



SCRUM

Februar 2010

Scrum: Entwickelt und fortgeführt von Ken Schwaber und Jeff Sutherland

Annerkennung

Allgemeines

Scrum basiert auf von der Industrie akzeptierten bewährten Verfahren, die seit Jahrzehnten verwendet und unter Beweis gestellt werden. Daraus wurde es in eine empirische Prozesstheorie eingebunden. Wie Jim Coplien einmal zu Jeff meinte: „Jeder wird Scrum mögen; das ist es, was wir bereits machen, wenn wir mit dem Rücken an der Wand stehen.“

Personen

Von den Tausenden Menschen, die zu Scrum beigetragen haben, sollten wir diejenigen besonders hervorheben, die Scrum in den ersten 10 Jahren vorangebracht haben. Als erste zählen dazu Jeff Sutherland in seiner Arbeit mit Jeff McKenna, und Ken Schwaber mit Mike Smith und Chris Martin. Scrum wurde zuerst formell präsentiert und veröffentlicht zur OOPSLA 1995. Während der folgenden fünf Jahre haben Mike Beadle und Martine Devos einen signifikanten Anteil geleistet. Und dann natürlich all die anderen, ohne deren Unterstützung Scrum nicht zu dem verfeinert worden wäre, was es heute ist.

Geschichte

Die Geschichte von Scrum kann bereits als lang in der Welt der Softwareentwicklung angesehen werden. Um die ersten Orte zu würdigen, an denen Scrum ausprobiert und verfeinert wurde, stellen wir Individual, Inc., Fidelity Investments und IDX (heute GE Medical) besonders heraus.

Übersetzung

Dieser Guide ist eine Übersetzung der ursprünglichen englischen Version von Ken Schwaber und Jeff Sutherland. Zu dieser Übersetzung beigetragen haben: Sabine Canditt, Andreas Schliep, Stefan Rook und Wolfgang Wiedenroth.

Zielsetzung

Scrum wird seit den frühen 1990ern zur Entwicklung komplexer Produkte eingesetzt. Dieser Leitfaden erklärt, wie man Scrum verwendet, um Produkte zu erstellen. Scrum ist weder ein Prozess noch eine Technik zur Erstellung von Produkten. Es ist eher ein Rahmenwerk, innerhalb dessen diverse Prozesse und Methoden eingesetzt werden können. Die Aufgabe von Scrum besteht darin, die Wirksamkeit Ihrer Entwicklungspraktiken zum Vorschein zu bringen, damit Sie diese verbessern können, und dabei ein Rahmenwerk zur Entwicklung komplexer Produkte zu bieten.

Scrum Theorie

Scrum basiert auf der Theorie der empirischen Prozesssteuerung. Es verwendet einen iterativen, inkrementellen Ansatz, um die Vorhersagbarkeit zu optimieren und Risiken zu beherrschen. Drei Säulen stützen jede Implementierung der empirischen Prozesssteuerung.

Die erste Säule ist Transparenz.

Transparenz sorgt für Sichtbarkeit der Prozessaspekte, die das Ergebnis beeinflussen - für diejenigen, die für die Ergebnisse verantwortlich sind. Diese Aspekte müssen nicht nur transparent sein, sondern es muss auch bekannt sein, was im Blick behalten werden muss. Das bedeutet, wann immer jemand bei der Inspektion eines Prozesses der Meinung ist, dass etwas „done“ (wirklich fertig) ist, muss es seiner Definition von „Done“ entsprechen.

Die zweite Säule ist Inspektion.

Die verschiedenen Aspekte des Prozesses müssen so häufig untersucht werden, dass inakzeptable Abweichungen entdeckt werden können. Bei der Häufigkeit der Inspektion gilt es in Betracht zu ziehen, dass alle Prozesse durch den Vorgang der Inspektion verändert werden. Ein Problem entsteht, wenn die erforderliche Häufigkeit von Inspektionen die Toleranz gegenüber der Inspektion des Prozesses überschreitet.

Das scheint zum Glück nicht für die Softwareentwicklung zuzutreffen. Der andere Faktor ist die Qualifikation und Sorgfältigkeit der Leute, die die Arbeitsergebnisse inspizieren.

Die dritte Säule ist Adaption.

Wenn der Untersuchende während der Inspektion feststellt, dass sich ein oder mehrere Aspekte des Prozesses außerhalb des akzeptablen Rahmens bewegen – und das resultierende Produkt inakzeptabel sein wird – muss der Untersuchende den Prozess oder die Arbeitsgegenstände anpassen. Die Anpassung muss so schnell wie möglich erfolgen, um weitere Abweichungen zu vermeiden.

Es gibt drei Inspektions- und Adaptionenpunkte in Scrum. Das Daily Scrum Meeting wird eingesetzt, um den Fortschritt in Richtung Sprintziel zu inspizieren und Anpassungen vorzunehmen, die den Wert des nächsten Arbeitstages optimieren. Die Sprint-Review und – Planning-Meetings sind dazu gedacht, den Fortschritt in Richtung des Release-Zieles zu inspizieren, und Anpassungen vorzunehmen, die den Wert des folgenden Sprints maximieren. Die Sprint-Retrospektive dient schließlich dazu, den vergangenen Sprint selbst zu untersuchen und Anpassungen vorzunehmen, die den kommenden Sprint produktiver, erfüllender und angenehmer gestalten.

Scrum Inhalte

Das Scrum-Rahmenwerk besteht aus dem Zusammenspiel von **Scrum Teams** und den damit verbundenen Rollen; **Timeboxen, Artefakten** und **Regeln**.

Scrum Teams haben die Aufgabe, Flexibilität und Produktivität zu optimieren. Um das zu erreichen, organisieren sie sich selbst, sind funktionsübergreifend aufgestellt und arbeiten in Iterationen. Jedes Scrum Team hat drei Rollen: 1) den **ScrumMaster**, der dafür verantwortlich ist, dass der Prozess verstanden und eingehalten wird; 2) den **Product Owner**, der dafür verantwortlich ist, den Wert der Teamarbeit zu maximieren; 3) **das Team**, das die Arbeit durchführt. Das Team besteht aus Entwicklern mit all den Fähigkeiten, die

notwendig sind, um die Anforderungen des Product Owners zum Ende jedes Sprints in ein potenziell auslieferbares Produktinkrement umzuwandeln.

