

5 Die Durchführung

Wenn alle Vorbereitungen abgeschlossen sind, startet der eigentliche Scrum-Prozess. Dieser Teil stellt gewissermaßen das Herzstück eines Scrum-Projekts dar, denn er wird wie ein rhythmischer Herzschlag so lange wiederholt, bis das Produkt veröffentlicht werden kann. Abbildung 5-1 veranschaulicht den Prozess.

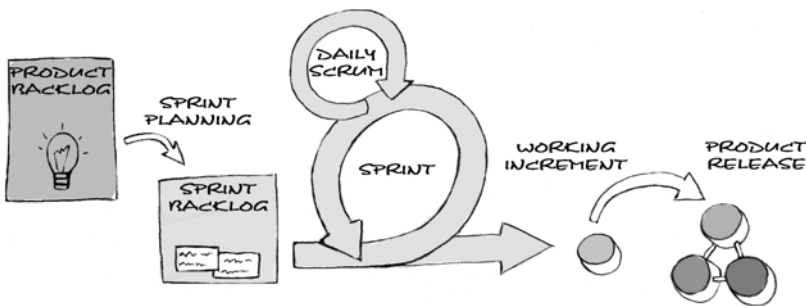


Abb. 5-1

Der Scrum-Prozess

In diesem Kapitel durchlaufen wir einen kompletten Sprint. Wir beginnen den Sprint mit dem Sprint Planning und gehen dann über in die tägliche Synchronisation des Entwicklungsteams, die im Daily Scrum stattfindet. Dieses kurze Treffen bietet erstaunliches Potenzial, aber auch jede Menge Stolperfallen, die wir ansprechen. Anschließend folgt die Pflege des Product Backlog im Backlog Refinement. Während dieses Events bearbeitet das Scrum-Team gemeinsam das Product Backlog, schätzt die Komplexität von Backlog Items und plant nächste Schritte. Nachdem wir die Vorteile eines gut vorbereiteten und interessant gestalteten Reviews herausgestellt haben, beschreiben wir mehrere Varianten für die Durchführung einer Retrospektive, die ein Scrum-Team bei der ständigen Verbesserung der Arbeitsweise unterstützt.

5.1 Sprint Planning I

Ready, Steady, Go!



Montagsmorgen, kurz vor 9:30 Uhr. Das Team kam erholt aus dem Wochenende und traf sich voller Tatendrang zum Sprint Planning für den anstehenden Sprint. Während Finn noch letzte Vorbereitungen erledigte, erzählten die anderen von ihren Unternehmungen am Wochenende. Sergio war am Samstag auf einer Party und war erst so spät zu Hause, dass er den ganzen Sonntag im Bett verbracht hatte. Mina und Lara waren zusammen im Kino und schwärmten von dem Film mit ihrem Lieblingsschauspieler. Punkt 10:00 Uhr unterbrach Finn freundlich die Gespräche und wies auf die Timebox hin. Das Team hatte maximal 3 Stunden für das Sprint Planning eingeplant, davon je 1,5 Stunden für Sprint Planning I und II.

Zu Beginn fragte Finn die Verfügbarkeit der Teammitglieder ab. Lara plante ein verlängertes Wochenende und steht daher zwei Tage weniger zur Verfügung. Jordi hatte einen Zahnarzttermin, ging aber davon aus, dass es keine Nachwirkungen gäbe, sodass er dem Team voll zur Verfügung stünde. Sergio merkte an, dass er im übernächsten Sprint eine ganze Woche Urlaub plane, aber Finn bat ihn, ihn nach dem Sprint Planning noch mal darauf anzusprechen, da es nichts mit dem aktuellen Sprint zu tun habe. Die von den Kollegen genannten Verfügbarkeiten notierte Finn auf einer Sprint-Übersicht für die kommenden zwei Wochen, die Finn vorab vorbereitet hatte.

Als Nächstes bat Finn Casper, dem Team das von ihm vorgeschlagene Sprint-Ziel zu erläutern und einen Überblick über die von ihm gewünschte Funktionalität für den Sprint zu geben. Bereits im Vorfeld hatten die beiden gemeinsam am Product Backlog gearbeitet und einen Block von User Stories identifiziert, der ihrem Empfinden nach in einen Sprint passen könnte. Casper erklärte die Funktionalität zunächst im Zusammenhang aus Sicht des späteren Benutzers. Anschließend ging er ins Detail und begann, einzelne User Stories vorzustellen. Dabei achtete er darauf, dass er die Reihenfolge so wählte, dass die ersten vier Stories die Grundfunktionalität beschrieben, während ab der vierten Story nur noch Ergänzungen hinzugefügt wurden. Für jede Story nannte er die Akzeptanzkriterien, die für eine erfolgreiche Erledigung notwendig waren.

