



Was ist Scrum?

Scrum ist ein einfaches Projektmanagement-Framework, in das Entwicklungsteams selbstbestimmt erprobte Praktiken einbetten.

Der Rahmen sieht einen empirisch, iterativen Prozess vor, bei dem in Zeitabschnitten von je 30 Tagen ein nächster vollständig fertiggestellter Produktstand dem Kunden zur Inspektion bereitgestellt wird. Termine sind nicht verschiebbar, sondern die Menge an Features, die in einer Zeitbox vollständig als „fertig“ vermerkt werden können, variiert. Da Zeitabschnitte, sogenannte Sprints, immer gleich lang sind, entwickelt das Team über die Zeit ein Gefühl und Statistiken darüber, wie viele Arbeitseinheiten es in einem Sprint erledigen kann. Damit wird die Planung in einem Team immer verlässlicher.

Scrum verfolgt einen evolutionären Ansatz. Vor jedem Sprint werden neue Prioritäten gesetzt und das Gelernte miteinbezogen. Das schafft Flexibilität in einer sich schnell wandelnden globalen Welt. Es ermöglicht dem Kunden den größten möglichen Wert, für seine Investitionen, zu bekommen.

Sich selbstorganisierende Teams entwickeln die höchste Produktivität, in der die Summe aller Teile mehr ergibt, als jeder Einzelne. Scrum stellt Prozesse und Methoden zur Verfügung, um Selbstorganisation zu organisieren.

Werte und Prinzipien der agilen Softwareentwicklung

Scrum setzt das Konzept der agilen Softwareentwicklung in einen konkreten Vorschlag für das Projektmanagement um. Das menschliche Potenzial jedes Einzelnen wird mit Selbst-Organisation, unbürokratischem Vorgehen, Disziplin und Wertschätzung am intensivsten genutzt. Transparenz und Flexibilität zeichnen den Prozess aus.

Der Wertzuwachs, des Produktes für den Kunden, steht im Mittelpunkt. Dies erfordert einen Lernprozess während der Entwicklung. Neue Erkenntnisse fließen jederzeit ein. Es steht nur fest, was produziert wird und nicht wie. Dies ist ein radikal anderer Ansatz, als wenn das Endprodukt zu Beginn des Projektes vollständig spezifiziert ist. Um das zu erreichen, arbeitet Scrum mit festen Zeiteinheiten, Sprints, von max. 30 Tagen. Die Menge an Features pro Sprint ist Team abhängig. Prioritäten werden klar gesetzt und Termine immer eingehalten. Dieses Vorgehen hat, bei genauer Betrachtung, den Vorteil, dass Termintreue der Entwickler möglich und klare Prioritäten des Kunden gefordert sind.

Da vor jedem Sprint neu begutachtet und geplant wird, fließen neueste Erkenntnisse in den Prozess ein. Es ist also ein empirisches Vorgehen, das der Evolution gleicht. Damit bekommt der Kunde den größten Wert für sein Geld.