Scrum beinhaltet mehrere Timeboxen, um Regelmäßigkeit herzustellen: das **Release-Planning Meeting**, das **Sprint Planning Meeting**, den **Sprint**, das **Daily Scrum**, den **Sprint Review** und die **Sprint-Retrospektive**. Das Herz von Scrum ist der **Sprint**, eine Iteration von einem Monat oder weniger, die während eines Entwicklungsvorhabens ihre Länge beibehält. Alle Sprints verwenden das gleiche Scrum-Rahmenwerk, und alle Sprints erzeugen ein Inkrement des Endprodukts, das potenziell auslieferbar ist. Ein Sprint beginnt unmittelbar nach dem Ende des vorigen Sprints.

Scrum beinhaltet vier wesentliche Artefakte. Das **Product Backlog** ist eine priorisierte Liste von allem, was in dem Produkt gebraucht werden könnte. Das **Sprint Backlog** ist eine Liste von Aufgaben, um aus dem Product Backlog für einen Sprint ein potenziell auslieferbares Produktinkrement zu machen. Ein Burndown-Diagramm misst das verbleibende Backlog über die Zeit. Ein **Release Burndown** misst das verbleibende Product Backlog über den Zeitraum eines Release-Plans. Ein **Sprint Burndown** misst verbleibende Sprint Backlog-Einträge während eines Sprints.

Regeln verknüpfen die Timeboxen, Rollen und Artefakte von Scrum. Die Regeln von Scrum werden in diesem Dokument dargestellt. Zum Beispiel gilt die Regel, dass nur Team-Mitglieder - diejenigen Leute, welche sich der Aufgabe verschrieben haben, das Product Backlog in ein Produkt-Inkrement zu überführen - während eines Daily Scrums

Tipp

Wenn keine Regeln angegeben sind, sollen die Scrum-Anwender selbst entscheiden, wie sie vorgehen. Versuchen Sie nicht eine perfekte Lösung zu finden, da sich das Problem normalerweise schnell ändert. Probieren Sie stattdessen etwas aus, und überprüfen Sie, wie es funktioniert. Die Inspektions- und Adaptionsmechanismen des empirischen Charakters von Scrum werden Sie leiten.

reden dürfen. Möglichkeiten, Scrum zu implementieren, die keine Regeln sind, werden als Vorschläge in den „Tipp-Boxen“ beschrieben.

Scrum Rollen

Das Scrum Team besteht aus dem ScrumMaster, dem Product Owner und dem (Entwicklungs-)Team. Mitglieder des Scrum Teams sehen wir als „Schweine“ an. Der Product Owner ist das „Schwein“ für das Product Backlog. Das Team ist das „Schwein“ für die Arbeit im Sprint. Der ScrumMaster ist das „Schwein“ für den Scrum-Prozess. Hühner dürfen Schweinen nicht vorschreiben, wie sie ihre Arbeit zu tun haben. Die Ausdrücke „Hühner“ und „Schweine“ entstammen der folgenden Geschichte:

Ein Huhn und ein Schwein treffen sich, und das Huhn sagt:

„Lass uns ein Restaurant aufmachen!“

Das Schwein denkt nach und sagt: „Wie würden wir das Restaurant denn nennen?“

Darauf das Huhn: „Schinken und Eier!“.

Das Schwein erwidert: „Nein Danke, ich würde da voll drinstecken, aber du wärest nur beteiligt.“

Der ScrumMaster

Der ScrumMaster muss sicherstellen, dass das Scrum Team die Werte, Praktiken und Regeln von Scrum einhält. Der ScrumMaster hilft dem Scrum Team und der Organisation Scrum einzuführen. Er zeigt dem Team durch Coaching und Anleitung, wie es produktiver werden und Arbeitsergebnisse von höherer Qualität erzielen kann. Der ScrumMaster hilft dem Team, Selbst-Organisation und

Tipp

Der ScrumMaster arbeitet mit den Kunden und dem Management, um den Product Owner zu identifizieren und in seiner Position zu etablieren. Der ScrumMaster vermittelt dem Product Owner, wie er oder sie seine oder ihre Aufgabe erfüllen kann. Product Owner sollen in der Lage sein, die Wertschöpfung bei der Produktentwicklung durch Scrum zu optimieren. Falls Ihnen dafür die erforderlichen Kenntnisse fehlen, nehmen wir den ScrumMaster in die Pflicht.

funktionsübergreifendes Arbeiten zu verstehen und anzuwenden. Der ScrumMaster sorgt dafür, dass das Scrum Team in einer Organisation sein Bestes geben kann, die sich eventuell noch nicht auf die komplexe Produktentwicklung optimal eingestellt hat. Wenn der ScrumMaster diese Veränderungen initiiert, nennen wir es die „Beseitigung von Hindernissen“. Der ScrumMaster entspricht dabei in seiner Rolle einem Servant-Leader für das Scrum Team.

Der Product Owner

Der Product Owner ist die einzige Person, die für das Management des Product Backlog verantwortlich ist und dafür den Wert der geleisteten Arbeit des Teams sicherzustellen. Diese Person pflegt das Product Backlog und stellt sicher, dass es für jedermann zugänglich ist. Jeder weiß, welche Einträge im Backlog die höchste Priorität haben, so dass auch jeder weiß, woran gearbeitet werden wird.

Der Product Owner ist eine Person, nicht ein Komitee. Er kann von einem Komitee

Tipp

Der ScrumMaster kann Teil des Teams sein, zum Beispiel ein Entwickler, der Aufgaben im Sprint erledigt. Dies führt allerdings häufig zu Konflikten, wenn der ScrumMaster sich zwischen dem Beseitigen von Hindernissen und der Arbeit an den Aufgaben entscheiden muss. Der ScrumMaster sollte niemals gleichzeitig der Product Owner sein.

