

Scrum Setup

BTI7081 - Software Engineering and Design
Frühlingssemester 2020

Berner Fachhochschule
Abteilung Technik und Informatik

Team Blau:

Timon Borter
Sven De Gasparo
Luca Mühlheim
Marc Muster
Elias Schmidhalter

Klienten:

Prof. Urs Künzler
Prof. Dr. Jürgen Vogel

Inhalt

Rollen.....	3
Product Owner	3
Scrum Master	3
Developer.....	3
Artefakte	3
Product Backlog.....	3
Sprint Backlog.....	4
Definition of Ready	4
User Story.....	4
Task	4
Definition of Done	4
Task.....	4
User Story.....	4
Burndown Chart	4
Meetings	5
Product (and release) planning	5
Sprint Planning	5
Daily Scrum	5
Code Review.....	5
Sprint Review	5
Product (and release) review	5
Ressourcen Planung	5
Anhang.....	6
Tipps und Tricks zu Scrum.....	6
Abbildungsverzeichnis	6

Rollen

Product Owner

Die Rolle der Product Owner wird von unseren Klienten, Prof. Urs Künzler und Prof. Dr. Jürgen Vogel wahrgenommen.

Scrum Master

Als Scrum Master haben wir Timon Borter nominiert.

Developer

Elias Schmidhalter, Sven De Gasparo, Luca Mühlheim und Marc Muster fungieren in diesem Projekt als Entwickler.

GitHub Alias

Vorname, Name	GitHub Alias
Timon Borter	bbortt
Elias Schmidhalter	calesanz
Sven De Gasparo	zeroplexer
Luca Mühlheim	lucamuh
Marc Muster	marcarrian
Urs Künzler	urskuenzler
Jürgen Vogel	vgj1

Artefakte

Product Backlog

Im Produkt Backlog werden wir User Stories festhalten. Dies sind grobkörnige Anforderungen, die oft eine genauere Definierung erfordern. Man kreiert also zu einer User Story mehrere Tasks, diese sind dann Teil des Sprint Backlogs.

Das Produkt Backlog wird in einem [GitHub Projekt](#) geführt. Dies hat den Vorteil, dass neue User Stories direkt zuunterst im Backlog hinzugefügt werden, also mit der tiefsten Priorität. Zusätzlich werden alle Mitglieder über Neuigkeiten informiert. Die Priorität kann anschliessend gemeinsam am Planning angeschaut werden, wobei die dringendsten User Stories den höchsten Platz im Backlog erhalten. Das Team (die Developer) werden dann die notwendigen Tasks für das Sprint Backlog definieren.

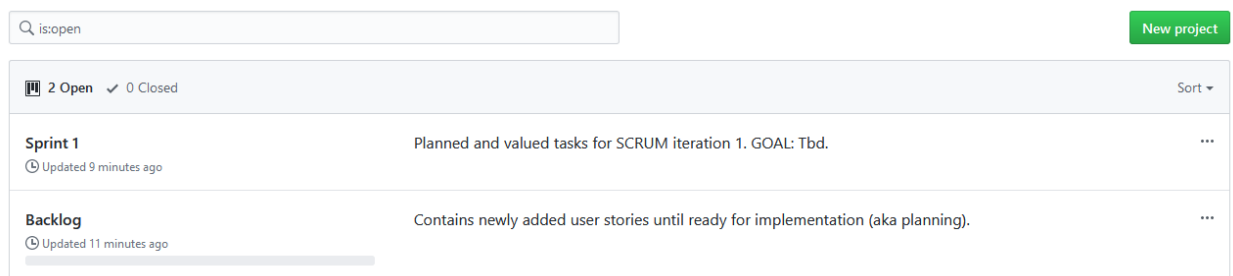


Abbildung 1: Die GitHub Projekte.

Sprint Backlog

Hier werden Tasks von den Developern von links nach rechts abgearbeitet. Das zu erfüllende Value wurde zusammen mit dem Sprint Ziel in gemeinsamen Konsens am Planning festgehalten.

