

Projektbericht

Experimenteverwaltung - Erweiterung

Buchungssystem (I2)

Alena Bischoff, Gregor Müller, Julius Schmidt, Richard Böhme, Taejun Kim, Tim
Würzburg, Tobias Karsch, William Elsner, Piet Gutsche

2021-07-01 11:31:08 +0200

Inhaltsverzeichnis

1. Experimenteverwaltung - Erweiterung Buchungssystem (I2)	1
2. Planung	2
2.1. Ausgangssituation in SE II	2
2.1.1. Teamaufstellung	2
2.1.2. Aufgabenstellung und Auftraggeber	2
2.2. Projektorganisation	2
2.2.1. Team	2
2.2.2. Rollenverteilung	4
2.3. Ablauf der Kommunikation unter Verwendung von Tools	4
2.4. Way of Working	5
3. Durchführung des Projekts	6
3.1. SE I	6
3.2. SE II	6
3.2.1. Start in das Neue Semester (22.03.2021-21.04.2021)	6
3.2.2. Iteration 05 (21.04.2021-05.05.2021)	9
3.2.3. Iteration 06 (05.05.2021-19.05.2021)	12
3.2.4. Iteration 07 (19.05.2021-02.06.2021)	14
3.2.5. Iteration 08 (02.06.2021-16.06.2021)	16
3.2.6. Iteration 09 (16.06.2021-30.06.2021)	18
4. Ergebnisse	21
4.1. Ziel: Einheitliches Buchungssystem für EMS mit Buchungsverwaltungsfunktionalitäten	21
4.1.1. Anlegen und Speichern einer Physikvorlesung	21
4.1.2. Bearbeitung von Vorlesungen	21
4.1.3. Darstellung der gebuchten Vorlesungen	21
4.1.4. Abgrenzung der Nutzer	21
4.2. Einfachere Auswahl eines Geräts bei der Erstellung eines Experiments	22
4.3. Buchungshäufigkeit auswerten	22
4.4. Fehlerbehebung	22
4.5. Begründung für das Nichterfüllen von Zielen	22
4.6. Reflexionen der Teammitglieder	22
4.6.1. Gregor Müller	22
4.6.2. William Elsner	23
4.6.3. Richard Böhme	24

1. Experimenteverwaltung - Erweiterung Buchungssystem (I2)

Dies ist der Projekt Bericht des Themas I2: Erweiterung des Systems zur Verwaltung von Physikexperimenten. Das Projekt wurde im Sommersemester 2021 von Richard Böhme (a.I.), William Elsner (W.I.), Piet Gutsche (a.I.), Gregor Müller (V.I.) betreut.

2. Planung

2.1. Ausgangssituation in SE II

2.1.1. Teamaufstellung

Am Anfang des Projekts bestand das Team aus acht Mitgliedern. Diese waren eine hochschulexterne Fachinformatikerin (alena.bischoff@htw-dresden.de), ein Informatikstudent (richard.boehme@htw-dresden.de), zwei Wirtschaftsinformatikstudenten (william.elsner@htw-dresden.de, taejun.kim@htw-dresden.de), ein Verwaltungsinformatikstudent (gregor.mueller@htw-dresden.de) und drei Wirtschaftsingenieurstudenten (tobias.karsch@htw-dresden.de, tim.wuerzburg@htw-dresden.de, juliusheiko.schmidt@htw-dresden.de). Geplant war das Ausscheiden der Wirtschaftsingenieure zu SE II, jedoch wurde das Team leider zusätzlich von alena.bischoff@htw-dresden.de und taejun.kim@htw-dresden.de verlassen. Deshalb wurde uns der Informatikstudent piet.gutsche@htw-dresden.de als Verstärkung zugewiesen. Dadurch haben wir zu Beginn von SE II zuerst eine neue Einteilung der Rollen des Teams durchgeführt. Mehr dazu findet man in der Rollenverteilung.

2.1.2. Aufgabenstellung und Auftraggeber

Das Hauptziel unseres Projekts war es ein Buchungssystem zu entwickeln um den Workflow zwischen den Dozenten und dem Admin zu erleichtern. Dazu mussten wir erst die Arbeitsweise der beteiligten Personen verstehen und konnten so die zu speichernden Daten herausarbeiten. Wir mussten eine geeignete Darstellungsform und eine einfach zu verstehende Umgangsweise mit dem Buchungssystem finden. Auch sollte es möglich sein Vorlesungen nach dem Anlegen noch zu bearbeiten, aber nicht nachdem die Vorlesung gehalten wurde. Zusätzlich mussten wir eine Möglichkeit finden um die einzelnen Anwender voneinander abgrenzen zu können. Hinzu kam noch der Wunsch nach einer Auslesemöglichkeit der Häufigkeit von verwendeten Physik-Experimenten und Bugfixing (Für eine genaue Auflistung, siehe Vision). Unser Auftraggeber, Herr Matthias Heisig, blieb der selbe aus SE I. Allerdings änderte er gleich beim ersten Themensteller-Meeting die Aufgabenpriorisierung. Es war von unserer Seite eigentlich geplant, dass erst die Use Cases nacheinander implementiert werden und anschließend ein Zeitfenster für Bugfixing und Änderungen des aktuellen Systems eingerichtet wird. Dies wurde verworfen. Durch den Kunden war nun gewünscht, dass nach der Implementierung des Buchungssystems direkt ein besserer Weg gefunden wird, die Geräte im vorhandenen System beim Anlegen eines Experiments auszuwählen. Die anderen Arbeiten sollten dann nach gewohnter Priorisierung aus SE I erfolgen. Eine genaue Auflistung der Aufgaben ist in der Vision zu finden.

2.2. Projektorganisation

2.2.1. Team

Da wir schon früh in SE I die Erfahrung gemacht haben, dass ein gutes Team der Schlüssel zu einem guten Projektabschluss ist, haben wir auch in SE II daran festgehalten. Wir meinen mit "gutem Team" nicht, dass die einzelnen Teammitglieder über große Expertisen im Bereich Software Engineering oder andere IT bezogene Qualifikationen verfügen müssen, sondern, dass eine gute

Gruppendynamik und ein starker Zusammenhalt innerhalb der Gruppe vorhanden ist.

Wir haben versucht, die Planung der einzelnen Teilaspekte so zu gestalten, dass die eben genannten Eigenschaften stets in hohem Ausmaß vorhanden sind.

Im Folgenden sind eingesetzte Praktiken aufgelistet, die dazu beigetragen haben die Zusammenarbeit in unserem Team zu verbessern.

Eingesetzte Praktik	Gewünschter Erfolg
Risiko des Scheiterns miteinbeziehen	Die Priorisierung der Reihenfolge der Arbeitsschritte wurde stets so gewählt, dass ein auftretendes Problem nie die Arbeit der anderen Gruppenmitglieder behinderte. So sollte der Erfolgsdruck gemindert werden.
Aufgabengebiete weit streuen	Es wurde versucht, die einzelnen Gruppenmitglieder Rollenunabhängig, in allen Teilgebieten mit kleinen Aufgaben einzusetzen. Hintergrund dieser Strategie war, dass jedes Gruppenmitglied auch die Aufgaben der anderen Rollen kennenlernen kann und so auch außerhalb der eigenen Rolle Erfahrungen sammeln konnte.
Zeit für Austausch einplanen	Durch geregelte Gruppendiskussionen kann jedes Mitglied seine Meinung oder Erfahrung bei Entscheidung einbringen und sollte dadurch auch mehr am Projekt beteiligt werden
Demokratische Entscheidungsfindung	Jede Entscheidung, die von uns getroffen werden konnte, wurde demokratisch und nicht autoritär entschieden. Dies sollte gewährleisten, dass alle Gruppenmitglieder mit der Entscheidung einverstanden sind und auch keine schlechten Entscheidungen, wie z.B. durch mangelhafte Expertise getroffen werden.
Einsetzen eines QMs und Puls-Checks	Durch Verwendung eines Qualitätsmanagements und Puls-Check-Umfragen konnte in der Planung das allgemeine Gruppenempfinden einbezogen werden. Hier sollten mögliche Probleme durch gezielte Fragen oder Austausch gelöst werden. (Hinweis: Es wurden alle Probleme stets anonymisiert gesammelt und die Diskussionsgrundlage allgemein formuliert)
Stundennachweis	Die Gruppenmitglieder wurden dazu angehalten, ihre Arbeitszeiten in einer Stundenliste zu protokollieren. So sollte in der Planung sichergestellt werden, dass eine gleichmäßige Verteilung der Arbeitszeit über alle Gruppenmitglieder stattfinden kann.
Wiederkehrende Aufgaben standardisieren	Immer wiederkehrende kleine Aufgaben wie z.B. das Erstellen einer Agenda für das Weekly wurden direkt ohne Dokumentation in der Work-Item-List ausgeführt oder wurden als konstant bestehender Issue aufgeführt. Dies sollte ermöglichen, dass die Work-Item-List übersichtlicher angezeigt werden konnte, trotz partiell sehr granularen Aufgliederungen von großen Aufgaben.