Tipp

Für kommerzielle Entwicklungen kann der Product Owner identisch mit dem Produktmanager sein. Für In-House Entwicklungen könnte der Product Owner der Manager der Geschäftseinheit sein, die durch die Software automatisiert wird.

Tipp

Der Product Owner kann auch als Teammitglied an der Entwicklung arbeiten. Die zusätzliche Verantwortung kann allerdings die Fähigkeit des Product Owners einschränken, mit Stakeholdern zu arbeiten. Jedoch darf der Product Owner niemals gleichzeitig der ScrumMaster sein.

unterstützt oder beeinflusst werden, aber jeder der die Priorität eines Eintrags geändert haben möchte, muss den Product Owner davon überzeugen. Unternehmen, die Scrum einsetzen, werden wahrscheinlich herausfinden, dass es nach einer Weile die Art und Weise beeinflusst, in der Prioritäten gesetzt und Anforderungen definiert werden.

Damit der Product Owner Erfolg haben kann, müssen alle in der Organisation seine oder ihre Entscheidungen respektieren. Niemand darf den Teams vorschreiben, dass sie nach anderen Prioritäten arbeiten müssen – und Teams dürfen sich auf niemanden einlassen, der das Gegenteil behauptet. Die Entscheidungen des Product Owners spiegeln sich im Inhalt und der Priorisierung des Product Backlogs wieder. Diese Sichtbarkeit verlangt dem Product Owner ab, sein oder ihr Bestes zu geben – und macht die Rolle des Product Owners anspruchsvoll und lohnend zugleich.

Das Team

Entwicklungsteams setzen in jedem Sprint Teile des Product Backlogs in potenziell auslieferbare Produktinkremente um. Teams sind funktionsübergreifend, mit all den Fertigkeiten, die für die Herstellung des Produktinkrements erforderlich sind. Teammitglieder haben oft spezielle Fertigkeiten für Programmierung, Qualitätskontrolle, Geschäftsanalyse, Architektur, User Interface Design oder Datenbankentwurf. Doch sind die gemeinsamen Fähigkeiten – die Fähigkeit eine Anforderung anzugehen und in ein nutzbares Produkt umzusetzen – meistens wichtiger als die einzelnen Kenntnisse. Wer sich weigert zu programmieren, weil er Architekt oder Designer ist, passt nicht gut in ein Scrum Team. Jeder ist zur Stelle und gibt sein Bestes, auch wenn es erforderlich ist, neue Fähigkeiten zu erlernen oder sich an alte zu erinnern. Es gibt keine Titel in Scrum Teams – und von dieser Regel gibt es keine Ausnahme. Scrum vermeidet vertikale Teams von Analysten, Designern, Qualitätssicherung und Softwareingenieuren. Teams enthalten auch keine Unter-Teams für spezielle Gebiete wie Test oder Geschäftsanalyse.

Teams organisieren sich zudem selbst. Niemand – auch nicht der ScrumMaster – schreibt dem Team vor, wie es aus dem Product Backlog potenziell auslieferbare Produktinkremente herstellen soll. Das Team erarbeitet dies eigenständig. Jedes Teammitglied wendet seine oder ihre Expertise auf alle Problemstellungen an. Die daraus resultierende Synergie verbessert die gesamte Effektivität und Effizienz des Teams.

Die optimale Teamgröße beträgt sieben Mitglieder, plus oder minus zwei. Bei Teams mit weniger als fünf Mitgliedern beschränkt die geringe Größe die mögliche Interaktion im Team - was die mögliche Produktivitätssteigerung behindert. Außerdem kann das Team eventuell kein auslieferbares Produktinkrement erstellen, weil es an erforderlichen Fertigkeiten fehlt. Für Teams mit mehr als neun Mitgliedern ist der Koordinationsaufwand einfach zu hoch. Große Teams erzeugen zu viel Komplexität für einen empirischen Prozess. Dennoch sind uns erfolgreiche Teams begegnet, die diese obere oder untere Grenze über- bzw. unterschritten haben. Product Owner und ScrumMaster zählen bei der Bestimmung der Teamgröße nicht mit, außer sie arbeiten selbst produktiv an der Umsetzung des Sprint-Backlogs mit.

Die Teamzusammensetzung kann sich am Ende eines Sprints ändern. Allerdings führt jede Änderung der Mitgliedschaft zu einer Minderung der Produktivität, die man durch Selbstorganisation erlangt hat. Die Änderung der Teamzusammensetzung sollte mit Vorsicht vorgenommen werden.

Timeboxen

Die Timeboxen in Scrum sind: das **Release Planning Meeting**, der **Sprint**, das **Sprint Planning Meeting**, der **Sprint Review**, die **Sprint-Retrospektive** und das **Daily Scrum**.

Release Planning Meeting

Der Zweck der Release-Planung besteht darin, einen Plan und Ziele zu etablieren, die das Scrum Team und die übrige Organisation

beiderseitig verstehen und kommunizieren können. Die Release-Planung beantwortet die Frage: „Wie können wir so gut wie möglich von der Vision zu einem erfolgreichen Produkt kommen? Wie können wir dabei die erwünschte Kundenzufriedenheit und Return On Investment erreichen oder übertreffen?“ Der Release-Plan legt das Ziel für das Release fest, die am höchsten priorisierten Backlog-Einträge, die Hauptrisiken und die generellen Features und Funktionalitäten im Release-Umfang. Er ermittelt auch einen wahrscheinlichen Lieferzeitpunkt und die veranschlagten Kosten, falls keine Änderungen aufkommen. Die Organisation kann dann den Fortschritt inspizieren und Sprint für Sprint Anpassungen vornehmen.

Das Release-Planning ist als optional anzusehen. Wenn Teams ohne dieses Meeting starten, erscheint das Fehlen seiner Arbeitsergebnisse als Hindernis, das es zu lösen gilt. Die Arbeit zur Beseitigung des Hindernisses wird als Eintrag im Product Backlog erscheinen.

