

Exemplarische Beschreibung der agilen Vorgehensweise mit SCRUM

User Stories

Vor dem Start eines Projekts müssen die Funktionsanforderungen in User Stories übersetzt werden. User Stories beinhalten die Anforderungen und beschreiben zusätzlich warum etwas getan werden muss.

Exemplarisch werden unten zwei User Stories genannt, die mithilfe der SOPHIST-Satzschablone Funktionsmaster erstellt wurden.

User Story 1:

Der Gepäckscanner muss fähig sein Handgepäck zu scannen.

User Story 2:

Wenn der Pin drei Mal falsch eingegeben wurde, muss der Gepäckscanner fähig sein den Ausweis zu sperren.

Definieren des Product Backlogs und der Velocity

Product Backlog

Das Product Backlog ist eine Liste mit Tätigkeiten, die für diese Projekt noch zu erledigen sind. Sie sind auf Basis des Return of Investments (ROI) priorisiert.

Velocity

Die Velocity ist die Arbeitsmenge die das Team pro Sprint schafft. So kann eingeschätzt wie viele User Storys pro Sprint bearbeitet werden können.

- 4 Mitarbeiter mit je 40 Stunden pro Woche
-> 160 Mannstunden pro Woche
- Bei einer Sprintdauer von 14 Tagen
-> 320 Mannstunden
- Tägliche Meetingzeit 0,25 Stunden pro Tag
-> 10 Mannstunden pro Sprint
- Kalkulierter Ausfall durch Krankheit, Urlaub oder Sonstiges
-> 38 Mannstunden pro Sprint
- Resultat: **272 Mannstunden** für die Abarbeitung der Sprinttasks innerhalb eines Sprints

Sprint – Sprint Planning Meeting

Das Team wählt Items aus dem Product Backlog aus, die sie in diesem Sprint bearbeiten wollen und definiert die Definition of Done

Planning Poker

Jedes Teammitglied schätzt wie viel Zeit für eine Aufgabe gebraucht wird. Treten Differenzen auf werden diese besprochen, bis eine Einigung eintritt. So wird sichergestellt dass alle Teammitglieder in die Diskussion mit einbezogen werden und der Aufwand zuverlässig geschätzt wird.

Bsp.:

Teammitglied A schätzt den Aufwand von User Story 1 auf 4 Stunden. Teammitglied B schätzt 7 Stunden. B erklärt A warum es soviel Zeit benötigt. Nach einer kurzen Diskussion einigen sich A und B auf 6 Stunden.

Definition of Done

Die Definition of Done legt fest welche Anforderungen das Ergebnis des Sprints erfüllen soll. Das Ergebnis wird am Ende des Sprints mit der Definition of Done verglichen.

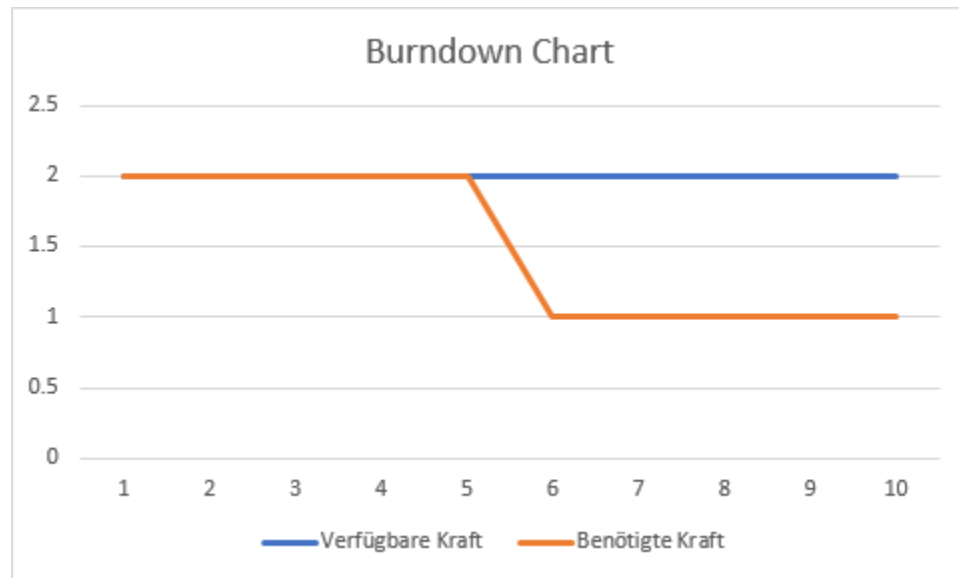
Sprint – Scrum Board

Das Scrum Board ist eine Tabelle mit den Spalten To-Do In-Progress und Done. In diese Spalten sind alle User Stories eingeordnet. Das Scrum Board gibt so einen Überblick über den aktuellen Stand des Projekts. Das Scrum Board wird täglich aktualisiert.

To Do	In Progress	Done
	User Story 1	
User Story 2		

Sprint – Burndown Chart

Das Burndown Chart besteht aus zwei Linien. Der benötigten und der verfügbaren Kapazität. Das Burndown Chart gibt wie das Scrum Board einen Überblick über den aktuellen Stand des Projekts. Das Burndown Chart wird täglich aktualisiert



Innerhalb der ersten fünf Tage werden beide User Storys bearbeitet werden. Beide verfügbaren Kräfte (A und B) werden gebraucht. A wird am 5. Tag fertig. Für die folgenden fünf Tage stehen zwei verfügbare Kräfte zur Verfügung.

Sprint – Daily Scrum

Jeden Tag hält das Team ein kurzes Meeting von etwa 15 Minuten ab. Während des Meetings soll jedes Teammitglied drei Fragen beantworten:

- Was habe ich seit dem letzten Daily Scrum Meeting gemacht?
- Was sind meine Aufgaben bis zum nächsten Daily Scrum Meeting?
- Welche Hilfe benötige ich um meine zugeteilten Aufgaben zu erledigen?

Ziel des Daily Scrums ist das jedes Teammitglied effektiv eingesetzt wird und das Ziel im Blick behält.

Sprint – Sprint Review/Retrospective

Sprint Review

Im Sprint Review stellt das Team ihr Inkrement, das Ergebnis des Sprints vor. Das Team überprüft ob das Ergebnis der zuvor festgelegten Definition of Done entspricht und alle Punkte erfüllt sind. Ist dies nicht der Fall muss nachgebessert werden. Das Sprint Review findet immer am Ende eines Sprints statt.

Retrospective

Die Retrospective ist der Abschluss eines Sprints. Hier bespricht das Team was während des letzten Sprints gut lief und was schlecht lief. Ziel der Retrospective ist es Probleme zu identifizieren und diese im nächsten Sprint zu verhindern. Was gut funktioniert soll dagegen im nächsten Sprint beibehalten werden.