

Pflichtenheft

Klassenratssitzungsplaner

Bearbeiter: Damian Gasser
Dennis Mairhofer
Lukas Kofler

Betreuer: Alexander Holzer

Datum: 27.04.2020

Dokumentenhistorie

Version	Datum	Bearbeiter	Kommentar	Status
1.0	20.01.2020	Alle	Initial commit	WIP
1.1	27.01.2020	Alle	Erweiterung des Hefts	WIP
1.2	09.02.2020	Lukas Kofler	Punkte 5.1 und 5.2 wurden hinzugefügt	WIP
1.3	10.02.2020	Dennis Mairhofer	Punkte 5.2 und 5.4 wurden hinzugefügt	WIP
2.0	19.02.2020	Damian Gasser	Klassen hinzugefügt	WIP
2.1	19.02.2020	Alle	Klassendiagramme wurden hinzugefügt	WIP
2.2	10.03.2020	Dennis Mairhofer	Einige Fehler ausgebessert	WIP
3	27.04.2020	Lukas Kofler	Final Release	Live

Inhaltsverzeichnis

Dokumentenhistorie	3
1 Anforderungen und Ziele	7
1.1 Überblick	7
1.2 Details Muss-Anforderungen	7
1.2.1 Klassenrat auswählen	7
1.2.2 Klassenrat verifizieren	7
1.3 Details Kann-Anforderungen	7
1.3.1 Button-Sounds	7
1.3.2 Speichern der Daten	7
1.4 Zielgruppe und Anforderungen an die Benutzerführung	7
2 Aktivitätsdiagramm	8
3 Use Cases	9
3.1 Akteure	9
3.1.1 Benutzer	9
3.1.2 System	9
3.2 Use Case Diagramm	9
3.3 Use Case „Eingabe Lehrer“	10
3.3.1 Kurze Beschreibung	10
3.3.2 Auslöser	10
3.3.3 Akteur(e)	10
3.3.4 Vorbereitung(en)	10
3.3.5 Schritte	10
3.3.6 Involvierte Klassen	10
3.4 Use Case „Eingabe anwesender Autoritäten“	10
3.4.1 Kurze Beschreibung	10
3.4.2 Auslöser	10
3.4.3 Akteur(e)	10
3.4.4 Vorbereitung(en)	10
3.4.5 Schritte	10
3.4.6 Involvierte Klassen	10
3.5 Use Case „Eingabe Klassenrat“	10
3.5.1 Kurze Beschreibung	10
3.5.2 Auslöser	11
3.5.3 Akteur(e)	11

3.5.4	Vorbereitung(en)	11
3.5.5	Schritte	11
3.5.6	Involvierte Klassen.....	11
3.6	Use Case „Einfügen Lehrer“	11
3.6.1	Kurze Beschreibung	11
3.6.2	Auslöser	11
3.6.3	Akteur(e).....	11
3.6.4	Vorbereitung(en)	11
3.6.5	Schritte	11
3.6.6	Involvierte Klassen.....	11
3.7	Use Case „Berechnung des Plans“	12
3.7.1	Kurze Beschreibung	12
3.7.2	Auslöser	12
3.7.3	Akteur(e).....	12
3.7.4	Vorbereitung(en)	12
3.7.5	Schritte	12
3.7.6	Involvierte Klassen.....	12
3.8	Use Case „Eingabe der Zeiten an denen die Sitzungen ausgeführt werden“	12
3.8.1	Kurze Beschreibung	12
3.8.2	Auslöser	12
3.8.3	Akteur(e).....	12
3.8.4	Vorbereitung(en)	12
3.8.5	Schritte	12
3.8.6	Involvierte Klassen.....	12
3.9	Use Case „Speichern der Pläne“	13
3.9.1	Kurze Beschreibung	13
3.9.2	Auslöser	13
3.9.3	Akteur(e).....	13
3.9.4	Vorbereitung(en)	13
3.9.5	Schritte	13
3.9.6	Involvierte Klassen.....	13
4	Klassenmodell.....	14
4.1	Klassendiagramm	14
5	Umsetzungs-Details.....	15
5.1	Zielpattform.....	15