Die Teammitglieder stellten Fragen zu den einzelnen User Stories, waren aber im Wesentlichen bereits damit vertraut, weil sie in einem der letzten Anforderungswshops an den Inhalten mitgearbeitet hatten. Nach jeder Story fragte Casper, ob diese im nächsten Sprint umgesetzt werden kann. Bis zur sechsten Story nickten alle Teammitglieder

einhellig. Bei der sechsten User Story wendete Lara ein, dass ihr nicht genügend Informationen vorliegen. Auch Mina zögerte, weil sie in den bereits ausgewählten User Stories bereits eine Menge Testaufwand sah. Das Team beschloss, es bei den sechs Stories zu belassen. Casper war glücklich, konnte er doch davon ausgehen, dass bis auf ein kleines Extra nach dem Sprint eine komplette Funktionalität zur Verfügung stehen würde. Im letzten Schritt überprüften alle zusammen noch einmal das Sprint-Ziel und einigten sich auf eine griffige Formulierung.

Praxistipp

Oft beginnen Sprints an einem Montag, vermutlich weil der Montag sowieso den Beginn einer Arbeitswoche symbolisiert. Es gibt aber keine Vorschrift, einen Sprint immer an einem Montag beginnen zu lassen. Manchmal gibt es äußere Rahmenbedingungen (zum Beispiel 4-Tage-Wochen, keine Verfügbarkeit wichtiger Stakeholder), die einen anderen Wochentag sinnvoller erscheinen lassen. Finden Sie gemeinsam mit dem Team den für die gegebene Situation idealen Wochentag.

Am besten planen Sie gemeinsam mit dem Team das Sprint Planning zum frühestmöglichen Termin am Vormittag ein, zu dem das komplette Team anwesend ist. Dadurch vermeiden Sie Leerlauf und Nachmittagsmüdigkeit.



Das Sprint Planning kennzeichnet den Beginn eines Sprints. Im ersten Teil des Sprint Planning vereinbaren Product Owner und Entwicklungsteam das *Was* für den neuen Sprint. Wichtigste Voraussetzung ist eine gute Vorbereitung des Product Owners. Er muss wissen, was er im nächsten Sprint umgesetzt haben möchte. Außerdem müssen die Akzeptanzkriterien klar sein. Je besser die Vorbereitung des Product Owners ist, desto einfacher ist es für das Entwicklungsteam, Backlog Items in den Sprint zu nehmen und eine Prognose abzugeben. Im Idealfall verfügt der Product Owner nicht nur über die Backlog Items für den nächsten Sprint, sondern auch über einen gut durchdachten und priorisierten Releaseplan, der alle zum jetzigen Zeitpunkt bekannten Backlog Items bis zum nächsten Release enthält (vgl. Abschnitte 4.2.4 und 4.4.1).

Das Team sollte im Sprint Planning niemals mit komplett neuen Anforderungen konfrontiert werden. Wir ermutigen unsere Teams, solche Backlog Items nicht in den Sprint zu übernehmen. Das Risiko ist einfach zu groß, dass man damit in ein zurzeit nicht erkennbares Wespennest sticht. Das Sprint Planning ist dazu da, bekannte Backlog Items noch einmal durchzusprechen, das gemeinsame Verständnis zu

verifizieren und sie für den Sprint einzuplanen. Für die inhaltliche Arbeit gibt es Anforderungsworkshops.



Praxistipp

Es ist nichts dagegen einzuwenden, wenn der Product Owner im Sprint Planning noch eine Ergänzung anfragt, die offensichtlich klein und unkritisch ist. Sollte es sich um ein neues Backlog Item handeln, dann muss dies im Sprint Planning noch geschätzt werden. Oft kann man solche Anforderungen aber auch als Akzeptanzkriterium in ein bereits bestehendes Backlog Item integrieren, z.B. eine vorher nicht geforderte Sortierreihenfolge in einer Liste oder die Anzeige einer zusätzlichen Information. Fragen Sie in dem Fall das Team, ob sich dadurch die ursprüngliche Schätzung ändert.

5.1.1 Teamverfügbarkeit

Zu Beginn des Sprint Planning I ist es wichtig, dass alle Teammitglieder über die Verfügbarkeit der Teammitglieder informiert sind, um die Inhalte des Sprints entsprechend zu planen. Wenn beispielsweise der einzige Frontend-Entwickler während des Sprints für eine Woche im Urlaub ist und niemand im Team einspringen kann, sollte dem Rechnung getragen werden, indem Backlog Items mit weniger komplexem Frontend-Anteil ausgewählt werden. Wir erfragen die Verfügbarkeit mit einer Granularität von einem halben Tag, da dies als Überblick vollkommen ausreicht.



