Gegenstand: SYP(PRE)

Themenpool: 2 Entwicklung von Softwaresystemen

Referatsthema: 2 SCRUM

Klasse: 5AHIFS

Schuljahr: 2018/19

Professor/in: Prof. Müller Stegmüller

Schüler: Kristian Rajic, Gabriel Webhofer

Inhaltsverzeichnis

[1) Prozessmodelle 3](#_Toc530580347)

[1.1) Agile Prozessmodelle 3](#_Toc530580348)

[1.2) Klassische Prozessmodelle 3](#_Toc530580349)

[2) Entstehung von Scrum 4](#_Toc530580350)

[3) Scrum Einführung 4](#_Toc530580353)

[4) Rollen 4](#_Toc530580354)

[4.1) Entwicklungsteam 4](#_Toc530580355)

[4.2) Product Owner 5](#_Toc530580356)

[4.3) Scrum Master 5](#_Toc530580357)

[5) Meetings 5](#_Toc530580358)

[5.1) Sprint Planning Meeting 1 – Anforderungen klären 5](#_Toc530580359)

[5.2) Sprint Planning Meeting 2 – Design und Planung 5](#_Toc530580360)

[5.3) Daily Scrum 5](#_Toc530580361)

[5.4) Estimation Meeting 6](#_Toc530580362)

[5.5) Sprint Review 6](#_Toc530580364)

[5.6) Sprint Retrospektive 6](#_Toc530580365)

[6) Artefakte 6](#_Toc530580366)

[6.1) Produktvision 6](#_Toc530580367)

[6.2) Product Backlog Item 6](#_Toc530580368)

[6.3) Product Backlog 7](#_Toc530580369)

[6.4) Sprint Goal 7](#_Toc530580370)

[6.5) Arbeitspakete (Tasks) 7](#_Toc530580371)

[6.6) Releaseplan 7](#_Toc530580372)

[6.7) Impediment Backlog 7](#_Toc530580373)

[6.8) Produktinkrement 7](#_Toc530580374)

[7) Scrum Struktur 7](#_Toc530580375)

[7.1) Begriffserklärung 7](#_Toc530580377)

[7.1.1) Sprint-Backlog 7](#_Toc530580378)

[7.1.2) In-Progress 8](#_Toc530580380)

[7.1.3) To-Verify 8](#_Toc530580382)

[7.1.4) Done 8](#_Toc530580384)

[7.2) Ablauf: 8](#_Toc530580386)

[8) Vorteile & Nachteile 8](#_Toc530580391)

[9) Beispiel 9](#_Toc530580392)

[10) Eigene Erfahrungen 9](#_Toc530580405)

[11) Quellen 10](#_Toc530580406)

# 1) Prozessmodelle

In der Software Entwicklung unterscheidet man zwischen zwei Arten von Prozessmodellen.

## 1.1) Agile Prozessmodelle

Agile Methoden bündeln Entwicklungsmaßnahmen, die mit leichtgewichtigen Vorgehensweisen (Lightweight Methods) eine höhere Flexibilität versprechen.

Zentrale Ideen sind:

* ein frühes Einsteigen in die Codierung
* starke Einbeziehung der Nutzer
* beständiges Testen
* die Weiterentwicklung der Architektur.

Diese Werte wurden im "Agilen Manifest“ niedergeschrieben. Sie beschreiben die grundlegende Philosophie der agilen Entwicklung, und zwar:

* Individuals and interactions over processes and tools
* Working software over comprehensive documentation
* Customer collaboration over contract negotiation
* Responding to change over following a plan

Ergänzt werden diese Werte durch sogenannte Prinzipien, wie Zweckmäßigkeit, Kundennähe oder dass der im Projekt erstellte Code allen Teammitgliedern "gehört" (im Sinne einer offen zugänglichen Codebasis für das Team).

## 1.2) Klassische Prozessmodelle

Klassische Methoden der Softwareentwicklung (Heavyweight Methods) übertragen ingenieurwissenschaftliche Ansätze der Projektdurchführung auf Softwareprojekte. Wie bei den „klassischen“ Ingenieurwissenschaften, wird hier viel Wert auf den Prozess gelegt. Dies bedeutet, dass die Planungsphase ausführlich ausfällt und allen Phasen eine detaillierte Dokumentation folgt, die wiederum als Eingabe für die darauffolgende Phase dient. Dieser Ansatz entspricht den „klassischen“ Software-Engineering.

