

MODUL 426

SOFTWARE MIT AGILEN METHODEN ENTWICKELN

Reto Glarner

Proktmethode Scrum

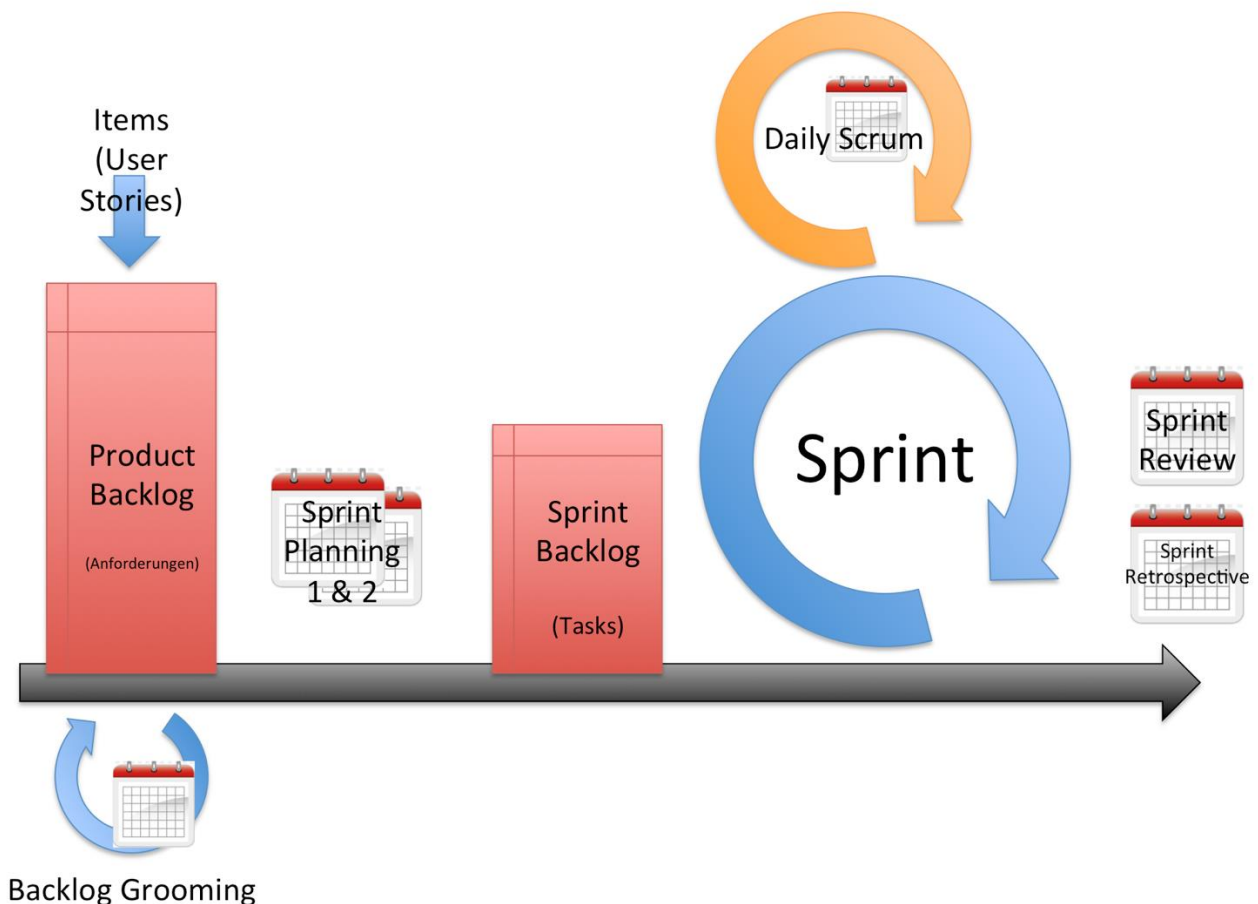
2/2

Lernziele

- › Kennt den Ablauf und die Bestandteile eines Releasezyklus
- › Kennt den Aufbau und Inhalt einer Releasepräsentation
- › Kennt die Bedeutung von lauffähigen Software releases
- › Kennt ein Vorgehen um die Ergebnisse eines Releasezyklus zu reflektieren
- › Kennt Möglichkeiten um das eigene Arbeitsvorgehen im Releasezyklus zu reflektieren

Das Scrum Projekt

Bevor die nächsten Tätigkeiten behandelt werden, sollten wir uns einen Überblick über die Abläufe in Scrum verschaffen:



Als Ausgangslage haben wir nun einen sauberen Product Backlog mit Anforderungen, welche priorisiert sind. Daraus gilt es nun "das Futter" für die Entwickler bereitzustellen, damit diese mit dem ersten Sprint beginnen können.

