Software Engineering Projekt

Entwicklung einer Anwendung zum Pokerspielen am Computer

Miriam Haas

Sophia Schmoll

Noah Hauke

Luca Rohmann

Felix Schmitt

Informationstechnik

Duale Hochschule Baden-Württemberg

Studienjahrgang: 2019

Bearbeitungszeitraum: 10.09.2020 – 19.11.2020

Inhaltsverzeichnis

[1 Allgemein 1](#_Toc55747493)

[1.1 Projektbeschreibung 1](#_Toc55747494)

[1.2 Vorgehensmodell 1](#_Toc55747495)

[1.3 1.3. Programmierstandards 2](#_Toc55747496)

[1.4 1.4 Qualitätsmanagement 2](#_Toc55747497)

[1.5 1.5 Gleichberechtigung 3](#_Toc55747498)

[2 2. Sprint-Planning 4](#_Toc55747499)

[2.1 2.1 Sprint Planning am 24.09.2020 4](#_Toc55747500)

[2.2 2.2 Sprint Planning am 08.10.2020 4](#_Toc55747501)

[2.3 2.3. Sprint Planning am 22.10.2020 5](#_Toc55747502)

[3 3. Sprint Review 8](#_Toc55747503)

[3.1 3.1 Sprint Review am 08.10.2020 8](#_Toc55747504)

[3.2 3.2 Sprint Review am 22.10.2020 8](#_Toc55747505)

[4 4. Sprint Retrospektive 9](#_Toc55747506)

[4.1 4.1 Sprint Retrospektive am 08.10.2020 9](#_Toc55747507)

[4.2 4.2 Sprint Retrospektive am 08.10.2020 10](#_Toc55747508)

[Abbildung 1: Definition of Done 2](file:///C:\Users\Miriam\Desktop\GitHub\SoftwareEngineering\Software_Engineering_Dokumentation.docx#_Toc55467578)

[Abbildung 2: Sailboat-Retrospektive 6](file:///C:\Users\Miriam\Desktop\GitHub\SoftwareEngineering\Software_Engineering_Dokumentation.docx#_Toc55467579)

[Abbildung 3: Helden-Retrospektive 7](file:///C:\Users\Miriam\Desktop\GitHub\SoftwareEngineering\Software_Engineering_Dokumentation.docx#_Toc55467580)

Abkürzungen:

DoD: Definition of Done

Kartensatz 1: Besteht aus Karten mit der Gewichtung 1, 2, 3, 5, 8, 13, 25

Kartensatz 2: Besteht aus Karten mit der Gewichtung 1, 2, 4, 8, 16, 32, 64

# 1 Allgemein

## Projektbeschreibung

Realisierung einer Computeranwendung zum Pokerspielen, in der Spielvariante Texas Holdem. Die Anwendung soll in der Programmiersprache C++ implementiert werden. Die Anwendung soll aus einem einfachen Spielermodus bestehen. Mögliche Erweiterungen dazu sind verschiedene Bots, eine graphische Oberfläche oder weitere Spielvarianten von Poker.

## Vorgehensmodell

Als Team haben wir uns einstimmig für ein agiles Vorgehensmodell entschieden. Vor allem aus dem Grund, dass bei unserer initialen Projektbesprechung noch keine vollumfängliche Beschreibung des Projekts möglich war. Aus dem Kreis der agilen Vorgehensmodelle haben wir uns dann konkret für den Scrum-Ansatz entschieden. Für diese Entscheidung sprachen vor allem, dass Scrum in vielen Unternehmen als Ansatz für die Softwareentwicklung verwendet wird. Wir haben uns dadurch erhofft erste Erfahrungen mit Scrum zu machen. Außerdem bietet Scrum viele Vorteile durch seine gute Struktur und Einfachheit.

Um diese Projektarbeit und die damit verbundenen Scrumprozesse in unseren Studienalltag einbringen zu können, haben wir uns in unserem initialen Projektmeeting dem Process Tailoring bedient, um das Vorgehensmodell auf unsere Bedürfnisse anzupassen.