# 2) Entstehung von Scrum

# Scrum wurde in der Mitte der 1990er Jahre von den US-amerikanischen Software-Entwicklern Jeff Sutherland und Ken Schwaber entwickelt. Da zu dieser Zeit rund 80 Prozent der produzierten Software laut der CHAOS-Studie scheiterten, sahen Sutherland und Schwaber die Lösung in einer Flexibilisierung der Projektprozesse.

# Scrum ist eine Adaption der sogenannten Lean Production, welche in der zweiten Hälfte des 20. Jahrhunderts vom Automobilhersteller Toyota vergelebt wurde.Der Kerngedanke besteht hier wie dort in der „Vermeidung von Verschwendung" bei konsequenter Nutzung impliziten Mitarbeiter-Know-hows.

# 3) Scrum Einführung

Das Prozessmodell von Scrum besteht aus *Rollen, Meetings und Artefakten* und steckt den Rahmen ab, in dem alle Aktivitäten der Produktentwicklung ablaufen.

Am Anfang der *Discovery-Phase* steht die Person mit der *Produktidee*. Häufig ist das ein Kunde oder eine andere Person, die einen Auftrag für eine Produktentwicklung geben kann. Diese Person bespricht ihre Idee mit dem *Product Owner*, der die Idee, oft auch direkt mit dem *Entwicklungsteam*, so lange bearbeitet, bis eine *Produktvision* entstanden ist. Die Produktvision enthält die grundlegende Idee für das Produkt. Der Product Owner erarbeitet, entweder alleine oder mithilfe der Teammitglieder, die Produktfunktionalitäten, *Product Backlog Items* oder auch *User Stories* genannt. Alle User Stories werden in eine Liste eingetragen, das *Product Backlog*. Der Product Owner bringt die User Stories in dieser Liste in eine Reihenfolge. Der gesamte Prozess wird vom *ScrumMaster* gemanagt. Die drei Rollen Product Owner, ScrumMaster und Entwicklungsteam agieren gemeinsam als das *Scrum-Team*.

# 4) Rollen

## 4.1) Entwicklungsteam

Das Entwicklungsteam arbeitet so dicht wie möglich am User und designt und entwickelt das Produkt. Ein Entwicklungsteam besteht aus den Personen, die für die Lieferung des Produkts erforderlich sind, diese können Tester, Frontend -oder Backend Entwickler sein. Diese sogenannten crossfunktionalen und multidisziplinären Teams sind so zusammengestellt, dass sie ohne den Einfluss der Organisation das Produkt liefern können. Ein Entwicklungsteam managt seine Angelegenheiten selbst. Es ist autorisiert, alles Zielführende zu tun, um das angestrebte Ergebnis zu erreichen. Gleichzeitig sollte es die Standards und Prozesse der Organisation einhalten. Das Entwicklungsteam steuert selbst die Arbeitsmenge, die es bewältigen kann. Dafür trägt es die Verantwortung für die Qualität der Lieferung und arbeitet so professionell wie möglich. Es ist immer am neuesten Stand der jeweiligen Technologie und liefert State of the Art.

## 4.2) Product Owner

Der Product Owner plant und lenkt die Produktentwicklung. Er steuert das Team, indem er eine deutliche Vision vorgibt, klare und eindeutige Rahmenbedingungen definiert und festhält. Er ist verantwortlich dafür, dass das Produkt den gewünschten Return on Investment bringt, indem er vorgibt, in welcher Reihenfolge das Entwicklungsteam die gewünschten Funktionalitäten liefern soll. Er sorgt dafür, dass die Projektergebnisse den finanziellen Aufwand für das Projekt rechtfertigen. Dazu arbeitet er täglich mit dem Entwicklungsteam, trifft zeitnah die notwendigen Entscheidungen und arbeitet kontinuierlich am Product Backlog und dem Releaseplan.

## 4.3) Scrum Master

Der ScrumMaster führt das Scrum-Team. Er ist dafür verantwortlich, dass die Produktivität des Scrum-Teams kontinuierlich steigt und arbeitet daran, dass alle Schwierigkeiten, Blockaden und Probleme, die das Team aufhalten, gelöst werden. Zwar ist der ScrumMaster nicht weisungsbefugt, er sorgt aber dafür, dass der Scrum-Prozess eingehalten wird. Seine Hauptaufgabe besteht darin, alle am Projekt beteiligten Personen so zu schulen, dass sie ihre Rolle verstehen und ausüben können.