Eingesetzte Praktik	Gewünschter Erfolg
Pair Programming	Die Implementierung des Systems wurde vorrangig in einer Zweiergruppe angegangen. Diese bestand aus dem Hauptentwickler mit Ruby-Expertise und dem Architekten. Ihm wurde es möglich, dass erste Programmiererfahrungen in Ruby gesammelt werden konnten, ohne, dass der Zeitaufwand zu groß wurde. Außerdem konnten spontane Probleme so durchaus schneller gelöst werden, was auch den Developer entlastete.

2.2.2. Rollenverteilung

Die Aufgaben wurden hauptsächlich nach Interessen- und Expertisengebieten verteilt. Das Team einigte sich darauf, dass der Rollenverantwortliche sich hauptsächlich um die Dokumentationspflichten und Arbeitspriorisierung kümmerte, er aber immer von anderen Teammitgliedern bei der eigentlichen Aufgabe unterstützt wird. So konnten wir eine perfekte Mischung aus Rollenspezialisierung und Informationsaustausch erreichen. Durch die starke Dezimierung des Teams in SE II waren wir gezwungen, dass einige Gruppenmitglieder mehrerer Rollen ausüben mussten. Somit wandelte sich die Teamarbeit in den Bereichen Dokumentation und Organisation zu spezialisierter Einzelarbeit. Der Informationsaustausch wurde durch präzise Reports in den Weeklys aufrecht erhalten. Eine genaue Übersicht zur Verteilung der einzelnen Gruppenmitglieder auf die einzelnen Rollen findet man in der Rollenübersicht.

2.3. Ablauf der Kommunikation unter Verwendung von Tools

Da während des Projekts die Corona Pandemie Deutschland beherrschte, wurde versucht, sämtliche Kommunikation online durchzuführen. Um dies umzusetzen, wurde der Messenger-Dienst WhatsApp für die direkte Kommunikation zwischen einzelnen Gruppenmitgliedern und zur schnellen Informationsübermittlung an alle Teammitglieder genutzt. Um Meetings zu realisieren, wurde zu erst ein Discord Server genutzt, im Laufe des Projekts dann aber auf Microsoft Teams gewechselt. Hintergrund hier war, dass Teams einige Zusatzfunktionen hat, wie z.B. "Polly", die für Umfragen genutzt wurden, die Discord leider nicht unterstützte. Des Weiteren konnte in Teams auch ein cloudbasiertes Datei-Management-Tool leicht verwendet werden, so mussten wir nicht mehr OneDrive nutzen, das privat von einem Gruppenmitglied gestellt wurde. Auch konnte in Teams eine bessere Rollenhierarchie genutzt werden, um Rollenspezifische Feed Anfragen zu nutzen, damit ein Brainstormeffekt entsteht. Die Dokumentation erfolgte geschlossen in ADOC Dateien, die über GitHub verwaltet wurden sind. Auch wurde über GitHub unsere Work-Item-List mit Hilfe der Issues verwirklicht. GitHub war der Schlüssel um Gruppen- und Parallelarbeiten zu organisieren und umzusetzen. Zur Überprüfung und zur Dokumentation unseres Projektfortschritts benutzten wir regelmäßig den Essence Navigator. Um mit dem Kunden in Kontakt treten zu können, wurden ausschließlich E-Mails, Microsoft Teams und Präsenz-Treffen genutzt. Die Präsenztreffen fanden erst in SE II statt, mit kompletter Einhaltung der Corona-Hygiene-Maßnahmen. Diese Änderung wurde von unserem Kunden sehr begrüßt, da er leider durch die eben genannten Techniken in seinem Workflow eher behindert als unterstützt wurde.

Die Online-Situation erleichterte die Planung von Meetings und Gruppenarbeiten erheblich, da nun

Fragen wie bspw. an welchem Ort das Treffen stattfinden könnte, ob dieser Ort auch zur gegebenen Zeit verfügbar ist oder wie lange die Anreise der einzelnen Mitglieder wohl wäre, nun alle obsolet waren. Es musste nun nur noch eine Uhrzeit gefunden werden, zu der alle verfügbar waren.

2.4. Way of Working

Unsere Arbeitsweise war sehr stark vom Open Unified Process geprägt. Wir haben stets versucht, alle vier Kernziele zu durchlaufen. Alternativ dazu haben wir auch versucht andere Praktiken aus anderen Bereichen zu verwenden, die wir aus eigener Erfahrung oder aus der Theorie kannten. Diese waren das QM oder die Puls-Checks. Wir haben mit Iterationen gearbeitet, die immer zwei Wochen andauerten.

Grundsätzlich war aber folgender Ablauf regelmäßig zu erkennen:

1. Planung der Iteration

- Zielsetzung anhand der Aufgabenpriorisierung des Kunden
- Die Ziele in Aufgaben unterteilen
- Bewertung der Aufgaben
 - Bewertung des Aufwands
 - Bewertung der Priorität
- Aufgabenverteilung an die Gruppenmitglieder

2. Durchführung der Iteration

- Die Aufgaben werden nach der Priorisierung unter Beachtung des Aufwands abgearbeitet
- Ständiger Austausch unter den Gruppenmitgliedern über den Status der Aufgabe

3. Ende der Iteration

- Vorstellung der erledigten Aufgaben
- Bewertung der Ergebnisse durch die Gruppe
- Puls-Check
- Iterationsnachbesprechung
- Lesson-Learned (wenn erforderlich)
- Dokumentation der Iteration

Wenn geplante Ziele oder Aufgaben nicht in der Iteration umgesetzt werden konnten, wurden diese in die nächste Iteration übernommen und versucht, die Aufwandsplanung anhand der neuen Erkenntnisse besser umzusetzen.

3. Durchführung des Projekts

3.1. SE I

Das Hauptaugenmerk in SE I unseres Projektes war, eine gute und solide Grundlage für die Constructions Phase in SE II zu schaffen. Deshalb bestand die erste Iteration hauptsächlich aus Teamorganisations Tätigkeiten und der Einarbeitung in das Projekt und die Aufgabenstellung. Hier haben wir auch zum ersten Mal Kontakt zum Kunden hergestellt.

Die Ziele der zweiten Iteration waren es, Einigkeit mit dem Kunden über die Vision herzustellen. Hierzu benutzten wir erstmals Wireframes, damit unserer Kunde sich unsere Ideen besser vorstellen konnte. Auch wurden hier die systemweiten Anforderungen aufgestellt und der GitHub Workshop abgehalten.

In Iteration 03 versuchten wir die Anforderungen genau zu spezifizieren, also die erstellten Wireframes genau mit dem Kunden durchzusprechen, um zu verifizieren ob alle Forderungen richtig verstanden wurden. Hier wurden auch die Test Cases zu den Use Cases erstellt. Der **LCO-Meilenstein** und der **LCA-Meilenstein** konnten hier abgeschlossen werden.

In der letzten Iteration von SE I fiel uns auf, dass wir mit den Dozenten, die einen großen Teil der Anwenderbasis darstellen, leider nie Kontakt aufgenommen haben. Das war zum einem unserer fehlenden Bewusstsein und zum anderen dem Wunsch des Kunden, Aufwand für die Dozenten zu vermeiden, zuzuschreiben. Um diesen Fehler zu beheben, versuchten wir die fehlenden Anforderungen mit der Unterstützung von Frau Prof. Krawietz zu ermitteln. Bei diesem Unterfangen bemerkten wir jedoch rechtzeitig, dass wir es zeitlich nicht mehr schaffen würden alle anwenderspezifischen Sichten richtig zu analysieren. Deswegen verschoben wir dieses Ziel in SE II. Auch wurden in dieser Iteration nochmals die Use Cases und die Testcases überarbeitet und das Abgabedokument angefertigt.

3.2. SE II

3.2.1. Start in das Neue Semester (22.03.2021-21.04.2021)

Der Start in das neue Semester verlief leider nicht optimal, da uns zwei Teammitglieder zusätzlich zu den Wirtschaftsingenieuren verließen. Somit hatten wir direkt am Anfang des Semesters (23.03.2021) das erste und einzige Kriesen-Meeting. Hier versuchten richard.boehme@htw-dresden.de, william.elsner@htw-dresden.de und gregor.mueller@htw-dresden.de die Ausgangslage zu sondieren und abschätzen zu können ob es im Worst-Case auch nur zu dritt möglich wäre, das Projekt erfolgreich betreuen zu können. Sie kamen zu dem Entschluss, dass die Expertise im Bereich Ruby on Rails von richard.boehme@htw-dresden.de ihnen einen immensen Zeitvorsprung ermöglichte und es somit möglich war, die fehlenden Teammitglieder ausgleichen zu können. Als dann dem Team auch noch ein neues Mitglied (pier.gutsche@htw-dresden.de) zugewiesen wurde, welches auch noch über Erfahrungen mit automatisierten Softwaretests verfügte, konnte das Team mit neuer Zuversicht die nächsten Schritte planen (30.03.2021):

- Anwendersicht auf das System validieren
- Ruby on Rails Workshop durchführen, um mit dem Implementieren starten zu können

- Altbelegkorrektur durchführen
- Neues Teammitglied einarbeiten
- Programmierkonventionen erarbeiten
- Login-Problematik für das Buchungssystem lösen
- Ruby on Rails Update des vorhandenen Systems
- Kundenmeeting

Parallel dazu wurde auch die Rollenverteilung überarbeitet. Hier mussten nun aufgrund von fehlenden Mitgliedern einige Rollen doppelt besetzt werden. Wir versuchten hier so logisch wie möglich vorzugehen, heißt Mitglieder, die eher technische Rollen besetzten, besetzen auch wieder eher Technische Rollen und Mitglieder, die eher organisatorische Rollen innehatten, werden weitere organisatorische Aufgaben dazubekommen. Eine genaue Übersicht zu den Rollen befindet sich in der Rollenübersicht.

