

# AGILO HOWTO

Agilo [1] ist ein auf Trac [2] basierendes Scrum [3] Tool. Im Folgenden soll eine kurze Übersicht gegeben werden, wie Agilo benutzt wird.

## ROLLEN IM TEAM

In Scrum hat jedes Teammitglied eine von drei möglichen Rollen:

- **PRODUCT OWNER**

Der Product Owner (PO) stellt die Verbindung zum Kunden her. Er ist das Bindeglied zwischen den Kunden und dem Team und wird häufig auch als „On-Site Customer“ bezeichnet.

Der PO ist für folgendes verantwortlich:

- Langzeitplanung
- Pflege des **PRODUCT BACKLOGS**. Der PO muss für jede Anforderung des Kunden ein **REQUIREMENT** anlegen. Für jedes Requirement legt er mehrere zu diesem Requirement passende **USER STORIES** an, die genau beschreiben, was in konkreten Situationen passieren soll.

- **SCRUM MASTER**

Der Scrum Master (SM) ist dafür zuständig, dass das Team reibungslos arbeiten kann. Treten Probleme auf, muss der SM sich dafür einsetzen, dass diese gelöst werden.

Der SM ist für folgendes verantwortlich:

- Erstellen von **TASKS** zu User Stories
- Reibungsloser Arbeitsablauf innerhalb des Teams. Problembeseitigung.
- Erstellung von Dokumentation für die Abgabe beim Kunden. Hier benötigt er insbesondere die Hilfe der Developer. Zur Dokumentation gehören:
  - Auszug der zum vergangenen Release gehörenden Informationen aus Agilo: **SPRINT BACKLOG**, User Stories, Diagramme zu den User Stories

- **DEVELOPER**

Die Entwickler sind für die Abarbeitung des Sprint Backlogs verantwortlich. Sie Arbeiten an Tasks und aktualisieren regelmäßig (jedes Mal wenn an einem Task gearbeitet wurde) die noch geschätzt verbleibenden Stunden.

Je nach Rolle stehen dem Teammitglied in Agilo unterschiedliche Tools zur Verfügung. So kann z.B. der SM keine Requirements und User Stories anlegen.

## ABLAUF

Im Folgenden der Ablauf wie er typischerweise für die Umsetzung eines Releases stattfindet.

### RELEASE PLANUNG

Da der PO für die Langzeitplanung verantwortlich ist, ist es seine Aufgabe, die Releases sowie SPRINTS anzulegen. Diese Aufgabe muss nur einmal für das ganze Projekt ausgeführt werden.



Es gibt insgesamt 4 Releases. Jedes Release hat eine Länge von 4 Wochen und besteht aus 2 Sprints zu je 2 Wochen.

Unter „PROJEKTPLAN“ erhält der PO in Agilo eine Projektübersicht. Hier sind folgende Schritte auszuführen:

1. In einer neuen Agilo Umgebung gibt es bereits 4 „Meilensteine“ durch Klick auf auf den Meilensteinnamen in „Release 1“ bis „Release 4“ umbenannt werden können.
2. Für jedes Release werden nun 2 Sprints über „ADD NEW SPRINT“ angelegt. **HIER DARF NICHT VERGESSEN WERDEN, DEM SPRINT EIN TEAM ZUZUORDNEN!**



Die Daten für die Sprints können dem Google Kalender zur Veranstaltung entnommen werden:  
<http://bit.ly/lmlW1D>

### 1. KUNDENTREFFEN

Kundentreffen dienen dazu herauszufinden, was genau der Kunde in seinem Produkt haben möchte.

- I. Vor jedem Release trifft sich der PO mit dem Kunden. Hierzu muss der PO einen Termin beim Kunden machen.



Bei SE 1 ist es durchaus erlaubt, dass der SM bei dem Kundentreffen dabei ist.

Beim Kundengespräch werden die Anforderungen des Kunden zum Release aufgenommen und versucht so viel wie möglich über die Wünsche des Kunden herauszufinden.

- II. Mit den Anforderungen kann der PO nun mehrere Requirements in Agilo anlegen. Hierzu wählt er in Agilo auf der linken Seite „NEW REQUIREMENT“. Die Kurzbeschreibung sollte möglichst treffend sein. In der Beschreibung muss der PO detailliert die Anforderung des Kunden formulieren. Der „Meilenstein“ (Release) wird gewählt. Der „BUSINESS VALUE“ gibt an, wie wichtig diese Anforderung dem Kunden ist.

### Beispiel für ein Requirement

Kurzbeschreibung: Graphical user interface for login

Beschreibung: The customer wants some graphical user interface which allows the user to login with his/her credentials. Also some kind of server status would be great (online/offline) as well meaningful error messages.