Praxistipp

Achten Sie darauf, dass Wissen im Team verteilt wird. Der Ausfall von Einzelpersonen darf nicht zu einem Stillstand im Team führen. Es hat sich bewährt, dafür zu sorgen, dass jedes Teammitglied zusätzlich zu seiner Spezialdisziplin noch Wissen in einer weiteren, sekundären Disziplin aufbaut. Unterstützen Sie das Team dahingehend.

Fremdaufwände

Erfahrungsgemäß stehen die Teammitglieder nie zu 100 % ihrer Anwesenheit für den Sprint zur Verfügung. Je nach Unternehmen liegt die Verfügbarkeit der Mitarbeiter meist zwischen 50 % und 80 %. Bei der nicht für den Sprint verfügbaren Zeit kann es sich zum Beispiel um Teammeetings aus der Linie handeln oder um regelmäßige Termine mit

Vorgesetzten. Besonders in kleinen Unternehmen haben die Scrum-Teams oft zusätzliche Wartungsaufgaben oder müssen sich nebenbei um den operativen Betrieb kümmern. Eine besondere Herausforderung stellte einmal ein Team dar, dessen Mitglieder zusätzlich zur Projektarbeit für die Wartung eines Legacy-Systems verantwortlich waren – jedes Teammitglied für ein anderes. Dies sind alles Aufwände, die nichts mit dem Sprint zu tun haben und die daher von der Verfügbarkeit der Mitarbeiter abgezogen werden müssen. Wenn man sich einmal die Mühe macht, diese Zeiten und Aufwände zu erfassen, ist man oft erstaunt, wie wenig Zeit tatsächlich noch für die Schaffung neuer Funktionalität zur Verfügung steht.

Praxistipp

Versuchen Sie, die zusätzlichen Aufgaben auf Kollegen außerhalb des Teams zu verteilen. Sofern es sich nicht um erkennbare Blöcke von Störungen handelt, bitten Sie das Team, während des Sprints alle Aufwände zu notieren, die nichts mit dem Sprint zu tun haben. Achten Sie auch während des Daily Scrum auf Formulierungen wie »Ich konnte nicht an dem Thema arbeiten, weil ...«. So kommen Sie Hindernissen auf die Spur, und die Auflistung bietet Ihnen eine gute Argumentationsgrundlage, die Verfügbarkeit zu erhöhen.

Eine weitere gute Möglichkeit, Störungen zu identifizieren, besteht darin, die Teammitglieder am Ende des Sprints nach ihrer tatsächlichen Verfügbarkeit im Sprint zu fragen. Liegt diese niedriger als die geplante Verfügbarkeit, besteht Handlungsbedarf für den Scrum Master. Entweder müssen die Störungen beseitigt werden oder die tatsächliche Kapazität muss transparent dargestellt werden.



Wenn man über ein achtköpfiges Team verfügt, das aber nur zu 50 % am Sprint arbeiten kann, hat man rechnerisch lediglich vier Personen zur Verfügung. Diese Situation sollten Sie unbedingt als Hindernis behandeln. Die Fokussierung des Teams wird stark beeinträchtigt, und eine Prognose ist nicht möglich, wenn die anderen Tätigkeiten spontan und ungeplant auftreten (z.B. User Support). Durch das aktive Angehen dieses Hindernisses beugen Sie auch Nachfragen des Managements vor, warum das (vermeintlich) achtköpfige, teure Scrum-Team denn so langsam sei und nur so wenig Ergebnisse liefere.

**Praxistipp**

Hängen Sie eine »Zeitfresser«-Übersicht auf. Das ist eine einfache Tabelle der Wochentage, in die Teammitglieder Post-its mit Störungsursachen und der dafür aufgewendeten Zeit hängen. Spätestens im Daily Scrum sollte diese Übersicht aktualisiert werden. Die dadurch entstehende Dokumentation wird genutzt, um wiederkehrende oder sehr zeitintensive Themen anzugehen.

Abwesenheitskalender

Neben Urlaub kann es viele Gründe für geplante Abwesenheiten geben, zum Beispiel Schulungen, Arzttermine, Handwerker im Haus. Diese Abwesenheiten im Sprint sollten zu Beginn des Sprint Planning offengelegt werden, damit das Team seine Planung danach ausrichten kann. Sollte im Sprint ein Teammitglied mit Spezialisten-Know-how fehlen, sollte dies natürlich schon vorher bekannt sein, damit der Product Owner ggf. den Sprint-Inhalt danach ausrichten kann oder der Scrum Master sich darum kümmern kann, längerfristige Abwesenheiten durch temporäre Unterstützung zu kompensieren.