Produkte werden mit Scrum iterativ erstellt, wobei jeder Sprint ein Inkrement des Produkts erzeugt – angefangen mit den wertvollsten und riskantesten Inkrementen. Weitere Sprints erzeugen zusätzliche Inkremente des Produkts. Jedes Inkrement ist ein potenziell auslieferbarer Bestandteil des Gesamtprodukts. Wenn genügend Inkremente für das Produkt erzeugt wurden, um einen Wert, einen Nutzen für die Investoren darzustellen, wird das Produkt ausgeliefert.

Die meisten Organisationen haben bereits einen Release-Planungs-Prozess. Meistens wird dabei der Hauptteil der Planung zu Beginn eines Releases vorgenommen und im Projektverlauf nicht geändert. In der Scrum-Release-Planung werden ein übergreifendes Ziel und mögliche Resultate festgelegt. Diese Release-Planung benötigt meistens nicht mehr als 15 – 20 % der für einen traditionellen Release-Plan aufgewendeten Zeit. Auf der anderen Seite werden in jedem Sprint-Planning- und Review-Meeting Just-In-Time-Planungen vorgenommen, so wie die tägliche Just-In-Time-Planung im Daily Scrum Meeting. In der Summe werden also die Aufwände für die Release-Planung in

Scrum wahrscheinlich die traditionellen Release-Planungsaufwände leicht übersteigen.

Die Releaseplanung erfordert die Schätzung und Priorisierung des Product Backlogs für das jeweilige Release. Es gibt diverse Techniken für diese Aufgabenstellung, die außerhalb des Definitionsbereichs von Scrum liegen, aber dennoch von hohem Nutzen sind.

Der Sprint

Ein Sprint ist eine Iteration. Sprints sind in Timeboxen, festen Zeitabschnitten, organisiert. Während des Sprints unterbindet der ScrumMaster alle Änderungen, die das Sprint-Ziel gefährden. Sowohl Teamzusammensetzung als auch Qualitätsziele bleiben während des Sprints unverändert. Sprints beinhalten und bestehen aus dem Sprint Planning Meeting, der Entwicklungsarbeit, dem Sprint-Review und der Sprint-Retrospektive. Sprints folgen direkt aufeinander, ohne zeitlichen Abstand zwischen den Sprints.

Ein Projekt wird angegangen, um ein bestimmtes Ziel zu erreichen. In der Softwareentwicklung handelt es sich dabei um ein Produkt oder System. Jedes Projekt besteht aus einer Definition was gebaut werden soll, einem Plan nach dem es gebaut werden soll, der dem Plan entsprechenden Arbeit sowie dem resultierenden Produkt. Jedes Projekt hat einen Horizont - den Zeitrahmen, für den der Plan angemessen ist. Wenn der Horizont zu weit ist, kann sich die Definition inzwischen geändert haben, zu viele

Tipp

Wenn das Team den Eindruck hat, sich zuviel vorgenommen zu haben, trifft es sich mit dem Product Owner, um den Umfang des für den Sprint ausgewählten Product Backlogs zu reduzieren. Wenn das Team das Gefühl hat, noch Zeit übrig zu haben, kann es mit dem Product Owner daran arbeiten, sich zusätzliche Aufgaben aus dem Product Backlog zu suchen.

Tipp

Wenn ein Team neu mit Scrum beginnt, erlauben ihm zweiwöchige Sprints zu lernen, ohne in Unsicherheit unterzugehen. Sprints dieser Länge können mit denen der anderen Teams synchronisiert werden, indem zwei Sprints zusammengefügt werden.

Ungewissheiten können sich eingeschlichen haben, das Risiko kann zu groß werden. Scrum ist ein Rahmenwerk für ein Projekt, dessen zeitlicher Horizont maximal ein Monat ist - in dem die Komplexität so hoch ist, dass ein längerer Horizont zu riskant wäre. Die Vorhersagbarkeit des Projektes muss zumindest jeden Monat kontrolliert werden. Das Risiko, dass das Projekt außer Kontrolle gerät oder unvorhersagbar wird, wird mindestens jeden Monat eingedämmt.

Sprints können vor dem Ablauf ihrer Timebox abgebrochen werden. Die Autorität zum Abbruch eines Sprints liegt einzig und allein beim Product Owner – auch wenn das eventuell unter dem Einfluss von Stakeholdern, dem Team oder dem ScrumMaster geschieht. Unter welchen Umständen kann die Situation aufkommen, dass ein Sprint abgebrochen werden muss? Das Management kann einen Sprint abbrechen, wenn das Sprint-Ziel obsolet wird. Das kann passieren, wenn ein Unternehmen insgesamt seine Ausrichtung ändert oder die aktuellen Marktbedingungen oder technologische Anforderungen eine schnelle Reaktion erfordern. Generell sollte ein Sprint dann abgebrochen werden, wenn die Fortführung unter den gegebenen Umständen keinen Sinn mehr ergibt. Dennoch ist es aufgrund der kurzen Dauer von Sprints selten sinnvoll, diese Maßnahme zu ergreifen.

Wenn ein Sprint abgebrochen wird, werden alle abgeschlossenen und „fertigen“ Backlog-Einträge untersucht. Sie werden abgenommen, wenn sie ein potenziell auslieferbares Produktinkrement darstellen. Alle anderen Backlog-Einträge werden mit ihren ursprünglichen Schätzungen wieder in das Product Backlog eingestellt. Sämtliche an ihnen verrichtete Arbeit wird als verloren angesehen. Sprint-Abbrüche verbrauchen Ressourcen, weil sich alle in einem weiterem Sprint-Planning-Meeting neu organisieren müssen, um einen neuen Sprint zu starten. Sprint-Abbrüche sind oft schmerzhaft für das Scrum Team, daher sind sie sehr unüblich.

Sprint Planning Meeting

Im Sprint Planning Meeting findet die Iterationsplanung statt. Es ist zeitlich auf 8 Stunden für einen einmonatigen Sprint begrenzt. Für kürzere Sprints sollten Sie proportional zur gesamten Sprintlänge weniger Zeit für dieses Meeting reservieren (zum Beispiel setzt man für zweiwöchige Sprints ein vierstündiges Sprint Planning Meeting an). Es besteht aus zwei Teilen. Im ersten Teil, wird festgelegt, was in dem Sprint erledigt werden soll. Im zweiten Teil (der maximal vier Stunden für einen vierwöchigen Sprint dauern sollte), macht sich das Team darüber Gedanken, wie diese Funktionalität während des Sprints in ein Produktinkrement umgesetzt wird.