5.2	Datenhandlung	15
5.3	Oberfläche	15
5.4	Systemarchitektur (grob)	15
5.5	Vorgehensmodell	15
5.5.1	Product Owner	15
5.5.2	Scrum-Master	15
5.5.3	Projektteam	16
5.6	Eigene Erfahrung	16
Abbildung 1: Activity diagram des aktuellen Ablaufes zur Planung der Klassenratssitzungen.....		8
Abbildung 2: Use Case Diagramm		9
Abbildung 3: Bild des Klassendiagramms.....		14
Abbildung 4: Bild des Scrum-Verlaufs		16

1 Anforderungen und Ziele

1.1 Überblick

Es soll ein System für die Planung von Klassenratssitzungen entwickelt werden. Nach einmaliger Eingabe der Lehrer und Klassen, werden diese in einer externen Datei gespeichert. Anschließend wird bei jedem Neustart diese Datei wieder eingelesen, wodurch eine Benutzerfreundliche Verwendung des Programms möglich ist. Mithilfe der vom Benutzer eingegebenen Daten wird bei Bedarf ein idealer Klassenratssitzungsplan erstellt.

Folgende Funktionen müssen durch das Programm abgedeckt werden:

- Dynamische Eingabe von Daten (anhand Tabellen und durch händische Eingabe)
- Berechnen eines effizienten Klassenratssitzungsplans, sodass die Lehrer möglichst wenig oft in die Schule kommen müssen
- Eingabe der Dauer für jeden Klassenrat individuell

1.2 Details Muss-Anforderungen

1.2.1 Klassenrat auswählen

Das System muss einen Klassenrat wählen, sodass die in ihm enthaltenen Lehrer nicht sehr oft in die Schule kommen müssen, sondern sehr effizient eingeteilt werden können. Dabei muss Tag, Zeit, Anwesenheit der Lehrer und der Autorität, sowie die bereits geplanten Sitzungen berücksichtigt werden.

1.2.2 Klassenrat verifizieren

Das System muss überprüfen, ob der Klassenrat bereits eingeteilt wurde, und ob alle Lehrer am gewünschten Tag Zeit haben

1.3 Details Kann-Anforderungen

1.3.1 Button-Sounds

Das Programm soll beim Drücken eines Knopfes oder einer Auswahl ein Sound ausgegeben werden.

1.3.2 Speichern der Daten

Das Programm soll das Ergebnis als CSV Datei exportieren können.

1.4 Zielgruppe und Anforderungen an die Benutzerführung

Potenzielle Kunden des Klassenratssitzungsplaners sind beispielsweise Lehrer, oder Sekretärinnen beziehungsweise Sekretäre des Sekretariats einer Schule.

