Inhalt

[1. Unsere Desktopanwendung 2](#_Toc39660243)

[2. Teambuilding und Organisation 2](#_Toc39660244)

[3. Scrum Vorbereitungen 3](#_Toc39660245)

[3.1 Definition of Ready 3](#_Toc39660246)

[3.2 Definition of Done 4](#_Toc39660247)

[3.3 Schätzgrößen 4](#_Toc39660248)

[4. Auswahl der Programmiersprache 4](#_Toc39660249)

[5. Sprint Inkremente 5](#_Toc39660250)

[6. Gestaltung der Anwendung 5](#_Toc39660251)

[7. Retrospektiven 5](#_Toc39660252)

[8. Anleitung 6](#_Toc39660253)

[9. Reflexionen der einzelnen Personen 6](#_Toc39660254)

[10. Links zu Trelloboards 6](#_Toc39660255)

[11. Lop Liste 6](#_Toc39660256)

# Unsere Desktopanwendung

Die Aufgabe in unserem Projekt ist es eine Desktop-App für Sekretariate zu entwickeln und zwar ein Excel-Moodle-“Gateway“ für Noten und Feedback. Die Sekretäre an Hochschulen haben gegen Ende eines Semesters oft viel Arbeit mit dem Eintragen der Noten für die Studierenden. Im Moment läuft dies alles händisch ab. Das heißt die Dozenten geben im Sekretariat meist eine Excel-Tabelle ab, in der die Noten für alle Studierenden eines Kurses eingetragen sind. Teilweise sind dort auch nicht die endgültigen Noten angegeben, sondern Teilnoten, die dann noch verrechnet werden müssen. In Studiengängen mit viel Praxisbezug kann die Zahl der Einzelnoten auch schnell sehr groß werden. Dagegen wollen wir Abhilfe schaffen indem wir eine Desktopanwendung entwickeln, die es einem Sekretär oder einer Sekretärin mit wenigen Schritten ermöglicht, Noten schnell und unkompliziert hochzuladen. Dazu haben wir uns einen genauen Plan gemacht, wie dies funktionieren soll. Wir erstellen Excelvorlagen, die die Dozenten nutzen können um darin die Noten für ihre Studenten einzutragen. Dazu haben wir verschiedene Vorlagen. Eine mit Matrikelnummern und eine mit Namen. Diese können in der Anwendung dann ausgewählt werden und ganz einfach per Klick in Moodle importiert werden. So ist es nicht notwendig die Noten für jeden Studenten einzeln einzutragen. Aus der Anwendung sollen diese dann einfach, über eine eingegeben URL, in Moodle zu dem richtigen Studierenden importiert werden. Das war unsere Grundidee, die wir durch unsere User Stories und innerhalb der einzelnen Sprint To-Do´s genauer definiert haben. Dadurch soll vor allem Sekretariaten das Leben erleichtert werden. Die Arbeit die vorher viel Zeit gekostet hat wird verkürzt und mit wenigen Klicks erledigt. So ist es auch möglich, dass Sekretäre ihren Fokus besser auf andere Dinge wenden können. Wichtig für die Anwendung ist aber auch, dass Dozenten die vorgegeben Excel-Tabellen verwenden. Das erleichtert auch diesen die Arbeit und die Originaldateien sind in Zukunft auch einheitlich. Das Entwicklungsprojekt wird hier mit der agilen Projektmanagement-Methode Scrum umgesetzt. Da es aber eine vorgegebene Anzahl an Sprints gibt, ist unser Projekt in dem momentanen Stand noch nicht vollendet. Die Dokumentation dient deshalb auch dazu, dass daran weitergearbeitet werden kann und verstanden wird, was bereits gemacht wurde und wie wir dabei vorgegangen sind. Zunächst soll deshalb erläutert werden, wie wir uns innerhalb Scrum organisiert haben und im zweiten Teil dann näher auf die Umsetzung eingegangen werden.