Anwendersicht auf das System validieren

Um die Anwendersicht zu validieren, wurde durch den Projektmanager der Kontakt mit Frau Prof. Krawietz gesucht. Dieser ließ sich schnell herstellen, da die Anfrage zu Ihrer Unterstützung schon im vergangenen Semester geklärt werden konnte. Frau Prof. Krawietz wurde eine ausführliche PDF mit allen Wireframes zur Verfügung gestellt, in der einzelne Abläufe textlich erklärt wurden. Bei Unklarheiten stand ihr der Projektmanager auf dem Wege des Emailverkehrs zur Seite. So wurde eine Liste mit Änderungsvorschlägen gesammelt, die den Workflow aus Anwendersicht verbessern würden.

Ruby on Rails Workshop

Der Ruby on Rails Workshop wurde von richard.boehme@htw-dresden.de vorbereitet und durchgeführt. Dazu fertigte er zwei von einander getrennte Dokumentationen an. Die erste behandelte Ruby im Allgemeinen, die zweite hatte die Schwerpunkt bei Ruby on Rails direkt. Bei den Dokumentationen wurde großer Wert auf Übersichtlichkeit und Wiederverwendbarkeit gelegt. Deshalb wurden nur kurze Einleitungen verwendet und einfach zu verstehende Code-Beispiele. Der Workshop an sich wurde auf zwei Teile aufgeteilt und war für alle Gruppenmitglieder verpflichtend. Der erste Teil wurde am 12.04.2021 durchgeführt und der zweite Teil am 19.04.2021. Beide Teile wurden aufgezeichnet und die Dokumente allen Gruppenmitgliedern zugänglich gemacht. Dazu werden beide Dokumente in der Betriebsdokumentation beigelegt um zukünftige Entwickler zu unterstützen.

Altbelegkorrektur und neues Mitglied einarbeiten

Nach der Konsultation (02.04.2021) mit Professor Dr. Anke, bezüglich des Belges von SE I, wussten wir, was im Abgabedokument aus SE I korrigiert werden mussten. Diese Aufgabe wurde unmittelbar angegangen, da wir zu diesem Zeitpunkt noch nicht direkt mit dem Entwickeln anfangen konnten, da wir noch warten wollten bis die Anwenderwünsche mit dem Kunden besprochen wurden. Hintergrund war, dass wir erst die Elaboration Phase abschließen wollten, bevor wir mit der Construction Phase anfangen. Um die Zeit sinnvoll zu nutzen, wurde mit der Korrektur des Belegs begonnen. Die orthografische Korrektur der überarbeiteten Dokumente überließen wir unserem neuen Teammitglied, um ihn so direkt sinnvoll im Projekt einsetzen zu können. Außerdem wurde er so mit unseren Ergebnissen vertraut. Mit dieser Arbeitsweise schafften wir es, die Einarbeitungs-

zeit für unser Projekt perfekt zu nutzen.

Programmierkonventionen

Um unserem QM gerecht zu werden, entschieden wir uns dazu Programmierkonventionen zu erstellen, die für alle Developer verpflichtend war. Den Vorteil hier sahen wir in der Übersichtlichkeit des Codes. Wir befürchteten, dass das Arbeiten durch mehrere Individuen mit unterschiedlichen Code-Schreibstilen zu Chaos und Uneinheitlichkeit führen könnte. Deshalb beauftragten wir richard.boehme@htw-dresden.de mit der Aufgabe, geeignete Konventionen zu erstellen. Um den Kosten/Nutzen Faktor nicht zu sehr zu strapazieren, entschied er sich dazu, sich bei der Ausarbeitung auf die geltenden Konventionen der Ruby-Community zu beziehen. Eine genauer Übersicht dazu findet man im QM.

Login-Problematik für das Buchungssystem lösen

Da unser Projekt in jedem Fall eine Multi-User-Anwendung sein musste, mussten wir auch eine geeignete Lösung für die Rechteverwaltung der unterschiedlichen Rollen finden. Da das System unserer Vorgänger schon über einen Login verfügte, kamen wir zu dem Entschluss auf eben diesem aufzubauen. Wir planten hier grob, dass beim Anlegen eines neuen Nutzers das System ein Passwort vergeben sollte und bei der ersten Anmeldung des Nutzers, er dies in sein persönliches ändern sollte. Diesen Vorschlag wollten wir dem Kunden bei dem Kundenmeeting unterbreiten.

Ruby on Rails Update

In der Vorbereitung auf die Construction Phase bemerkten wir, dass die Ruby on Rails Version des Systems nicht mehr die aktuellste war. Dies wollten wir durch ein Update der Version ändern. Diese Überlegung möchten wir mit möglichen Sicherheitslücken und Performanceneinbußen begründen. Da wir zu so einem Schritt die Einwilligung des Kunden benötigen, wollten wir ihm dies im kommenden Meeting vorschlagen.

Kundenmeeting

Da wir in unserer Vorarbeit nun einige Punkte (siehe vorgegangene Kapitel) ausgearbeitet hatten, die wir dem Kunden vorstellen wollten, nahmen wir am 13.04.2021 Kontakt zum Kunden auf und baten um ein Meeting. Dies wurde uns am 27.04.2021 gewährt. Die großen Abstände zwischen Erstkontakt und Meeting können durch einen krankheitsbedingten Ausfall des Kunden erklärt werden.

Retrospektive

Die ersten vier Wochen des Semesters haben wir ohne Iterationen gearbeitet. Wir hatten unser Mindset noch nicht wirklich auf Software Engineering umgestellt und versuchten immer noch ein Schulprojekt zu betreuen. Deshalb waren wir uns unsicher ob unsere Korrektur- und Vorbereitungsarbeiten überhaupt schon als SE II Arbeiten galten. Aus heutiger Sicht (am Ende des Semesters) war dies eine falsche Entscheidung. Wir haben zwar unterbewusst einige Iterationsabläufe bei unserer Arbeitsweise beibehalten, wie z.B. Protokolle anfertigen oder Ziele formulieren, andere jedoch wie z.B. Work-Item-List anfertigen oder Puls-Checks durchführen wurden vergessen. Dies betrachten wir als Fehler, welchen wir uns eingestehen müssen.

Die Einarbeitung des neuen Teammitglieds verlief unglaublich einfach und intuitiv und wir alles sind sehr froh uns gefunden zu haben.

Die deutliche Mehrfachbelegung der Rollen führte zu einer gewissen Abgrenzung der Teammitglieder, da nun sehr Fachspezifisch gearbeitet wurde und der Gruppenaustausch sich fast ausschließlich auf die Weeklys begrenzte.

3.2.2. Iteration 05 (21.04.2021-05.05.2021)

Iterationsziele:

- Login-Lösung weiterverfolgen
- Validierung der Benutzerwünsche mit dem Kunden
- Besuch des Ruby on Rails Workshops durch alle Gruppenmitglieder
- Überarbeitung der UCs, TCs, Wireframes, Vision und des Glossars
- Installation der Entwicklungsumgebung durch alle Developer
- Evaluation der alten TCs, Entscheidung fällen über das zu verwendende Testframework
- C4 Modell um UC01 erweitern
- Durchführung von Ruby on Rails Update

Login-Lösung weiterverfolgen

Es wurde die Möglichkeit vorgeschlagen, den Login mit dem HTW Login zu verknüpfen. Dazu konsultierten wir Herrn Schubert, um uns bezüglich des Aufwands und der Abstimmung mit dem Rechenzentrum einen Überblick zu verschaffen. Herr Schubert riet uns von dieser Lösung ab, weil es seiner Meinung nach möglich wäre durch unser System die Passwörter abgreifen zu können. Dies war nicht als Vorwurf gemeint, sondern nur als Hinweis. Des Weiteren sah er ein Problem in der Authentifizierung, da wir hier administrative Informationen des Nutzerverwaltungs-Server des Rechenzentrums benötigen würden und diese sehr wahrscheinlich nicht bekommen werden. Er empfahl uns den Login mit Hilfe einer Datenbank umzusetzen, da es hier auch einfacher wäre, den Mechanismus zu ändern, wenn es erforderlich sei.

Dank dieser Informationen kamen wir zu dem Entschluss, vom HTW Login abzusehen, wollten den Kunden aber dennoch über diesen Vorschlag informieren.