# 5) Meetings

## 5.1) Sprint Planning Meeting 1 – Anforderungen klären

In diesem Meeting eines Sprints sind der Product Owner, das Entwicklungsteam, das Management, der Anwender und der ScrumMaster anwesend. Der Product Owner erläutert die Product Backlog Items und definiert gemeinsam mit den Teammitgliedern und dem Management das Ziel für den anstehenden Sprint. Dann werden die Backlog Items ausgewählt, die zu diesem Ziel passen und die das Team schaffen kann. So entsteht das Selected Product Backlog (Sprint Backlog). Das Entwicklungsteam alleine bestimmt, wie viele Backlog Items es auswählt.

## 5.2) Sprint Planning Meeting 2 – Design und Planung

Im Sprint Planning Meeting 2 plant das Entwicklungsteam, wie es das im Sprint Planning Meeting 1 vereinbarte Ziel erreichen will. Dazu beraten die Teammitglieder, wie die Applikation aufgebaut sein soll, welche Architektur gewählt werden muss, welche Interfaces geschrieben werden sollen, ob bereits Test Cases erstellt und geschrieben werden sollen, kurz: Sie besprechen detailliert, was getan werden muss. Am Ende dieses Meetings liegt eine Liste aller notwendigen Arbeitspakete(Tasks) vor: das Sprint Backlog.

## 5.3) Daily Scrum

Jeden Tag treffen sich die Mitglieder des Entwicklungsteams zur gleichen Zeit am selben Ort für 15 Minuten zu einem vom ScrumMaster moderierten Meeting. In diesem Meeting sprechen sie ab, wer an diesem Tag welche Aufgaben übernimmt. Dabei wählen die Teammitglieder selbst die Aufgabe aus, die sie als Nächstes übernehmen wollen. Sie informieren den ScrumMaster über Blockaden und Probleme, damit dieser sie so schnell wie möglich zur Lösung bringen kann.

## 5.4) Estimation Meeting

Der Product Owner und die Mitglieder des Entwicklungsteam aktualisieren mindestens einmal im Sprint das Product Backlog. Dabei werden User Stories mit neuen Schätzungen versehen und neue Backlog Items in das Product Backlog aufgenommen. Gleichzeitig wird die Reihenfolge der Backlog Items unter Berücksichtigung der neuen Informationen angepasst. Dieses Meeting dient auch dazu, den Releaseplan des Projekts zu aktualisieren und zu vervollständigen.

### 5.4.1) Estimation (Berechnung)

Bei der Estimation bewertet das Entwicklungsteam jede anstehende Story im Sprint. Die Bewertung hängt dann von der Schwierigkeit der Story ab. Wie schwer eine Story ist, entscheidet das Team.

Bei der Bewertung werden Zahlen aus der Fibonacci-Reihe benutzt. Die Zahl wird Story-Points genannt. Die Summe aller Story-points ergibt dann die Gesamtschwierigkeit des Sprints.

Am Ende des Sprints sieht man, wie viele Story-Points man wirklich erreicht hat. Dies zeigt wie gut man die Schwierigkeit abgeschätzt hat und man gewinnt an Erfahrung für das nächste Mal.

## 5.5) Sprint Review

Am Ende des Sprints präsentiert das Entwicklungsteam die erarbeiteten Funktionalitäten. Das Team zeigt nur die Tasks, die so weit sind, dass sie sofort produktiv gesetzt werden könnten. Nicht getestete oder instabile Arbeitspakete werden nicht angezeigt und gelten als nicht geliefert.

## 5.6) Sprint Retrospektive

Die Sprint Retrospektive ermöglicht dem Team das systematische Lernen. Hier wird analysiert, welche Arbeitsprozesse verbessert werden müssen, damit das Team effektiver arbeiten kann. Die Resultate aus der Retrospektive werden im Impediment Backlog festgehalten und lassen sich so als Verbesserungsvorschläge in das Sprint Planning einbringen.

# 6) Artefakte

## 6.1) Produktvision

Eine Vision ist ein klares, deutliches, attraktives und emotionalisierendes Bild des angestrebten Produkts.

## 6.2) Product Backlog Item

In Scrum werden die zu liefernden Funktionalitäten, die in einem Product Backlog aufgelistet sind, als Product Backlog Items bezeichnet. Diese können sein User Stories, Modellierungsaufgaben, Dokumentationsaufgaben, Organisationsaufgaben, Konfigurationsaufgaben, Bugfixing-Aufgaben

## 6.3) Product Backlog

Das Product Backlog ist eine Liste von Backlog Items. Alle Items sind in diesem Backlog nach ihrer Wichtigkeit sortiert.