In einer »Sprintübersicht« [URL:Wiechmann a] werden zu Beginn des Sprint Planning die geplanten Abwesenheiten für den Sprint visualisiert und anschließend sichtbar neben dem Scrum-Board aufgehängt. Jeder im Team sollte sich verantwortlich dafür fühlen, den Kalender während des Sprints sowohl für sich selbst als auch für andere (z. B. bei Krankmeldungen) zu aktualisieren. Ergänzt wird diese Übersicht gerne um Informationen, wie die aktuelle Sprint-Nummer, das Sprint-Ziel, das Releasedatum, die aktuelle Velocity sowie das Commitment.

**Praxistipp**

Führen Sie zusätzlich einen Jahres-Teamkalender, der offen im Teamraum aushängt. Markieren Sie auf dem Kalender die nächsten Sprints und bitten Sie das Team, alle Abwesenheiten dort einzutragen. Auf diese Weise sieht man früh, wer in welchem Sprint nicht oder nur teilweise anwesend ist, und kann den Releaseplan entsprechend daran ausrichten.

Sonderaufgaben

Wir treffen oft auf Teams, die neben den Entwicklungsaufgaben noch weitere regelmäßige Aufgaben haben, zum Beispiel den produktiven Betrieb des eigenen Produkts zu monitoren. Es hat sich bewährt, diese Aufgaben klar zu identifizieren und bereits im Sprint Planning zu entscheiden, wer sich darum kümmert.

Praxistipp

Wenn immer dieselben Teammitglieder eine bestimmte Aufgabe erledigen, entstehen Wissensinseln. Schlagen Sie dem Team vor, die Aufgaben immer für einen Sprint wahrzunehmen und dann rotieren zu lassen. Dies hilft auch dabei, eintönige Aufgaben gerecht im Team zu verteilen.



5.1.2 Planung ohne Story Points

Die Velocity eines Teams berechnet sich über die Teamleistung vergangener Sprints. Diese Teamleistung schwankt und wird für die Berechnung der Velocity nivelliert. Da die Schätzungen der Backlog Items nur sehr grob sind und eine Menge Unsicherheit enthalten, hat es keinen Sinn, im Sprint Planning I bis zur exakten Erreichung der Velocity zu planen. Ein Team mit einer Velocity von 30 Story Points, das im Sprint Planning I Backlog Items mit einer Summe von 27 Story Points auswählt, sollte nicht aufgefordert werden, noch ein weiteres Backlog Item mit drei Story Points auszuwählen. Das Aufaddieren der Story Points der Backlog Items bis zum Erreichen der Velocity gaukelt an dieser Stelle eine vermeintliche Planungspräzision vor, die de facto aber nicht vorhanden ist. Erinnern wir uns an das Estimation (vgl. Abschnitt 4.3): Jedes Backlog Item enthält eine gewisse Unsicherheit, die umso größer wird, je komplexer das Backlog Item ist. Wenn sich die 27 Story Points in obigem Beispiel aus drei mal acht Story Points und einmal drei Story Points zusammensetzen, bedeutet dies genaugenommen »irgendwo zwischen 17 und 44 Story Points«. Es ist also nicht sinnvoll, die Velocity »bis zum Anschlag« auszureizen.

Wir haben sehr gute Erfahrungen damit gemacht, dem Entwicklungsteam während des Sprint Planning die Story Points der Backlog Items nicht zu zeigen, sondern es allein aufgrund fachlicher Inhalte und einer gesunden Portion Bauchgefühl entscheiden zu lassen. Wir arbeiten oft mit der TTM-Matrix (vgl. Abschnitt 4.2.6) und bieten dem Team im Sprint Planning I diese Information als Basis für die Auswahl der Backlog Items für den Sprint an.



Praxistipp

Prüfen Sie nach Ende des Sprint Planning I für sich, ob die Summe der Story Points der ausgewählten Backlog Items in etwa der Velocity entspricht. Unsere Erfahrung zeigt, dass die Teams ein sehr gutes Bauchgefühl dafür haben, was im Sprint machbar ist. Sollte die Summe tatsächlich signifikant von der Velocity abweichen, beobachten Sie das Sprint Planning II sehr genau und sprechen Sie das Team am Ende darauf an. Möglicherweise waren Schätzungen zu hoch, vielleicht gibt es aber auch wirklich ein Problem. Machen Sie die Situation transparent und besprechen Sie sie.

5.1.3 Akzeptanzkriterien (How to demo)

In Abschnitt 4.2.3 haben wir die Wichtigkeit von Akzeptanzkriterien ausführlich besprochen. Im Sprint Planning kann das Team nun die Früchte guter Vorarbeit ernten, da viel weniger Nachfragen kommen.