Validierung mit dem Kunden, TSM4 (27.04.2021)

Hintergrund des Treffens mit dem Kunden war die Abklärung der Kundenwünsche, die Absprache des Logins und der Vorschlag des Updates. Die Vorschläge der Kundenwünsche wurden dem Kunden unterbreitet und gemeinsam mit ihm überlegt, welche Änderungen sinnvoll sind und welche nicht. Daraus ergaben sich folgende Beschlüsse:

Nr.	Anwenderwunsch	Kundenentscheidung
AW01	Es sollte möglich sein, die Studiengänge, welche bei der Voreinstellung auswählbar sind, durch den Admin bearbeiten zu können	Bewilligt

Nr.	Anwenderwunsch	Kundenentscheidung
AW02	Es sollte möglich sein, auch alternative Vorlesungen wie z.B. die "Kinderuni", die keinem Studiengang zuordbar sind, mit dem System abzudecken.	Bewilligt
AW03	Es sollte möglich sein, nach jedem Experiment einen Kommentar zu hinterlegen.	Verweigert
AW04	Es sollte möglich sein, auch Experimente, die nicht in der DB vorhanden sind im Wochenplan abzubilden.	Bewilligt
AW05	Es sollte möglich sein das Journal mit Hilfe eines Akkordeon-Menüs übersichtlicher zu gestalten.	Bewilligt

Die Idee der Loginvergabe wurde vom Kunden nicht positiv aufgenommen, weder der HTW Login, noch das zu verändernde Passwort. Er will Benutzername und Passwort der Nutzer selbst vergeben und den die Details den Nutzern dann per Email zukommen lassen. Der Projektmanager, der das Meeting führt wies mehrmals darauf hin, dass dies datenschutztechnisch wahrscheinlich nicht der beste Weg ist. Die Meinung des Kunden ändert sich daraufhin aber nicht.

Der Vorschlag des Updates von Ruby on Rails wurde von dem Kunden sehr positiv aufgenommen und wir erhielten die Erlaubnis, dies auf der VM durchzuführen. Falls alles funktionieren sollte, sollten wir dies dann auch auf dem Hauptsystem umsetzen und in der Betriebsdokumentation beschreiben, wie auch in Zukunft ein solches Update umzusetzen sei.

Der Kunde wünschte auch eine Neupriorisierung der Aufgaben, die Implementiert werden sollten. Eine genaue Übersicht zur Priorisierung ist in der Vision zu finden. Auch sollte die Ansicht der Statistik angepasst werden, da das vorgeschlagene Design von SE I dem Kunden nun nicht mehr entsprach. Das selbe galt für die Dozentenwoche, welche ebenfalls vom Design nochmals angepasst werden sollte.

Abschluss des Ruby on Rails Workshops

Der letzte verpflichtende Termin des Workshops war noch außerhalb der I05, die Nachbesprechung erfolgte allerdings in ihr. Das Feedback war einstimmig sehr positiv und richard.boehme@htw-dresden.de wurde von allen Teammitgliedern mit Dankbarkeit und Lob überschüttet.

Überarbeitung der UCs, TCs, Wireframes, Vision und des Glossars

Das TSM4 ergab, dass einige Dokumente im Beleg von angepasst werden mussten. Der UC01 musste so verändert werden, dass er nun auch AW02 und AW04 abdeckte. Im UC02 musste AW05 mit aufgenommen werden.

Die TCs mussten dementsprechend auch abgeändert werden, um die neu hinzugekommen Funktionen auch testen zu können. Bei dem Wireframe WF_1 wurde der optionale Titel hinzugefügt (AW02), bei WF_2 wurde ein Button hinzugefügt, um ein benutzerdefiniertes Experiment außerhalb der Datenbank zur Buchung hinzuzufügen(AW04) und bei WF_8 wurde zum Titel das Semester hinzugefügt. Es wurden die Experimentenummern hervorgehoben. Es wurde der Abstand zwischen den Experimenten verkleinert. Es wurde Abstand zum Abheften erzeugt (Änderungen des

Kunden zur Wochenplanansicht).

Alle Änderungen der Wireframes wurden vom Kunden akzeptiert, außer WF_8. Die Kundenforderung war, dass wir das Design von Frau Prof. Krawietz übernehmen sollten, was wir auch taten. Es wurde davon abgesehen, nochmals einen Wireframe hierfür anzufertigen.

Die Neupriorisierung wurde in der Vision dokumentiert.

Das Glossar wurde um einige Wörter erweitert.

Installation der Entwicklungsumgebung aller Developer

Um die Vorbereitungen für die Construction Phase abzuschließen, setzen wir uns das Ziel, dass alle Developer bereit waren mit Ruby on Rails zu arbeiten. Die Kriterien waren hier, über eine Linux-System und mindestens über ein Textverarbeitungsprogramm zu verfügen, um Ruby on Rails schreiben zu können.

Evaluation der alten TCs, Entscheidung fällen über das zu verwendende Testframework

Da wir nun langsam in die Construction Phase kommen würden, mussten wir uns nun auch für ein Testframework entscheiden. Der Entwickler hatte bisher nur Erfahrung mit dem Testframework Minitest und der Tester hatte bisher mit pythonbasierten Frameworks (ähnlich zu Minitest) gearbeitet. Allerdings gab es eine beträchtliche Menge Legacy Code für automatisierte Tests bzw. Regressionstests, die das Framework Rspec benutzten. Somit war es eine Abwägung der Zeitaufwände, zwischen dem Einarbeiten in ein neues Framework und dem kompletten überarbeiten der alten Tests. Nach längerer Diskussion entschieden sich Tester und Entwickler dann Rspec beizubehalten und als Testframework zu verwenden. Das hatte zur Folge, dass der Entwickler ebenfalls Einarbeitungszeit für Rspec benötigte, da die Unittests von ihm geschrieben werden. Allerdings wurde so die Gesamtarbeitszeit für das Team geringer gehalten, da die alten TCs übernommen werden konnten. Eine anfängliche Evaluation der TCs der Vorgänger ergab, dass wir deren Regressionstestsuite auf jeden Fall benutzen sollten. Durch deren Benutzung konnten wir nach Änderungen an alten Modulen bzw. Ergänzung von neuen Modulen sicher sein, dass keine Fehler am Bestandssystem entstanden sind. Somit ergaben sich die Aufgaben: Einarbeitung in Rspec, komplette Erfassung und Evaluation der alten TCs. Dies sollte in I06 umgesetzt werden. Da wir diesen Schritt als Teil des **LCA-Meilenstein** sehen, betrachten wir diesen als wieder geöffnet.

C4 Modell um UC01 erweitern

Um mit der Implementierung an sich so schnell wie möglich in der Construction Phase anfangen zu können, wurde hier das C4 Modell um UC01 erweitert.

Durchführung von Ruby on Rails Update

Der Code wurde von richard.boehme@htw-dresden.de angepasst, so dass er mit den neuen Versionen kompatibel ist.

Abschluss der Iteration 05

- Puls-Check I05 (siehe Puls Check im QM)
- Retrospektive

- Projektstatus: **grün**
- Die Funktion des Redeführers wurde abgeschafft, da diese bei einer Gruppengröße von vier Personen nicht benötigt wird.
- Die andauernden Änderungen des Kunden und unser fehlerhaftes Verständnis der agilen Softwareentwicklung hat bei uns zwar viele negative Gefühle ausgelöst und für viel Diskussionsbedarf gesorgt, dennoch haben wir es durch unsere Professionalität und durch das teilweise Einschreiten von Teammitgliedern geschafft, die Iteration abzuschließen.
- Der beabsichtigte Zeitaufwand konnte erfüllt werden.
- Das Gruppen- und Arbeitsklima ist im gelben Bereich.
- Lesson Learned
 - Agile Software Entwicklung wurde verstanden.
 - Unsere Arbeitszeit ist begrenzt: Heißt: wir Arbeiten nicht, bis das Projekt abgeschlossen ist, sondern bis die verfügbare Zeit aufgebraucht ist.
 - Unser Kunde darf jederzeit Änderungen wünschen, wir müssen aber selektieren, welche "In-Scope" oder "Out-Scope" sind.

3.2.3. Iteration 06 (05.05.2021-19.05.2021)

Iterations Ziele:

- Entscheidung über den Login treffen
- Entscheidung treffen, welche alte Test Cases entfernt werden und welche repariert werden
- Update des Systementwurfs im Architecture-Notebook
- Das C4 Modell von Use Case 01 um die Änderung des "Dummy Experiments" erweitern
- Das C4 Modell um Use Case 03 erweitern
- Die Implementierung von Use Case 01
- Fortlaufende Korrektur des Belegs
- Abschluss des **LCA-Meilensteins** und Beginn des **IOC-Meilensteins**

Entscheidung über den Login treffen

Nach dem Kundenmeeting war des Team stark verwundert über die Forderung des Kunden. Es wurden einige Möglichkeiten diskutiert, wie damit umzugehen sei. Es wurde auch der Vorschlag gebracht, sich mit dem Datenschutzbeauftragten der HTW in Verbindung zu setzen. Davon haben wir aber abgesehen, weil wir zu dem Entschluss gekommen sind, dass der Datenschutz in der Verantwortlichkeit des Kunden liegt und es auch nicht die Aufgabe des Entwicklungsteams sei, bezüglich Forderungen oder Wünschen des Kunden, mit dem Kunden zu diskutieren oder gar zu streiten. Wir haben den Kunden darauf hingewiesen, dass wir es bezüglich des Datenschutzes für keine gute Idee halten, wenn der Admin Benutzername und Passwort für die Anwender vergibt. Nochdazu ist diese Forderung für uns auch eine erhebliche Erleichterung, da sich somit der Entwicklungsaufwand bezüglich des Logins drastisch reduziert. Denn dadurch konnten wir den Login unsere Vorgänger verwenden und mussten keine eigene Nutzerverwaltung entwickeln. Hierdurch würde sich auch die Installation auf das Hauptsystem vereinfacht werden, da dadurch auch keine Nutzermigra-

tion erforderlich sein würde.

