Scrum

## Einleitung

Ist ein Rahmenwerk zur Entwicklung komplexer Produkte.

Scrum baut auf hoch qualifizierten, interdisziplinär besetzten Entwicklungsteams auf, die zwar eine klare Zielvorgabe bekommen, für die Umsetzung jedoch allein zuständig sind. Dadurch bekommen die Entwicklungsteams den nötigen Freiraum, um ihr Wissens- und Kreativitätspotenzial in Eigenregie zur Entfaltung zu bringen

Scrum verkörpert Werte des Agilen Manifests:

1. *Individuen* und *Interaktionen* gelten mehr als *Prozesse* und *Tools*.
2. *Funktionierende Programme* gelten mehr als *ausführliche Dokumentation*.
3. Die stetige *Zusammenarbeit mit dem Kunden* steht über *Verträgen*.
4. Der Mut und die Offenheit für *Änderungen* steht über dem *Befolgen* eines festgelegten *Plans*.

empirischer (Experimente, Beobachtung oder Befragung) Ansatz: die meisten Entwicklungsprojekte sind zu komplex, um durchgängig planbar zu sein

3 Prinzipien zur Reduzierung der Komplexität

1. Transparenz: Fortschritt und Hindernisse eines Projektes täglich und sichtbar festhalten
2. Überprüfung: Produktfunktionalitäten regelmässig liefern und prüfen
3. Anpassung: Anforderungen an Produkt nicht zu Beginn schon festgelegt, bei Bedarf anpassbar

## Rollen

3 Rollen mit Verantwortungen

* Scrum-Team (Product Owner, Entwicklungsteam, ScrumMaster)
* Externe Rollen(Customer, User, Management)

### Product Owner

|  |  |
| --- | --- |
| Verantwortlichkeit | Strategische Produktentwicklung:   * Vision * Festlegung + Priorisierung Produkteigenschaften dazu User Stories, welche eine Priorisierung enthalten, in Product Backlog eintragen * Entscheid ob Produktfunktionalität akzeptabel |
| Ziel | wirtschaftlicher Nutzen für eigenes Unternehmens maximieren |
| Weiteres | * hält Rücksprache mit Kunde (kennt dessen Bedürfnisse, Wünsche) * bündelt Interessen und Anforderungen im eigenen Unternehmen |
| Probleme | Bei der Implementation von Scrum passiert es häufig, dass Product Owner nicht bevollmächtigt sind, um die notwendigen Entscheidungen verbindlich zu treffen. Ein weiteres Problem aus der Praxiserfahrung ist die Überlastung der Product Owner mit fremden Aufgaben |

### Entwicklungsteam

|  |  |
| --- | --- |
| Verantwortlichkeit | * Produktfunktionalitäten (Anzahl Funktionalitäten pro Sprint wählbar) * Einhaltung Qualitätsstandard |
| Zusammensetzung | 7±2 Personen, interdisziplinäre Besetzung (Teammitglieder aus allen Bereichen), Entwicklungsteam ist eine Einheit |
| Weiteres | * User Stories: Umfang schätzen, in Arbeitsschritte aufteilen * Wichtig: das Team organisiert sich selbst, keine Vorgaben von Product Owner oder ScrumMaster |
| Probleme | In der Implementierungsphase haben Team-Mitglieder bisweilen Schwierigkeiten, die interdisziplinären Anforderungen zu akzeptieren. So kann z.B. ein Entwickler nicht verstehen, warum er nun auch noch die Arbeit eines Testers leisten soll. Hinter diesen Anforderungen steht jedoch der Gedanke, dass ein starkes Entwicklungsteam den mannigfachen Unwägbarkeiten eines Projektes wesentlich besser gewachsen ist als eine Sammlung individueller Talente. Falls beispielsweise jemand mit einer Aufgabe nicht zurechtkommt, kann ein anderer aushelfen und so die Einhaltung des Sprint-Ziels gewährleisten. Und fällt jemand aus privaten Gründe für einige Zeit aus, ist ein interdisziplinär aufgestelltes Entwicklungsteam besser in der Lage, die fehlende Expertise zu kompensieren. |

### ScrumMaster

|  |  |
| --- | --- |
| Verantwortlichkeit | * Scrum muss gelingen, enge Zusammenarbeit mit Entwicklungsteam * Scrum-Regeln und deren Einhaltung * Moderation Meetings, Überwacht Team, entscheidet was zumutbar ist * Lösen von Problemen (intern und von aussen) |
| Position | Führungskraft aber nicht Vorgesetzter |

### Customer

Auftraggeber

Soll bereits nach den ersten Sprints Gelegenheit haben, sich die neuen Funktionalitäten anzuschauen und Feedback zu geben (gewünschte Produktfunktionalitäten dürfen sich während Entwicklungsphase ändern)

### User

Anwender

Kann zugleich der Kunde sein

Sollte beim Planen des 1. Sprints sowie bei den Sprint Reviews dabei sein, Produkt ausprobieren, Feedback geben

### Management

Stellt materielle Ressourcen bereit, bietet generelle Unterstützung, unterstützt ScrumMaster in seiner Arbeit als Problemlöser

## Ereignisse

### Sprint Planungs-Meetings

Meeting 1 und 2 sollen am gleichen Tag stattfinden

#### Sprint Planung 1

Fokus: Was

User Stories werden vorgestellt, Entwicklungsteam kann sich klares Bild verschaffen, Anforderungen werden geklärt, Kriterien zur Abnahme der zu entwickelnden Funktionalität(en)

Entwicklungsteam setzt fest, wie viele (der Reihen nach) User Stories es umsetzen können wird

Anwesende: Product Owner, Entwicklungsteam, ScrumMaster, User

Ziel: usable software, hinreichend getestet und integriert, für Benutzer freigegeben

#### Sprint Planung 2

Fokus: Wie

Entwicklungsteam organisiert/plant technische Umsetzung

Ergebnis: Tasks/Aufgaben

### Sprint

Zu Beginn jedes Sprints finden die Meetings zu Sprint Planung 1 und 2 statt.

Am Ende des Sprints wird eine Produktfunktionalität geliefert, nach Plan. Es folgt ein Sprint Review.

Es wird eine User Story nach der anderen (in der Reihenfolge der Priorisierung) im Team abgearbeitet. Man geht erst zur nächsten Funktionalität über, wenn die vorige ausreichend getestet wurde, damit sie vom Product Owner abgenommen werden wird.

Die Dauer von Sprints sollte immer gleich lang (1-4 Wochen) sein und darf nicht verlängert werden.

Ist das Ziel nicht zu erreichen, so kann ein Sprint abgebrochen werden. Nun folgt eine Retrospektive.

Daily Scrum

Zu Beginn des Arbeitstages 15 min Informationsaustausch, Überblick über aktuellen Stand der Arbeiten.

Dauert die Erledigung eines Tasks länger als einen Tag, so soll dieser heruntergebrochen werden in Untertasks

Alle Aktivitäten sind timeboxed und dienen zur Vorbereitung der jeweils nächsten Aktivität

Ziel: schnelle, kostengünstige, qualitativ hochwertige Fertigstellung eines Produktes, entspricht der Vision

Umsetzung: in max. 4 Wochen langen Sprints (sich wiederholende Intervalle)

Am Ende eines jeden Sprints: Lieferung einer fertigen Software-Funktionalität (useable software)

„Scrum akzeptiert, dass der Entwicklungsprozess nicht vorherzusehen ist