# Teambuilding und Organisation

Noch während der Vorlesung zu Scrum haben wir bereits Maßnahmen zum Teambuilding getroffen. Folgende Regeln haben wir für die Zusammenarbeit definiert:

* Respektvoller Umgang miteinander
* Ausreden lassen und zuhören
* Pünktlich kommen und Aufgaben erledigen (unmittelbar, rechtzeitig absagen)
* Offene Kommunikation
* Aufgaben sorgfältig erledigen
* Gleichberechtigung
* Kompromissbereitschaft
* 10 Minuten vor einem Meeting Internetverbindung checken und einloggen

Diese sind sehr wichtig, damit wir untereinander ein gutes Arbeitsklima schaffen. Außerdem helfen diese Regeln die Zusammenarbeit zu fördern und Probleme leicht zu beheben, beziehungsweise sie gar nicht erst entstehen zu lassen.

Zur Organisation der Dokumente und Kommunikationsmittel haben wir uns auch Gedanken gemacht. Als Kommunikationskanäle haben wir uns für Whatsapp, Discord, Whereby und Trello entschieden. Die Übersicht zeigt genauer, für was wir welchen Kanal benutzen möchten.

|  |  |  |  |
| --- | --- | --- | --- |
| Whatsapp | Discord | Whereby | Trello |
| Für spontane Treffen und kurze Fragen,  🡪 Erleichterung dieser Kommunikation | Gruppentreffen (Video, Screenshare), gespeicherter Chat | Gruppentreffen während Vorlesung und Treffen mit Stakeholdern | Organisation von Aufgaben, Übersicht |

Außerdem haben wir folgende Kanäle für unsere Dokumentenablage ausgewählt.

|  |  |  |
| --- | --- | --- |
| GitHub | E-Mail | Google Drive |
| Dokumentenablage, versionskontrolliertes Speichern, gleichzeitiges Arbeiten | Daten teilen, Dokumente nur für einzelne Mitglieder | Dokumentenablage/ - austausch |

Die Kommunikation über diese Kanäle hat sehr gut funktioniert. Jedes Mitglied war dadurch immer auf dem neuesten Stand und hatte immer die aktuelle Version des Projekts.

Als nächstes hat sich jedes Teammitglied seine Stärken vor allem in Hinsicht auf dieses Projekt überlegt. Daraus haben wir auch eine Übersicht erstellt.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Pascal Feinauer | Christian Dänzer | Jonas Althoff | Marco Scotellaro | Katharina Schmitt |
| Menschen überzeugen, Planung | Für Gruppe sprechen, Programmieren | Technik, Programmieren | Design, Präsentation | Organisation, Beratung |

Darauf aufbauend haben wir zum Schluss die Rollen des Scrum Teams auf unsere Mitglieder verteilt. Diese sieht folgendermaßen aus.

Product Owner: Pascal Feinauer

Scrum Master: Katharina Schmitt

Entwicklungsteam: Christian Dänzer, Jonas Althoff, Marco Scotellaro

# Scrum Vorbereitungen

Zur Scrum Vorbereitung gehört auch das definieren der Definition of Ready und der Definition of Done. Diese haben wir ebenfalls gemeinsam festgelegt, damit jedes Mitglied des Scrum-Teams das gleiche Verständnis davon hat.

## 3.1 Definition of Ready

Unsere User Stories sind unabhängig voneinander. Zusätzlich sind sie klein genug, damit sie innerhalb eines Sprints umgesetzt werden können. Dadurch haben wir zu jedem Release eine funktionsfähige Software. (Sie können, wenn das Entwicklungsteam den Anspruch hat, vom Product Owner zerlegt werden)

Sie sind ausreichend weit gefasst, damit wir genug Spielraum haben, um zu besprechen, was genau entwickelt werden soll und uns Änderungen vorbehalten sind. (Gespräch zwischen Entwicklungsteam und Product Owner)

Außerdem liefern alle unserer User Stories einen Wert, das heißt auch, sobald eine User Story überholt wird und diese somit keinen Wert mehr hat, wird sie gelöscht.

