Software Engineering Projekt

Entwicklung einer Anwendung zum Poker spielen am Computer

Informationstechnik

An der Dualen Hochschule Stuttgart

Eingereicht von: 3819525

5247874

6499003

7750470

Studienjahrgang, Kurs: 2019, 19IN

**Bearbeitungszeitraum:** 10.09.2020 bis 19.11.2020

**Inhalt:**

[1 Einleitung 1](#_Toc56251170)

[1.1 Projektbeschreibung 1](#_Toc56251171)

[1.2 Vorgehensmodell 1](#_Toc56251172)

[1.3 Organisation 2](#_Toc56251173)

[1.4 Programmierstandards 3](#_Toc56251174)

[1.5 Qualitätsmanagement 4](#_Toc56251175)

[1.6 Gleichberechtigung 4](#_Toc56251176)

[1.7 Epics und Product Backlog 5](#_Toc56251177)

[2 Sprint 1 7](#_Toc56251178)

[2.1 Sprint Planning am 24.09.2020 7](#_Toc56251179)

[2.2 Sprint Review am 08.10.2020 8](#_Toc56251180)

[2.3 Sprint Retrospektive am 08.10.2020 8](#_Toc56251181)

[3 Sprint 2 10](#_Toc56251182)

[3.1 Sprint Planning am 08.10.2020 10](#_Toc56251183)

[3.2 Sprint Review am 22.10.2020 11](#_Toc56251184)

[3.3 Sprint Retrospektive am 22.10.2020 11](#_Toc56251185)

[4 Sprint 3 13](#_Toc56251186)

[4.1 Sprint Planning am 22.10.2020 13](#_Toc56251187)

[4.2 Sprint Review am 05.11.2020 15](#_Toc56251188)

[4.3 Sprint Retrospektive am 05.11.2020 15](#_Toc56251189)

[5 UML Diagramm 17](#_Toc56251190)

[6 Ergebnis 18](#_Toc56251191)

[7 Zusammenfassung 22](#_Toc56251192)

**Abbildungen:**

[Abbildung 1 PlannerBoard in Teams 2](#_Toc56249425)

[Abbildung 2 BugFixing Liste 3](#_Toc56249426)

[Abbildung 3 Definition of Done 4](file:///C:\Users\sophia.schmoll\github\SoftwareEngineering\Doku_v2.docx#_Toc56249427)

[Abbildung 4 Sailboat Retrospektive 8](file:///C:\Users\sophia.schmoll\github\SoftwareEngineering\Doku_v2.docx#_Toc56249428)

[Abbildung 5 Helden Retrospektive 11](file:///C:\Users\sophia.schmoll\github\SoftwareEngineering\Doku_v2.docx#_Toc56249429)

[Abbildung 6 Amazon Produktbewertung Retrospektive 15](file:///C:\Users\sophia.schmoll\github\SoftwareEngineering\Doku_v2.docx#_Toc56249430)

[Abbildung 7 UML Diagramm 16](#_Toc56249431)

[Abbildung 8 Spielbeginn 17](#_Toc56249432)

[Abbildung 9 Eingabe der Sachen ToDo 18](#_Toc56249433)

[Abbildung 10 GameController 19](#_Toc56249434)

[Abbildung 11 Spiel und so 19](#_Toc56249435)

[Abbildung 12 Showdown 19](#_Toc56249436)

[Abbildung 13 winners 20](#_Toc56249437)

**Tabellen:**

[Tabelle 1 Sprint Backlog 1 11](#_Toc56248844)

[Tabelle 2 Sprint Backlog Review Liste 1 11](#_Toc56248845)

[Tabelle 3 Verbesserungsliste 1 12](#_Toc56248846)

[Tabelle 4 Sprint Backlog 2 14](#_Toc56248847)

[Tabelle 5 Sprint Backlog Review Liste 2 14](#_Toc56248848)

[Tabelle 6 Verbesserungsliste 2 15](#_Toc56248849)

[Tabelle 7 Sprint Backlog 3 17](#_Toc56248850)

[Tabelle 8 Sprint Backlog Liste 3 18](#_Toc56248851)

[Tabelle 9 Verbesserungsliste 3 18](#_Toc56248852)

**Abkürzungen:**

DoD: Definition of Done

GUI: Graphical User Interface

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

UML: Unified Modeling Language

# Einleitung

## 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.

**Graphical user interface, application

Description automatically generated**

Abbildung 1 PlannerBoard in Teams

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. Zu sehen ist diese Registerkarte in Abbildung 2. Dies hätte auch über Issues in GitHub geregelt werden können, jedoch war dieses Vorgehen auch für die Mitglieder mit geringerer GitHub Erfahrung einfacher.