Sprint Planung



Sprint Backlog

SCRUM
täglich

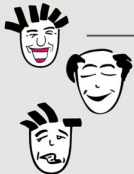
Sprint

30 Tage

Review

Retrospektive

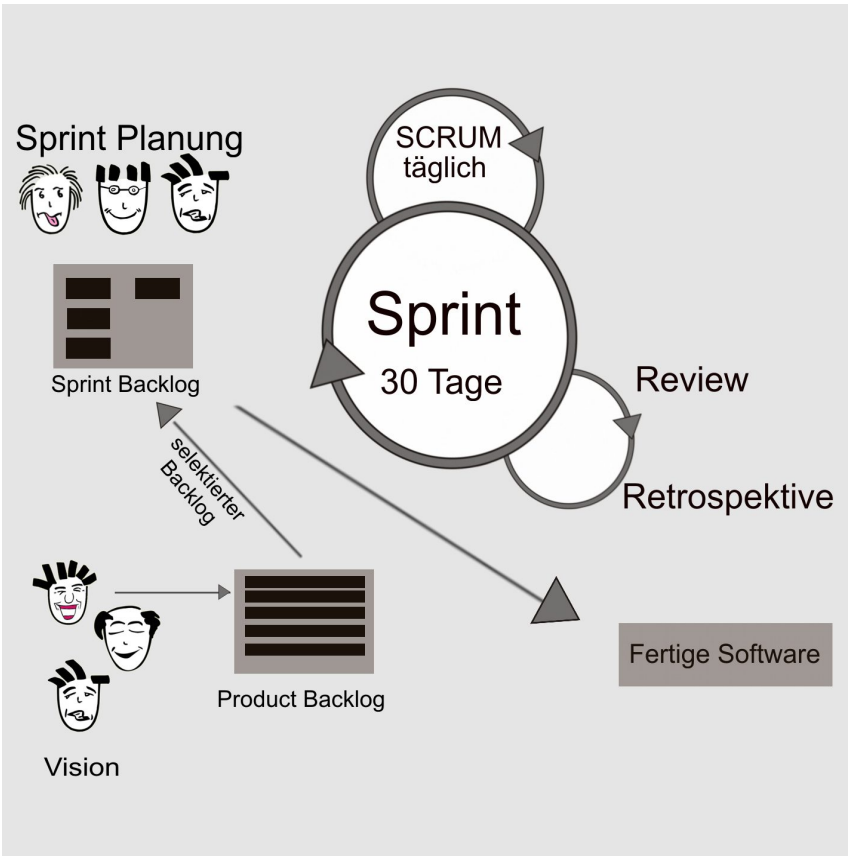
selektierter
Backlog



Product Backlog

Fertige Software

Vision



Scrum Rollen

ScrumMaster - Er dient dem Team als Moderator und sorgt dafür, dass die Scrum Regeln beachtet werden. Zusammen mit dem ProductOwner setzt er sich für die Interessen des Kunden ein und vermittelt zwischen Team und Managern.



TeamMitglied - Dies sind alle Personen die direkt an der Produktion beteiligt sind. Es schließt Programmierer, Softwarearchitekten, Tester, Systemadministratoren und Technische Redakteure ein.



ProductOwner - Es ist der Fachvertreter und zuständig für die Anforderungen des Kunden. Er liefert die Anforderungen für die Tests und sorgt für einen möglichst hohen ROI.



Kunde - Er ist der Auftraggeber und bestimmt das Budget und die Produkthanforderung. Der Kunde gibt seine Wünsche und Anforderungen an den ProductOwner weiter, der als Vermittler zum Team dient.



Benutzer - Dies sind die Personen, die ein Produkt benutzen. Sie sind nur indirekt am Entwicklungsprozess beteiligt und werden vom ProductOwner vertreten.



Daily Scrum

Jeden Tag um dieselbe Zeit trifft sich das Team zu einem Kurzmeeting von maximal 15 Minuten. Hier werden jedem Teammitglied folgende Fragen gestellt:

- Was habe ich seit dem letzten Daily Scrum erreicht?
- Was plane ich bis zum nächsten Daily Scrum?
- Welche Hindernisse gibt es?

In diesem Meeting synchronisiert sich ausschließlich das Team. Es ist nicht dazu gedacht, Probleme im Detail zu diskutieren.

Selbstorganisation

Das Kernkonzept von Scrum ist die Selbstorganisation des Entwicklerteams. Es steht bei allen Planungssitzungen im Mittelpunkt. Der ScrumMaster hat nicht die Führungsrolle eines Projektmanagers, sondern eher die eines Moderators. Relative Schätzungen des Umfangs von Aufgaben werden gemeinsam im Team entschieden. Dazu wird der *Planungspoker* verwendet. Da Scrum nur einen Rahmen aufzeigt, verbessert das Team den Prozess im Detail ständig, um die Produktivität zu steigern.

