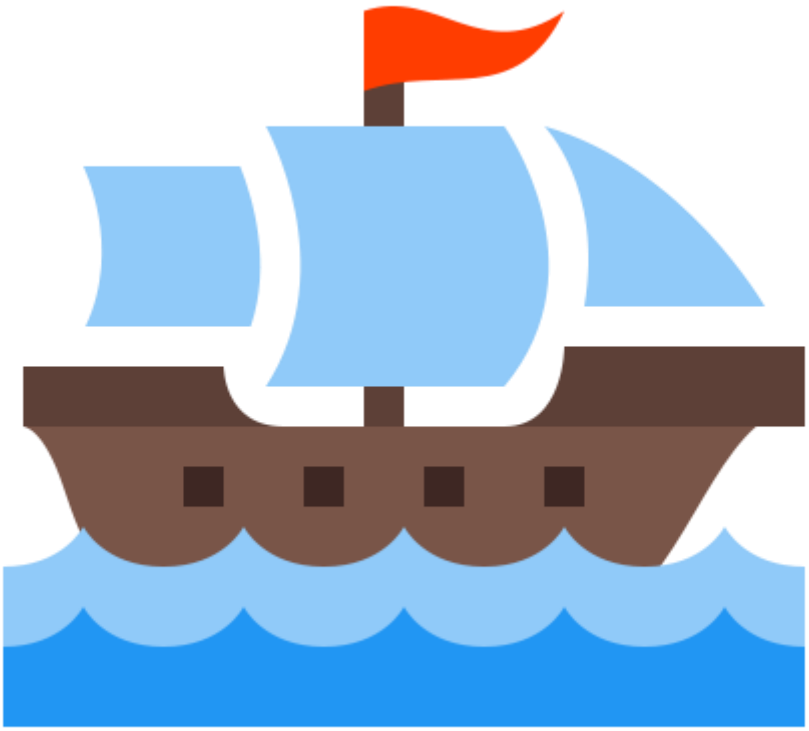

Scrum-Guideline

-

Guideline für den Scrum-Prozess

Version 1.3

Inhalt

0.1 Ziel des Dokumentes

Die Zielsetzung des Dokumentes besteht darin, den Scrum-Prozess eindeutig festzulegen und zu dokumentieren. Dabei werden auch Anpassungen an dem Scrum-Prozess aufgrund von Prozessoptimierungen festgehalten.

0.2 Changelog

Version	Datum	Änderung	Geändert von
1.0	21.10.2018	Initial	Steffen Sassalla
1.1	24.10.2018	Scrum Daily Meeting	Steffen Sassalla
1.2	28.10.2018	Retroperspektive aktualisiert	Steffen Sassalla
1.3	09.11.2018	Hinweise zu Taiga	Steffen Sassalla

1. Scrum-Framework – Grundlegendes

Scrum ist ein Vorgehensmodell im Projektmanagement und wird in der agilen Softwareentwicklung verwendet. Dabei baut es auf folgende grundlegende Prinzipien auf:

1. Erstellung Product-Backlog
2. Sprint-Planning
3. Sprints
4. Daily-Meeting
5. Sprint-Review

Die Definitionen der aufgelisteten Begriffe werden in den folgenden Kapiteln festgelegt.

2. Product-Backlog

Das Product-Backlog wird aus den User-Stories erstellt. Hierbei behält sich der Proudct-Owner[1] vor, das Product-Backlog auch mitten im Entwicklungsprozess kontinuierlich anzupassen, auch aufgrund eventueller Änderungen. Somit ist das Product-Backlog stetig im Wandel.
Für die Verwaltung des Product-Backlogs und die damit einhergehenden User-Stories behalten wir uns vor das Projektmanagement-Tool Taiga zu verwenden. Taiga bietet uns die Möglichkeit das Product-Backlog granular anzulegen und auch kontinuierlich anzupassen.

Taiga ist erreichbar unter: https://swtpra.cs.upb.de/project/jniclasg-swtpra-4/

Dabei steht für das Product-Backlog der Verantwortliche Product-Owner Lukas Gehring für aufkommende Fragen zur Verfügung. Unteranderem koordiniert der Verantwortliche das Product-Backlog im Umfang des Projektes.

Während der Erstellung des Product-Backlogs ist es wichtig die User-Stories in Taiga so granular wie möglich zu verfassen, damit im Sprint-Planning die Ressourcen optimal eingesetzt werden können. Ein Beispiel für eine User-Story im Product-Backlog soll dies veranschaulichen:

„As a player, I want to know the score of the game.“
„As a game host, I want to be able to import and export game configurations.“
„As a connected observer, I want to be able to see all scores of players.“

User-Stories werden also immer im folgenden Muster und in englischer Sprache im Product-Backlog erstellt:

Ich als	*<Rolle>*
möchte	*<nicht funktionale Anforderung>*
damit	*<Grund für die User-Story*

Zu jeder User-Story sind die zu umsetzenden Tasks definiert. User-Stories sind zu sogenannten Epics gruppiert, um eine gewisse Strukturierung im Product-Backlog sowie im Sprint-Planning zu haben. Abgeschlossenheit von Tasks, User-Stories und Epics sind in der Definition-of-Done festgehalten.

