Die Rolle des *Product Owner* im Entwicklungsprozess einer Software-Applikation für das GSI-Operating

Inhaltsverzeichnis

Der Software-Entwickungsprozess als Agile Methode

Im Bereich der Software-Entwicklung werden häufig Agile Methoden dem Wasserfall-Modell orientierten Vorgehensweisen vorgezogen. So auch bei der hier betrachteten Methode der Applikationsentwicklung für das GSI-Operating.

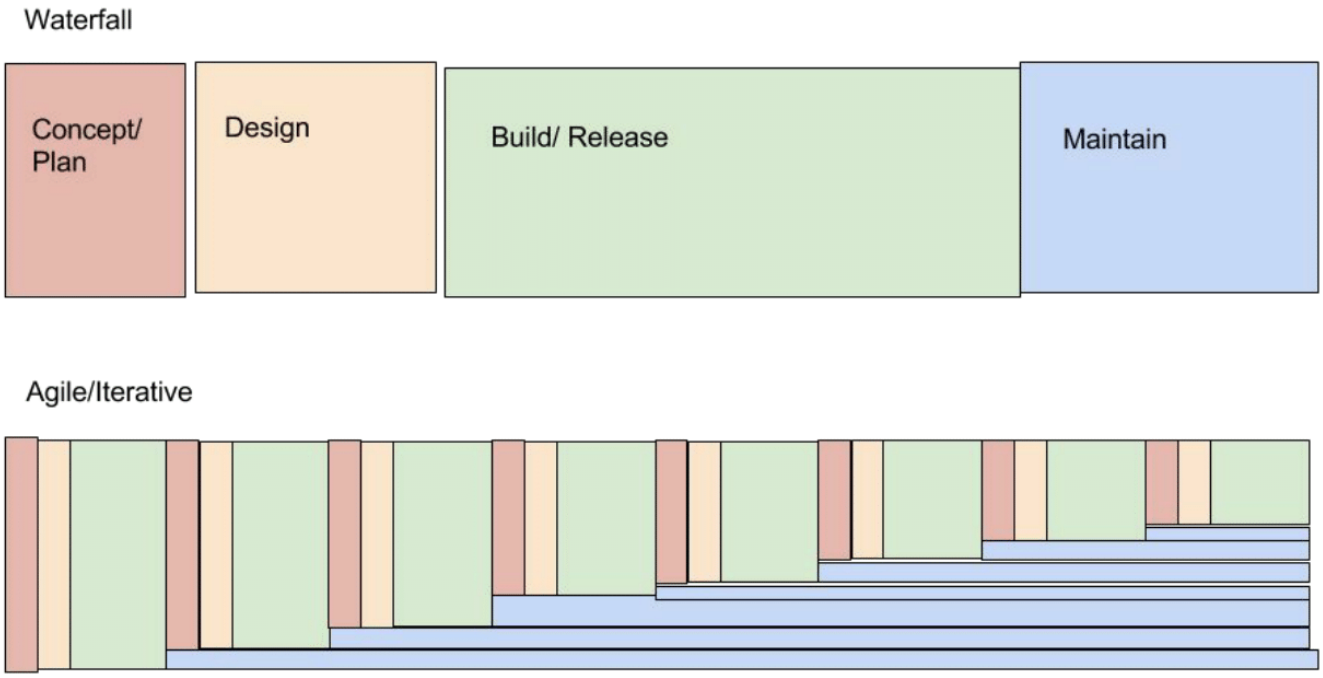

Figure 1. Wasserfall vs. Agile/Interative Vorgehensweise. KSmith (WMF),CC BY-SA 4.0 https://creativecommons.org/licenses/by-sa/4.0, via Wikimedia Commons

Ein Block aus roter Planungsphase, gelber Designphase und grüner Herstellungsphase entspricht einer Iteration bei Agilen Methoden und wird im folgenden als *Sprint* bezeichnet.

Die verschiedenen Rollen im Entwicklungsprozess einer Softwareapplikation

Product (Team)

Der Anglizismus *Product* ist im Folgenden mit dem Wort Software-Applikation gleichzusetzen.

Die Rolle des *Stake Holder* außerhalb des *Product Team*

Als erstes widmen wir uns der Rolle der *Stake Holder*, den Personen, für die die Software-Applikation gemacht wird oder entwickelt wurde.

Betriebsleiter, Maschinen Koordinatoren, Mitglieder der FAIR Commissioning Group und Operateure sind nicht direkt Mitglieder des Product Teams. Sie sind sogenannte *Stakeholder* oder Nutzer und ihre Interessen werden vom *Product Owner* im *Product Team* vertreten.

Traditionell kommt die Rolle des *Stake Holder* der Rolle des Kunden nahe. Im Kontext einer innerbetrieblichen Entwicklung kommen die Bezeichnungen Nutzer, Nutznießer oder Anwender dieser Rolle aber meistens näher. Diesen Bezeichnungen untersteht letztlich die Forderung, dass

Wünsche und Anforderungen dieses Personenkreises in das zu entwickelnde Produkt im Rahmen des rational möglichen einzufließen haben.