Sprint Planung (Sprint Planning Meetings)

Die Sprint Planung wird bei Anwesenheit aller 3 Rollen abgehalten und ist zentral für eine erfolgreiche Anwendung der Scrum Methode. Die Planung wird einmal vor *jedem* Sprint durchgeführt. Die Resultate einer Sprint Planung:

- › Festlegen des Sprint Zieles: Auswahl der Einträge aus dem Product Backlog, welche in diesem Sprint umgesetzt werden.
- › Klarstellen der Akzeptanzkriterien jedes einzelnen Punktes aus dem Product Backlog.
- › Erstellen des **Sprint Backlogs** in Form von Aufgaben (Tasks)

Die erzielten Resultate werden während des Sprints nicht verändert!

Diese Planung wird in 2 Teile gegliedert:

1. Auswahl der PBI's, welche in den Sprint Backlog übernommen werden (*Was Frage*)
2. Erstellen der Aufgaben zu den PBI's (*Wie Frage*)

Die Sprint Planung bedingt echtes Teamwork, jedes Mitglied soll seine Meinung vertreten dürfen, offene Fragen müssen in dieser Besprechung geklärt werden. Am Ende des Meetings haben alle den gleichen Wissensstand.

Hier ist der geeignete Zeitpunkt für kurze, administrative Inhalte wie zum Beispiel Ferienabwesenheiten, da diese Einfluss auf die Planung haben. Da die Sprints iterativ sind, wird in der Einleitung kurz Bezug genommen auf den vorherigen Sprint.

Sprint Planung Teil 1

Der Product Backlog ist ja bereits priorisiert, also geht es primär darum herauszufinden **wie viele** PBI in den Sprint Backlog übernommen werden sollen. Der Product Owner erläutert die PBI im Detail, wichtig sind vor allem die Akzeptanzkriterien, damit die Entwickler sich ein genaues Bild machen können über den Umfang und die Ziele der PBI.

Anschliessend müssen die Aufwände der einzelnen PBI geschätzt werden, nur so kann der Umfang des Sprints im gewünschten Rahmen gehalten werden. Es wird empfohlen dabei die Technik des "**Planning Poker**" einzusetzen.

Übung: Erklären Sie den **Ablauf** und den **Sinn** des Planning Poker: (15 Minuten)



Anhand der Daten des Planning Pokers kann nun entschieden werden, wie viele PBI's in den nächsten Sprint übernommen werden. Es muss versucht werden, die **Sprintlänge aller Sprints möglichst konstant** zu halten, unerfahrene Teams wählen ca. 1 Woche; Profi Scrum Teams können Sprints mit bis zu 30 Tagen planen. (Bei den ersten Versuchen werden Sprints tendenziell zu lang gewählt.)

Sprint Planung Teil 2

In diesem Teil geht es darum, die PBI's in den Sprint Backlog zu übernehmen. Die Anwesenheit des Product Owners ist nicht zwingend notwendig, solange keine nennenswerten Abweichungen zum Teil 1 auftreten. Die Arbeit kann in Teams aufgebrochen werden, auf keinen Fall aber sollte diese Planung durch den Teamleader *alleine* übernommen werden. (Akzeptanz, gemeinsames Wissen, Lernprozess aller usw.)

Es wird jedes PBI dieses Sprints einzeln abgehandelt. Dabei werden Aufgaben (Tasks) definiert, welche dann im Sprint Backlog abgebildet werden. Denken Sie bei den Tasks nicht nur an die Entwicklung, es können folgende Aufgabentypen vorkommen:

- › Design / Architektur
- › Umsetzen der Datenbank
- › Aufteilung der Tasks nach Schichten (GUI, Logik, Persistenz)
- › Testen (Unit Test, Benutzertest, Performancetest usw.)