Graphical user interface, application

Description automatically generated

Abbildung 2 BugFixing Liste

## 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 3 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:

|  |  |
| --- | --- |
| Nummer | User Story |
| 1 | Als Spieler möchte ich bequem von Zuhause aus Poker spielen können, damit ich mich nicht mit Corona anstecke. |
| 2 | Als Spieler möchte ich eine Erklärung der Regeln vor dem Spiel bekommen können, damit ich nicht im Internet suchen muss und damit die Regeln klar definiert sind. |
| 3 | Als Spieler möchte ich die Eingabemöglichkeiten über die Konsole auswählen können, damit das Spiel weiß was ich möchte. |
| 4 | Als Entwickler möchte ich das Guthaben der Spieler in Form von Chips darstellen, um das Spiel so realitätsnah wie möglich zu gestalten. |
| 5 | Als Spieler möchte ich sowohl gegen reale Player als auch Bots spielen können, um das Spiel abwechslungsreich zu gestalten und um auch mit Freunden/alleine spielen zu können. |
| 6 | Als Entwickler möchte ich die Ausgabe in englischer Sprache gestalten, um eine sprachliche Einheit darzustellen und die Ausgabe ordentlich zu gestalten. |
| 7 | Als Spieler möchte ich die Höhe der Blinds und das Tischmaximum festlegen können, um das Spiel an meine Bedürfnisse anpassen zu können. |
| 8 | Als Spieler möchte ich die Anzahl der Mitspieler festlegen können, um das Spiel abwechslungsreich machen zu können. |
| 9 | Als Entwickler möchte ich das Startguthaben der Bots zufällig festlegen, um ein aufregendes Spiel zu haben. |
| 10 | Als Spieler möchte ich einen schlauen und dummen Bot zur Auswahl haben, um mein Spiel aufregend zu gestalten. |
| 11 | Als Entwickler möchte ich, dass die anderen Entwickler Code kommentieren, um ihn besser verstehen zu können. |
| 12 | Als Entwickler möchte das Menü und den Gamecontroller miteinander verknüpfen, um das Spiel spielbar zu machen |
| 13 | Als Spieler möchte ich das Startguthaben festlegen können, um länger spielen zu können und das Spiel interessanter zu gestalten. |

Sprint 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.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Sprint Backlog 1 | |  |  |  |  |  |
|  |  |  |  |  |  |  |
| Priorisierung | User Story |  |  |  |  |  |
|  |  |  |  |  |  |  |
| 1 | Als Spieler möchte ich eine Erklärung der Regeln vor dem Spiel | | | | | |
|  | bekommen können, damit ich nicht im Internet suchen muss und damit die Regeln klar definiert sind. | | | | |  |
|  |  | | | | |  |
| 2 | Als Spieler möchte ich sowohl gegen reale Player | | | | |  |
|  | als auch Bots spielen können, um das Spiel abwechslungsreich | | | | | |
|  | zu gestalten und um auch mit Freunden/alleine spielen zu können. | | | | | |
|  |  |  |  |  |  |  |
| 3 | Als Spieler möchte ich die Anzahl der Mitspieler festlegen können, | | | | | |
|  | um das Spiel abwechslungsreich machen zu können. | | | | |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
| Task |  |  |  | Aufwand |  | Verantwortlich |
|  |  |  |  |  |  |  |
| Grundlegende Architektur | | |  | 3 |  | Luca |
|  |  |  |  |  |  |  |
| Spielregeln definieren | |  |  | 8 |  | Luca |
|  |  |  |  |  |  |  |
| Spielregeln grob programmieren | | |  | 8 |  | Felix |
|  |  |  |  |  |  |  |
| Zufallsgenerator | |  |  | 3 |  | Miriam |
|  |  |  |  |  |  |  |
| UML-Diagramm erstellen | |  |  | 13 |  | Noah |
|  |  |  |  |  |  |  |
| Menü aufsetzen | |  |  | 5 |  | Sophia |
|  |  |  |  |  |  |  |
| Spielregeln ausgeben | |  |  | 3 |  | Miriam |
|  |  |  |  |  |  |  |
| Klassen anlegen | |  |  | 5 |  | Miriam |
|  | |  |  |  |  |  |