## 6.4) Sprint Goal

Der Product Owner legt gemeinsam mit dem Entwicklungsteam das Sprint Goal, das Ziel des Sprints fest.

## 6.5) Arbeitspakete (Tasks)

Mit Tasks bezeichnen wir alles, was getan werden muss, um das Ziel des Sprints zu erreichen und die geforderten Funktionalitäten zu entwickeln.

## 6.6) Releaseplan

Der Release Plan zeigt an, in welchem Sprint welche Backlog Items vom Entwicklungsteam geliefert werden können. Er is kein Instrument, um einem Team zu sagen, was es wann zu liefern hat, sondern er zeigt nur an, wann das entsprechende Backlog Item vom Team zu erwarten ist. Er ist also Informationsinstrument und kein Planungsinstrument im klassischen Sinn.

## 6.7) Impediment Backlog

Das Impediment Backlog ist die Liste aller Hindernisse, die einem Team aus dem Weg geräumt werden müssen, damit es produktiver werden kann.

## 6.8) Produktinkrement

Ein Entwicklungsteam strebt an, am Ende eines Sprints etwas zu liefern, das man präsentieren kann. Die Funktionalitäten, die am Ende des Sprints präsentiert werden, müssen tatsächlich in einem Zustand sein, in dem sie ausgeliefert und verwendet werden können.

# 7) Scrum Struktur

In der Praxis besitzt jedes Entwicklungsteam ein Scrumboard. Dies ist eine Pinwand oder eine Tafel, die in verschiedene Sektionen aufgeteilt sind. Üblicherweise sind diese Bereiche folgende: Sprint-Backlog (ToDo), In-Pogress, To-Verify und Done.

## 7.1) Begriffserklärung

### 7.1.1) Sprint-Backlog

Das sind jene Backlog Items mit all ihren geplanten Tasks, welche im Sprint-Planning-Meeting für den derzeitigen Sprint ausgewählt wurden.

### 7.1.2) In-Progress

In diesem Bereich sind jene Tasks, welche schon von einem Mitglied des Entwicklungsteams bearbeitet werden.

### 7.1.3) To-Verify

In diesem Bereich befinden sich alle Tasks, welche schon fertig entwickelt wurden und darauf warten, von einem Product-Owner akzeptiert zu werden.

### 7.1.4) Done

Das sind alle Backlog Items, welche fertig entwickelt wurden und auch schon vom Product-Owner akzeptiert wurden.

#### **Definition of Done**

Die Definition of Done (DoD) ist eine Liste von Fertigstellungskriterien, die das Development-Team zur Erstellung des Produktes zu beachten hat. Meist sind damit Wünsche des Product Owners an Qualität, Skalierbarkeit etc. verbunden.

## 7.2) Ablauf:

Nachdem die Sprint–Backlog-Items für den jeweiligen Sprint ausgewählt wurden und auf dem Scrumboard hängen, werden einzelne User Stories an Mitglieder im Team vergeben bzw. Mitglieder suchen sich die User Stories aus welche sie bearbeiten.

Nachdem eine UserStory einem Mitglied zugewiesen ist, kommt diese in den ,,In-Progress'’-Bereich. Das bedeutet, dass das Mitglied angefangen hat, das Problem der UserStory zu bearbeiten.

Nachdem das Mitglied mit der Entwicklung fertig ist, kommt die UserStory in den Bereich ,,To-Verify'’. Hier wird dann darauf gewartet, dass ein Product-Owner sich die abgeschlossene UserStory ansieht, und beurteilt ob diese so verkauft werden kann. Falls ja, kommt diese in den ,,Done’’-Bereich. Falls der Product-Owner nicht überzeugt ist, überarbeitet der Entwickler seine Lösung.

User Stories, welche sich nach dem Ende des Sprints noch im Backlog stehen, bleiben entweder bestehen und werden in den nächsten Sprint übertragen, oder sie werden an eine andere User Story angehängt.