Das Sprint Planning Meeting besteht also aus zwei Teilen, dem „Was“ und dem „Wie“-Teil. Einige Teams vereinen die beiden Teile. Im ersten Teil geht das Team die „Was“-Frage an. Hier präsentiert der Product Owner dem Team die am höchsten priorisierten Product Backlog-Einträge. Sie finden zusammen heraus, welche Funktionalität im nächsten Sprint entwickelt werden soll. Als Input für dieses Meeting dienen das Product Backlog, das letzte Produktinkrement, die Kapazität des Teams und die letzte Performance des Teams. Die Menge der Einträge, die das Team auswählt, hängt einzig und allein vom Team ab. Nur das Team kann einschätzen, was es im Laufe des kommenden Sprints bewerkstelligen kann.

Nach der Auswahl des Product Backlogs wird ein Sprint-Ziel entworfen. Das Sprint-Ziel ist ein Ziel, das durch die Implementierung des Product Backlogs erreicht werden soll. Es handelt sich um eine Aussage, die dem Team vor Augen hält, warum es dieses Inkrement erstellt. Das Sprint-Ziel stellt eine Untermenge des Release-Ziels dar.

Der Grund für die Aufstellung des Sprintziels ist es, dem Team etwas Spielraum in Bezug auf die Funktionalität zu geben. Zum Beispiel könnte das Ziel des oben erwähnten Sprints lauten: „Automatisiere die Funktionalität für Kundenkonto-Modifikationen durch eine sichere, wieder herstellbare Transaktions-Middleware-Funktionalität.“ Bei der Arbeit behält das Team dieses Ziel im Kopf. Um das Ziel zu erfüllen,

implementiert das Team die entsprechende Funktionalität und Technologie. Wenn sich die Arbeit als aufwändiger herausstellt, als das Team erwartet hatte, erarbeitet es zusammen mit dem Product Owner eine Alternative und implementiert die Funktionalität nur teilweise.

Im zweiten Teil des Sprint Planning Meetings beschäftigt sich das Team mit der Frage „Wie?“. Während des zweiten Teils des Sprint Planning Meetings (eine vierstündige Timebox für einen Monatsprint) überlegt sich das Team, wie es das im Sprint Planning 1 ausgewählte Product Backlog (Was) in ein „fertiges“ Inkrement überführen kann. Das Team startet normalerweise mit dem Design der Arbeit. Während es das Design erstellt, identifiziert es Aufgaben (Tasks). Diese Aufgaben sind die detaillierten Arbeitsschritte, die man zur Überführung des Product Backlogs in lauffähige Software benötigt. Aufgaben sollten so heruntergebrochen werden, dass sie in weniger als einem Tag erledigt werden können. Diese Aufgabenliste nennt man das Sprint Backlog. Das Team organisiert sich selbst, um die Arbeit am Sprint Backlog durchzuführen, entweder noch im Sprint Planning oder nach Bedarf während des Sprints.

Tipp

Normalerweise werden 60-70% des endgültigen Sprint Backlogs im Sprint Planning Meeting ausgearbeitet. Der Rest wird zur späteren Verfeinerung grob umrissen oder mit großen Schätzungen versehen, um dann im Sprint weiter heruntergebrochen zu werden.

Der Product Owner steht während dieses zweiten Teils des Sprint Planning Meetings zur Verfügung, um Fragen zum Product Backlog zu klären oder bei notwendigen Abstrichen zu helfen. Wenn das Team erkennt, dass es zu viel oder zu wenig Arbeit hat, kann es das Product Backlog mit dem Product Owner neu aushandeln. Das Team kann außerdem Dritte einladen, um sich Rat bezüglich der Technologie oder Domäne einzuholen. Ein neues Team wird oft zum ersten Mal in diesem Meeting erkennen, dass es nur als Team – nicht als Individuen – besteht oder untergeht. Das Team erkennt, dass es sich auf sich selbst verlassen muss. Mit dieser Erkenntnis fängt es an, sich selbst zu

organisieren, um die Charakteristiken und das Verhalten eines wirklichen Teams einzunehmen.

Sprint Review

Am Ende des Sprints wird ein Sprint Review-Meeting abgehalten. Es handelt sich um ein auf vier Stunden beschränktes Meeting für einmonatige Sprints. Für kürzere Sprints sollten Sie dementsprechend weniger Zeit ansetzen (zum Beispiel ein zweistündiges Sprint Review Meeting für zweiwöchige Sprints). Während des Sprint Reviews arbeiten das Scrum Team und Stakeholder zusammen an der Begutachtung der Arbeitsergebnisse. Auf dieser Basis – und den Änderungen am Product Backlog während des Sprints – erarbeiten sie die nächsten möglichen Schritte. Das Sprint Review ist ein informelles Meeting; die Vorführung der Funktionalität ist dazu gedacht, die Zusammenarbeit für die weiteren Aktivitäten zu fördern.

Das Meeting beinhaltet zumindest die folgenden Elemente. Der Product Owner identifiziert, was fertig gestellt wurde, und was nicht. Das Team diskutiert, was während des Sprints funktioniert hat und wobei es Probleme gegeben hat; und wie diese Probleme gelöst worden sind. Das Team demonstriert dann die geleistete Arbeit und beantwortet Fragen dazu. Der Product Owner diskutiert dann den aktuellen Stand des Product Backlogs. Er oder sie ermittelt wahrscheinliche Fertigstellungsdaten anhand verschiedener Annahmen über die Geschwindigkeit des Teams. Schließlich erarbeiten alle Teilnehmer, was ihr Eindruck über das gerade Gesehene ist und was es in Bezug auf die weitere Planung bedeutet. Das Sprint Review bietet wertvolle Informationen für das folgende Sprint Planning- Meeting.