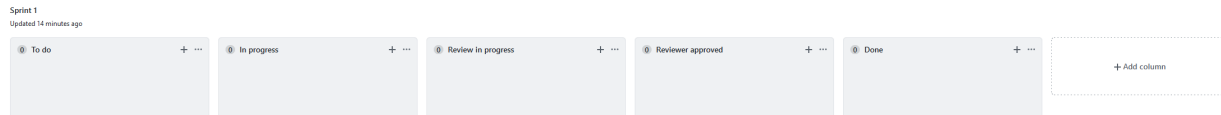


Abbildung 2: Sprint Backlog.

Definition of Ready

User Story

Eine User Story zählt als bereit für das Planning wenn:

- Der Product Owner die User Story entsprechend priorisiert hat
- Das Developer Team ein gemeinsames Verständnis der Anforderung hat
- Ein Value in Story Points geschätzt wurde

Task

Ein Task zählt als bereit umgesetzt zu werden, wenn folgende Punkte erfüllt sind:

- Der Task ist in gemässiger Zeit durch einen Developer unabhängig umsetzbar
- Testbare Akzeptanzkriterien sind festgelegt
- Der Task ist von den Developern in Story Point geschätzt worden

Definition of Done

Task

Ein Task kann abgeschlossen werden, wenn:

- Der Build im Pull-Request erfolgreich mit Status «passed» abgeschlossen hat
- Die statische Code Analyse keine groben Fehler aufweist
- Der Pull-Request mit mindestens einem Approval gemergt wurde
- Die Dokumentation falls nötig aktualisiert oder erstellt wurde

User Story

Eine User Story wird abgeschlossen, sobald:

- Alle dafür definierten Tasks abgeschlossen wurden
- Der Product Owner am Review eine positive Rückmeldung abgegeben hat

Burndown Chart

Die GitHub Projekte bieten leider keine Möglichkeiten, Burndown Charts zu generieren. Dies muss manuell nachgeführt werden.

Meetings

Product (and release) planning

Wird via Teams durchgeführt. Teilnehmende sind die Produkt Owner, der Scrum Master sowie die Entwickler. Hierbei geht es darum, die Priorität und die nächsten User Stories für das Team festzulegen.

Sprint Planning

Die Entwickler diskutieren über die Art der Umsetzung und definieren detaillierte Tasks, um das Sprint Ziel zu erreichen.

Daily Scrum

An Unterrichtstagen (Montag, Dienstag) werden die Developer jeweils zu Beginn der Lektion ein kurzes Standup abhalten. Tägliche Treffen an sonstigen Tagen sind wegen des berufsbegleitenden Studiums nicht möglich.

Code Review

Code Reviews werden in GitHub Pull-Requests gemacht. Ein Pull-Request darf dabei erst gemerged werden, wenn alle Überprüfungen positiv sind und mindestens ein Approval gegeben wurde.

Sollte eine bilaterale Absprache vom Ersteller des Pull-Requests gewünscht sein, so darf dies individuell abgesprochen werden.

Sprint Review

Am Sprint Review werden die Entwickler die erreichten Ziele den Product Owner (stolz) präsentieren. Dieser darf dabei kontinuierlich Feedback geben, welches gegebenenfalls in den nächsten Sprint Zyklus einfließt.

Product (and release) review

Schlussendlich, in diesem Fall nach drei geplanten Sprints, wird die umgesetzte Applikation noch einmal abschliessend mit dem Product Owner angeschaut.

Ressourcen Planung

Für die Entwickler werden drei Sprints mit folgenden Ressourcen eingeplant:

- Sprint 1: 8 Stunden/Teammitglied
- Sprint 2: 14 Stunden/Teammitglied
- Sprint 3: 6 Stunden/Teammitglied

Anhang

Tipps und Tricks zu Scrum

[scrum.org](https://www.scrum.org) bietet einen einfachen und übersichtlichen Einstieg in den Umgang mit Scrum.

Abbildungsverzeichnis

Abbildung 1: Die GitHub Projekte.	3
Abbildung 2: Sprint Backlog.	4