2 Aktivitätsdiagramm

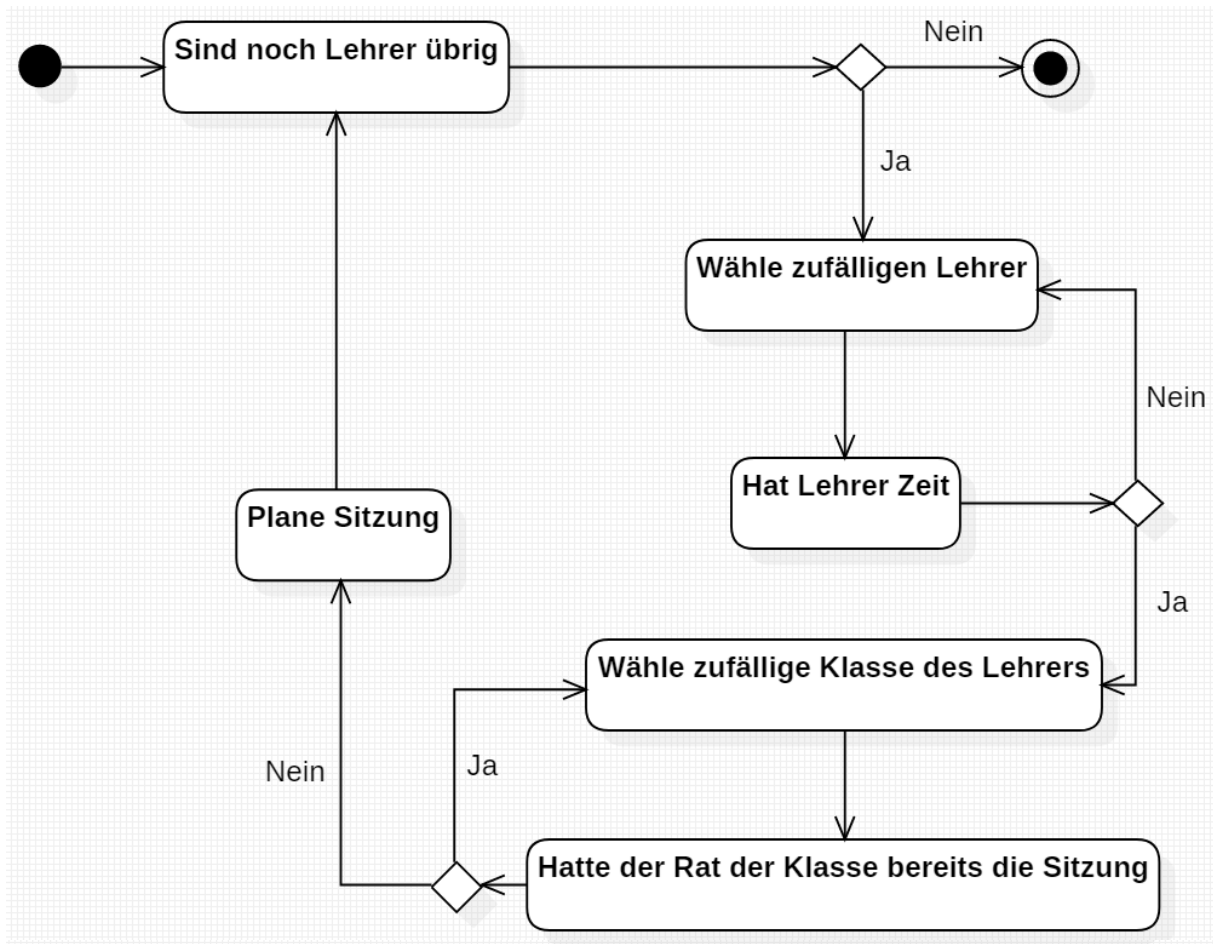


Abbildung 1: Activity diagram des aktuellen Ablaufes zur Planung der Klassenratssitzungen

3 Use Cases

3.1 Akteure

3.1.1 Benutzer

Benutzer, die dieses Programm erworben haben können, neue Lehrer hinzufügen, Klassenräte erstellen und diesen unbestimmt viele Lehrer hinzufügen. Zudem kann der Benutzer verschiedene Optionen auswählen, wie beispielsweise ob eine Führungskraft anwesend sein soll, oder ob es einen Tag geben soll, an dem keine Sitzung stattfinden soll.

3.1.2 System

Das System bekommt die Eingabedaten des Benutzers und muss mit diesen einen passenden Plan erarbeiten, sodass die Lehrer beziehungsweise alle beteiligten so wenig wie möglich zu den Sitzungen kommen müssen.