Meist liegen die Akzeptanzkriterien als Liste vor und beschreiben, was wie funktionieren soll. Da am Ende des Sprints jedoch ein Review stattfindet, in dem die neue Funktionalität *vorgestellt* werden soll, ist diese Listendarstellung nur bedingt hilfreich. Wir schlagen unseren Teams im Sprint Planning I daher vor, die Akzeptanzkriterien unter der Perspektive des späteren Zeigens zu betrachten: »Wie zeigen wir dem Product Owner und den Reviewteilnehmern, dass die Funktionalität tatsächlich wie erwartet umgesetzt wurde?« Das Ergebnis ist ein kleiner Workflow aus der Sicht eines Anwenders, der beispielsweise folgendermaßen aussehen könnte:

■ User Story

- Als Konferenzbesucher möchte ich interessante Vorträge markieren und als Zeitplan anzeigen, um sie später schnell wiederzufinden.

■ Akzeptanzkriterien

- Markierungsmöglichkeit in der Vortragsliste
- Falls parallel liegende Vorträge ausgewählt werden, soll eine Warnung erscheinen.
- Es soll einen Zeitplan geben, der die markierten Vorträge in der Reihenfolge ihres Stattfindens anzeigt.

■ How to demo

- App aufrufen
- Sechs Vorträge auswählen, davon zwei, die sich mit anderen überschneiden
- Warnung bei der Auswahl der sich überschneidenden Vorträge anzeigen
- Zeitplan aufrufen, die sich überschneidenden Vorträge anzeigen lassen
- Entscheiden für einen der parallelen Vorträge und Löschen des jeweils anderen
- Zeigen des Zeitplans ohne Überschneidungen

Wie man leicht erkennt, wird plötzlich das Löschen eines Elements erwähnt, was jedoch nicht in den Akzeptanzkriterien stand. Das Ändern der Perspektive hilft dabei, solche Dinge bereits im Sprint Planning aufzudecken.

5.1.4 Sprint-Ziel

Zu jedem Sprint Planning gehört das Festlegen eines Sprint-Ziels. Das Sprint-Ziel ist das Maß, an dem das Team am Ende des Sprints gemessen wird. Selbst wenn ein niedriger priorisiertes Backlog Item nicht erledigt werden konnte: Solange das Sprint-Ziel erreicht wurde, gilt der Sprint als erfolgreich.

Im Verhältnis gesehen ist ein Sprint-Ziel für einen Sprint das, was für das Gesamtprojekt bzw. das nächste Release die Vision ist. Ein Sprint-Ziel beschreibt fachliche Anforderungen aus Sicht des Benutzers, die ihm einen Nutzen bringen und die am Ende des Sprints vorzeigbar sind. Es stellt gewissermaßen einen Meilenstein auf dem Weg zur Vision dar. Weiter hilft es dem Team, sich während des Sprints auf das Wesentliche zu fokussieren und Anforderungen bzw. Ergänzungen, die keinen Einfluss auf das Sprint-Ziel haben, in Absprache mit dem Product Owner niedriger zu priorisieren.

Das Sprint-Ziel wird zu Beginn des Sprint Planning vom Product Owner auf Basis der von ihm gewünschten Backlog Items vorgeschlagen und am Ende vom Entwicklungsteam akzeptiert oder gemeinsam angepasst.



Praxistipp

Bitten Sie den Product Owner zu Beginn des Sprint Planning, dem Entwicklungsteam zu erläutern, welchen Einfluss das Sprint-Ziel auf die Produkt- oder Releasevision hat. Bitten Sie ihn außerdem im weiteren Verlauf des Sprint Planning immer wieder, die Relevanz des gerade besprochenen Backlog Items für das Sprint-Ziel zu erklären. Diese Ausführungen helfen dem Product Owner und dem Team, sich auf die wirklich wichtigen Features zu konzentrieren und sogenanntes »Goldplating«, also das für den Geschäftswert unwichtige Aufpolieren, zu vermeiden.

Wie man es schon aus dem traditionellen Projektmanagement kennt, sollten auch Sprint-Ziele SMART sein (vgl. Abb. 5–2).

Abb. 5–2
SMART-Kriterien

SPEZIFISCH	EINDEUTIG UND KONKRET, SODASS JEDER DAS GLEICHE VERSTÄNDNIS HAT
MESSBAR	DOPPELT SO SCHNELL WIE V1.1, MINDESTENS 50 GLEICHZEITIGE BENUTZER, INNERHALB VON 2 SEKUNDEN
AKZEPTIERT	VOM TEAM UND VOM PRODUCT OWNER AKZEPTIERT
RELEVANT	MIT AUSWIRKUNG AUF DIE PRODUCT- ODER RELEASE-VISION
TERMINIERT	RIS ZUM ENDE DES SPRINTS ERFÜLLT

»Die nächsten fünf Backlog Items umsetzen« ist ein sehr schlechtes Sprint-Ziel, »Am Ende des Sprints wird die Seitenauslieferung doppelt so schnell sein wie in V1.1« hingegen erfüllt die SMART-Kriterien.



