Scrum-Dokumentation

# Überblick

## Rollenverteilung

**Product Owner :**

* Amato Amedeo

**Scrum Master**:

* Jede Woche wird gewechselt

**Team:**

* Amato Amedeo
* Koller Remo
* Smelt Alexander
* Solis Christoph
* Yesil Egemen

## Scrum-Planung

Zu Beginn wurden aus der Problembeschreibung, die verschiedenen Hauptfunktionalitäten abgeleitet. Diese sogenannten User Stories wurden im Product Backlog („Product\_Backlog.xlsx“) festgehalten, wo diese von Product Owner auch nach Aufwand und Priorität sortiert wurden.

Die Dauer der Sprints wurde auf eine Woche festgelegt. D.h. jede Woche wurden beim Sprint Planning Meeting einige User Stories aus dem Product Backlog ausgewählt, und bearbeitet.  
Dazu wurden die Stories in einzelne kleinere Teilfunktionalitäten zerlegt und im Team verteilt. Dies wurde in den Sprint Backlogs („Sprint\_Backlog.xlsx“) festgehalten, für jeden Sprint gibt es ein eigenes Excel-Blatt.

Bei diesen Meetings wurde jeweils auch das Sprint Review des vorherigen Sprints durchgeführt.

Auf die Daily Scrums wurde aus Zeitgründen verzichtet.

# Erfahrungen

## Sprint 0:

**Überblick**:

* Das Team wurde gegründet.
* Die Einzelnen Rollen im Team wurden verteilt.
* Das Projekt und das Vorgehen wurden besprochen.
* Gemeinsam wurde ein Erstentwurf vom Produktbacklog erstellt.

## Sprint 1 :

**Problem:**

Zusammenarbeit im Team ist schlecht, da einige Teammitglieder nicht zum Meeting erscheinen sind.

User Stories mussten die Teammitglieder (Egemen und Amedeo) unter sich ausmachen.

**Feststellung:**

Teamarbeit und Verlässlichkeit der einzelnen Teammitglieder muss besser werden.

## Sprint 2:

**Problem:**

Da wieder das Team nicht vollzählig war, wurde beschlossen Verbesserungsmassnahmen einzuleiten.

**Feststellung:**

Scrum baut auf ein gutes Team auf. Wenn das Team nicht funktioniert und mehrere Teammitglieder nicht effektiv arbeiten, ist das gesamte Projekt gefährdet und kann scheitern.

## Sprint 3:

**Feststellung:**

Das Team erscheint wieder vollständig. Das Projekt kommt gut voran.

## Sprint 4:

**Feststellung:**Die Erfahrung zeigt, dass Scrum in dieser Schul-Umgebung nicht funktioniert. Teilfunktionalitäten einer User Story können nicht gut von mehreren Personen implementiert werden. Dazu müsste eine bessere Kommunikation (auch während dem Programmieren) stattfinden, was in 2 Stunden pro Woche nicht möglich ist.

## Sprint 5:

**Feststellung:**Ein Teilnehmer ist und wird aufgrund eines Unfalls eine Weile ausfallen. Ein weiterer hatte ein Vorfall in der Familie was ein Treffen mit den restlichen Teilnehmer unmöglich machte.

Ist noch einer Krank wie in dieser Woche, so ist es unmöglich ein persönliches Treffen über zwei Standorte zu organisieren. Es bleibt also nur die Kommunikation über Email oder Telefon, was an der Teammotivation und Effizienz nagt.

Trotzdem dank zwei sehr motivierenden und hilfsbereiten Studenten sind wir auf gutem Wege das Projekt fertigzustellen.

## Sprint 6:

**Feststellung:**Bis auf einige Punkte, an welchen leider schon seit mehreren Sprints ohne nennenswerten Fortschritt gearbeitet wird, wurde das Spiel soweit fertig gestellt.

## Sprint 7:

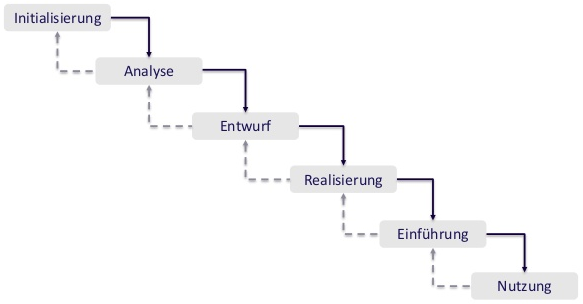
**Feststellung:**Die restlichen Punkte wurden schlussendlich anderen Teammitgliedern zugewiesen, welche diese nun auch rechtzeitig umgesetzt haben. Das Spiel wurde nach sieben Sprints somit fertiggestellt!

Das Spiel wurde von allen auf Fehler getestet. Es wurden keine gefunden.  
Ausserdem wurde der Code auch etwas aufgeräumt und fertig dokumentiert.

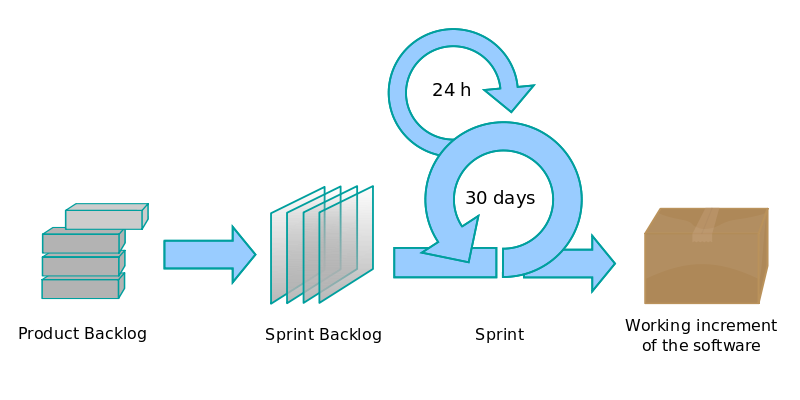
# Vergleich der Prozesse

## Scrum vs. Wasserfallmodell

Beim Wasserfallmodell wird auf feste Anforderungen und klare Phasenabgrenzung gesetzt.



Scrum hingegen setzt auf flexible Anforderungen und dynamische Implementierung, die während dem Projektverlauf fortlaufend angepasst werden.



### Vorteile

**Scrum:**

* Flexibilität
* Gegenseitige Kontrolle im Team
* Keine aufwändige Planung im Voraus nötig
* Feedback vom Kunden während Projekt möglich

**Wasserfall:**

* Klare Planung
* Durch stabile Anforderungen sind gute Kosten- und Aufwandschätzungen möglich

### Nachteile

**Scrum:**

* Keine Analysephase
* Kein Pflichtenheft (juristische Absicherung)
* Kein Überblick über die Software-Architektur
* Hoher Kommunikationsaufwand mit dem gesamten Team nötig

**Wasserfall:**

* Klare Abgrenzung und Abfolge der Phasen nicht realistisch
* Aufwändige Analyse am Anfang
* Anpassung des Konzepts in Nachhinein sehr aufwändig