3.2 Use Case Diagramm

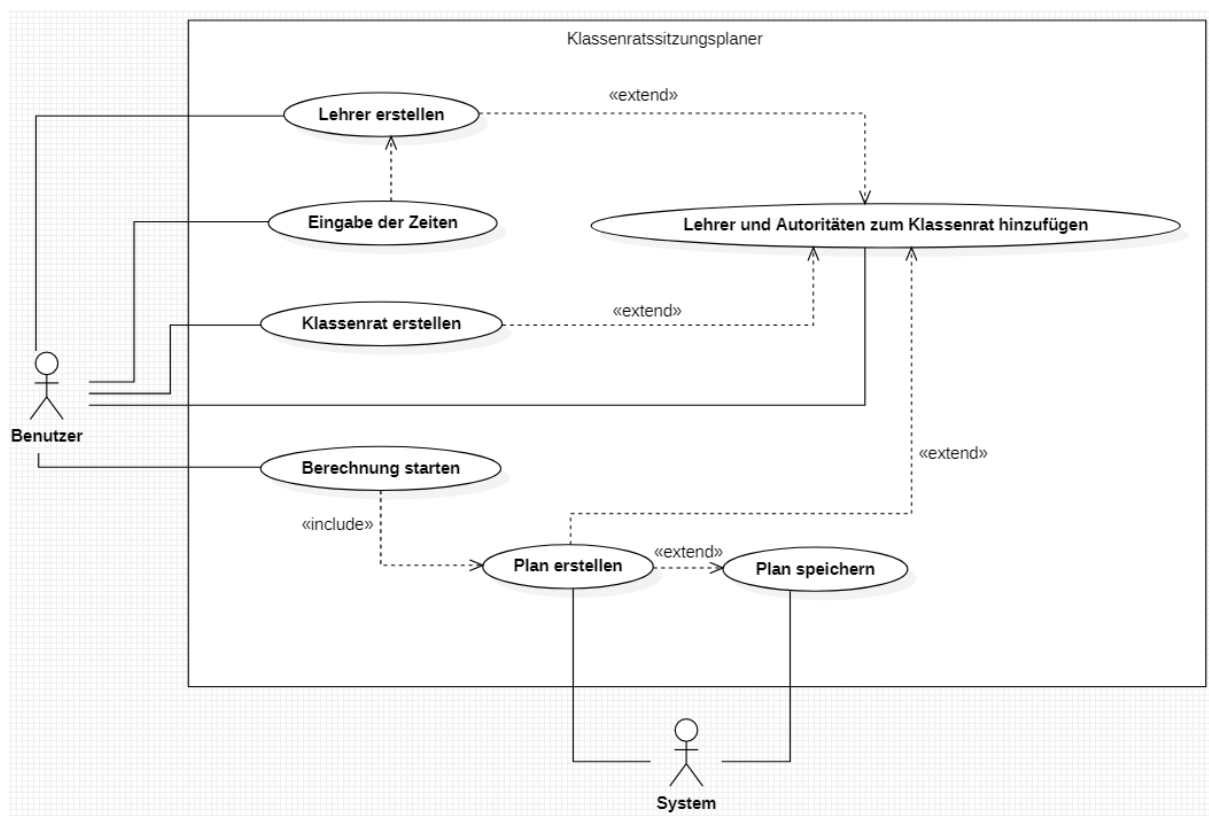


Abbildung 2: Use Case Diagramm

3.3 Use Case „Eingabe Lehrer“

3.3.1 Kurze Beschreibung

Der Benutzer kann einen Lehrer zur globalen Liste hinzufügen. Dabei kann er diesen Lehrer einen Namen geben und entscheiden, wann dieser frei hat beziehungsweise kann dieser keine Zeit hat.

3.3.2 Auslöser

Benutzer klickt auf das „+“ Symbol in der rechten Seite des Menüs.

3.3.3 Akteur(e)

Benutzer

3.3.4 Vorbereitung(en)

Programm ist gestartet

3.3.5 Schritte

1. Benutzer klickt auf das „+“ Symbol
2. Benutzer gibt Namen und den Stundenplan ein

3.3.6 Involvierte Klassen

Teacher/Authority, Person

3.4 Use Case „Eingabe anwesender Autoritäten“

3.4.1 Kurze Beschreibung

Der Benutzer kann angeben, ob entweder der/die Direktor/in, der/die Vize-Direktor/in, beide oder keiner der beiden anwesend sein muss. Dies geschieht über die Ansicht, in der der Benutzer auch die Lehrer zum Klassenrat hinzufügen kann. Der/Die Direktor/in und der/die Vize-Direktor/in sind bereits vordefinierte Lehrer, die nicht gelöscht werden können.

3.4.2 Auslöser

Benutzer fügt diese zum Klassenrat hinzu

3.4.3 Akteur(e)

Benutzer

3.4.4 Vorbereitung(en)

Programm ist gestartet

Benutzer befindet sich in den Eigenschaften eines Klassenrats

3.4.5 Schritte

1. Benutzer fährt über den (Vize-)Direktor
2. Benutzer hält die linke Maustaste gedrückt
3. Benutzer zieht das Objekt in den Bereich, in dem die Lehrer des Rates stehen
4. Benutzer lässt die Maustaste los

3.4.6 Involvierte Klassen

Authority, Draggable, Council

3.5 Use Case „Eingabe Klassenrat“

3.5.1 Kurze Beschreibung

Der Benutzer soll über eine Oberfläche in der Lage sein, einen Neuen Klassenrat zu erstellen. Dabei soll der Benutzer den Namen der Klasse eingeben und Lehrer dem Klassenrat hinzufügen können.