Tabelle 1 Sprint Backlog 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.

|  |  |  |  |
| --- | --- | --- | --- |
| Sprint Backlog Review Liste 1 | |  |  |
| 1 | UML-Diagramm: Änderungen an der Struktur vornehmen | | |
| 2 | Menue: Zahlendreher auflösen |  |  |
| 3 | Rules: Datentypen verändern von String zu Enum | | |
|  |  | | |

Tabelle 2 Sprint Backlog Review Liste 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.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Verbesserungsliste 1 | |  |  |  |
| 1 | Richtiges spielfähiges Programm schreiben bis zum nächsten Sprint | |  |  |
| 2 | DailyScrum nicht mehr verpassen |  |  |  |
| 3 | Tests zur Sprintplanung hinzufügen | Software muss getestet werden | | |
| 4 | C++-Kenntnisse auf einen Stand bringen | (z.B. durch Pair-Programming) | | |
| 5 | Projektplanung in Software einbetten | (z.B. Planner von Teams) | | |
| 6 | Definition of Done erstellen | für bessere Codequalität | | |

Tabelle 3 Verbesserungsliste 1

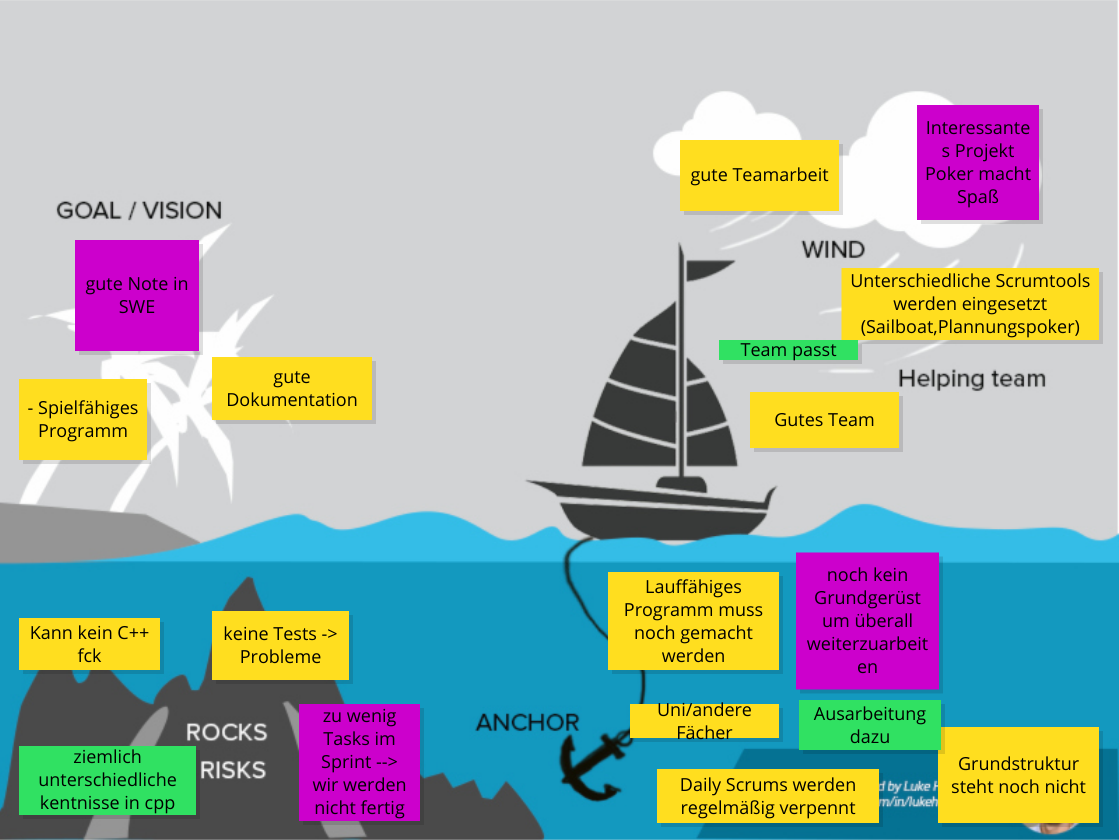


Abbildung 4 Sailboat Retrospektive

Sprint 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.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Sprint Backlog 2 | |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
| Priorisierung | User Story |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
| 1 | Als Spieler möchte ich bequem von Zuhause aus Poker spielen können, | | | | | | |
|  | damit ich mich nicht mit Corona anstecke. | | | |  |  |  |
|  |  |  |  |  |  |  |  |
| 2 | Als Spieler möchte ich sowohl gegen reale Player | | | | |  |  |
|  | als auch Bots spielen können,um das Spiel abwechslungsreich | | | | | |  |
|  | zu gestalten und um auch mit Freunden/alleine spielen zu können. | | | | | |  |
|  |  |  |  |  |  |  |  |
| 3 | Als Spieler möchte ich die Höhe der Blinds und das Tischmaximum | | | | | |  |
|  | festlegen können, um das Spiel an meine Bedürfnisse anpassen zu können. | | | | | | |
|  |  |  |  |  |  |  |  |
| 4 | Als Spieler möchte ich das Startguthaben festlegen können, | | | | | |  |
|  | um länger Spielen zu können und das Spiel interessanter zu gestalten. | | | | | | |
