzt und daher auch aus dem Klassendiagramm entfernt.

Auch die GUI-Entwürfe konnten mehr oder weniger so umgesetzt werden. Es war sehr angenehm von Beginn an nicht nur Entwürfe für die Platzierung der einzelnen Komponenten, sondern auch für das Design an sich (Farben, Header, Menü, etc.) zu haben.

Zum Qualitätsmanagement lässt sich sagen, dass die Qualitätsattribute eines Softwareentwicklungsprozesse (Planbarkeit, Transparenz, Überprüfbarkeit, Teamfähigkeiten) alle erfüllt und eingehalten werden konnten. Die Planbarkeit würde beispielsweise durch das ausführliche Konzept garantiert, die Überprüfbarkeit durch die hohe Testabdeckung (ca. 70%). Auch die Teamfähigkeiten waren mehr als zufriedenstellend, da die Arbeit im Team wirklich gut funktioniert hat und jeder einen fairen Beitrag zum Projekt geleistet hat.

3. Ursachenanalyse

Zwei Punkte, die bei einer nächsten Projektarbeit besser gemacht werden könnten, sind einerseits die Zeiteinteilung, aber auch andererseits die Häufigkeit von Teammeetings, die zu Beginn recht selten stattgefunden haben.

Bei der Zeiteinteilung lässt sich zunächst sagen, dass prinzipiell eine Planung stattgefunden hat und dass wir uns auch daran gehalten haben. Allerdings wurde der Umfang mancher Meilensteine unterschätzt, weshalb die ersten Wochen zunächst trotz gutem Fortschritt recht stressfrei waren. In den darauffolgenden Wochen kamen allerdings umfangreichere Meilensteine auf uns zu, die viel Zeit benötigten und zu größerem Druck führten. Der unterschiedliche Umfang der einzelnen Bereiche des Systems hätte bei der Planung berücksichtigt werden sollen.

Ein weiterer Punkt ist die Anzahl der Teammeetings, die von uns durchgeführt worden sind. Auch hier hätten am Anfang häufigere Treffen stattfinden müssen. Nachdem dieser Punkt aber schon recht schnell bei einem Meeting aufgekommen ist, haben wir dann auch vermehrt Meetings (mindestens einmal pro Woche, oft häufiger) durchgeführt, in denen wir uns gegenseitig die von uns implementierten Systemkomponenten vorgestellt haben. So wusste jedes Teammitglied jederzeit darüber Bescheid, wie der aktuelle Stand ist und es konnten während des Entwicklungsprozesses aufgetretene Fragen beantwortet beziehungsweise für die Rückfrage im Proseminar gesammelt werden.

Ein weiteres, allerdings unvermeidbares Problem war die durch das Distance Learning und den Lockdown auferlegte Distanz. So konnte nur eine Person den TimeFlip-Würfel sowie den Raspberry Pi nutzen und musste diesen Bereich des Systems alleine implementieren. Da es sich dabei um einen großen Bereich des Projekts handelte, wäre es sicher besser gewesen, wenn zwei Personen an diesem Teil arbeiten hätten können.

4. Erfahrungen mit den eingesetzten Werkzeugen

Im Grunde war uns das Skeleton-Gerüste eine große Hilfe. Es bot alles, das man benötigt, um ein individuelles Projekt mit guter Grundstruktur zu erschaffen. Zudem diente es teilweise auch als Vorlage, um beispielsweise eigene Models, Repositories, Services und Controller zu implementieren.

Größere Probleme hatten wir wiederum mit PrimeFaces. Viele der angegebenen PrimeFace-Elemente funktionierten zum Teil nur unter bestimmten Voraussetzungen oder verhielten sich komplett irrational. Jedoch war PrimeFaces trotz häufiger Probleme sehr hilfreich und erleichterte die Implementierung der Weboberfläche enorm. Durch die Anpassung der bereits existierenden Elemente an unsere Entwürfe der GUI konnten wir zudem trotz der Verwendung vorgefertigter Elemente eine individuelle und einzigartige Oberfläche erzielen.

Besonders Sonar Qube gab uns einen Einblick in bestimmte Probleme unseres Projektes. Beispielsweise konnten wir durch die Verwendung dieser Plattform überprüfen, ob wir die Testabdeckung bereits erreicht hatten oder auch "Code Smells" identifizieren und anschließend beheben.