3.5.2 Auslöser

Benutzer doppelklickt einen Klassenrat oder drückt das „+“ Symbol im Bereich der Klassenräte

3.5.3 Akteur(e)

Benutzer

3.5.4 Vorbereitung(en)

Programm ist gestartet
Benutzer befindet sich im Hauptmenü

3.5.5 Schritte

1. Benutzer klickt zweimal schnell hintereinander mit der linken Maustaste auf einen bereits existierenden Klassenrat oder er drückt auf das „+“ Symbol, wodurch ein neuer Klassenrat erstellt wird.

3.5.6 Involvierte Klassen

Council

3.6 Use Case „Einfügen Lehrer“

3.6.1 Kurze Beschreibung

Der Benutzer soll über eine Oberfläche Lehrer zu einem bestimmten Klassenrat hinzufügen können. Dies geschieht über die Drag-and-Drop Methode und kann in den Eigenschaften des Klassenrats vollzogen werden.

3.6.2 Auslöser

Benutzer schiebt über ein Drag-and-Drop Verfahren einen Lehrer von der Auswahl in den Vorgegebenen Bereich

3.6.3 Akteur(e)

Benutzer

3.6.4 Vorbereitung(en)

Programm ist gestartet
Benutzer befindet sich in den Eigenschaften eines Klassenrats oder im Menü für das Erstellen eines neuen Klassenrats.

3.6.5 Schritte

1. Benutzer fährt mit der Maus über einen bereits Erstellten Lehrer
2. Benutzer hält die linke Maustaste gedrückt
3. Benutzer lässt die Maustaste gedrückt und schiebt das Objekt in den Vorgesehenen Bereich
4. Benutzer lässt die linke Maustaste nun los

3.6.6 Involvierte Klassen

Teacher, Draggable, Authority

3.7 Use Case „Berechnung des Plans“

3.7.1 Kurze Beschreibung

Das System soll mit den Einggegebenen Daten des Benutzers einen sehr effizienten Plan für die Sitzungen erstellen. Hierfür verwendet das System ein sogenanntes Backtracking Verfahren, das bedeutet, es geht alle Möglichkeiten durch, bis ein Fehler auftritt, oder eine Lösung gefunden wurde.

3.7.2 Auslöser

Benutzer klickt auf den Knopf, um die Berechnung zu starten

3.7.3 Akteur(e)

Benutzer, System

3.7.4 Vorbereitung(en)

Das Programm ist gestartet

Benutzer befindet sich im Hauptmenü

Es ist mindestens ein Klassenrat vorhanden, der mindestens einen Lehrer beinhaltet

3.7.5 Schritte

1. Benutzer klickt auf den Knopf zum Starten
2. Das System evaluiert den Besten Plan mit einem Backtracking Verfahren

3.7.6 Involvierte Klassen

Teacher, Authority, Council, Interval

3.8 Use Case „Eingabe der Zeiten an denen die Sitzungen ausgeführt werden“

3.8.1 Kurze Beschreibung

Der Benutzer soll eingeben können, wann ein bestimmter Lehrer Zeit hat und wann nicht.

3.8.2 Auslöser

Benutzer erstellt einen neuen Lehrer oder bearbeitet einen bereits existierenden.

3.8.3 Akteur(e)

Benutzer

3.8.4 Vorbereitung(en)

Programm wurde gestartet

3.8.5 Schritte

1. Benutzer klickt entweder auf das „+“ Symbol in der Lehrer Liste oder doppelklickt einen Lehrer, um diesen zu bearbeiten
2. Benutzer gibt Intervalle an, an denen die Person Zeit hat

3.8.6 Involvierte Klassen

Interval, Teacher, Authority

3.9 Use Case „Speichern der Pläne“

3.9.1 Kurze Beschreibung

Das System soll nach dem Erstellen eines Plans diesen als Grafik speichern können

3.9.2 Auslöser

Das System hat den Plan fertiggestellt

3.9.3 Akteur(e)

System

3.9.4 Vorbereitung(en)

Programm wurde gestartet

Start-Knopf wurde gedrückt

Ein Plan wurde evaluiert

3.9.5 Schritte

1. Benutzer klickt auf den Knopf für die Evaluierung
2. Das Programm erstellt die Pläne
3. Das Programm speichert die Pläne als Bild