Sprint-Retrospektive

Zwischen dem Sprint Review und dem nächsten Sprint Planning hält das Scrum Team eine Sprint-Retrospektive ab. Die Sprint Retrospektive ist für monatliche Sprints auf drei Stunden begrenzt (setzen Sie dementsprechend weniger Zeit bei kürzeren Sprints an). In diesem Meeting ermutigt der ScrumMaster das Team dazu, seinen

Entwicklungsprozess innerhalb des Scrum-Prozess-Rahmenwerks und der Praktiken zu revidieren, um ihn für den kommenden Sprint effektiver und angenehmer zu gestalten. Es gibt viele Bücher, in denen nützliche Techniken für Retrospektiven dokumentiert sind.

Der Sinn der Retrospektive ist es zu inspizieren, wie der letzte Sprint in Bezug auf die Menschen, Beziehungen, den Prozess und die Werkzeuge gelaufen ist. Die Inspektion sollte die wesentlichen Punkte identifizieren und priorisieren, die gut funktioniert haben - und jene die noch besser gelaufen wären, wenn sie anders angegangen worden wären. Die Themen beinhalten die Teamzusammensetzung, Meeting-Arrangements, Werkzeuge, die Definition von „Done“, Kommunikationsmethoden sowie die Prozesse, um aus dem Product Backlog etwas potenziell auslieferbares herzustellen. Am Ende der Sprint Retrospektive sollte das Scrum Team umsetzbare Verbesserungsmaßnahmen identifiziert haben, die sie im nächsten Sprint in die Tat umsetzen. Diese Änderungen entsprechen der Adaption zur empirischen Inspektion.

Daily Scrum

Jedes Scrum Team trifft sich täglich zu einem 15-minütigem Inspektions- und Adaptionmeeting, dem Daily Scrum. Das Daily Scrum findet während des Sprints stets zur gleichen Zeit und am gleichen Ort statt. Während des Meetings erklärt jedes Teammitglied:

1. Was er oder sie seit dem letzten Meeting erreicht hat;
2. Was er oder sie bis zum nächsten Meeting tun wird;
3. Welche Hindernisse ihm oder ihr im Weg stehen.

Daily Scrums verbessern die Kommunikation, machen andere Meetings überflüssig, identifizieren und beseitigen Hindernisse für die Entwicklung, betonen und fördern die schnelle Entscheidungsfindung und verbessern den Kenntnisstand über das Projekt für alle.

Der ScrumMaster sorgt dafür, dass das Team dieses Meeting durchführt. Das Team ist für die tatsächliche Durchführung des Daily Scrums verantwortlich. Der ScrumMaster bringt dem Team bei, wie es das Daily Scrum kurz hält, indem er die Regeln durchsetzt und dafür sorgt, dass sich die Teilnehmer kurz fassen. Der ScrumMaster setzt auch die Regel durch, dass „Hühner“ während des Daily Scrums nicht sprechen oder sich auf andere Weise in das Daily Scrum einmischen.

Das Daily Scrum ist kein Status-Meeting. Es ist nicht für jeden gedacht, sondern nur für die Mitarbeiter, die aus den Product Backlog-Einträgen ein Produkt-Inkrement erstellen (das Team). Das Team hat sich einem Sprint-Ziel und diesen Backlog-Einträgen verschrieben. Das Daily Scrum ist eine Inspektion des Fortschritts in Richtung auf dieses Sprintziel (die drei Fragen). Folgemeetings werden häufig angesetzt, um Anpassungen an der aufkommenden Arbeit im Sprint vorzunehmen. Die Zielsetzung ist es, die Wahrscheinlichkeit zu erhöhen, dass das Team sein Ziel erreicht. Somit handelt es sich um ein Schlüsselmeeting zur Inspektion und Adaption im empirischen Scrum-Prozess.

Scrum Artefakte

Die Scrum Artefakte beinhalten das Product Backlog, das Release Burndown Diagramm, das Sprint Backlog sowie das Sprint Burndown Diagramm.

Tipp

Product Backlog-Einträge werden häufig in der Form von User Stories notiert. Use Cases sind ebenfalls anwendbar, sollten aber besser für lebens- oder missionskritische Software verwendet werden.

Product Backlog und Release Burndown

Die Anforderungen für das vom Scrum Team / den Teams zu entwickelnde Produkt sind im Product Backlog aufgelistet. Der Product Owner ist für das Product Backlog und dessen Inhalte, Verfügbarkeit und Priorisierung verantwortlich. Ein Product Backlog ist niemals vollständig und zeigt im ersten Schritt nur die anfangs bekannten und gut verstandenen Anforderungen auf. Das

Product Backlog entwickelt sich weiter, wie das Produkt und die Umgebung, in der es eingesetzt werden soll, sich weiter entwickeln. Ein Backlog ist insofern dynamisch, dass es sich beständig ändert, um Anforderungen bzgl. der Angemessenheit, Wettbewerbsfähigkeit und Nützlichkeit des Produkts zu identifizieren. So lange ein Produkt existiert, existiert auch das Product Backlog.

Das Product Backlog enthält alles, was für die Entwicklung und die Inbetriebnahme eines erfolgreichen Produkts erforderlich ist. Es listet alle Features, Funktionalitäten, Technologien, Verbesserungen und Fehlerbehebungen, die die Änderungen an dem Produkt in zukünftigen Releases ausmachen. Die Einträge im Product Backlog haben die Attribute Beschreibung, Priorität und Schätzung. Die Priorität wird durch Risiko, Wert und Notwendigkeit (nicht-funktionale Anforderungen) bestimmt. Es gibt viele Techniken, diese Attribute aufzustellen.

Das Product Backlog ist nach der Priorität sortiert. Die Einträge mit der höchsten Priorität treiben die unmittelbaren Entwicklungsaktivitäten. Je höher die Priorität, desto dringender ist es, desto mehr hat man darüber nachgedacht, desto mehr Konsens besteht über den Wert eines Eintrags. Höher priorisierte Backlog-Einträge sind klarer und mit mehr Detailinformationen versehen als niedriger priorisierte. Größere Klarheit und Detailtiefe führen zu besseren Schätzungen. Je niedriger die Priorität, desto weniger Details – bis hin zu sehr vagen Einträgen.