- III. Für jedes Requirement erstellt der PO nun mehrere User Stories. User Stories sind konkrete Szenarien, die für ein bestimmtes Requirement auftreten können. Um für ein Requirement eine User Story zu erstellen ist folgendes zu tun:
  - a. Über „TICKETS“ -> „ACTIVE TICKETS“ das richtige Requirement auswählen
  - b. „Bearbeiten“ wählen
  - c. „CREATE A NEW REFERENCED 'USER STORY'“ wählen
  - d. Kurzbeschreibung und Beschreibung ausfüllen

### Beispiel für eine User Story

Kurzbeschreibung: Login (successful)

Beschreibung:

Start situation: Alice started the client and the login screen opened.

Step 1: Alice enters her credentials

Step 2: Alice clicks on „Login“

Result situation: Alice is successfully logged in and is now in the lobby. No error messages occurred.

Die Requirements sowie die dazu gehörigen User Stories ergeben zusammen das „PRODUCT BACKLOG“.

## 2. SPRINTPLANUNG

Sobald der PO mit dem Erstellen des Product Backlogs fertig ist, stellt er das Product Backlog dem Team vor.

- I. Das komplette Team (PO, SM und alle Developer) treffen sich
- II. Der PO stellt dem Team das Product Backlog vor, indem er nach und nach jedes Requirement sowie die dazu gehörigen User Stories präsentiert. Hierzu wählt der PO in Agilo „Product Backlog“ auf der linken Seite.
- III. Für jede User Story schätzt das Team nun gemeinsam die Komplexität der User Story mit Hilfe von „STORY POINTS“. Je höher die Zahl, desto komplizierter ist diese User Story umzusetzen.
- IV. Der PO weist jeder User Story nun einen Sprint zu.



Typischerweise wird einer der beiden Sprints des aktuellen Releases ausgewählt!

Nachdem der PO mit der Vorstellung des Product Backlogs fertig ist, ist seine Aufgabe zunächst abgeschlossen. Alle User Stories die Sprints zugeordnet wurden, wurden automatisch aus dem Product Backlog entfernt.



Idealerweise sollte der Product Backlog nun leer sein (da nicht für das komplette Produkt sondern immer nur für ein Release geplant wird). Falls dem nicht so ist, können offenbar nicht alle Anforderungen des Kunden umgesetzt werden. In diesem Fall sollte noch einmal Rücksprache mit dem Team und im Zweifel mit dem Kunden gehalten werden!

**TRETEN IM WEITEREN VERLAUF FRAGEN AUF (Z.B. „WELCHE FARBEN MÖCHTE DER KUNDE GERN HABEN?“), IST ES DIE AUFGABE DES PO'S DIESE FRAGEN MIT HILFE DES KUNDEN ZU KLÄREN.**

### 3. TASKPLANUNG

Der SM ist nun dafür verantwortlich zusammen mit den Developern für jede User Story mehrere „TASKS“ zu erstellen. Hierzu ist folgendes zu tun:

- I. In Agilo wählt der SM auf der linken Seite den Sprint aus und drückt auf „VIEW“.
- II. Die User Story auswählen (auf die Zahl in der „ID“ Spalte klicken)
- III. In der User Story „Bearbeiten“ wählen.
- IV. Über „CREATE A NEW REFERENCED 'TASK'“ einen neuen Task erstellen.
- V. Kurzbeschreibung und Beschreibung ausfüllen.
- VI. Nötige Zeit in Stunden schätzen.

#### **Beispiel für einen Task der User Story „Login (successful)“**

Kurzbeschreibung: Switch screen from login to lobby

Beschreibung: After a successful login, the screen must switch from the login screen to the lobby screen. Possible solution: Tell ScreenManager to switch the screen.

### 4. KAPAZITÄTSPLANUNG

Zuletzt schließt der SM die Sprintplanung ab und erzeugt noch „KONTINGENTE“ für unvorhergesehene Aufgaben (z.B. Bugfixing). Hierzu ist folgendes zu tun:

- I. In Agilo wählt der SM auf der linken Seite den Sprint aus und drückt auf „VIEW“.
- II. Über den Button oben links „Show or hide contingents“ können die Kontingente eingeblendet werden.
- III. Es können beliebig viele Kontingente über den „+“ Button in der Kontingente-View angelegt werden. Hier muss ein „Contingent title“ sowie „Total amount of time“ (in Stunden) angegeben werden.

Ist die Planung für Kontingente abgeschlossen, schließt der SM die Planung durch den Klick auf den Button „Confirm Commitment“ ab.

## 5. TÄGLICHE ARBEIT

Die Entwickler fangen nun an, die erstellten Tasks nach und nach abzuarbeiten. Nachdem eine Funktionalität für einen Task implementiert wurde, wird die noch verbleibende Zeit geschätzt und im Task aktualisiert.

Ist der Task abgeschlossen wird die noch verbleibende Zeit auf 0 gesetzt und der Task geschlossen.

[1] Agilo Software GmbH, [Online]. Available: <http://agilosoftware.com/>. [Zugriff am 27 04 2012].

[2] E. Software. [Online]. Available: <http://trac.edgewall.org/>. [Zugriff am 27 04 2012].

[3] [Online]. Available: <http://de.wikipedia.org/wiki/Scrum>. [Zugriff am 27 04 2012].