Für die Datenbank verwendeten wir die HeidiSQL-Workbench. Diese bietet eine Vielzahl von Features, wie zum Beispiel die einfache Verwendung von SQL-Abfragen, was den SQL-Alltag vereinfacht hat. Die Umstellung von der H2-Datenbank auf eine MySQL-Datenbank stellte sich jedoch am Anfang als Herausforderung dar.

Wir einigten uns am Anfang des Projekts auf eine gemeinsame IDE (IntelliJ). Dadurch konnten wir Probleme, die bei Einzelnen auftraten, schnell und effizient lösen.

Für unsere Team-Treffen verwendeten wir Discord, dass uns in Zeiten der Fernlehre eine große Hilfe war. Mit Hilfe der Bildschirmübertragung konnten wir anderen Teammitgliedern über die Schultern schauen und bei Problemen helfen. Damit konnte auch der wöchentliche Fortschritt präsentiert werden.

Der verwendete Raspberry funktionierte einwandfrei. Meistens wurde direkt auf in gearbeitet und es wurde nur selten mittels ssh darauf zugegriffen. Deshalb wurde auch IntelliJ und andere nötige Software darauf installiert.

Der TimeFlip Würfel führte immer wieder zu Problemen. Die bezüglichen Workshops waren gerade am Anfang eine große Hilfe bezüglich des Würfels. Die Verbindung via Bluetooth funktionierte zwar immer, allerdings stellte das eher schlecht dokumentierte Bluetooth Protokoll und der Würfel an sich zeitweise ein Problem dar, wie zum Beispiel, dass das Ausschalten des Würfels zum Namensverlust oder auch das Neustarten des Würfels in manchen Fällen zu einer neuen Einteilung aller Würfelseiten führt. Problematisch war zudem, dass die Würfelseiten nicht einstellbar sind oder mit Zahlen größer 12 nummeriert sein können. Dies führt zu einigen Problemen, vor allem wenn man an ein realistisches Szenario denkt, wie beispielsweise der Einsatz in einem Unternehmen.

5. Feedback zur Proseminar-Organisation

Das Projekt an sich war eine tolle Idee. Eine persönliche und individuelle Zeiterfassung mit der Hilfe des TimeFlip-Würfels finden wir sehr praktisch und realitätsnah. Vor allem auch die Einbindung der Hardware (TimeFlip-Würfel sowie Raspberry Pi) machte das Projekt spannend, aber auch herausfordernd.

Durch die Situation um COVID-19 mussten wir alle Team-Meetings online durchführen und konnten nicht gemeinsam an der Implementierung der Software für TimeFlip und Raspberry Pi arbeiten.

Somit blieb dieser – recht umfangreiche – Teil des Projekts leider nur einer Person über und konnte nicht wie anfangs geplant von zwei Personen durchgeführt werden.

Hilfreich waren auch die zweiwöchigen Status-Quo Besprechungen, um Probleme und offene Fragen zu klären, Feedback zu erhalten und auch um zu sehen, wie andere Teams die Aufgabenstellung umgesetzt haben und auf welchem Stand sie sich befinden. Auch die Vorlagen und Beispieldokumente und generell ein Großteil der Materialien, die zur Verfügung gestellt wurden, haben den Planungs- sowie Dokumentationsprozess und auch den Testprozess (z.B.: Abnahmetest, Testprotokoll) sehr erleichtert.

Auch die Workshops an sich waren eine gute Idee, um einen zusätzlichen Einblick in spezielle Themen zu erhalten und sich in diesem Gebiet zu vertiefen. Allerdings war die Umsetzung oft nicht ganz ausgereift und auch durch die Situation um COVID-19 konnten alle Workshops nur online durchgeführt werden, was das Mitkommen bei den praktischen Übungen in den Workshops durch die fehlende oder versetzte Kommunikation (beispielsweise Nachrichten im Chat, die erst nach einigen Minuten gelesen wurden) erschwerte. Aber auch ohne Kommunikationsprobleme war es schwierig, dem Workshopleiter zu folgen.

Sehr schade fanden wir, dass die “FUNshops” mit Simon Haller nicht stattfinden konnten.