Tipp

Scrum Teams wenden oft 10% ihrer Zeit für die Pflege des Product Backlogs auf, damit die oben genannten Anforderungen an das Product Backlog erfüllt werden. Nach der Präzisierung auf diese Detailtiefe werden die obersten Backlog-Einträge (höchste Priorität, größter Wert) weiter herunter gebrochen, so dass sie in einen Sprint passen. Bei der Pflege des Backlogs wurden diese Einträge analysiert und durchdacht. Im Sprint Planning Meeting sind solche Einträge gut verstanden und können leicht für den Sprint ausgewählt werden.

Mit der Nutzung eines Produkts, seiner Wertsteigerung und dem Feedback des Marktes entwickelt sich das Product Backlog hin zu einer längeren und ausführlicheren Liste. Anforderungen hören nie auf, sich zu ändern. Das Product Backlog ist ein lebendes Dokument. Änderungen an Geschäftsanforderungen, Marktgegebenheiten, Technologiedurchdringung und Personalstruktur führen zu Anpassungen des Product Backlogs. Zur Minimierung von Nacharbeiten werden nur die Einträge mit der höchsten Priorität detailliert oder weiter aufgebrochen. Die Backlog-Einträge, die die Teams in den kommenden Sprints beschäftigen werden, werden weiter aufgebrochen: so aufgeteilt, dass jeder Eintrag innerhalb der Laufzeit eines Sprints abgeschlossen werden kann

Häufig arbeiten mehrere Scrum Teams zusammen an dem gleichen Produkt. Ein gemeinsames Product Backlog wird verwendet, um die anstehende Arbeit zu beschreiben. Ein zusätzliches Attribut dient der Gruppierung von Backlog-Einträgen. Die Gruppierung kann dabei nach Feature Sets, Technologie oder Architektur erfolgen – eine geläufige Grundlage der Aufteilung der Arbeit je nach Scrum Team.

Tipp

Abnahmetests werden häufig als weiteres Attribut im Produkt Backlog verwendet. Sie können detailliertere Informationen mit einer testbaren Beschreibung enthalten, die der Backlog-Eintrag zur Abnahme erfüllen muss.

Der Release Burndown Graph zeichnet die Summe des verbleibenden geschätzten Aufwands im Product Backlog über die Zeit auf. Der geschätzte Aufwand wird in der jeweils von Scrum Team und Organisation vereinbarten Einheit gemessen. Die Zeiteinheiten sind üblicherweise Sprints.

Schätzungen von Product Backlog-Einträgen werden anfangs während der Release-Planung vorgenommen, bei neuen Backlog-Einträgen bei der Erstellung. Während der Überarbeitung des Product Backlogs werden sie begutachtet und angepasst. Zudem können sie jederzeit aktualisiert werden. Das Team ist für alle Schätzungen verantwortlich. Der Product Owner kann seinen Einfluss darauf ausüben, indem er beim Verständnis und bei der Auswahl von Einschränkungen hilft, aber die endgültige Schätzung obliegt dem Team. Der Product Owner kommuniziert jederzeit eine aktualisierte Product Backlog-Liste und ein Release Burndown. Eine Trendlinie kann auf der Basis der Änderungen an der verbleibenden Arbeit gezogen werden.

Tipp

In einigen Organisationen wird dem Backlog mehr Arbeit hinzugefügt, als fertig gestellt wird. Das kann zu einer flachen oder sogar ansteigenden Trendlinie führen. Um dies zu kompensieren und für Transparenz zu sorgen, kann eine neue Basislinie angesetzt werden, wenn Arbeit hinzugefügt oder weggenommen wird. Diese Basislinie sollte nur um signifikante Änderungen erhöht oder herabgesetzt werden, was gut dokumentiert werden sollte.

Tipp

Die Trendlinie kann für die ersten zwei oder drei Sprints eines Releases noch unzuverlässig sein, außer das Team hat vorher bereits zusammen gearbeitet, kennt das Produkt gut und hat ein Verständnis für die zugrunde liegende Technologie.

Sprint Backlog und Sprint Burndown

Das Sprint Backlog besteht aus den Aufgaben, die das Team zur Überführung von Product Backlog-Einträgen in ein „fertiges“ Inkrement erfüllen muss. Viele dieser Aufgaben werden im Sprint Planning Meeting (Wie) aufgestellt. Es handelt sich um die gesamte Arbeit, die das Team zur Erreichung des Sprint-Ziels als notwendig ansieht. Die Aufteilung ist vollständig genug, so dass Änderungen des Fortschritts

im Daily Scrum bemerkt werden können. Daher sollte ein Eintrag im Sprint Backlog üblicherweise nicht größer als ein Tag sein.

Das Team verändert das Sprint Backlog während des Sprints, so wie sich das Sprint Backlog während des Sprints entwickelt. Bei der Arbeit an einzelnen Aufgaben kann das Team herausfinden, dass mehr oder weniger Arbeitsschritte erforderlich sind, oder dass ein Arbeitsschritt mehr oder weniger Zeit als veranschlagt beanspruchen wird. Wenn neue Aufgaben auftauchen, werden sie dem Sprint Backlog vom Team hinzugefügt. Wenn Aufgaben bearbeitet oder abgeschlossen werden, wird die geschätzte verbleibende Arbeit aktualisiert. Wenn Aufgaben als überflüssig erkannt werden, werden sie aus dem Sprint Backlog entfernt. Nur das Team darf sein Sprint Backlog während des Sprints ändern. Nur das Team kann die Inhalte oder die Schätzungen ändern. Das Sprint Backlog ist ein deutlich sichtbares Echtzeitabbild der Arbeit, die das Team im Sprint bewältigen will; es gehört einzig und allein dem Team.

Tipp

Wann immer es geht, zeichnen Sie den Sprint Burndown Graphen auf ein großes Blatt Papier, das im Teambereich aushängt. Teams sehen eher ein großes, sichtbares Diagramm, als ein Sprint Burndown Diagramm in Excel oder einem anderen Werkzeug.