3.9.6 Involvierte Klassen

/

4 Klassenmodell

4.1 Klassendiagramm

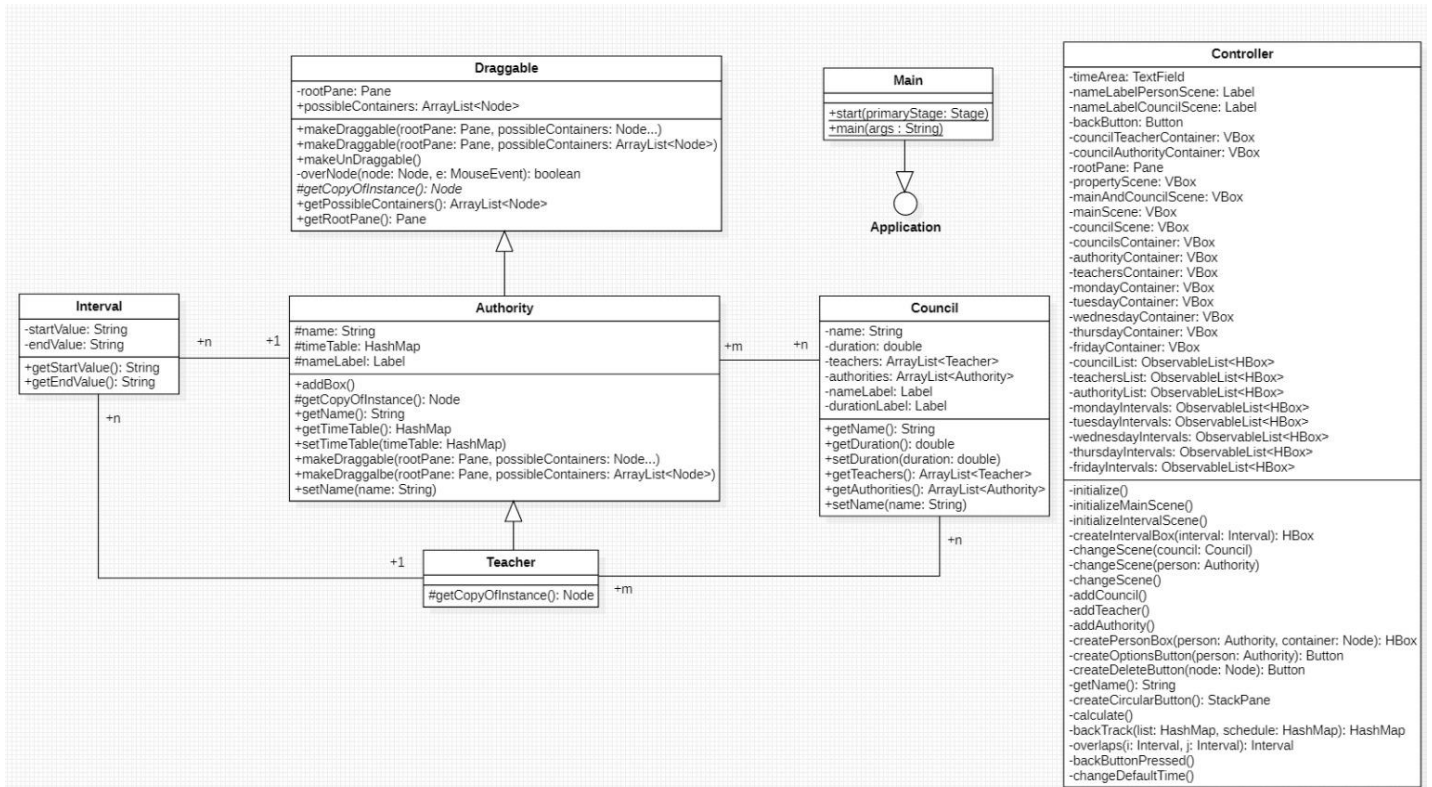


Abbildung 3: Bild des Klassendiagramms

5 Umsetzungs-Details

5.1 Zielplattform

Die Klassenratssitzungsplaner-Software wird als JavaFX-Applikation umgesetzt. Zielplattform ist Java 11 und damit alle gebräuchlichen Betriebssysteme. Besondere Hardware-Anforderungen sind eine aktuelle Java Version, dies bedeutet Java 11 oder neuer, weitere Anforderungen werden nicht gestellt. Auf älteren Computermodellen wird nur bedingt Rücksicht genommen.