Praxistipp

Drucken Sie das Sprint-Ziel in großer Schrift über mehrere DIN-A4-Seiten aus (Banner) und hängen Sie es über das Scrum-Board. So kann es sehr einfach für Diskussionen und Entscheidungen herangezogen werden. Sofern Sie über einen großen TV-Monitor verfügen, der z.B. über rotierende Browser-tabs den Systemzustand darstellt, zeigen Sie das Sprint-Ziel auch dort mit an.

Ein Feature muss nicht gleich von Anfang an mit allen Komfortfunktionen ausgestattet sein. Es reicht zum Beispiel aus, in einem Sprint eine Bezahlfunktion via Vorkasse anzubieten, um das Sprint-Ziel »Als Kunde möchte ich meine Bestellung bezahlen« zu erfüllen, um in Folge-Sprints Überweisung, PayPal und Kreditkarten anzubieten. Hier ist Fingerspitzengefühl des Product Owners gefragt, der Backlog Items entsprechend zuschneiden muss.

Praxistipp

Kommunizieren Sie mit den Stakeholdern bzw. dem Management auf Basis des Sprint-Ziels, nicht auf der Detailebene der Backlog Items. Leider kommt es häufig vor, dass ein nicht erledigtes Backlog Item mehr Aufmerksamkeit bekommt und länger diskutiert wird als ein erreichtes Sprint-Ziel, obwohl das Backlog Item nichts mit dem Sprint-Ziel zu tun hatte.

**5.1.5 Slack**

Manchmal gibt es im Team Diskussionen, ob ein weiteres Backlog Item in den Sprint aufgenommen werden sollte oder nicht. Meist läuft es dann darauf hinaus, dass es nicht mit hineingenommen wird, damit sich alle Teammitglieder auf das Sprint-Ziel einigen können. Für ein Team ist es frustrierend, sich auf ein Backlog Item zu verpflichten und es dann nicht erledigen zu können.

Um unseren Teams hier zu helfen, bieten wir ihnen an, das umstrittene Backlog Item als sogenannten Slack (dt. Schlupf, Puffer) mit in den Sprint zu nehmen. Das bedeutet, dass das Backlog Item als Letztes an das Scrum-Board kommt und Bestandteil des Sprint Backlog wird. Das Entwicklungsteam nimmt es jedoch nicht offiziell in die Prognose auf, sondern es wird im Fall der frühzeitigen Erledigung der anderen Backlog Items als Erstes nachgezogen. Am Scrum-Board werden das oder die optionalen Backlog Items gesondert markiert, zum Beispiel, indem die Karte eine andere Farbe hat oder in einem größeren Abstand zu den anderen Backlog Items aufgehängt wird. Alternativ könnte das Team die fraglichen Backlog Items zunächst nicht in den Sprint aufnehmen und bei Bedarf den Product Owner nach weiteren Backlog Items fragen.

Praxistipp

Achten Sie darauf, dass die Benutzung von Slack Items nicht permanent zu niedrigen Prognosen beim Team führt. Es darf nicht der Eindruck entstehen, dass man sich auf nichts festlegen müsse, sondern einfach das abliefern könne, was gemächlich fertig geworden ist. Der Slack ist ein Angebot an das Team, sich mit der Prognose wohlfühlen und möglicherweise »noch einen draufsetzen« zu können. Wenn der Slack missbraucht wird, schaffen Sie ihn wieder ab.



5.1.6 Häufige Probleme

Prognose zu niedrig

Es kann vorkommen, dass ein Entwicklungsteam sich wiederholt deutlich weniger Backlog Items für den Sprint auswählt, als es zu leisten imstande ist. Diese Backlog Items sind oft schon nach der Hälfte des Sprints erledigt, und das Team fragt beim Product Owner nach Nachschub. Oberflächlich betrachtet scheint dies zunächst kein Problem zu sein, da ja letzten Endes eine Menge geliefert wird und der Product Owner seinen Releaseplan basierend auf der tatsächlichen Velocity erstellen kann. Wenn man aber mal hinter die Kulissen blickt, kommt man zu anderen Erkenntnissen.

Diese Situation kann bedeuten, dass das Team unsicher ist und mit Störungen im Sprint rechnet. Diese Beeinträchtigungen gilt es offenzulegen und das Vertrauen des Teams in die Schutzfunktion des Scrum Masters und den Prozess zu stärken.

Es kann auch sein, dass ein Team die Backlog Items nicht gut genug kennt und sich seiner eigenen Schätzungen nicht sicher ist. In diesem Fall muss früher angesetzt und die Zusammenarbeit des Teams und des Product Owners im Vorfeld des Sprint Planning verbessert werden.