# 8) Vorteile & Nachteile

Vorteile von Scrum

* Schnelle Einführung da es wenig Regeln gibt und es leicht verständlich ist
* Hohe Flexibilität
* Hohe Effektivität durch Selbstorganisation
* Hohe Transparenz durch regelmäßige Meetings und Backlogs
* Der Prozess wird dauerhaft verbessert durch die Erfahrungen
* Geringer Administrations- und Dokumentationsaufwand
* Stufenweise Einführung ist möglich

Nachteile von Scrum

* Kein Gesamtüberblick über die komplette Projektstrecke
* „Tunnelblick-Gefahr" bei ausschließlicher Fokussierung auf Tasks
* Potenzielle Verunsicherung aufgrund fehlender Zuständigkeiten und Hierarchien
* Potenzielle Unvereinbarkeit mit bestehenden Unternehmensstrukturen
* Keine direkte Kommunikation mit dem Auftraggeber

# 9) Beispiel

Szenario:

Eine Programmierfirme hat ein Jahr zuvor den Auftrag bekommen, einen Webshop für Kleidung zu entwerfen. Das Team besteht aus 4 Personen, wobei es 2 Backendentwickler, 1 Frotendentwickler und 1 Tester gibt. Ein Sprint dauert 2 Wochen.

Im SP wird geplant, dass in diesem Sprint die Kauf-Funktion entwickelt wird. Aufgrund der Anfrage des Users, entstehen folgende User Stories:

* Der Warenkorb kann eingesehen werden
* Die Produkte können gelöscht werden.
* Die Quantität kann verändert werden.
* Die Bezahlmethode kann ausgewählt werden, welche den Kauf abschließt
  + Nachdem auf den Kaufen-Button gedrückt wurde, wird man zur jeweiligen Bezahlseite weitergeleitet

Im Estimation-Meeting einigt das Team sich, dass die User-Story eine Schwierigkeit von 5 bekommt, da sie der Meinung sind, dass die verschiedenen Bezahlmethoden Schwierigkeiten aufweisen können.

Das Team programmiert die Kauf-Funktion aus und der Product-Owner wird gerufen, um sich das Ergebnis anzusehen. Wenn dieser mit dem Ergebnis zufrieden ist, nimmt er das Backlog-Item ab und es gilt als Done.

Das Team stellt die Funktionen, welche in dem Sprint fertiggestellt wurden, im Sprint-Review vor.

Am Ende des Sprints wird bei der Retospektive darüber geredet, was bei dem Sprint gut bzw. schlecht gelaufen ist und welche Erfahrungen man in zukünftige Sprints mitnehmen soll.

# 10) Eigene Erfahrungen

Scrum wurde bei uns in einem Projekt-Team aus 8 Personen durchgeführt. Dabei wurde am Anfang eines Sprints in den ersten 2 Tagen das SP1 & 2 und das Estimationmeeting durchgeführt.

Im Büro des Teams stand ein Scrumboard mit den einzelnen User Stories. Vor diesem wurde der Daily-Scrum ausgeführt, welche in diesem Fall aus 3 Teilen bestand. Als erstes hat jedes Teammitglied erzählt, was am Vortag gemacht wurde. Im zweiten Teil präsentierte ein Mitglied, welches an dem sogenannten Green-CI-Meeting teilgenommen hat, allen anderen, welche Punkte beim Meeting besprochen wurden, die für das Team wichtig sind und im dritten Teil wurde festgelegt, welches Mitglied, welchen Task erledigt.

Am Ende des 2-wöchigen Sprints kam es dann zum Review und der Retrospektive.

Meiner Meinung nach ist die Verwendung von Scrum sehr gut, da man mit einem Blick auf das Board sieht, welches Mitglied was macht. Zudem haben die Mitglieder den Vorteil, dass sie am Anfang des Tages selbst festlegen, was zu machen ist. Das bringt zum einen den Vorteil, dass man genau weiß was man zu tun hat, aber auch dass man selber einschätzen kann, wieviel man an einem Tag schafft.

Zudem sieht jedes Teammitglied beim Daily Scrum, ob ein anderer in eine Zwickmühle sitzt und kann dann sofort reagieren und diesem helfen. Dadurch wird vermeidet, dass sich ein Problem über einen längeren Zeitraum hindurchzieht.

# 11) Quellen

# Scrum: Produkte zuverlässig und schnell entwickeln (Buch)

# <http://www.codesco.com/fileadmin/bilder_grafiken/Sonstiges/KlassischVSagil_Ausarbeitung.pdf>

# <http://www.enzyklopaedie-der-wirtschaftsinformatik.de/lexikon/is-management/Systementwicklung/Vorgehensmodell/Agile-Vorgehensmodelle/index.html>

# <https://www.scrum-events.de/was-ist-die-definition-of-done-dod.html>

# <http://agiles-projektmanagement.org/scrum-geschichte/>