Pro Task wird wieder ein Zeitaufwand vergeben, Ein Task sollte **nicht länger als einen Tag** dauern! Natürlich bekommt jeder Task später einen Verantwortlichen und einen Zustand. Dies wird aber *nicht* beim Sprint Planning festgelegt.

Falls nach diesem Schritt Fragen oder Differenzen auftauchen, müssen diese mit dem Product Owner abgesprochen werden. Im schlimmsten Fall muss die Sprint Planung nochmals zurück zu Teil 1 gehen.

Ansonsten ergibt sich aus der Sprint Planung eine Liste mit allen Aufgaben, dem **Sprint Backlog**.

Typisch amerikanisch wird am Ende der Planung ein "Team Commitment" abgegeben, das Team (und nicht jeder einzelne) verpflichtet sich die gemeinsam geplanten Anforderungen zeitgerecht umzusetzen. Alle Tasks des Sprint Backlogs werden auf einem **Taskboard** grafisch ansprechend dargestellt. Das Taskboard wird zentral aufgehängt. So können Sie sich ein Taskboard abstrakt vorstellen:

	Offen	Angefangen	Abgeschlossen
User Story 1	Task 5 / 7h Task 2 / 6h Task 6 / 8h	Task 3 / Klaus M. Task 7 / Maria T.	Task 1 Task 4
User Story 2	...		

Lösen Sie nun die entsprechende Aufgabe ➔1

Der Sprint

Prinzipiell geht es bei einem Sprint darum, die Aufgaben des Sprint Backlogs möglichst effizient zu erledigen. Fixe Elemente eines Sprints sind:

- › Daily Scrum (tägliche Kurzbesprechung)
- › Sprint Review (einmal am Schluss des Sprints)
- › Sprint Retrospective (einmal am Schluss des Sprints)

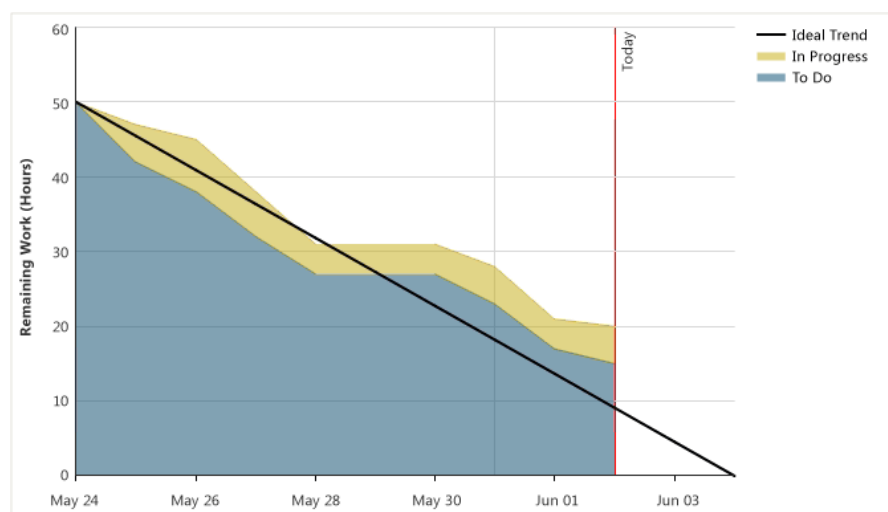
Der **Daily Scrum** ist ein wichtiges Element der Scrum Methode, deshalb gehen wir genauer darauf ein. Oberster Zweck des Meetings ist der Informationsaustausch. (Was war, was wird). Falls es mehrere Scrum Teams gibt, wird pro Team ein Daily Scrum abgehalten, die Koordination wird dann in einem **Scrum of Scrums Meeting** garantiert. Ein paar Tipps aus der ganzen Welt zum Daily Scrum:

- › Der Leader ist der Scrum Master
- › jeden Tag um die gleiche Zeit, am besten als erstes am Morgen
- › Dauer maximal 15 Minuten
- › Alle offenen Fragen und Probleme gehören auf den Tisch, grössere Fragen werden ausserhalb des Meetings gelöst, ebenso die Problemlösung im einzelnen Fall (→ Hindernisliste / **Impediment Backlog**)
- › Das Taskboard ist das zentrale Planungsmittel
- › Rechtzeitig da sein, die Sitzung fängt immer pünktlich an! (ggf. Teamküsseli machen)
- › Stehende Besprechungen sind dynamischer als sitzende
- › In der Kürze liegt die Würze, jeder erzählt nur was für alle bzw. für das Projekt von Wichtigkeit ist. (Notfalls eine Klingel oder ein "Magic Word" einführen um "abzuklemmen")
- › Keine Telefone (oder sogar Laptops) zugelassen!
- › Immer positiv beenden, es soll eine gewisse Energie fließen und zum erfolgreichen Weiterarbeiten ermutigen

Jeder Teilnehmer bereitet sich individuell vor mit den 3 Fragen:

1. Was habe ich gestern erledigt?
2. Was plane ich heute zu erledigen?
3. Könnte es Hindernisse geben?

Ein Dokumenten - Hilfsmittel, welches einen schnellen Einblick in den Sprint Fortschritt gibt ist die **Sprint Burndown Chart**.



Selbstverständlich wird diese Grafik vom TFS automatisch erstellt. (Quelle Microsoft)

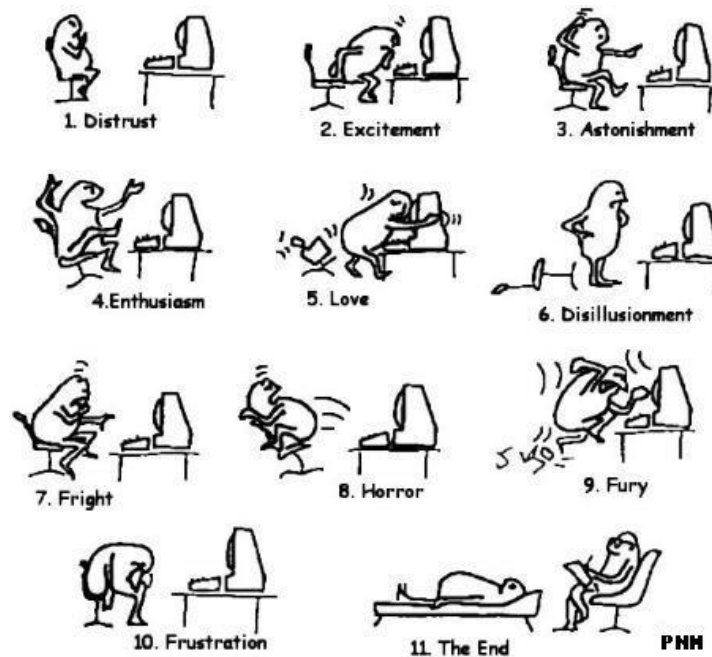
Auf der Sprint Burndown Chart kann man sehr einfach sehen:

- › Wie viel Arbeit ist noch offen im aktuellen Sprint
- › Ist das Team im Zeitplan

- › Wann werden wir voraussichtlich fertig
- › Wie viele Arbeiten sind gerade pendent

Übungszeit ➔2

Im zeitlichen Ablauf käme hier der Grossteil der Arbeit, nämlich die **Softwareentwicklung**.



Sprint Review

Die Ergebnisse des fertigen Sprints werden im **Sprint Review** dem Kunden (inkl. Benutzern) vorgestellt. Teilnehmer sind also alle Teammitglieder und der Kunde, die Leitung liegt beim Product Owner. Es werden alle PBI's (User Stories) nacheinander abgehandelt. Wichtig ist es, zuerst die Akzeptanzkriterien in Erinnerung zu rufen, damit keine Missverständnisse entstehen. Es werden nur *fertiggestellte und funktionierende* PBI's gezeigt! Der Kunde kann durchaus interaktiv involviert werden, um die Akzeptanz möglichst hoch zu halten.