|  |  | | | | | | |
|  |  |  |  |  |  |  |  |
| Task |  |  |  | Aufwand |  | Verantwortlich | |
|  |  |  |  |  |  |  |  |
| Gamecontroller schreiben | | |  | 32 |  | Luca |  |
|  | Dealer-Klasse | |  |  |  |  |  |
|  | Ausgabe-Klasse | |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
| Human Player erstellen | |  |  | 16 |  | Noah |  |
|  |  |  |  |  |  |  |  |
| Rules schreiben | |  |  | 32 |  | Felix |  |
|  | Funktion um Hand zu bewerten | | |  |  |  |  |
|  | Funktion um Hände zu vergleichen | | |  |  |  |  |
|  |  |  |  |  |  |  |  |
| Menü |  |  |  | 8 |  | Sophia |  |
|  | Tischmaximum | |  |  |  |  |  |
|  | Startguthaben | |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
| Tests schreiben | |  |  | 16 |  | Miriam |  |
|  |  |  |  |  |  |  |  |
| UML-Diagramm | |  |  |  |  | Team |  |
|  |  |  |  |  |  |  |  |
| Dokumentation | |  |  |  |  | Miriam + Sophia | |
|  | |  |  |  |  |  | |

Tabelle 4 Sprint Backlog 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.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Sprint Backlog Review Liste 2 | | |  | |  | | |  |
| 1 | UML-Diagramm: An aktuelles Programm anpassen | | | | | | |  |
| 2 | Ausgabe: Sonderzeichenausgabe an ASCII-Zeichensatz anpassen | | | | | | | |
| 3 | Tests: ausbauen |  | |  | |  |  | |
| 4 | Rules: fertigstellen um sie einbinden zu könenn | | | | | | |  |
| 5 | Menue: Verbindung zum GameController herstellen | | | | | | |  |
|  |  | | | | | | |  |

Tabelle 5 Sprint Backlog Review Liste 2

## Sprint Retrospektive am 22.10.2020

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.

|  |  |  |  |
| --- | --- | --- | --- |
| Verbesserungsliste 2 | |  | |
| 1 | häufiger auf Pairprogramming zurückgreifen | v.a. zu Projektbeginn und bei komplexen Funktionen | |
| 2 | UML-Diagramm in neuen Projekten zu Beginn gemeinsam erstellen | einzelne Bausteine werden besser aufeinander abgestimt und verbessert Testbarkeit der Software | |
| 3 | im Sprint früher mit der Fertigstellung von Aufgaben beginnen | |  |
| 4 | Abhängigkeit von Aufgaben von anderen Teammitgliedern reduzieren | | |