Praxistipp

Wenn Sie häufig eine zu niedrige Prognose in Ihrem Team feststellen, gehen Sie der Ursache auf dem Grund. Lassen Sie sich nicht dazu verleiten, es als nicht so problematisch abzutun, denn wie oben gezeigt liegen die Ursachen in der Regel tiefer und holen Sie später ein. Nutzen Sie Methoden wie »5 Whys« [URL:5 Whys], um Ursachen herauszufinden, oder spielen Sie »Fearless Journey« [URL:Hartmann Preuss] mit dem Team.

Intransparente Restaufwände

Es kann der Fall eintreten, dass ein Backlog Item im letzten Sprint nicht erledigt werden konnte und nun als Kandidat für den neuen Sprint bereitsteht. Diese Backlog Items werden meist ohne weitere Besprechung mit in den anstehenden Sprint übernommen.



Praxistipp

Bitten Sie den Story Owner (vgl. Abschnitt 5.2.2) des Backlog Items, die noch ausstehenden Aufgaben kurz für das gesamte Team zu erläutern, damit alle Teammitglieder auf dem gleichen Stand sind.

Anteilige Story Points bei Restarbeiten

Manchmal wurde ein großes Backlog Item im letzten Sprint nicht abgenommen und auch nicht gezeigt, weil es noch nicht 100 % »Done« war. Manche Teams erwarten, dass die bereits geleistete Arbeit dem letzten Sprint in Form von Story Points anteilig zugerechnet und im anstehenden Sprint nur mit den restlichen Story Points geplant wird. Dies widerspricht aber dem Konzept der Velocity, denn konsequenterweise fließen die Story Points eines Backlog Items erst dann in die Velocity des Teams ein, wenn das Backlog Item komplett erledigt wurde und damit auch der Geschäftswert bereitgestellt wurde. Die Berechnung der Velocity ist so ausgelegt, dass solche Schwankungen ausgeglichen werden.

Praxistipp

Lassen Sie keine anteiligen Story Points zu. Erläutern Sie dem Team stattdessen die Berechnung der Velocity anhand von Beispielen und zeigen Sie den Ausgleich über die Zeit auf.



Nichtbeachtung der Definition of Ready

Idealerweise hat das Team mit dem Product Owner eine Definition of Ready (vgl. Abschnitt 4.1.4) vereinbart, sodass die Erwartungen für das Sprint Planning klar sind. Manchmal stellt man aber fest, dass aus unterschiedlichen Gründen nicht alle vereinbarten Vorarbeiten erledigt wurden, die DoR also nicht erfüllt wurde.

Praxistipp

Die Entwickler tun sich keinen Gefallen damit, solche Backlog Items trotzdem zu akzeptieren. Erfahrungsgemäß sorgen die ungeklärten Punkte im Sprint für Verzögerung und Diskussionen und können letztendlich sogar dazu führen, dass das Sprint-Ziel nicht erreicht wird. Ermutigen Sie das Team, diese Backlog Items abzulehnen und als Konsequenz daraus ggf. sogar ein anderes Sprint-Ziel zu vereinbaren. Nicht zuletzt wird auch dem Product Owner klar, dass er seine Vorbereitung verbessern muss.



Technische Diskussionen

Technische Diskussionen im Sprint Planning I sind ein Spezialfall der Nichtbeachtung der DoR, den wir aufgrund der Häufigkeit seines Auftretens hier noch einmal gesondert herausheben. Im Sprint Planning I

geht es um das Verstehen der fachlichen Anforderungen. Technische Diskussionen über die Umsetzung haben hier nichts verloren, dafür gibt es das Sprint Planning II oder die vorausgehende Besprechung der Backlog Items im Backlog Refinement. Falls ein Backlog Item ohne vorhergehende technische Diskussion nicht ausgewählt werden kann, ist die Definition of Ready nicht erfüllt.

Viele große Backlog Items

Wenn ein Sprint nur aus großen Backlog Items besteht, erhöht dies das Risiko, dass das letzte Backlog Item in der Liste nicht mehr umgesetzt werden kann und damit möglicherweise sogar das Sprint-Ziel verfehlt wird. Außerdem implizieren große Backlog Items in der Regel mehr Codereviews, Nachbesserungen, Nachfragen usw.



Praxistipp

Achten Sie als Scrum Master auf eine ausgewogene Auswahl an Backlog Items für den Sprint. Schlagen Sie vor, einige kleinere Backlog Items mit aufzunehmen, deren Nichterledigung das Sprint-Ziel an sich nicht gefährdet. Machen Sie dabei aber klar, dass die Prognose auch diese Backlog Items beinhaltet und dass es sich nicht um Slack (vgl. Abschnitt 5.1.5), sondern um ein Sicherheitsnetz handelt.

Eventuell können die großen Backlog Items auch weiter aufgebrochen werden (vgl. Abschnitt 4.2.5).