Falls beim Sprint Review Differenzen zu Tage treten welche Anpassungen erfordern, werden diese in den Product Backlog aufgenommen und entsprechend der Dringlichkeit priorisiert. Oft kommen neue Ideen ins Spiel, dies sollte nicht als Kritik verstanden werden! Bei dieser Gelegenheit sollte auch überprüft werden, ob das Projekt zeitlich und kostenmässig im Rahmen geblieben ist.

Am Ende des Meetings wird gemeinsam besprochen was als nächstes zu tun ist, daraus ergibt sich ein aktueller, verbindlicher **Product Backlog**.

Zu diesem Meeting gibt es keine Übung in diesem Moment. Im praktischen Teil wird dieses dann geprobt.

Sprint Retrospective

Einen "internen Abschluss" findet der Sprint in der sogenannten **Sprint Retrospective**, eine Besprechung, in der nur das Scrum Team zusammensitzt. Ziel ist eine offene Kommunikation der Erfahrungen aus dem Sprint. Es wird die Möglichkeit geboten Dinge zu nennen, die beim nächsten Sprint verbesserungswürdig erscheinen. Der Scrum

Master nimmt die vom Team als gutgeheissenen Punkte in seine Hindernisliste (Impediment Backlog) auf und kümmert sich um Lösungen. Alle Teammitglieder sollten Einsicht in den Fortschritt dieser Verbesserungen haben, da sonst die Gefahr besteht dass die Sprint Retrospective nicht mehr ernst genommen wird. Dabei ist sie die Möglichkeit zur Verbesserung der Umstände, Arbeitsqualität, Produktequalität usw.



Leseübung: Eine Moderationstechnik für die Scrum Retrospektive | agile coding.pdf im Ordner 60_Artikel

Abschluss des Arbeitsblattes ➔3

Aufgabe 1

45 Minuten

Führen Sie im Team eine komplette Sprintplanung (1 & 2) anhand des Product Backlog aus dem AB01 durch.

Notieren Sie sich zuerst das Szenario:

Anzahl Entwickler

Anzahl Stunden pro
Entwickler pro Woche

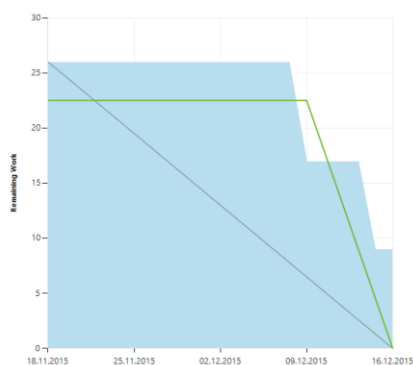
Sprintlänge in Tagen

Konsultieren Sie die Lehrperson bevor Sie mit dem Sprint Planning Meeting 2 beginnen!
Die Ergebnisse (Sprint Backlog und oder Taskboards) werden wiederum auf dem Klassenshare abgegeben. Die Lehrperson steht Ihnen bei Fragen zur Seite!

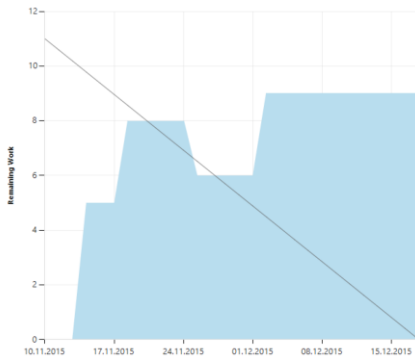
Aufgabe 2

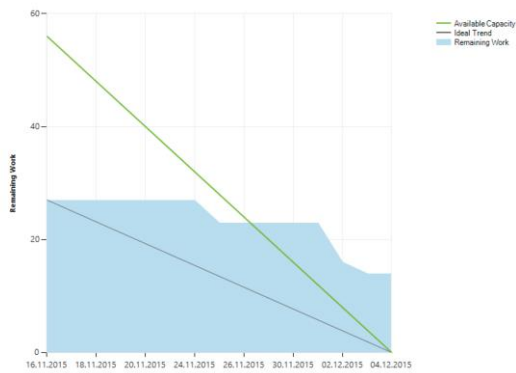
5 Minuten

Interpretieren Sie folgende Sprint Burndown Charts









Aufgabe 3

5 Minuten

Markieren Sie die bereits behandelten Themen auf dem Modulleitfaden Seite 2 in der Grafik!