Festgelegt wurde dabei, dass die DailyScrum Meetings nur zwei Mal die Woche stattfinden sollen. Grund dafür war, dass das Projekt nicht 100% unserer Zeit beansprucht, und somit auch nicht regelmäßig bzw. täglich am Projekt gearbeitet wird.

Die Sprintlänge wurde auf zwei Wochen initial festgelegt, um auch hier einige Zyklen bis zum Projektende zu durchlaufen, auch wenn dadurch die Größe des Sprint Backlogs kleiner ist als eventuell sinnvoll.

Die Scrum-Rollenverteilung sieht folgende Rollen vor, die vom Team folgendermaßen besetzt wurden:

Scrum Master: Noah Hauke

Product Owner: Felix Schmitt  
Die restlichen Projektbeteiligten bilden das Team. In dieses Team sind auch Scrum Master und Product Owner integriert.

## Organisation

Organisiert hat sich das Team über Microsoft Teams. Die Wahl für die erste Besprechung fiel auf Teams, da bei einem Gruppenmitglied Discord nicht funktionierte. Alle weiteren DailyScrums, Sprint Plannings und Sprint Retrospektiven wurden aufgrund dessen auch in Teams abgehalten. Für die Programmierung wurde ein Git**-**Repository erstellt. Aufgrund keiner wirklichen Kenntnis von Alternativen und daein Großteil des Teams schon Erfahrungen in GitHub hatte. Dadurch, dass Teams viele weitere Microsoft Produkte wie zum Beispiel Microsoft Planner integriert hat, entschieden sich das Team dazu, sich nicht in eine Software wie Jira einzuarbeiten, sondern in das Team ein Planner Board zu integrieren. Ein Screenshot des Planner Bords ist in Abbildung 1 zu sehen. Das bot auch den Vorteil, dass man so alles in einer Software halten konnte. Im Laufe des Projektes wurden die Registerkarten immer wieder erweitert, zum einen durch das hinzufügen einer Definition of Done oder einer Registerkarte mit Bugs. Diese erwies sich als nützlich, da dort dann jeder einen Bug und eine potenziell verantwortliche Person eintragen konnte. Dies hätte auch über Issues in GitHub geregelt werden können, jedoch war dieses Vorgehen auch für die Mitglieder mit geringerer GitHub Erfahrung einfacher.

**Ein Bild, das Screenshot, Computer, Monitor, computer enthält.

Automatisch generierte Beschreibung**

Abbildung 1: PlannerBoard in Teams

## Programmierstandards

Zur Vereinfachung des Prozesses und zur einheitlichen Darstellung der Ergebnisse hat sich das Team auf verschiedene Standards geeinigt.

Die Dateienablage und der Dateienaustausch findet über ein eigens für das Projekt angelegtes Git-Repository statt. Dadurch wird eine konsistente Datenhaltung gewährleistet und ein Datenaustausch kann schnell und einfach vollzogen werden.

Programmiert wird in der Sprache C++, da diese einem Großteil der Teammitglieder bekannt ist.

Als Programmierumgebung wurde Visual Studio ausgewählt, welche bereits bei allen Teammitgliedern installiert war.

Ebenfalls hat sich das Team auf verschiedene Programmierstandards geeinigt:

* Allen Variablen werden englische Namen vergeben
* Alle Variablen haben sprechende Namen
* Ausgabe wird in Englisch verfasst

## Qualitätsmanagement

Im Rahmen der Sprint Retrospektive am 08.10.2020 wurde die Dringlichkeit einer sinnvollen Code-Review vor dem Merge mit dem Masterbranch deutlich. Daher entschied sich das Team dazu, eine Definition of Done zu schreiben, um die Codequalität zu verbessern. Durch die Definition of Done soll vor allem das 4-Augen-Prinzip erfüllt werden. Dabei wird der entwickelte Code von mindestens einem Teammitglied vor der Veröffentlichung bestätigt.