Ungleichmäßige Auslastung des Teams

Die Auswahl der Backlog Items für den Sprint sollte so erfolgen, dass nicht alle die Unterstützung des erfahrensten Entwicklers benötigen. Das Team läuft sonst Gefahr, dass es sich dann nur in Ansätzen um die Themen kümmern kann und viele Änderungen und Nachbesserungen nötig werden.



Praxistipp

Wirken Sie als Scrum Master darauf ein, dass Backlog Items so ausgewählt werden, dass die Verfügbarkeit und fachliche Kompetenz aller Teammitglieder möglichst optimal ausgenutzt werden. Setzen Sie zum Beispiel die TTM-Matrix (vgl. Abschnitt 4.2.6) ein, um dem Team eine Entscheidungshilfe zu geben.

Deployment nicht geplant

Oft enden Sprints an einem Freitagnachmittag mit einem Review und einer Retrospektive. Das vorgestellte Inkrement noch am gleichen Tag zu deployen ist vielen Firmen zu riskant und wird oftmals in der Folgewoche durchgeführt. Insbesondere bei kleinen Firmen sind die Entwickler auch gleichzeitig diejenigen, die das Deployment durchführen, sodass die dafür benötigte Zeit sowie die Zeit für eventuell auftretende Probleme vom nächsten Sprint abgeht. Dieser Aufwand wird in der Regel nicht geplant oder wird in die Verfügbarkeit des Teams eingerechnet. Dies hat jedoch den Nachteil, dass der Aufwand damit »unsichtbar« wird.

Praxistipp

Nehmen Sie ein Deployment-Item in jeden Sprint mit auf. Dieses Item wird ebenso mit Tasks am Scrum-Board geplant wie alle anderen Backlog Items.



Ungutes Bauchgefühl

Die oben beschriebenen Probleme bei der Durchführung eines Sprint Planning lassen sich in der Regel klar erkennen und benennen. Es kann aber auch vorkommen, dass man einfach nur ein ungutes Bauchgefühl hat, das man nicht weiter beschreiben kann. In solch einem Fall empfehlen wir, das Sprint Planning als solches gemeinsam mit dem Team zu besprechen und die Meinung des Teams dazu zu hören.

Praxistipp

Nehmen Sie das Sprint Planning als Thema für eine Retrospektive auf. Auch wenn Scrum ein Sprint Planning als Muss vorschreibt, kann es von Zeit zu Zeit hinterfragt und verbessert werden. Die Retrospektive bietet durch ihre Struktur einen idealen Rahmen für derartige Fragestellungen.



5.1.7 Checklisten

Sprint Planning I



Für die Planung und Durchführung eines Sprint Planning gibt es so manches zu beachten. Nachfolgend haben wir Ihnen eine Prüfliste mit den wichtigsten Punkten zusammengestellt.

■ Vorbereitung

- Hinreichend großen Raum buchen, der dem gesamten Scrum-Team und ggf. Gästen Platz bietet. Den Raum am besten für die gesamte Projektlaufzeit im Voraus buchen.
- Raum bereits für eine halbe Stunde vor Beginn des Sprint Planning buchen, um ggf. Vorbereitungen treffen zu können
- Termin möglichst früh am ersten Tag des Sprints planen
- Verfügbarkeit der Teammitglieder ermitteln
- Durchführung planen und vorbereiten
- Benötigte Hilfsmittel für die Moderation bereitstellen
- Verfügbarkeit der technischen Ausrüstung sicherstellen, z.B. Projektor, Videokonferenzsystem
- Existenz eines geordneten Product Backlog und ausreichend vorbereiteter Backlog Items sicherstellen

■ Durchführung

- Pünktlich beginnen, auch wenn noch Teilnehmer fehlen
- Notebooks und Mobiltelefone verbannen
- Gegebenenfalls Timer ablaufen lassen, um die Timebox zu visualisieren
- Teamverfügbarkeit abfragen und visualisieren, dabei Feiertage, Urlaub und sonstige geplante Abwesenheiten berücksichtigen
- Einhaltung der Definition of Ready einfordern
- Auf zwischen den Zeilen geäußerte Hindernisse achten
- Technische Diskussionen unterbinden
- Bei der Planung möglichst auf Story Points verzichten, sondern diese am Ende nur zur Überprüfung heranziehen
- Prognose des Entwicklungsteams erfragen
- Gegebenenfalls Slack vereinbaren
- Sprint-Ziel mit SMART-Kriterien festlegen

■ Nachbereitung

- Ausgewählte Backlog Items für das Sprint Planning II ausdrucken
- Gegebenenfalls elektronische Tools aktualisieren
- Gegebenenfalls Sprint-Dokumentation aktualisieren, z.B. im Wiki