Product Backlog

Ausgehend von der gemeinsamen Vision für das Produkt erstellt der ProductOwner eine Liste mit allen Anforderungswünschen für das Projekt. Einzelne *User Stories* beschreiben diese Anforderungen aus Sicht des Kunden und in der Sprache des Kunden. Die Liste aller für das Projekt zu erledigen User Stories ergeben den Product Backlog.

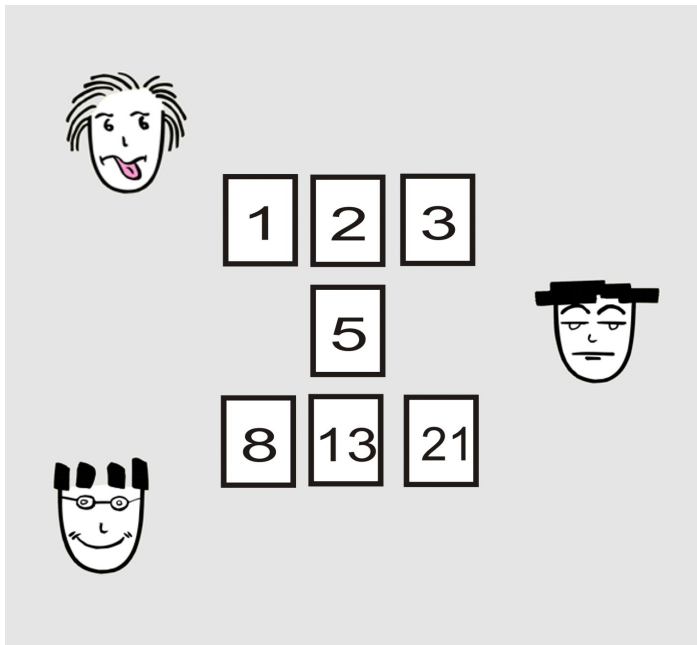
Der ProductOwner ergänzt diesen ständig.

Schätzungs-Meeting

In dieser Sitzung schätzt das Team die relative Größe einer jeden User Story. Hier wird kein Zeitaufwand in Stunden oder Tagen geschätzt, sondern jeder Story eine fiktive Zahl zugeordnet, die seiner Größe, relativ zu einer gewählten Referenzstory, entspricht. Stories, die zu umfangreich sind, werden aufgebrochen. Unklare werden diskutiert.

Damit die Schätzung eines jeden Teammitglieds das gleiche Gewicht erhält, wird ein Planungspoker durchgeführt. Jede Story wird hierzu einzeln betrachtet, eventuell diskutiert und jeder legt eine Karte, mit seiner persönlichen Größeneinschätzung, in Story Punkten, auf den Tisch. Schätzungs-Meetings finden regelmäßig statt und dauern ca. 90 Minuten.

Planungspoker



Release Planung

Auch in Scrum bekommt der Kunde eine Aussage darüber, wann ein Meilenstein erreicht ist. Dies lässt sich an der Anzahl an Sprints ablesen, die notwendig sind, um die Features zu implementieren. Um den Releaseplan einzuhalten, ist es erforderlich, dass für jedes neu hinzukommende Feature ein bestehendes herausgenommen wird.

Sprint Planung

Vor jedem Sprint wird neu geplant und geschätzt. Es wird ein Sprint Ziel definiert. Das Team selektiert so viele Aufgaben, wie in 30 Tagen „fertig“ werden. Unwichtiges fällt weg. Was als fertig gilt, wird gemeinsam definiert. Es gibt 2 Teile:

Sprint Planung 1

Diese Analysesitzung dauert 4 Stunden. Die zu erledigenden User Stories oder Anforderungen werden aus dem priorisierten und geschätzten Product Backlog selektiert. Einzelne Stories werden für das Sprint Backlog verfeinert und zu umsetzbaren Arbeitseinheiten geschnitten.