5.2 Datenhandlung

Die Datenhaltung erfolgt in einem normalen Ordner, der vom Programm direkt erstellt wird. Das Programm kann die Daten anschließend ändern oder für die Berechnung der Klassenratssitzung verwenden. Speichermedium ist eine normale Textdatei, worin alle erforderlichen Objekte gespeichert werden.

5.3 Oberfläche

Die Oberfläche wird in JavaFX erstellt dazu wird eine FXML Datei mit dem Scene-Builder für eine einfache Implementierung verwendet.

Beim Aufbau soll es mehrere verschiedene Oberflächen eingeteilt werden, um die Übersichtlichkeit zu steigern, dazu wird die Steuerung mit dem „drag and drop“ feature deutlich vereinfacht.

5.4 Systemarchitektur (grob)

Das Programm ist sehr einfach aufgebaut und benötigt daher keine speziellen externen Medien und Informationen zu speichern oder einzulesen.

5.5 Vorgehensmodell

Um unser Projekt so gut wie möglich zu planen, entschieden wir uns für das agile Vorgehensmodell Scrum, da es eine höhere Erfolgsrate hat als andere Vorgehensmodelle. Scrum basieren auf einfachen Regeln, definierten Rollen und täglichen Abstimmungen im Team.

Es gibt bei Scrum drei Rollen:

- der Produkteigner (Product Owner)
- der Scrum-Master
- das Mitglied im Projektteam

5.5.1 Product Owner

Der Produkt-Owner vertritt die Anwender des Produkts. Das sind alle, die betroffen sind und ein Interesse am Erfolg des Projekts haben.

5.5.2 Scrum-Master

Der Scrum-Master trägt die Verantwortung für den Scrum-Prozess. Er ist Moderator und Unterstützer für das Projektteam. Zu seinen Aufgaben gehört das Beseitigen von Hindernissen und das Fördern einer guten Zusammenarbeit.

5.5.3 Projektteam

Ein Scrum-Team besteht normalerweise aus zwischen fünf und zehn Mitarbeitern. Die Teammitglieder organisieren alle Aufgaben selbst, dabei gibt es keine Hierarchie. Jeder hat dieselben Rechte und Pflichten, aber durchaus unterschiedliche Kompetenzen.

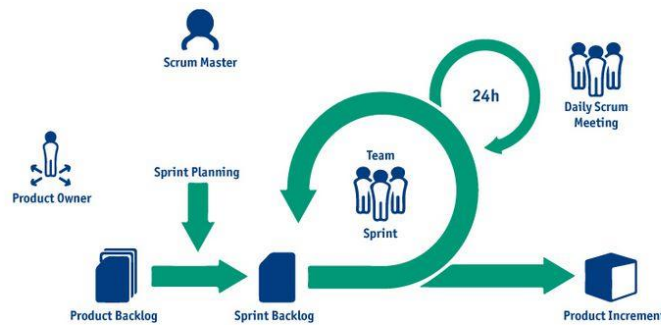


Abbildung 4: Bild des Scrum-Verlaufs

Sprints sind die typischen Entwicklungszyklen die meist 2 Wochen dauern. Am Ende jeden Sprints wird dem Kunden ein funktionsfähiger Stand des entwickelten Produktes vorgeführt und die weiteren Aufgaben für den nächsten Sprint festgelegt. Mit diesem Verfahren hat der Kunde die Kontrolle und Übersicht der Entwicklung und kann schnell bei eventuellen Marktveränderungen und Fehlentwicklungen reagieren.

5.6 Eigene Erfahrung

Nachdem wir uns zusammengesetzt haben und zuerst grob das Projekt geplant haben, begannen wir auch schon mit dem ersten Sprint. Nach dem ersten Sprint hatten wir ein Meeting untereinander. Dort besprachen wir die weiteren Schritte. Leider haben wir es nur einmal geschafft unseren Fortschritt dem Kunden zu präsentieren. Dann ging es immer weiter mit den Sprints und wir haben uns regelmäßig zu Meeting getroffen.