Zudem entschied sich das Team dazu Tests zu schreiben, da das manuelle Testen der Regeln einen erheblichen Zeitaufwand mit sich bringen würde und dieser von keinem Teammitglied geleistet werden kann. Der Zeitaufwand resultiert aus der zufälligen Kartenverteilung und dadurch, dass das Spiel gespielt werden muss und jede Kartenkombination mal aufgetreten sein muss, um von einer Testung aller Funktionen sprechen zu können.

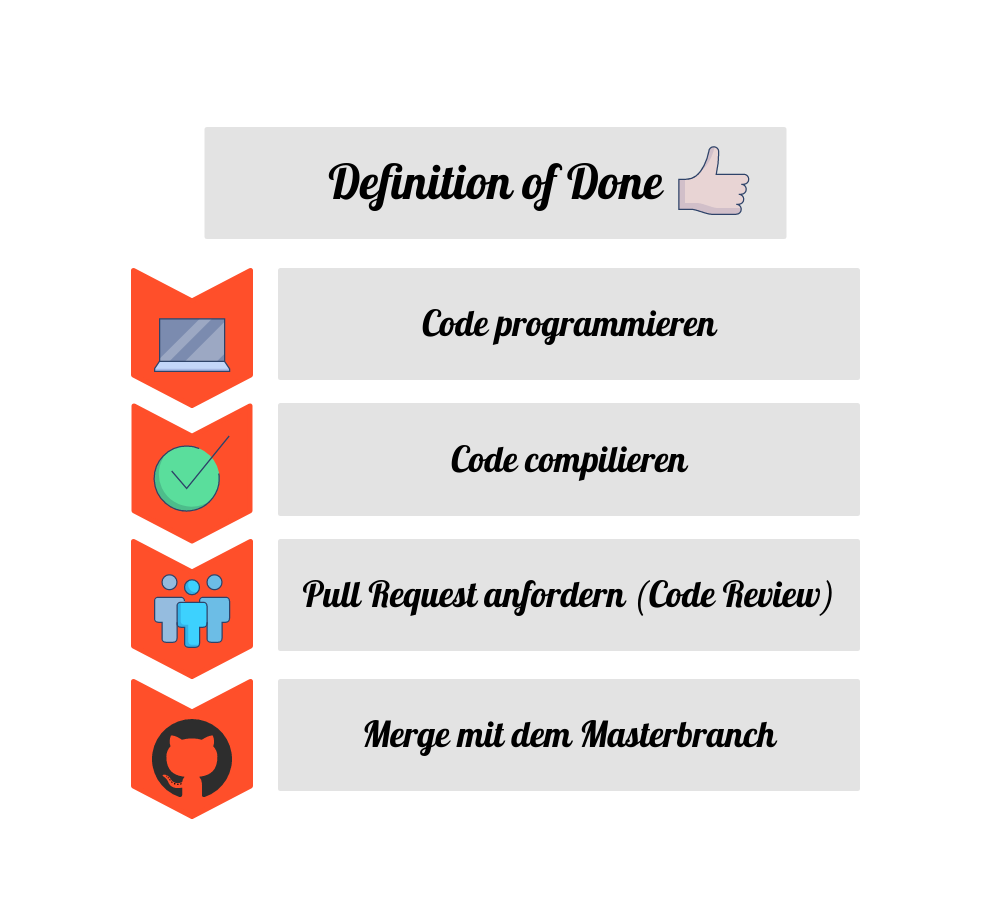


Abbildung 2: Definition of Done

## Gleichberechtigung

Aufgrund der umständlichen und schwer verständlichen Schreibweise zur Gleichberechtigung zwischen Mann und Frau wird in dieser Dokumentation auf die weibliche Form verzichtet. Dies soll aber in keinster Weise diskriminierend gemeint sein.

## Epics und Product Backlog

Es wurde für das Projekt ein Epic festgelegt:

Epic 1: Anwendung zum Pokerspielen am Computer

Außerdem wurde ein umfassender Product Backlog angelegt, der mit jedem Sprint erweitert wurde.