Unser Aufwand der Umsetzung ist gut zu schätzen, da wir genügend Informationen zu den Anforderungen unseres Stakeholders haben. (Herr Mester)

Wir haben zu jeder User Story ein Akzeptanzkriterium, damit wir überprüfen können, ob die Umsetzung planmäßig erfolgt ist. So kann sichergestellt werden, ob es einen Wert für den Stakeholder hat.

## 3.2 Definition of Done

* User Stories wurden in Tasks geteilt
* Alle Akzeptanzkriterien wurden erfüllt
* Der Code ist vollständig implementiert und kommentiert
* Der Code ist immer aktuell auf Git
* Ein Code Review wurde durchgeführt
* Der Code wurde getestet und funktioniert
* Es gibt keine kritischen Fehler

Hier haben wir einmal einen Text und einmal Stichpunkte geschrieben, damit wir beides ausprobieren können und sehen, womit wir besser zurechtkommen. Wir konnten mit beidem gut arbeiten. Jedoch waren die Stichpunkte etwas leichter, da hier die einzelnen Punkte, die erfüllt sein müssen, direkt erkannt werden können.

## 3.3 Schätzgrößen

Außerdem gehörte zur Vorbereitung das Festlegen unserer Schätzgrößen. Dabei haben wir uns auf Kuchen geeinigt. Unsere Skala sieht folgendermaßen aus.

Cakepops

Sehr einfach

Sehr schwer

Macarons

Muffins

Cupcakes

Kuchen

Torten

Hochzeitstorte

Nach dieser Skala haben wir dann auch alle unsere User Stories geschätzt. Im Verlauf unseres Projekts haben wir außerdem noch Magic Estimation ausprobiert und auch hiermit nochmal neu geschätzt. Dabei sind wir aber auf gleiche beziehungsweise sehr ähnliche Ergebnisse gekommen. Ab dem zweiten Sprint haben wir dann beide Methoden verwendet und unsere „Kuchenskala“ noch um Zahlen der Fibonacci-Reihe erweitert:

1 - Cakepops

Sehr einfach

Sehr schwer

2 - Macarons

3 - Muffins

5 - Cupcakes

8 - Kuchen

13 - Torten

20 - Hochzeitstorte

## 3.4 Priorisieren

Für das Priorisieren der einzelnen User-Stories haben wir die Methode „Buy a Feature“ kennengelernt. Damit haben wir dann unsere User-Stories neu priorisiert. Jeder hatte hier zu Beginn 556“€“ zur Verfügung. Dabei sind wir dann auf Folgendes Ergebnis gekommen, bei dem eine verkaufte User-Story durch den Namen des Mitglieds unseres Scrum-Teams markiert ist.

Als User möchte ich eine Vorlage haben, in die ich meine Noten eintragen kann.

10 Jonas

Als User möchte ich eine Anwendung mit einer ansprechenden Benutzeroberfläche haben, damit ich mich gut zurecht finden.

250

Als User möchte eine plattformübergreifende Desktopanwendung, damit ich das Programm benutzen kann.

800 Passi und Marco

Als User möchte ich in der Anwendung problemlos auf den Darkmode wechseln können.

10 Jonas

Als User möchte ich eine Excel-Datei, die Noten enthält, in Moodle importieren, damit ich die in Excel stehenden Noten dem in Moodle hinterlegten Studenten für eine bestimmte Leistungserhebung nicht alle einzeln eintragen muss.

1000 Chris und Katha

Als User möchte ich mich über die Anwendung in Moodle einloggen können, damit ich dann von dort aus den Import starten kann.

900 alle

Als User möchte ich Dateien in einem Auswahlfeld auswählen können und per Mausklick z.B. in Button-Form in Moodle exportieren können, damit ich nicht extra die Anwendungen verlassen muss.