[1] Product-Owner ist Lukas Gehring

3. Sprint-Planning

Die grobe Sprint-Planung ist im Angebot unter Projektplan definiert. Dabei findet die genaue Sprint-Planung in Taiga statt, da hier aus dem Product-Backlog die Sprint-Planung erzeugt werden kann.

Jeder Sprint ist genau eine Woche lang!

Somit ergeben sich genau 10 Sprints über der gesamten Projektlaufzeit. Die Sprint-Dauer wurde auf eine Woche gelegt, damit das gesamte Team agil genug ist auf Änderungen und Komplikationen zu reagieren, vor allem auch da sich die grundlegende Softwarekomponente Interface Änderungen vorbehält.

User-Stories wurden zu Beginn des Projekts vom Product-Owner akzeptiert. Am Anfang des Sprints wurden nach der Retroperspektive des letzten Sprints im Team User-Stories zur Bearbeitung ausgewählt und die Zuständigkeiten eingeteilt. Dies stellt sicher, dass die Auswahl im ganzen Team abgestimmt ist und mögliche Fragen zu den User-Stories direkt für alle geklärt werden können. Außerdem können so Unklarheiten oder nicht fertiggestellte User-Stories des letzten Sprints mit einfließen. Die Abgeschlossenheit einer User-Story ist in der Definition-of-Done definiert.

Das Sprint-Planning ist agil gestaltet. Es wird jeweils maximal 2 Stunden je Sprint-Woche dauern und ist in zwei Teilen gegliedert:

1. Was kann im kommenden Sprint entwickelt werden?
2. Wie wird die Arbeit im kommenden Sprint erledigt?

Das Dokument der Definition-of-Done unterstützt dabei das Sprint-Planning um genau festzulegen, wann die entwickelten Features und Fixes als fertig anerkannt werden und auch um den Aufwand genauer abschätzen zu können.

4. Sprints

Ziel jedes Sprints ist es, ein hinreichend getesteter und integrierter Teil des Softwareproduktes fertigzustellen. Das Ergebnis jedes Sprints wird im Sprint-Review festgehalten. Die Sprint-Dauer ist auf genau eine Woche festgelegt, daraus ergeben sich für die gesamte Projektlaufzeit genau 10 Sprints.

5. Daily-Meeting

Das Standard-Scrum-Modell sieht ein tägliches 15-minütiges Meeting vor, in der folgende 3 Fragen jedes Teammitglieds beantwortet werden soll:

1. Was habe ich erledigt/bearbeitet?
2. Welche Probleme/Störer sind dabei aufgetreten?
3. Was werde ich bis zum nächsten Meeting erledigen/Was sind die heutigen Tagesziele?

Tägliche Meetings sind aufgrund der unterschiedlichen Tagespläne der Teammitglieder nicht umzusetzen. Das Daily-Meeting wird somit auf einen Zyklus von zwei Tagen angehoben und kann unter Umständen auch Online abgehalten werden.

6. Sprint-Review und Retroperspektive

Das Sprint-Review soll Überblick über den beendeten Sprint für das gesamte Team geben. Die Retroperspektive soll Erfahrungen des gesamten Teams sammeln und Probleme identifizieren, sodass im nächsten Sprint diese Erfahrungen mit einbezogen werden können. Es ist also eine Art

Prozessoptimierung. Das Sprint-Review und die Retroperspektive werden somit immer am folgenden Wochentag abgehalten:

dienstags, 11:00 – 13:00 Uhr, Tutorenstunde O1 252

Im Sprint-Review wird live am entwickelten Produkt gezeigt, welche Änderungen im Sprint umgesetzt wurden, welche Fehler aufgetreten sind und was daraus gelernt werden kann. Verbesserungen fließen somit im nächsten Sprint mit ein. Verbesserungen können aber auch in allen festgehaltenen Guidelines mit einfließen, wie z.B. das DevOps-Dokument. Das Sprint-Review ist für jedes Teammitglied pflicht.

In der Retroperspektive sind diese grundlegenden Fragen zu beantworten:

- Wie zufrieden sind wir mit dem Ergebnis des letzten Sprints? Haben wir unsere Sprint-Ziele erreicht und wenn nein: Warum nicht?
- Wie können wir die Zusammenarbeit im Team und auch mit dem Product-Owner verbessern?
- Wie können wir die Produktqualität optimieren?

Die Ergebnisse der Retroperspektive werden in einem gesonderten Dokument nach einer vordefinierten Struktur festgehalten.