# 2. Sprint-Planning

## 2.1 Sprint Planning am 24.09.2020

Zu Beginn des Meetings wurden vom Product Owner folgende Priorisierung der User Stories vorgenommen. Daraus wurden dann vom Team verschiedene Tasks definiert (siehe Sprint Backlog 1).

Zur Einschätzung der Tasks wurde auf die Methode des Planning Pokers zurückgegriffen. Um dieses durchzuführen, wurde das Online-Tool <https://www.scrumpoker.online> verwendet und der darin implementierte Kartensatz 1. Dieses Tool erwies sich als sehr nützlich. Damit konnte die Einschätzung der Tasks sehr schnell und effizient vorgenommen werden. Über Uneinigkeiten wurde diskutiert, um auf ein gemeinsames Ergebnis zu kommen.

Der letzte Punkt im Meeting war die Verteilung der einzelnen Tasks auf die Mitglieder des Teams. Dabei wurde sowohl auf eine gleichmäßige Verteilung des Arbeitsvolumens geachtet, als auch auf die einzelnen Vorkenntnisse der Teammitglieder.

## 2.2 Sprint Planning am 08.10.2020

Erster Punkt des Meetings war die Priorisierung der User Stories für den kommenden Sprint durch den Product Owner.

Aus den priorisierten Stories konnte das Team dann verschiedene Tasks für den Sprint ableiten (siehe Sprint Backlog 2).

Zur Schätzung der Tasks wurde erneut auf die Methode des Planning Pokers zurückgegriffen, allerdings diesmal mit dem Kartensatz 2. Bei Unstimmigkeiten über die Gewichtung eines Tasks, wurde offen im Team über die Unterschiede diskutiert um eine Einigung zu erzielen.

Zuletzt wurden dann noch die Tasks an die einzelnen Teammitglieder verteilt. Dabei konnte sich jedes Teammitglied abhängig von seinem Können, einen oder mehrere Tasks auswählen.

## 2.3 Sprint Planning am 22.10.2020

Zu Beginn des Meetings musste die Rolle des Product Owners neu vergeben werden, da der vorherige Product Owner exmatrikuliert wurde. Als neuer Product Owner wurde Sophia Schmoll ernannt. Danach wurden neue User Stories in den Product Backlog eingebracht und durch den Product Owner priorisiert. Daraus wurden vom Team verschiedene Tasks abgeleitet (siehe Sprint Backlog 3), die dann mit Planning Poker bewertet wurden. Erneut wurde der Kartensatz 2 gewählt. Dieser hat sich im letzten Sprint als gut und genau erwiesen. In den letzten Sprint Plannings hat es sich bewährt, bei Unstimmigkeiten nochmals den genauen Umfang des Tasks zu besprechen, um dadurch eine Einigung zu erzielen.

Abschließend wurden die Tasks noch auf die einzelnen Teammitglieder verteilt. In dieser späten Phase des Projektes wurde vor allem bei der Verteilung darauf geachtet, dass die neuen Tasks an die Teammitglieder verteilt wurden, die in den vergangenen Sprints schon in diesem Bereich gearbeitet haben.

## 2.4 Sprint Planning am 05.11.2020

Zu Beginn des Meetings wurden die User Stories aus dem Product Backlog vom Product Owner neu priorisiert. Aus den wenigen verbleibenden UserStories wurden dann von den Teammitgliedern Tasks abgeleitet. Aufgrund der geringen Anzahl an Tasks hat sich das Team dazu entschlossen, den Sprint auf eine Woche zu begrenzen. Ebenfalls wurde beschlossen die Tasks auch nicht zu schätzen. Da die Tasks im größten Teil auf alle Teammitglieder aufgeteilt sind, und jedem Mitglied nur einzelne kleine Aufgabe zugewiesen wurden, die mit dem Programmcode des jeweiligen Teammitglied zusammenhängen. Das Ergebnis des Sprint Plannings wurde im Sprint Backlog 4 zusammengefasst.

# 3. Sprint Review

## 3.1 Sprint Review am 08.10.2020

Als erster Punkt des Meetings wurde der im letzten Sprint geschriebene Code vorgestellt. Außerdem wurde jedes erstellte Codeelement zusätzlich nach dem 4-Augen-Prinzip allen Teammitgliedern vorgestellt.