Entscheidung über die Reperatur oder die Entfernung alter TCs

Nachdem sich der Tester in RSpec eingelese hatte, begann er damit die alte Testsuite genauer zu evaluieren. Dabei stellte er fest, dass nicht alle Tests funktionierten. Ohne das wir Änderungen am Programm durchgeführt hatten, waren 30% der alten Tests fehlerhaft. Um aber einzuschätzen zu können, ob von uns durchgeführte Änderungen zu Fehlern im Bestandssystem führen, wurde diese Testsuite dringend benötigt. Deshalb musste der Tester, mit Unterstützung vom Entwickler, sämtliche fehlerhafte Tests durchgehen. Das Hauptproblem dabei war, dass der alte Testcode kaum dokumentiert war und dessen Zweck deshalb zum Teil durch händisches durchklicken erfasst werden musste. Es gelang uns dann alle relevanten Tests zu reparieren bzw. einige auszusetzen da diese für Programmteile geschrieben waren, die das alte Team nicht mehr implementiert hatte. Insgesamt ergab sich durch die fehlerhaften, schlecht dokumentierten und schlecht gepflegten alten Tests für uns ein Mehraufwand von ca. 10 Stunden. Da der Tester in dieser Phase auch Shutdown-bedingte Zeitmanagement Problem hatte (siehe Probleme im Abschluss der Iteration) wurde das Ziel nicht erreicht.

Update des Systementwurfs im Architecture-Notebook

Durch die Änderungen des Kunden im TSM4 musste auch das Architecture-Notebook entsprechend angepasst werden. Die Änderungen wurden unter Aufsicht des Analysten vorgenommen, da er denn besten Überblick hatte.

C4 Modellerweiterungen

Das C4 Modell wurde um AW04 erweitert und der UC03 wurde mit aufgenommen. Die Arbeit an UC03 ist nicht abgeschlossen worden.

Implementierung von UC01

Der UC01 wurde erfolgreich implemntiert. Um bei der Implementierungen einen perfekten Fach-austausch zu gewährleisten, verwenden wir hier und bei jeder weiteren Implementierung die Pair-programmierung. An der Pairprogrammierung waren der Developer und der Architekt beteiligt.

Fortlaufende Korrektur des Belegs

Wir kamen zu dem Entschluss, dass es eine Verbesserung des Organisationsaufwands wäre, wenn wir die Verbesserungen des Belegs nun nicht mehr durchgehend planen würden, sondern dazu übergehen, sie mit normalen Arbeiten am Beleg zu verbinden. Dies wurde auch in der Work-Item-List mit einem durchgehend Issue realisiert und es wurde ein Dokument in Microsoft Teams erstellt mit den noch zu erledigten Verbesserungen, sodass nichts vergessen wird.

Abschluss des LCA-Meilensteins und Beginn des IOC-Meilensteins

Der **IOC-Meilenstein** konnte durch die Implemntierung von UC01 erfolgreich begonnen werden. Der **LCA-Meilenstein** konnte nicht erfolgreich abgeschlossen werden, da das Ziel bezüglich der alten TCs nicht erreicht wurde.

Abschluss der Iteration 06

Probleme

Problem	Status	Notizen
Stark erhöhter Zeitaufwand bei der Reparatur der alten Test Cases	Gelöst	Da unsere Vorgänger leider keine genau Aufführung Ihrer Test Cases erstellt haben, müssen wir uns nun langsam die alten Test Cases erarbeiten. Dieser Umstand des erhöhten Zeitaufwandes wurde als Problem bei der Kosten-Nutzen-Abschätzung erkannt und in der Risklist festgehalten. Nach der Mitigations-Strategie der Risk List wurde eine Notfalllösung gefunden: die Arbeit an den alten Test Cases wird nun erst einmal stillgelegt, bis eine Lösung dafür gefunden wurde (Coach und Projektvorgänger wurden um Hilfe gebeten). Es wird die Arbeit an unseren aktuellen Test Cases vorgezogen.
Corona bedingte Zeitmanagementprobleme	Gelöst	Da das Corona Hygienekonzept zur erneuten Schließung von Kindergärten geführt hat, kann piet.gutsche@htw-dresden.de , der Vater von zwei Kindern ist, sich nicht in dem Umfang dem Projekt widmen, wie er es möchte. Dies führt zu dem Problem, dass er bei seinem Workload unterstützt werden muss, was ein großes Zeitproblem mit sich bringt. (Coach wurde über dieses "außergewöhnliche" Problem informiert). Das Problem wurde in der Risklist vermerkt und als Lösung wurde beschlossen, dass wenn Zeitprobleme jeder Art auftauchen sollten, diese so schnell wie möglich der Gruppe mitzuteilen sind, dass so schneller gehandelt werden kann.

- Retrospektive
 - Projektstatus: **gelb**
- Unsere Arbeitsschätzungen betrachten wir als GUT, obwohl nicht alle Iterationsziele erreicht werden konnten. Dies begründen wir mit dem oben beschriebenen Corona-Umständen. Dennoch werden wir nun unserer Planung für die nächste Iteration überdenken und die Arbeitskräfte besser verteilen.
- Das Gruppen- und Arbeitsklima ist im grünen Bereich. Siehe [Iterations-Umfrage](#)

3.2.4. Iteration 07 (19.05.2021-02.06.2021)

Iterations Ziele:

- Zusammenführen des UC05, UC06, UC07 in den UC03

- Reparatur oder Entfernung alter TCs
- UC03 im C4 Modell abschließen
- TCs für UC01 werden umgesetzt und formuliert
- UC03 Implementierung
- Erstellung von UC10 bezüglich AW01

Reparatur oder Entfernung alter TCs

Dieses Ziel konnte direkt am Anfang der Iteration erreicht werden, da richard.boehme@htw-dresden.de den Tester pier.gutsche@htw-dresden.de unterstützte. Durch die Expertise im Bereich Coding und Ruby on Rails von richard.boehme@htw-dresden.de konnten die alten TCs validiert werden. Somit betrachten wir hier nun die Elaboration Phase als abgeschlossen und haben den **LCA-Meilenstein** erreicht.

Unsere Entscheidung, die alten TCs erst einmal ruhen zu lassen, die wir in der letzten Iteration getroffen hatten, wurde vom Coach positiv bestätigt.

Ein weiterer Teil unsere Notfallmanagements war Vincent Thelang zu kontaktieren, ob er uns vielleicht weiterhelfen könne. Obwohl wir seine Hilfe bezüglich der alten TCs zum Meetingzeitpunkt (01.06.2021) glücklicherweise nicht mehr benötigten, hielten wir dennoch einen Fachaustausch ab. Hier ergaben sich zwar keine neuen Ergebnisse, aber Herr Thelang meinte, dass wir auf einem guten Weg wären. An dem Fachaustausch beteiligten sich der Projektmanager und der Developer, da hier die Bezugsgößen von Projektorganisation und Projektumsetzung am größten waren. Die alten TCs werden im Beleg dokumentiert, damit zukünftige Entwickler einen besseren Überblick über alle TCs haben und wir beschreiben können, was wir repariert haben oder warum wir TCs entfernt haben.

Zusammenführen der UCs

Bei der Betrachtung der Use-Cases UC05, UC06 und UC07 bemerkten wir, dass wir bei Ihrer Erstellung zu granular vorgegangen waren. Da sich alle Use-Cases auf die Dozentenwoche bezogen, überführten wir diese in UC03. Somit war ein leichteres Verständnis und größere Übersichtlichkeit gegeben. Hier beteiligte sich der Analyst tatkräftig, dass keine Anforderungen verloren gingen.

UC03 im C4 Modell abschließen

Der UC03 wurde im C4 Modell erfolgreich umgesetzt.

TCs für UC01 werden umgesetzt und formuliert

Die TCs für UC01 wurden erfolgreich implementiert. Das Resultat war erfolgreich.

UC03 Implementierung

Der UC03 wurde Implementiert. Hier haben Developer und Architekt Pairprogrammierung betrieben.

Erstellung für UC10 bezüglich AW01

Der UC10 wurde erstellt um die Studiengangsverwaltung zu ermöglichen.