Um abzuschätzen, wie viele Aufgaben in einem Sprint zu

schaffen sind, wird die relative Größe einer Aufgabe mit dem Erfahrungswert, für die Fließgeschwindigkeit, des Projektes in Story Punkten, multipliziert. Dazu werden zwei Arbeitseinheiten, bei denen das Team relativ sicher weiß, welchen Zeitaufwand sie beanspruchen, gewählt. Division durch die relative Größe der Arbeitseinheiten ergibt den Zeitaufwand pro Story Punkt. Damit ergibt die Summe der Story Punkte aller Arbeitseinheiten, multipliziert mit diesem Wert, den Zeitaufwand für den Sprint.

Sprint Planung 2

Hier geht es um das „wie“. Dies ist ein Designmeeting. Es dauert ca. 4 Stunden und folgt in Anschluss an Sprint Planung 1. Hier bestimmt das Team, wie es die Funktionalität für den Sprint implementiert. Es muss ein klares gemeinsames Verständnis darüber geben, wie umgesetzt wird. Das Team verwendet seine eigenen Tools und Praktiken.

Bei diesem Meeting können auch Werkzeuge, wie UML-Tools verwendet werden. Grundsätzlich gilt hier, dass nur die Aufgaben erledigt werden, die das gesamte Team erfordern. Teilaufgaben die von kleinen Gruppen aus 2-3 Teammitgliedern zu erledigen sind, werden nach dem Meeting besprochen.

Fehler-Backlog

Dieser wird getrennt vom Product Backlog geführt, muss aber beim den Sprint Planungsmeetings berücksichtigt werden. Der Umgang mit Fehlerkorrekturen und deren eventuelle Auslieferung, als Patch, wird in jedem Projekt speziell definiert.

Technischer Backlog

Für Anforderungen, die zur technischen Infrastruktur gehören, wie Serverarchitekturen, Datenbanken oder Entwicklungsumgebungen, muss genau so geplant werden, wie für die Anforderungen des Kunden.

Definition von Done „Fertig“

Jedes Team sollte sich auf eine gemeinsame Definition von Fertig einigen. Welche Kriterien muss eine Arbeitseinheit des Sprint Backlogs erfüllen, um als erledigt zu gelten?

Beispiele sind:

- programmiert, getestet und integriert
- Unit-Test erstellt
- funktional getestet
- keine „technischen“ Schulden
- fehlerfrei lauffähig
- kann ausgeliefert werden

Nicht fertige User Stories kommen zurück in den Product Backlog und werden im nächsten Sprint neu bewertet.



Scrum Best Practices

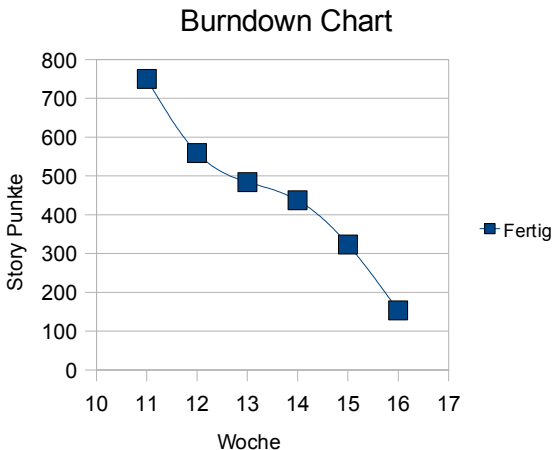
Die Erfahrung aus vielen Scrum Projekten zeigt, dass es sich lohnt, die Spielregeln des Scrum-Frameworks unbedingt einzuhalten.