Dabei wurden einige Elemente gefunden, die im nächsten Sprint noch bearbeitet werden müssen.

Diese sind in der Sprint Backlog Review Liste 1 dargestellt.

## 3.2 Sprint Review am 22.10.2020

Zu Beginn des Meetings konnte diesmal ein lauffähiges Programm vorgestellt werden. Daraus abgeleitet wurden verschiedene Verbesserungen, die noch am Programm gemacht werden sollten.

Diese wurden in der Sprint Backlog Review Liste 2 zusammengefasst.

## 3.3 Sprint Review am 05.10.2020

Eingeleitet wurde das Meeting mit einer Demonstration der lauffähigen Software. Dabei konnten noch einige Punkte festgehalten werden, die im kommenden Sprint verbessert werden sollten. Diese wurden in festgehalten in der Sprint Backlog Review Liste 3.

# 4. Sprint Retrospektive

## 4.1 Sprint Retrospektive am 08.10.2020

Begonnen wurde das Meeting mit einer Sailboat-Retrospektive, in die jedes Teammitglied seine Gedanken zum vergangenen Sprint eintragen konnte.

Aus dieser Grafik wurde dann eine Liste an Verbesserungen für den nächsten Sprint abgeleitet, diese wurden dann in der Verbesserungsliste 1 festgehalten.