Tabelle 6 Verbesserungsliste 2

Helden-Retrospektive


Automatisch generierte Beschreibung

Abbildung 5 Helden Retrospektive

Sprint 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.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Sprint Backlog 3 | |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
| Priorisierung | User Story |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
| 1 | Als Entwickler möchte das Menü und den Gamecontroller miteinander | | | | | | |
|  | verknüpfen, um das Spiel spielbar zu machen | | | | |  |  |
|  |  |  |  |  |  |  |  |
| 2 | Als Spieler möchte ich einen schlauen und dummen Bot zur Auswahl haben, | | | | | | |
|  | um mein Spiel aufregend zu gestalten | | | |  |  |  |
|  |  |  |  |  |  |  |  |
| 3 | Als Spieler möchte ich sowohl gegen reale Player als auch Bots | | | | | |  |
|  | spielen können, um das Spiel abwechslungsreich zu gestalten | | | | | |  |
|  | und um auch mit Freunden/alleine spielen zu können. | | | | |  |  |
|  |  |  |  |  |  |  |  |
| 4 | Als Entwickler möchte ich das Startguthaben der Bots zufällig festlegen, | | | | | | |
|  | um ein aufregendes Spiel zu haben | | | |  |  |  |
|  |  |  |  |  |  |  |  |
| 5 | Als Entwickler möchte ich, dass die anderen Entwickler Code kommentieren, | | | | | | |
|  | um ihn besser verstehen zu können | | | |  |  |  |
|  |  |  |  |  |  |  |  |
| 6 | Als Spieler möchte ich die Höhe der Blinds und das Tischmaximum | | | | | |  |
|  | festlegen können, um das Spiel an meine Bedürfnisse anpassen zu können. | | | | | | |
|  |  |  |  |  |  |  |  |
| 7 | Als Spieler möchte ich die Eingabemöglichkeiten über die Konsole | | | | | |  |
|  | auswählen können, damit das Spiel weiß was ich möchte. | | | | | |  |
|  |  |  |  |  |  |  |  |
| 8 | Als Entwickler möchte ich die Ausgabe in englischer Sprache gestalten, | | | | | | |
|  | um eine sprachliche Einheit darzustellen | | | |  |  |  |
|  | und die Ausgabe ordentlich zu gestalten. | | | |  |  |  |
|  |  |  |  |  |  |  |  |
| 9 | Als Spieler möchte ich bequem von Zuhause aus Poker spielen können | | | | | | |
|  | damit ich mich nicht mit Corona anstecke. | | | |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
| Task |  |  |  | Aufwand |  | Verantwortlich | |
|  |  |  |  |  |  |  |  |
| Gamecontroller schreiben | | |  | 8 |  | Luca |  |
|  | Gewinnausgabe | |  |  |  |  |  |
|  | Regeln abfragen | |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
| Dummen Bot erstellen | |  |  | 8 |  | Luca |  |
|  |  |  |  |  |  |  |  |
| Schlauen Bot erstellen | |  |  | 64 |  | Noah + Luca | |
|  |  |  |  |  |  |  |  |
| Rules schreiben | |  |  | 32 |  | Felix |  |
|  |  |  |  |  |  |  |  |
| Menü |  |  |  | 16 |  | Sophia |  |
|  | Verbindung zum Gamecontroller | | |  |  |  |  |
|  | dummen Bot erstellen | |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
| Tests schreiben | |  |  | 16 |  | Miriam |  |
|  |  |  |  |  |  |  |  |
| Kommentare schreiben | |  |  |  |  | Team |  |
|  |  |  |  |  |  |  |  |
| UML-Diagramm | |  |  |  |  | Team |  |
|  |  |  |  |  |  |  |  |
| Dokumentation | |  |  |  |  | Miriam + Sophia | |
|  | |  |  |  |  |  | |