100

Als User möchte ich Noten für nur einen Studenten eintragen können damit noch nicht finale Noten ausgespart werden.

700

Als User möchte ich Excel-Tabellen kombinieren können und die Prozentanteile der Noten zuteilen können, damit ich das nicht händisch machen muss.

400

# Produktvision und Anforderungen an das Produkt

# Auswahl der Programmiersprache

Für unser Entwicklungsprojekt haben wir uns für die Programmiersprache JavaScript mit Electron entschieden. Warum?....

# Sprint Inkremente

In unserem ersten Spint war unser Inkrement eine GUI in XD sowie dies bereits in HTML umgesetzt und der Funktion Excel-Tabellen auszuwählen.

Der zweite Sprint hatte als Inkrement das Hochladen der Noten in Moodle zum richtigen Studenten mit Einloggvorgang aus der Desktopanwendung heraus. Und zusätzlich einen dynamischen Darkmode.

# Gestaltung der Anwendung

Unser Entwicklungsteam hat hierzu ein Styletile sowie eine GUI in XD ausgearbeitet an denen wir uns orientieren können. Außerdem haben wir uns einen Namen für die Anwendung überlegt: The Gate. Das Entwicklungsteam hat hierzu auch ein Logo erstellt, an dem die Anwendung schnell erkannt werden kann.

# Umsetzung

# Ergebnis

Mit screenshots

# Problemdokumentation

# Anleitung

# Retrospektiven

Während unseren Sprints sind wir natürlich auch auf verschiedene Probleme gestoßen, die uns bei der Arbeit behindert haben. Diese haben wir immer in der Sprint Retrospektive analysiert und besprochen. Dabei sind wir in der ersten Retrospektive zu folgenden Ergebnissen gekommen.

Was ist gut gelaufen?

* Teamkommunikation
* Zeitliche Einhaltung
* Aufgabenverteilung
* Bestätigung der richtigen Rollenverteilung
* Jeder hatte gleiches Verständnis der User Stories

Was ist schlecht gelaufen?

* Verständnisproblem mit Herr Mester 🡪 mehrmals Sprint Planning neu machen
* Sprint Backlog nicht immer direkt nach Erledigung der Aufgabe verändert
* Product Owner keine Idee was als nächstes kommen soll
* Termine zu spät vorbereitet

Was wollen wir ändern?

* Entwicklungsteam soll Sprint Backlog immer aktuell halten und direkt anpassen
* Product Owner vor Sprint Review schon Gedanken über nächsten Schritt machen

Wie setzen wir Änderung um?

* Jeder Entwickler seine Aufgaben direkt anpasst und weiterverschiebt, wenn es erledigt ist 🡪 dazu in Whatsapp Erinnerungen an Entwicklungsteam geben
* Product Owner soll sich vor Sprint Review nochmal das Product Backlog anschauen

Was wollen wir beibehalten?

* Kommunikation beibehalten
* Schauen, dass wir weiterhin zeitlich im Rahmen bleiben

In der Retrospektive 2 hatten wir dann dieses Ergebnis:

Was ist gut gelaufen?

* Teamarbeit
* Kommunikation
* Aufteilung Aufgaben und Hilfe
* Abnehmen von Aufgaben, wenn einer zu viel hat
* Fehlersuche und Alternativen finden
* Hilfe angefragt

Was ist schlecht gelaufen?

* Externer Support
* Überforderung mit Umsetzung
* Ziele nicht erreicht
* Wissen noch nicht genau, wie wir weitermachen sollen

Was wollen wir ändern?

* Zu Alternativen direkt Beispiele zeigen
* Nachfragen wie es weitergehen soll

Wie setzen wir Änderung um?

* Direkt in Kontakt mit Herr Mester treten

Was wollen wir beibehalten?

* Kommunikation im Team
* Gute Zusammenarbeit

# Reflexionen der einzelnen Mitglieder

# Trelloboards

# Lop Liste



# Quellen

Dir für Entwicklung verwendet wurden