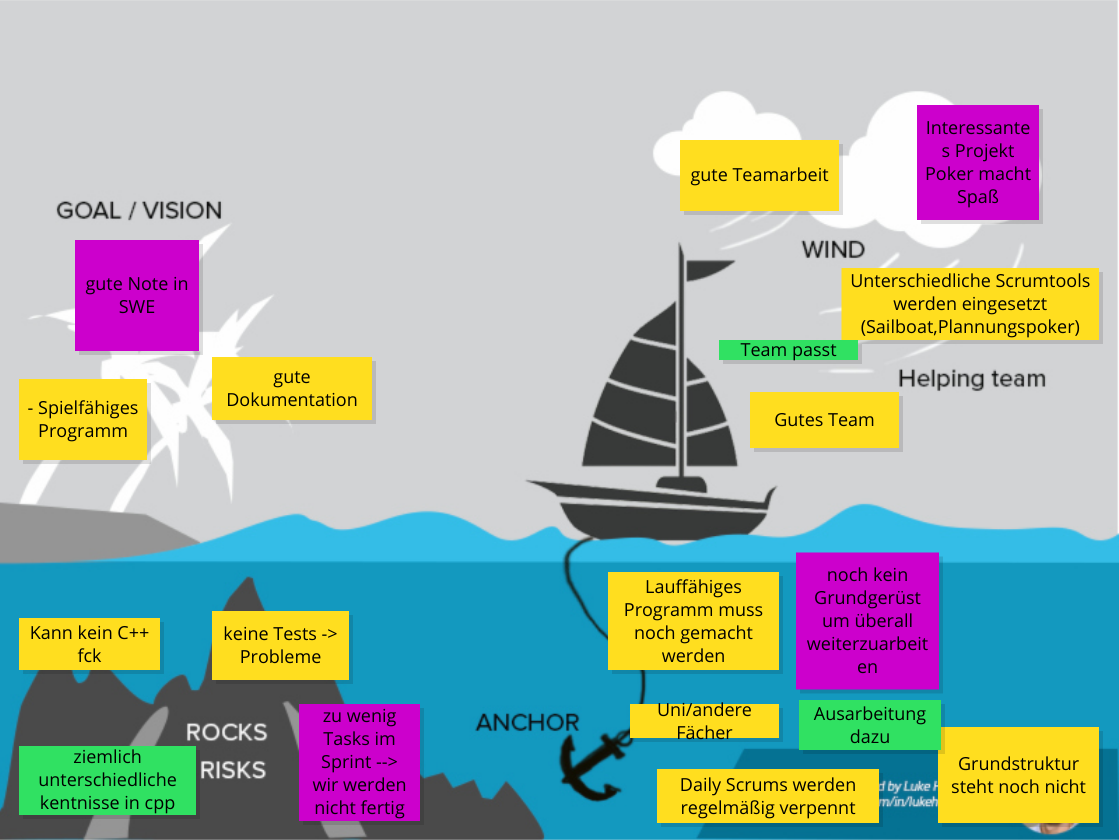


Abbildung 3: Sailboat-Retrospektive

## 4.2 Sprint Retrospektive am 08.10.2020

Zur Durchführung dieser Review wurde eine Hero-Retrospektive ausgewählt, in der die verschiedenen Schlagworte zum abgelaufenen Sprint gesammelt werden konnte.

Daraus abgeleitet wurde dann eine Verbesserungsliste 2, die Punkte enthält, welche dann in den nächsten Sprints oder auch in zukünftigen Projekten umgesetzt werden sollten.

Helden-Retrospektive


Automatisch generierte Beschreibung

Abbildung 4: Helden-Retrospektive

## 4.3 Sprint Retrospektive am 05.11.2020

Die dritte Sprint Retrospektive wurde im Stil einer Amazon Produktbewertung durchgeführt. Dazu war jedes Teammitglied aufgefordert, eine Rezension zum Sprint zu schreiben und diesen mit einer Menge aus 5 Sternen zu bewerten. Die abgeleiteten Ergebnisse aus der Retrospektive wurden in der Verbesserungsliste 3 festgehalten.

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Abbildung 5: Amazon Produktbewertung Retrospektive

5 Ergebnis:

Im Folgenden wird das fertige Produkt vorgestellt. Fertig ist dabei auf unsere User Stories von Beginn des Projektes zu beziehen. Es ist natürlich noch einiger Spielraum für Verbesserungen gegeben. Zum Beispiel könnte noch ein schlauer Bot implementiert werden oder ein Graphical User Interface.

Ein Bild, das Text enthält.

Automatisch generierte BeschreibungZu Spielbeginn hat der Nutzer die Möglichkeit auszuwählen, ob er die Regeln angezeigt haben möchte (siehe User Story xx). Anschließend kann die Zahl der Spieler, das Tischmaximum und die Art der Spieler festgelegt werden. Die Art der Spieler bezieht sich dabei auf einen menschlichen Spieler oder einen Bot. Für menschliche Spieler muss ein Name und ein Chipstack eingegeben werden. Für einen Bot kann optional ein Name eingegeben werden. Wie das in der Anwendung realisiert wurde, ist in Abbildung 6 zu sehen.

Abbildung 6 Spielbeginn

Anschließend wird das eigentliche Spiel gestartet, die Bots spielen dabei automatisch und jeder menschliche Spieler hat die Möglichkeit über die Eingabe in die Konsole seine Aktionen auszuwählen. Das ist in Abbildung xx zu sehen.

Spielende ....

6. Zusammenfassung

Zusammenfassend lässt sich sagen, dass das Projekt eine gute Plattform bot, sich mit verschiedenen Punkten der Softwareentwicklung auseinanderzusetzen.

Zum einen konnten Erfahrungen in dem Scrum-Modell gesammelt werden. Welches dann gegen Projektende immer gewohnter für die Teammitglieder wurde und somit auch immer besser funktioniert hat.

Desweitern musste zu Beginn eine Plattform gefunden werden, um zusammen an einer Softwareanwendung zu arbeiten. Durch die Wahl von GitHub, arbeiteten manche Teammitglieder erstmalig mit GitHub und andere konnten ihre Erfahrungen vertiefen.

Zudem konnten wir unsere Programmierkentnisse erweitern. Das Projekt bot für den Teil der Gruppe, welcher im zweiten Semester Java gelernt hatte, die Möglichkeit sich mit C++ zu beschäftigen. Der Teil der Gruppe, welcher im zweiten Semester die Vorlesung C++ belegt hatte, konnte durch das Projekt die Vorhandenen Kenntnisse und Fertigkeiten erweitern und weitergeben.