Abschluss der Iteration 07

- Puls-Check I07 (siehe Puls Check im QM)
- Retrospektive
 - Projektstatus: **grün**
 - Das Feedback des Coaches und des Fachaustausch wurden sehr positiv vom Team aufgenommen. Wir fühlen in unserem Tun mehr bestärkt und stehen sicherer hinter unseren Entscheidungen.
 - Durch das gute Voranschreiten im Entwicklungsprozess, sind wir der Meinung die Hauptaufgabe des Buchungssystems erfolgreich abschließen zu können.
 - Der beabsichtigte Zeitaufwand konnte erfüllt werden.
 - Das Gruppen- und Arbeitsklima ist im grünen Bereich.

3.2.5. Iteration 08 (02.06.2021-16.06.2021)

Iterationsziele:

- Implementierung von TCs zu UC03
- Implementierung von UC02, UC08, UC09
- Erstellung von Anwender und Admin Tests
- Inkrement bei TSM5 vorstellen

Implementierung von TCs zu UC03

Die TCs wurden erfolgreich implementiert und durchgeführt.

Implementierung der Use Cases

Bei der Implementierung wurde wieder Pairprogramming angewendet.

Erstellung von Anwender und Admin Tests

Um nochmals die Funktionalität und auch das intuitive Verständnis unseres System überprüfen zu können, erstellten wir geführte Anwender- und Admin Testaufgaben. Der Anwender hatte die Aufgabe eine Vorlesung anzulegen und davon eine Wochenplan-PDF zu erstellen. Der Admin hatte die Aufgabe sich den Wochenplan des Anwenders anzusehen und auszudrucken. Die Aufgaben waren in mehrere Unteraufgaben unterteilt und enthielten mit fortschreitender Aufgabenstellung immer weniger detaillierte Beschreibungen der auszuführenden Aufgaben. Hintergrund hier war, dass wir testen wollten, ob unsere Funktionen des Systems auch verstanden werden, wenn es wenig Hilfestellung dazu gibt.

Inkrement bei TSM5 abhalten (10.06.2021)

Am Anfang wurde das System vom Projektmanager vorgestellt. Anwesend waren hier Frau Prof. Krawietz, Herr Heisig und Herr Zirkelbach (Coach). Die Vorstellung verlief sehr gut, bis auf einen Fehler, dass sich die Buttons der Menüleiste übereinandergeschoben haben und somit die Menüinhalte verdeckt wurden, bei nicht ausreichender Bildschirmbreite.

Anschließend wurde vom Projektmanager der Vorschlag unterbreitet, dass durch den Verzicht auf die Umsetzung von UC10 mehr Zeit für andere Anforderungen entstehen würde. Die Lösung wäre hier, dass die Studiengänge hart ins System gecodet werden und wenn es Änderungen bei den Studiengängen geben sollte, die nach Meinung des Teams selten seien, diese auch wieder im System geändert werden. Eine entsprechende Beschreibung und Anleitung sollte in der Betriebsdokumentation beigelegt werden.

Ein weiterer Vorschlag war auch, auf die Filterfunktion im Journal zu verzichten, um auch hier mehr Zeit zu gewinnen. (Die Vorschläge waren im Vorfeld mit dem Team abgesprochen worden). Der Kunde stimmte bei beiden zu, bestand aber darauf, dass die Filterfunktion in die Zukunftsaussicht mitaufgenommen werden würde. Der Projektmanager bestätigte dies. Der Kunde wünschte nach der Vorstellung noch einige Änderungen:

- Inscope Änderungen
 - Einpflegen der Studiengangsnamen
 - Bei der Veränderung einer Vorlesung im Journal soll ein Button erscheinen mit "Experiment hinzufügen" damit es intuitiver wird
 - Bei der Dozentenwoche soll ein Button erscheinen mit "Drucken" für die Druckfunktionalität
- Outscope Änderungen
 - Es soll eine Notification geben die den Admin darauf hinweist, dass an einem bereits ausgedruckten Wochenplan noch eine Änderung durch einen Dozenten geschehen ist
 - Es soll möglich sein, die Voreinstellung auch noch im Nachhinein zu verändern.

Die gewünschten Änderungen wurde alle akzeptiert, nur die Änderungen bezüglich der Notification löste beim Projektmanager Unverständnis aus, da genau dieser Vorschlag dem Kunden bei TSM2 gemacht wurde und auf Ablehnung traf.

Zum Abschluss des Meetings teilte der Kunde nochmals seine Anerkennung und Dankbarkeit lautstark mit dem Projektmanager. Der Kunde schien über das bisherige System sehr glücklich zu sein.

Aus Zeitgründen wurden die Anwender- und Admintests nicht im Meeting durchgeführt. Frau Prof. Krawietz und Herr Heisig wurden gebeten, diese Zuhause zu erledigen. Beide akzeptierten den Vorschlag und fanden die Idee mit den Usertests gut.

Auf Wunsch des Kunden wurde nach dem TSM5, dem Kunden und Frau Prof. Krawietz Zugang zu dem Testsystem gewährt. Einmal natürlich um die Usertests durchführen zu können, aber auch um das System besser kennenlernen zu können. Wir hielten dies für eine gute Idee, da wir glaubten, es sei besser selbst mit dem System arbeiten zu können, als nur auf eine Präsentation angewiesen zu sein. Der Zugang wurde gewährt.

Zusätzlich konnte der Kunde sich so auch mit der Userverwaltung besser vertraut machen.

Abschluss der Iteration 08

- Puls-Check I08 (siehe Puls Check im QM)
- Retrospektive
 - Projektstatus: **grün**
 - Das sehr positive Feedback des erste Inkrement mit dem Kunden freut uns sehr. Da wir im Laufe des Projektes ja immer wieder Verständigungsschwierigkeiten hatten, war bei uns immer das Risiko präsent, dass der Kunde nicht zufrieden wäre, mit unser Arbeit. Dass dieser Fall nun nicht eingetreten war, erleichterte uns sehr.
- Lesson Learned
 - Inkremente sind unglaublich wichtig, für den Kunden sowie für die Entwickler. Kein Wireframe oder Protokoll kann mit der Informationsdichte oder dem Verständnisaustausch einer wirklichen Vorführung mithalten.

3.2.6. Iteration 09 (16.06.2021-30.06.2021)

Iterations Ziele:

- Einfügen der gewünschten Buttons
- Experimentefeld "Geräte" optimieren
- Projekt mit Kunde abschließen
- Auswertung der User Tests
- Installation des Systems
- Auslieferung des Systems

Einfügen der gewünschten Buttons

Der "Drucken" Button bei der Wochenplanansicht und der "Experiment Hinzufügen" Button bei dem Journal wurden erfolgreich implementiert.

Experimentefeld optimieren

Da wir die Forderung nach einem Buchungssystem als abgeschlossen betrachteten, kamen wir der Aufgabe nach das Experimentenfeld Geräte zu optimieren. Das Ziel war es die Ausgabe der Geräte in der Übersichtsliste zu verbessern, da der Kunde viel Zeit damit verschwendet die Geräte zu finden.

Wir analysierten das Problem und kamen zu dem Schluss, dass hier eine Stringsuche am besten geeignet war. Hier war der Vorteil, dass die Geräte beliebig erweitert werden konnten, ohne die Suche anpassen zu müssen.

TSM6 Projekt mit dem Kunden abschließen

Die Vorführung der Buttons und des optimierten Gerätefelds wurden von dem Kunden erfreut

angenommen. Allerdings wurden von Ihm noch einige Änderungen gewünscht. Die meisten davon waren visueller Natur, wie die Umbenennung und Umfärbung von Buttons. Eine jedoch war direkt so groß, dass der Projektmanager sie direkt im Meeting als Outscope deklarieren konnte. Der Kunde forderte noch die Einführung einer zusätzlichen Rolle, die des Assistenten. Eine genaue Auflistung der Forderungen findet man in der Vision.

Auch wurde hier gleich die Auslieferung und deren Ablauf mit dem Kunden besprochen. Der Kunde war mit den Abläufen einverstanden. Die Installation wurde auf den 28.06. und den 29.06.2021 festgelegt. Der Ausliefertermin sollte am 01.07.2021 stattfinden.