Das Sprint Burndown Diagramm zeigt die Menge an Arbeit im Sprint Backlog, die über die Zeit in einem Sprint noch ansteht. Um diesen Graphen zu erstellen, ermitteln Sie an jedem Tag im Sprint die Menge der restlichen Arbeit durch die Aufsummierung der Sprint-Backlog-Schätzungen. Die Menge der Arbeit, die für einen Sprint noch ansteht ist das gleiche wie die Summe der verbleibenden Arbeit für das gesamte Sprint Backlog. Notieren Sie diese Summen jeden Tag, um einen Graph zu erstellen, der die restliche Arbeit im Zeitverlauf darstellt. Das Team kann seinen Fortschritt beim Fertigstellen der Aufgaben des Sprints verfolgen, indem es eine Linie durch die Punkte im Graphen zieht. Die Bearbeitungsdauer wird in Scrum nicht betrachtet. Verbleibende Arbeit und das Datum sind die einzigen Variablen von Interesse.

Eine der Regeln von Scrum betrifft das Ziel jedes Sprints, Inkremente von potenziell auslieferbarer Funktionalität zu liefern, die der vereinbarten Definition von „Done“ genügen.

Done

Scrum verlangt von Teams, dass sie in jedem Sprint ein Inkrement an Produktfunktionalität bauen. Dieses Inkrement muss potenziell auslieferbar sein, damit der Product Owner die Möglichkeit hat, die Funktionalität sofort in den Einsatz zu bringen. Deshalb muss das Inkrement ein kompletter Teil des Produkts sein. Es muss „done“ sein. Jedes Inkrement sollte auf den vorigen aufsetzen und gründlich getestet sein, damit die Integration aller Inkremente gewährleistet ist.

Bei der Produktentwicklung gehen einige davon aus, dass Funktionalität, die als „done“ bezeichnet wird, zumindest sauber codiert, refaktoriert, mit Unit Tests versehen, gebaut und abgenommen ist. Andere könnten annehmen, dass nur der Code erstellt wurde. Wenn nicht jeder weiß, was die Definition von „done“ bedeutet, dann funktionieren die anderen beiden

Säulen der empirischen Prozesssteuerung nicht. Wann immer jemand etwas als „done“ bezeichnet, muss jeder verstehen was „done“ heisst.

„Done“ ist die Definition dessen, was das Team meint, wenn es sich dazu verpflichtet, einen Product Backlog-Eintrag in einem Sprint zu „erledigen“. Einige Produkte enthalten keine Dokumentation, so dass ihre Definition von „done“ nicht die Dokumentation beinhaltet. Ein komplett „done“ Inkrement enthält die Analyse, das Design, die Überarbeitung, die Programmierung, die Dokumentation und die Tests für das Inkrement und alle darin enthaltenen Backlog-Einträge. Das Testen umfasst Unit-, System-, Benutzer- und Regressionstests wie

Tipp

„Undone“ Arbeiten werden häufig in einem Product Backlog-Eintrag namens „Undone Work“ oder „Implementierungsaufgaben“ gesammelt. Mit der Ansammlung dieser Aufwände bleibt das Product Backlog Burndown-Diagramm akkurater, als wenn die Arbeit nicht aufgerechnet werden würde.

auch die Überprüfung der nicht-funktionalen Anforderungen wie Performance, Stabilität, Sicherheit und Integration. „Done“ beinhaltet alle Internationalisierung. Einige Teams sind noch nicht in der Lage, alles was für die Implementierung erforderlich ist, in ihre Definition von „done“ aufzunehmen. Das muss dem Product Owner klar sein. Diese übrige Arbeit muss abgeschlossen werden, bevor das Produkt implementiert und genutzt werden kann.

ABSCHLIESSENDE GEDANKEN

Einige Organisationen sind nicht in der Lage, ein komplettes Inkrement innerhalb eines Sprints fertig zu stellen. Sie haben eventuell noch nicht die Infrastruktur für automatisierte Tests zur Verfügung, um alle Tests abzuschließen. In diesem Fall werden für jedes Inkrement zwei Kategorien angelegt: „done“ und „undone“ Arbeit. Die „undone“ Arbeit ist der Anteil an jedem Inkrement, der zu einem späteren Zeitpunkt fertig gestellt werden muss. Der Product Owner weiß ganz genau, was er oder sie am Ende des Sprints inspizieren wird, da das Inkrement der Definition von „done“ entspricht – und der Product Owner die Definition versteht. „Unfertige“ Arbeit wird einem Product Backlog-Eintrag namens „Undone Work“ zugeordnet, damit sie mit eingerechnet und in dem Release Burndown Graphen widergespiegelt werden kann. Diese Vorgehensweise erzeugt Transparenz in Bezug auf den Fortschritt in Richtung eines Releases. Die Inspektion und Adaption im Sprint Review ist so akkurat wie diese Transparenz.

Wenn ein Team zum Beispiel nicht in der Lage ist, Performance-, Regressions-, Stabilitäts- und Integrationstests für jeden Backlog-Eintrag vorzunehmen, so wird der Anteil dieser unerledigten Arbeiten zu der handhabbaren Arbeit (Analyse, Design, Refaktorisierung, Programmierung, Dokumentation) berechnet. Nehmen wir einmal an, diese Proportion beträgt sechs Teile „done“ und vier Teile „undone“. Wenn das Team einen Backlog-Eintrag von sechs Arbeitseinheiten fertig gestellt hat, werden vier zu dem Backlog-Eintrag „Unfertige Arbeit“ hinzugezählt.

Die „undone“ (nicht fertig gestellte) Arbeit sammelt sich Sprint für Sprint an und muss angegangen werden, bevor das Produkt ausgeliefert werden kann. Die Arbeit wird linear aufsummiert- obwohl sie tatsächlich einen exponentiellen Anstieg haben kann, je nach den Charakteristiken der Organisation. Release Sprints werden zum Ende eines jeden Releases eingeplant, um diese „undone“ Arbeit zu erledigen. Die Anzahl der Sprints ist bis zu dem Grad unvorhersagbar, zu dem der Anteil der „undone“ Arbeit nicht-linear ansteigt.