Die Rollen innerhalb des *Product Team*

- *Mitglied des Entwicklerteams* (z.B. aus dem Developer Pool)
- *Product Owner*

Mitglied des Entwicklerteams

Mitglied des Entwicklerteams sind Personen, die direkt an den verschieden Entwicklungsaktivitäten beteiligt sind. Sie führen u.a. Aktivitäten und Leistungen durch in den Bereichen:

- Software- / System-Design
- Benutzeroberflächen-Design
- Erstellung von sowohl technischer als auch Benutzer-Dokumentationen
- Programmierung
- Aufgaben der Qualitätssicherung

Product Owner

Der *Product Owner* ist eine Art Bindeglied zwischen *Stake Holdern* und *Entwicklerteam* und steht dem *Entwicklerteam* u.a. durch folgende Aktivitäten zur Seite.

1. Erkennen, Entdecken, Sammeln und Beschreiben von Anforderungen, um zusammen mit dem *Entwicklerteam*, ein gemeinsames Verständnis über die Anforderungsumsetzungen, die Probleme und deren Lösungen zu entwickeln.
2. Priorisierung der Anforderungen zur Optimierung der (zu einem bestimmten Zeitpunkt) auszuliefernden Software-Applikation. Dabei ist häufig eine Konsensfindung zusammen mit dem *Entwicklerteam* und im Zweifel mit den *Stakeholdern* notwendig.
3. Beurteilung, ob alle (bis zu einem bestimmten Zeitpunkt) eingeplanten Anforderungen zufriedenstellen abgedeckt wurden und korrekt ausgeliefert wurden. Die Ergebnisse dieser Beurteilungen werden, wenn nötig und wenn eingeplant, in die nächste Arbeitsphase des *Product Teams* mitgenommen.

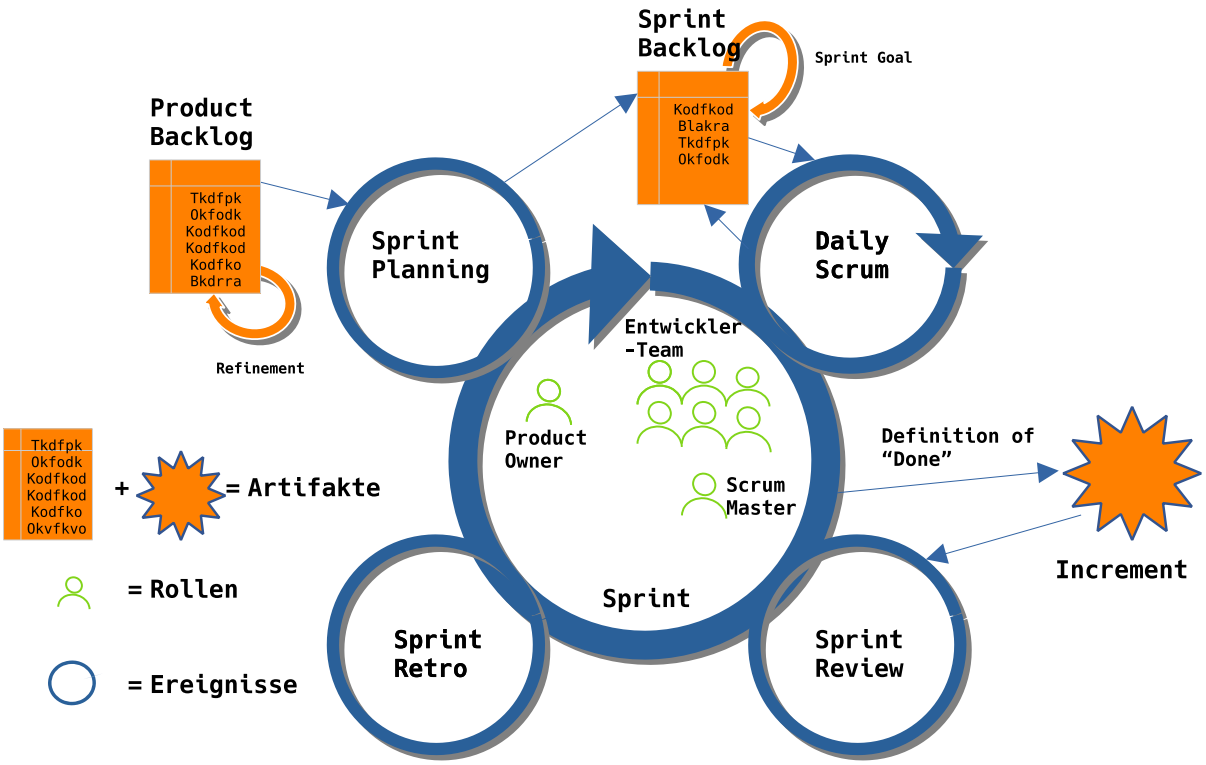

Figure 2. Agiler, Scrum orientierter Entwicklungsprozess aus der Sicht des Product Teams

Product Backlog

Der Fokus dieser Dokumentation ist auf die Rolle des *Product Owner* gerichtet ist. Deshalb wenden wir unseren Blick zuerst auf ein *Artifakt* , welches eine zentrale Rolle für die Arbeit des *Product Owner*s spielt. Den sogenannten *Product Backlog*.