Entscheidung über Ablehnung des Änderungswunschs vom Kunden

Beim Treffen am 23.07.2021 teilte uns der Kunde mit, dass er zahlreiche Änderungen wünscht. So sollten mehrere Buttons umbenannt sowie farblich und in der Form verändert werden. Außerdem wünschte er sich eine zusätzliche Rolle und die Programmlogik sollte auch an einer Stelle geändert werden. Darüber hinaus zum Umfang dieser Änderungen waren Benennungswünsche widersprüchlich zur Funktionalität der Buttons. Nach längerer Beratung entschieden wir uns, diese Änderungswünsche abzulehnen und diese in der Zukunftsvision zu dokumentieren. Wir begründeten diese Entscheidung mit folgenden Argumenten:

- Installation auf das Produktivsystem war für den 28.7.2021 geplant und sollte nicht verschoben werden
- somit noch 2 Werkzeuge Zeit für Umsetzung der Wünsche
- Umsetzung hätte aufgrund ihrer Widersprüchlichkeit und Tiefe aber eine komplette Iteration benötigt
- Dokumentation der Testcases und des Anwenderhandbuchs wären inkonsistent geworden da diese bereits erstellt wurden und aus zeitlichen Gründen nicht mehr geändert werden könnten

Zusammenfassend lässt sich also sagen das wir die Änderung ablehnten da ihre Umsetzung das brechen mit unseren Entwicklungsprinzipien und Best Practices bedingt hätte. Ein Umsetzen wäre nur im sprichwörtlichen Schnelldurchlauf möglich gewesen, was die Wahrscheinlichkeit von unentdeckten Fehlern im Code und am Produktivsystem erhöht hätte. Diese wären aber ab dem 02.07.2021 für einen längeren Zeitraum nicht behebbar gewesen, da dann die Betreuung durch unser Team endet. Deshalb war es für uns wichtiger sorgfältig getesteten Code mit einer stimmigen Dokumentation abzugeben als auf die Wünsche einzugehen. Alle vom Wunsch betroffenen Teile wurden dem Kunden vorher mehrmals gezeigt und von ihm bestätigt. So waren alle z.B. alle Bezeichnungen bereits in den vorgelegten Wireframes enthalten. Auch beim dem vom Kunden durchgeführten Usertests hätte er diese Wünsche äußern können. Das Aufkommen eines solchen Änderungswunsches 2 Werkzeuge vor dem Aufspielen auf das Livesystem zeigt aber auch, dass es uns trotz intensiven Bemühungen nicht gelungen ist, dem Kunden unser Arbeitsweise so zu erläutern das er sie ausreichend verstanden hat. Über die gesamte Dauer des Projekts haben wir die Beobachtung gemacht, das der Kunde Änderungen in Bereichen wünscht die er bereits abgenommen hatte, die Priorisierung von Abschnitten anpassen möchte und Vorschlägen, Beratungen und Erklärungen von uns kaum Beachtung schenkt.

Auswerten von User Tests

Die Auswertung hat ergeben, dass Frau Prof. Krawietz mit den von uns gestellten Aufgaben, sehr gut

zurecht kam und Ihr auch keine Fehler aufgefallen sind. Wir sehen uns hier bestätigt, dass wir die geforderte innovative Arbeitsweise auf dem System erreichen konnten, da Frau Prof. Krawietz keine Hilfestellung hatte und auch keinen Zugriff auf eine Anwenderdokumentation. Von Herr Heisig haben wir leider keine ausgefüllten Unterlagen erhalten, aber auch er betonte im Meeting, dass er mit den Aufgaben keine Probleme hatte.

Installation des Systems

Die Installation des Systems wurde am 28.06.2021 von unserem Developer richard.boehme@htw-dresden.de durchgeführt und konnte erfolgreich, ohne Probleme abgeschlossen werden.

4. Ergebnisse

4.1. Ziel: Einheitliches Buchungssystem für EMS mit Buchungsverwaltungsfunktionalitäten

4.1.1. Anlegen und Speichern einer Physikvorlesung

Diese Ziel wurde erreicht, es ist nun mit unserem System möglich eine komplette Vorlesung, mit dem Studiengang, einem optionalen Titel, dem Datum, der Uhrzeit und den entsprechenden Experimenten, die in der Vorlesung vorgeführt werden sollen, zu planen und zu buchen. Dies wurde erreicht mit Hilfe der **Voreinstellung**, welche die Kerndaten der Vorlesung, wie Studiengang, optionaler Titel, Datum und Uhrzeit festlegen. Die Auswahl der Experimente erfolgt über ein Overlay in der Experimentenübersicht, in dem man durch Klicken auf ein Einkaufswagensymbol Experimente zu Vorlesung hinzufügen oder abwählen kann. Beim Buchungsabschluss wird dem Nutzer eine Übersicht der ausgewählten Experimente im **Warenkorb** angezeigt, wo er nun auch die Reihenfolge der Experimente verändern kann. Zusätzlich kann er bei Bedarf auch einen Kommentar pro Vorlesung hinterlassen.

4.1.2. Bearbeitung von Vorlesungen

Auch dieses Ziel wurde erreicht und wird von uns durch das **Journal** umgesetzt. Hier kann sich der Dozent alle bisher gebuchten Vorlesungen ansehen und diese, wenn die Vorlesung noch nicht gehalten wurde, verändern. Hier können Experimente hinzugefügt werden oder aus der Buchung gelöscht werden. Auch lässt sich hier auch die Reihenfolge der Experimente und der Kommentar bearbeiten. Die gesamte Buchung kann hier auch gelöscht werden, wenn sie noch in der Zukunft liegt.

4.1.3. Darstellung der gebuchten Vorlesungen

Ein weiteres Ziel, dass wir durch die **Dozentenwoche** erreichen konnten, bestand in einer strukturierten Darstellung der Buchungen. Durch klicken auf die Dozentenwoche, kann der Dozent sich nach Auswahl des Jahrs und der Kalenderwoche eine Wochenplanübersicht erstellen lassen. Diese wird dann online angezeigt und kann über die Druckfunktion des Browsers gedruckt oder als PDF generiert werden.

Der Admin verfügt über die gleiche Möglichkeit, kann aber zusätzlich noch den Dozenten auswählen, dessen Dozentenwoche er einsehen möchte.

4.1.4. Abgrenzung der Nutzer

Ziel erreicht durch den **Login**. Durch unterschiedliche Userrollen ist es möglich, die Nutzer mit ihren unterschiedlichen Funktionen und Rechten voneinander zu trennen. Auch wird es so ermöglicht, dass der Admin keine Vorlesungen der Dozenten bearbeiten kann, oder die Dozenten Vorlesungen anderer Dozenten verändern können. Weiterhin hat der Admin auf alle Funktionen Zugriff, die der Experimentenbearbeitung oder allgemeinen Systemverwaltung (Nutzerverwaltung etc.) entsprechen. Der Login wurde von uns nicht selbst erstellt, sonder wir verwendeten die vorhandene

4.2. Einfachere Auswahl eines Geräts bei der Erstellung eines Experiments

Durch Hinzufügen einer Suchleiste in der Eingabemaske, kann der Admin sich nun gezielt Geräte anzeigen lassen. Das Ziel wurde erreicht.

4.3. Buchungshäufigkeit auswerten

Diese Ziel wollten wir mit Hilfe der **Statistik** umsetzen. Die entsprechenden Use Cases und Test Cases wurden erstellt. Sie wurde aber nicht implementiert. Das Ziel wurde nicht erreicht.

4.4. Fehlerbehebung

Es wurde gewünscht zwei Bugs auf dem Vorhanden System zu beheben. Der eine Bug bezog sich auf einen Anzeigefehler von Textfeldern, dieser wurde behoben. Der andere war ein Problem bei der Organisation von Bildern, der Experimente. Dieser wurde nicht behoben. Wir betrachten dieses Ziel nur als teilweise erreicht.

4.5. Begründung für das Nichterfüllen von Zielen

Wir hatten nicht genug Zeit alle Ziele zu erfüllen. Dadurch, dass zwei Teammitglieder uns am Anfang des Semesters verlassen haben und uns nur ein neues Teammitglied hinzugefügt wurde, fehlte uns die komplette Arbeitskraft eines Teammitglied über das Semester. Auch die Einarbeitungszeit des neuen Mitglieds konnte nicht produktiv für das Projekt genutzt werden. Auch die vielen Änderungen des Kunden, behinderten uns stark am eigentlichen Arbeiten an der Aufgabe, da eigentlich beschlossene Aufgaben immer wieder neu analysiert und dokumentiert werden mussten. Auch die Korrektur des Belegs am Anfang des Semesters kostete viel Arbeitszeit, die aber für das System keine Fortschritte gebracht hat.

4.6. Reflexionen der Teammitglieder

4.6.1. Gregor Müller

Bei dem Projekt in SE gab es für mich einige Höhen und einige Tiefen. Ich durfte durch das Projekt, einige tolle Bekanntschaften machen und auch den Lehrkörper aus einer anderen Perspektive kennenlernen. Zum Beispiel wurde mir erst hier Bewusst, dass es auch an unsere Hochschule einige Menschen gibt, die sich gegen Veränderung sträuben und es ein harter Kampf sein kann, neue Systeme in den Arbeitsalltag zu integrieren. Diese Erfahrung machte ich so zum ersten Mal und dachte bis jetzt immer, diese typischen Aussagen wie „aber früher war alles besser“, würden sich auf mein Berufsleben begrenzen. Und ich bin auch dankbar dafür, immer wieder argumentiert haben zu müssen, warum und wieso und weshalb und weswegen wir unsere Entscheidungen getroffen haben. Dies lenkte meinen Fokus auf die wirklichen Hintergründe einer Forderung des Kunden. Es war zwar öfter mehr als frustrierend, das gebe ich zu, aber im Nachhinein betrachtet glaube ich,

dass ich so auch auf vielleicht noch schwierigere oder komplizierte Kunden vorbereitet wurde.