Tabelle 7 Sprint Backlog 3

## Sprint Review am 05.11.2020

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.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Sprint Backlog Review Liste 3 | |  |  |  |  |  |
| 1 | Menue: Die Konsolenausgabe soll nach dem Anzeigen der Regeln wieder geleert werden | | | | | |
| 2 | Menue: Schreibfehler beheben |  |  |  |  |  |
| 3 | Menü: Defaultnamen für den Bot anlegen | |  |  |  |  |
| 4 | Output: Die Ausgabe der Währung soll auch in der Konsole ein $ darstellen | | | | |  |
| 5 | Human Player: Reihenfolge der Auswahl ändern | | |  |  |  |
| 6 | GameController: Bug in der Spielfunktion (-> Endlosschleife) | | | |  |  |
|  |  | | | |  |  |

Tabelle 8 Sprint Backlog Liste 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.

|  |  |
| --- | --- |
| Verbesserungsliste 3 | |
| 1 | Tasks zum Sprintende auch wirklich fertigstellen |
| 2 | Task am Anfang des Sprints beginnen, nicht erst zum Ende des Sprints hin |
| 3 | Offener Reden und dann direkt erkennen, dass ein Task zu kompliziert wird für den Sprint |
| 4 | Branches aktuell halten |
| 5 | Nur kommentierten und compilierten Code pushen |

Tabelle 9 Verbesserungsliste 3

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Abbildung 6 Amazon Produktbewertung Retrospektive

# UML Diagramm

Wie aus den Sprint Plannings hervorgeht, sollte des UML Diagramm eigentlich im Sprint 1 erstellt werden. In der Retrospektive wurde jedoch klar, das es zum damaligen Stand nicht möglich war das UML Diagramm fertig zu stellen. Es fehlten noch zu viele Verbindungen und einige Implementierungsschritte waren noch nicht klar.

Das Team einigte sich darauf, das UML Diagramm im Laufe des Projektes mit zu entwickeln. Dies wurde in Teilen auch umgesetzt. Nach einem Software-Problem wurde das UML Diagramm im letzten Sprint erneut begonnen und dann auch fertig gestellt. Das vollständige UML Diagramm ist in Abbildung 7 gezeigt. Zusätzlich ist es in GitHub abgelegt da somit eine Qualitativ hochwertigere Abgabe gewährleistet werden kann.

Diagram

Description automatically generated

Abbildung UML Diagramm

# Ergebnis

Im Folgenden wird das fertige Produkt vorgestellt. Fertig ist dabei auf unsere User Stories von Beginn des Projektes zu beziehen, welche bis zum Ende des letzten Sprints großteils umgesetzt werden konnten. Es ist natürlich noch einiger Spielraum für Verbesserungen gegeben, diese waren aber für die Dauer des Projektes nicht vorgesehen. Eine mögliche Erweiterung wäre ein Graphical User Interface. In User Story 3 wurde deshalb definiert, dass die Eingabe über die Konsole geregelt werden soll.

Zu Spielbeginn hat der Nutzer die Möglichkeit auszuwählen, ob er die Regeln angezeigt haben möchte (siehe User Story 2).

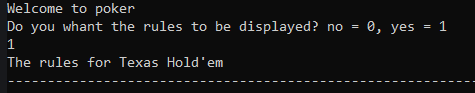


Abbildung 8 Spielbeginn

Anschließend kann die Zahl der Spieler, das Tischmaximum und die Art der Spieler festgelegt werden. Dies setzt die User Stories 5, 7 und 8 um. 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, dies wurde in User Story 13 beschrieben. Für einen Bot kann optional ein Name eingegeben werden. Das Startguthaben der Bots wird automatisch festgelegt (siehe User Story 9). Wie das in der Anwendung realisiert wurde, ist in Abbildung 9 zu sehen.