Folgende „Best Practices“ haben sich bewährt:

- eine klare gemeinsame Vision für das Projekt definieren
- Menschen stehen im Mittelpunkt, nicht Technik
- ein aktueller Product Backlog
- ein definierter Releaseplan
- ein nach Geschäftswert priorisierter Product Backlog
- während des Sprints bleibt das Sprint Backlog stabil
- Team schätzt Größe der User Stories gemeinsam
- täglicher „Daily Scrum“ als Meeting zur Synchronisation
- während des Sprints arbeitet das Team ungestört
- die ausgelieferte Software ist „fertig“
- Sprint Reviews werten den Projektstand aus
- Sprint Retrospektiven optimieren den Prozess
- projektbezogene Burndown-Charts und Metriken

Burndown Charts

Das Burndown Chart zeigt die Fließgeschwindigkeit des Projektes an. Wie viele Story Punkte hat das Team in einem Sprint erledigt?



Sprint Review

Das Ergebnis eines Sprints ist immer ein potenziell auslieferbares Produkt. Das Scrum-Team präsentiert in dieser Revision das lauffähige Produkt den Kunden und Endbenutzern, um Feedback und neue Ideen zu bekommen. Das Product Backlog wird danach ergänzt und verändert. Neue Prioritäten werden hieraus abgeleitet. Da alle 30 Tage ein Review ansteht, ist die Möglichkeit der Korrektur in einem komplexen Umfeld gegeben.

Sprint Retrospektive

Der Prozess an sich wird in Scrum ständig beobachtet und verbessert. Dazu wird das gesamte Team eingeladen.

Was lief gut? Was ist verbesserungsfähig? Das sind hier die zentralen Fragen. Jedes Teammitglied ist aufgefordert, sein Feedback zu liefern. Der ScrumMaster kann hier für einen reibungslosen Ablauf sorgen. Dazu gehört vor allem das Trennen von Fakten und Emotionen.

Software Tools

Wenn das Scrum Team mehrere Sprints erfolgreich durchgeführt hat, oder über mehrere Standorte verteilt arbeitet, kommt der Wunsch nach elektronischer Unterstützung für das Führen des Product Backlog und Sprint Backlog, sowie das automatische generieren von Burndown Charts, auf. Moderne Collaboration-Plattformen decken dies ab und unterstützen zusätzliche Funktionen für alle Phasen des Application Lifecycle.

Eine Liste von nützlichen Tools finden Sie unter:

<http://www.utakapp.de/collaborationtools>

Scrum Patterns

Erfahrungen mit Scrum zeigen, dass es viele soziale und organisatorische Muster gibt, die wiederholbar sind. Diese Patterns ermöglichen Scrum Teams die Erfahrungen von Anderen zu nutzen.

Nützliche solcher Muster finden Sie unter:

<http://www.utakapp.de/scrumpatterns>.



Uta Kapp ist seit über 20 Jahren in der Softwareentwicklung tätig. Sie arbeitet als Beraterin, Trainerin und Coach um ihre eigenen Erfahrungen auf diesem Wege weiter zu geben. Als Entwicklerin hat sie erkannt, dass der Schlüssel zum Erfolg immer die beteiligten Menschen sind. Scrum trägt diesem Rechnung, indem es den Raum für menschliche Potenziale öffnet und gleichzeitig

Ordnung und Struktur bereithält.

„Der kleine ScrumMaster“ fasst die wichtigsten Prinzipien und Abläufe von Scrum zusammen.

Kontakt

Uta Kapp
Chopinstraße 23
70195 Stuttgart
www.utakapp.de
info@utakapp.de

„Der kleine ScrumMaster“
Release 1.3.1
© Uta Kapp, 2010

Uta Kapp

Der kleine ScrumMaster

Gute Software schnell und kostengünstig entwickeln, das scheint nur eine technische Aufgabe zu sein. Tatsächlich ist es eine Herausforderung, die sehr viel Organisations- und Sozialkompetenz erfordert. Dies wird von Scrum, einem kleinen, übersichtlichen Projektmanagement-Framework, das sich auf wenigen Seiten beschreiben lässt, berücksichtigt. Seine Größe erhält es durch die Menschen die es mit Leben erfüllen. Damit funktioniert Scrum für jedes Projekt und in jedem Team anders und doch gleich.