Die ewige Anfertigung von Dokumentationsbelegen empfand ich teilweise mehr als hinderlich, als förderlich. Ich dachte oft, es ist doch klar, warum wir dies oder das machten, oder das der Detailgrad einfach zu groß war, der in den Belegen gefordert wurde. Aber als dann der Punkt im Projekt kam, an dem wir die alten TCs bearbeiten wollten und wir uns kaum erklären konnten, was hier eigentlich was macht und warum überhaupt, überdachte ich meine Einstellung. Man kann einfach nie sagen, wie es nach einem Projektabschluss weiter geht und was ein zukünftiger Entwickler vielleicht noch alles für Zusatzinformationen benötigt. Heute sage ich, dass die Dokumentation mehr als wichtig ist und auch stets, auch im Privatbereich eingehalten werden sollte.

Zu guter Letzt möchte ich noch ein paar loben Worte an mein Team richten. Ich durfte lernen wie wichtig es ist, ein Team zu haben mit dem man wirklich gut und eifrig zusammenarbeiten kann. Wie wichtig es ist, dass man sich auf Absprachen verlassen kann oder wenn Fehler passieren, man nicht damit beschäftigt ist einen Schuldigen zu finden, sondern wie man das Problem lösen kann. Ein Team bei dem man ehrlich sein kann und auch mal gemäßregelt wird, wenn etwas über die Stränge schlägt. Bei Problemen der anderen Teams, habe ich oft gehört „ja, die anderen“, deshalb bin ich so froh, dass bei meinem Team alles seit der ersten Minute auf der menschlichen Seite funktioniert hat. Das Einzige was ich bedauere ist nur, dass ich durch Corona noch nie einem der anderen, die mich nun seit fast einem Jahr begleiten wirklich ins Gesicht sehen konnte.

Ja ich bin stolz auf das, was wir erreicht haben. Ich bin stolz darauf, dass wir fast alle Hürden und Problem erfolgreich meistern konnten und ich bin stolz drauf was ich lernen konnte und auch darauf, dass ich für das Projekt viel einbringen konnte und durfte.

Gregor Müller

4.6.2. William Elsner

Diese Reflexion bezieht sich auf das Gruppenprojekt im Rahmen der Prüfungsleistung des Moduls Software Engineering II. Für die Belegarbeit im Rahmen von SE I wurde bereits eine Reflexion abgegeben, welche sich fast ausschließlich auf die Anforderungsanalyse bezieht. Da diese in SE II nicht mehr primär im Vordergrund stand, möchte ich mich in diesem Dokument ausschließlich auf dieses Semester und die eigentliche Entwicklung beziehen.

Das Ziel war es weiterhin, ein bereits bestehendes, HTW-internes Informationssystem bezüglich der Experimente der Physikfakultät um ein nutzerfreundliches Buchungssystem zu erweitern. Dies diente dem Zweck, dem Themensteller und den anderen Anwendern manuelle Arbeit und überflüssigen Mailverkehr abzunehmen. Des Weiteren sollten kleinere Bugs gefixt werden.

Anfangs möchte ich betonen, dass das Projekt mir wirklich Spaß gemacht hat. Im Laufe der zwei Semester hat sich unser Team so gut eingearbeitet, dass sogar der Verlust der Wirtschaftsingenieure und zwei weiterer Teammitglieder keine unüberwindbare Hürde war. Auch die Einarbeitung eines neuen Teammitglieds hat dementsprechend gut funktioniert. Zusammenfassend möchte ich hierzu unterstreichen, dass, meiner Meinung nach, sogar gute persönliche Verhältnisse entstanden sind, die sich nicht nur sehr positiv auf unsere Zusammenarbeit und Kommunikation als Team ausgewirkt, sondern auch für modulexterne Kooperationen gesorgt haben.

Des Weiteren fällt mir auf, dass sich mein entwicklungsbezogener Kenntnisstand gefestigt und

erweitert hat. Durch den konstanten Austausch mit Teammitgliedern mit erhöhter Expertise wurde mir viel über Architektur, Konventionen und sämtliche Methoden vermittelt, vom Testing bis hin zum Qualitätsmanagement. Auch habe ich eine neue Programmiersprache (Ruby) und das verwendete Framework (Ruby on Rails) kennengelernt, sodass meine Einarbeitungsaufwand für das eventuelle Nutzen dieser Sprache verringert wurde, parallel zur gestiegenen Neugier.

Das anfängliche Missverständnis der agilen Entwicklung hat sich auch aufgeklärt. Hierfür könnte man unseren Kunden als Grund nennen, welcher ohne Rücksicht auf Verluste, aber selbstlos im Hinblick auf „seine“ Dozenten, für ständige Anforderungsänderungen oder -zugänge sorgte. Um ehrlich zu sein, hat dies oft für Verwirrung und Frustration innerhalb des Teams und auf meiner Gemütsseite gesorgt. Ich denke jedoch, dass genau das dafür verantwortlich war, dass das Risiko- und agile Verständnis (vor allem im Kontext des OUP) gesteigert wurden und wir sozusagen den realistischen „Extremfall“ erlebt haben.

Der große Dokumentationsaufwand schien mir teilweise etwas überflüssig. Dennoch bin ich der festen Überzeugung, dass ich nur daraus gelernt habe – zu viel ist meistens besser als zu wenig.

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, sodass sich der große Zeitaufwand definitiv gelohnt hat. Ich bin durchaus zufrieden.

William Elsner

4.6.3. Richard Böhme

Während des zweisemestrigen Moduls zum Thema Software Engineering durfte ich mit einem wechselnden Team ein erstes eigenes Softwareprojekt umsetzen. Solch eine langwierige Gruppenarbeit war für mich aus Sicht der Universität etwas komplett Neues und bisher Einzigartiges.

Trotz gewissen Vorkenntnissen aus meiner Tätigkeit als Werksstudent konnte ich eine Menge Neues lernen und auch zum Team beitragen. So konnte ich mich mit Themen der Anforderungsanalyse oder der Architektur eines Systems und dessen Dokumentation tiefgehend beschäftigen und lernen, wie man solch eine Arbeit iterativ und kollaborativ durchführt.

Natürlich war es auch von Nutzen vorherige Erfahrungen gesammelt zu haben. Ich konnte mein Team maßgeblich bei der Entwicklung der Anwendung unterstützen und leiten. Außerdem kannte ich bereits Konzepte wie iterative Entwicklung und Tools wie Git oder Github. Zudem war mir sowohl die, doch recht selten benutzte, Programmiersprache und das entsprechende Webframework bekannt. So war es mir möglich mein bereits bestehendes Wissen in diesen Bereichen mit meinem Teammitgliedern in der Form von Workshops und Dokumentationen zu teilen. Es ermöglichte ein deutlich angenehmeres Arbeiten an dem System, weshalb ich dankbar bin, dieses Projekt zugeordnet bekommen zu haben.

Allerdings brachten die Vorkenntnisse auch einen kleinen Nachteil mit sich. Zum einen war mir so bewusst, dass ich im Thema Programmiersprachen oder Frameworks keinen Wissensgewinn erlangen werde, da ich mich bereits gut mit dem verwendeten Techstack auskannte. Außerdem kosteten mich sowohl Dokumentationen oder Workshops als auch die komplette Umsetzung deutlich mehr

Zeit als ich am Anfang für den Beleg in Software Engineering eingeplant hatte. Dadurch, dass kein anderes Teammitglied Erfahrung in der Entwicklung mit den verwendeten Technologien hatte, blieb viel der Arbeit daran an mir hängen. Im Endeffekt ist dies für mich allerdings kein Problem gewesen. Die Arbeit hat mir Spaß gemacht und ich kann natürlich auch nachvollziehen, dass es nicht sinnvoll ist einem Teammitglied eine Aufgabe zu geben, die ich in einem Viertel der Zeit erledigen könnte. Das Team hat dafür automatisch darauf geachtet meinen Aufwand bei anderen Aufgaben gering zu halten, was ich sehr zuvorkommend und nett finde. Ich habe stets versucht meine Begeisterung der Programmiersprache mit den anderen Entwicklern zu teilen. Ob das funktioniert hat, kann ich nicht mit aller Sicherheit sagen, allerdings hat es auch Spaß gemacht Ruby und Rails den anderen vorzustellen.

Auch ich empfand den Dokumentationsaufwand für die Größe unseres Projektes etwas übertrieben. Dadurch, dass das Projekt klein war, hatte man häufig das Gefühl sich Inhalte „aus den Fingern ziehen“ zu müssen. Das ist natürlich damit begründet, dass wir das generelle Format lernen sollen und größere Projekte nicht in zwei Semestern umsetzbar sind.

Letztendlich kann ich sagen, dass ich viel aus diesem Projekt gelernt habe. Sei es die erfolgreiche und sehr entspannte Zusammenarbeit mit einem Team bestehend aus Menschen, die ich weder vor dem Projekt noch nach dem Projekt nur einmal gesehen habe oder die große Verantwortung die ich als Hauptentwickler übernommen habe und wie ich damit zum Projekterfolg beitragen konnte. Ich habe mich von Anfang an auf das Projekt gefreut und auch wenn es natürlich viel Zeit gekostet hat, bin ich immer noch glücklich es durchgeführt haben zu können.

Richard Böhme