Artifakt (Scrum)

Die im folgenden vorgestellte Softwareentwicklungsmethodologie ist stark an die agile Scrum Methode angelehnt. Bei Scrum ist ein *Artifakt*, eine Art Prozessdokumentation. Zu den Artifakten gehören mindestens der *Product Backlog*, der *Sprint Backlog* und das *Increment*.

Der *Product Backlog* ist im Prinzip eine Liste von Anforderungen. Der *Product Owner* ist zuständig für den *Product Backlog*, pflegt ihn und entwickelt ihn ständig weiter.

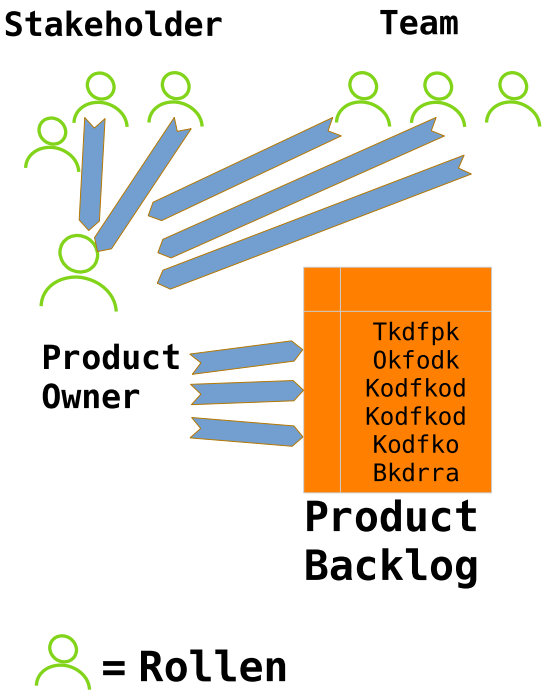

Figure 3. Product Backlog als Schnittstelle zwischen den Rollen

Zur Erfüllung dieser Zuständigkeit braucht der *Product Owner* regelmäßigen Input sowohl vom *Entwicklerteam*, als auch von den *Stakeholdern*. Die einzelnen Einträge des *Product Backlog* werden *Product Backlog Item (PBI)* genannt. Welche formalen Regeln zu beachten sind, um *PBI*s zu erstellen, soll **zunächst nicht Bestandteil dieser Dokumentation** sein, muss aber prinzipiell geklärt sein. Ein *Product Backlog* entwickelt sich nach und nach. Seine Vollständigkeit ist keine Zielvorgabe, da man von einer dynamischen Entwicklung der Anforderungen während der gesamten Projektdauer ausgeht. Die Einträge (*PBI*s) des *Product Backlog* müssen regelmäßig unter Berücksichtigung von Faktoren wie Wert, Risiko, Machbarkeit und Abhängkeit priorisiert werden. Der *Product Owner* ist neben der Priorisierung gefordert bei der Beschreibung der einzelnen *Product Backlog Items (PBI)*. Er muss versuchen sicherzustellen, dass sowohl die *Stakeholdern* als auch das *Entwicklerteam* das gleiche Verständnis über die *Product_Backlog Items (PBI)*s entwickeln. Der *Product Backlog* wird während der gesamten Projektlaufzeit gepflegt. Die *Product_Backlog Items (PBI)* werden Schritt für Schritt in *Sprints* bearbeitet.

Hier ist ein unvollständiges Beispiel eines *Product Backlog* unter der Benutzung von sogenannten *User Stories* zur Beschreibung der gewünschten Funtionalitäten und Features.

Card #	Title	Description	SP

2	Profile grid visual representation as a component with one or two graphs and aditional user interface components like amplification level, aquisition time and so on.		XL (20)
3	Profile grid data collector passes data to area graph to visualize profile grid.		XL (20)
9	Profile Grids in main window should be ordered in particle flight direction, e.g. from bottom to top.		S (2)
8	User can choose a PG belonging to a context (chain, particle transfer, virtual accelerator).	Test: Check if PG list looks feasible and selected PG is visualized.	M (5)
6	User can set amplification level of a choosen PG to improve measurement and visualization of PG data graph.	Test: Switch amplification level in GUI and check if PG graph is changing accordingly	M (5)
7	User can choose existing context or virtual accelerator for a specific PG to measure a particular beam.	Test: Switch contexti GUI and check if PG graph is changing accordingly	S (2)
11	X-Axis of the visualized PGs must optionally scale accordingly to visualized PG with the largest X dimensions.	X-Axis of the visualized PGs must optionally scale accordingly to the visualized PG with the largest X dimensions. Same is true for Y-axis. Let the user easily compare profile of various PGs.	M (5)

Sprint

Sprint Backlog

Sprint Planning, Sprint Review