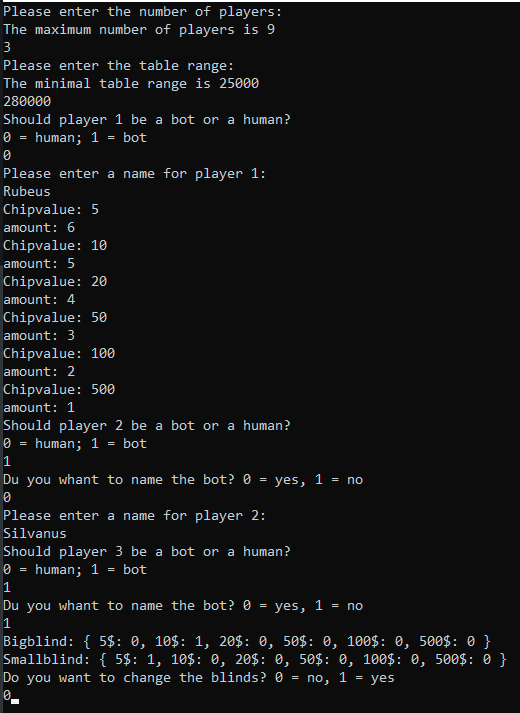


Abbildung 9 Spielinitialisierung

Anschließend wird das eigentliche Spiel gestartet, die Bots spielen dabei automatisch und jeder menschliche Spieler hat die Möglichkeit über die Eingabe in der Konsole seine Aktionen auszuwählen. Somit wurde auch die Verknüpfung von Menue und GameController wie in User Story 12 beschrieben wurde realisiert. Das ist in Abbildung 10 zu sehen.

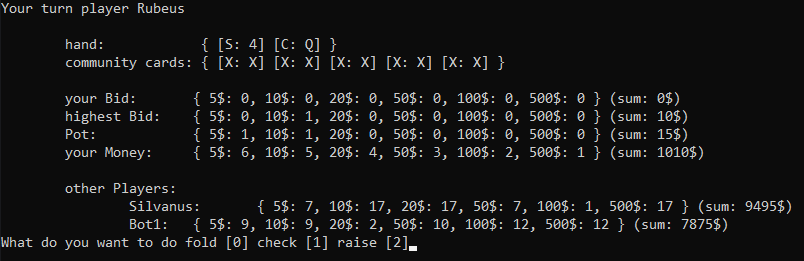


Abbildung 10 GameController

Die Spielmöglichkeiten für den Spieler sind ebenfalls in Abbildung 10 dargestellt. Um seinen Spielzug bestmöglich zu planen werden dem Spieler wie in Abbildung 10 dargestellt verschiedene Informationen über das Spiel aufgezeigt. In Abbildung 11 wurde beispielhaft erhöht.

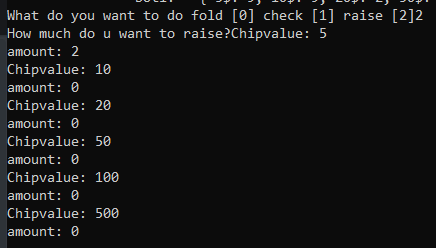


Abbildung 11 Erhöhung des Wetteinsatzes

Das Spielende ist durch einen Showdown (siehe Abbildung 12) dargestellt. Anschließend wird der Gewinner der Runde angezeigt (siehe Abbildung 13)

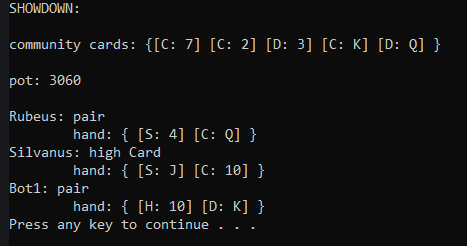


Abbildung 12 Showdown

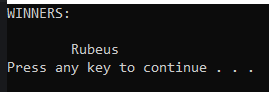


Abbildung 13 winners

Wie aus dem Abgleich des eben vorgestellten Ergebnisses mit der anfänglichen Beschreibung durch die User Stories hervorgeht, wurden alles User Stories umgesetzt. Bis auf User Story 10, da kein schlauer Bot implementiert wurde. Diese Aufgabe war im vorletzten Sprint gestellt und nach überlegen von Implementierungsmöglichkeiten fallengelassen, da den BugFixes eine höhere Priorität zugeordnet wurde und somit die Zeit fehlte diesen Bot noch zu implementieren. Einen schlauen Bot zu implementieren wäre genau wie ein GUI eine mögliche Erweiterung des Projektes.

# 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 Programmierkenntnisse 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.