Scrum in der Produktwartung

Martin Heilemann
Lynx-Consulting GmbH
Bielefeld

Schlüsselworte:
Produktwartung, Scrum, Softwarequalität, Software Maintenance

Einleitung
Das Vorgehensmodell Scrum wird schon seit einiger Zeit erfolgreich in der Softwareentwicklung eingesetzt. Dieser Vortrag beschreibt wie sich die Produktwartung einer Applikation vom Arbeiten mit langfristiger Planung in ein agiles Vorgehen mit Scrum entwickelt.

Was ist Scrum?
Scrum entwickelte sich um 1991 geprägt durch die schlanke Produktentwicklung japanischer Unternehmen. Scrum ist ein agiler Prozess für Softwareentwicklung und Projektmanagement. Durch den Einsatz von Scrum wird eine Organisation befähigt flexibel, aktiv, anpassungsfähig und mit Initiative zu agieren und auf ständig veränderte (Markt-)Bedingungen zu reagieren.
Scrum ist ein agiles Managementframework zur Entwicklung von Software, das aus wenigen klaren Regeln besteht: Mit den drei Rollen Product Owner, Team und ScrumMaster, der Verwendung eines priorisierten Product Backlogs und dem Erstellen von Produktinkrementen innerhalb eines Sprints. Scrum ist nicht technologie- oder toolorientiert und basiert auf einer engen Zusammenarbeit zwischen den drei Rollen. Scrum schafft ein hohes Maß an Transparenz aller Aktivitäten, die zur Erstellung einer Software notwendig sind.
Das Scrum Projekt besteht aus mehreren Iterationen, den sogenannten Sprints. Jede Iteration wandelt Anforderungen aus dem Product Backlog in ein auslieferbares Produktinkrement. Das Product Backlog enthält alle Funktionen und ihre technischen Abhängigkeiten des zu entwickelnden Produkts. Die Dauer eines Sprints beträgt maximal 30 Tage und endet zu einem vereinbarten Termin (SC09).

Die Rollen
Product Owner
Der Product Owner erfasst und beschreibt die Anforderungen der Kunden, bearbeitet das Product Backlog kontinuierlich und priorisiert die Anforderungen. Der Auslieferungszeitpunkt, die Funktionalität und die Kosten steuert der Product Owner. In dieser Rolle repräsentiert er die Kundenbedürfnisse und ist für das erreichen der Projektziele verantwortlich.

Team
Das Team führt alle Arbeiten aus, die zur Umsetzung der Anforderungen in auslieferbare Produktinkremente notwendig sind. Das Team ist mit unterschiedlichen Funktionen besetzt und organisiert sich selbst. Welche Anforderungen innerhalb eines Sprints als Produktinkrement geliefert werden entscheidet das Team, ebenso organisiert es die notwendigen Arbeitsschritte.

ScrumMaster
Der ScrumMaster ist der Coach. Er unterstützt das Team und das Unternehmen Scrum einzusetzen.
Weitere Aufgaben sind die Unterstützung des Product Owners bei der Erstellung des Product Backlogs oder der Beseitigung von Hürden im Sprint. Er sorgt dafür dass der Product Owner und das Team direkt zusammenarbeiten.

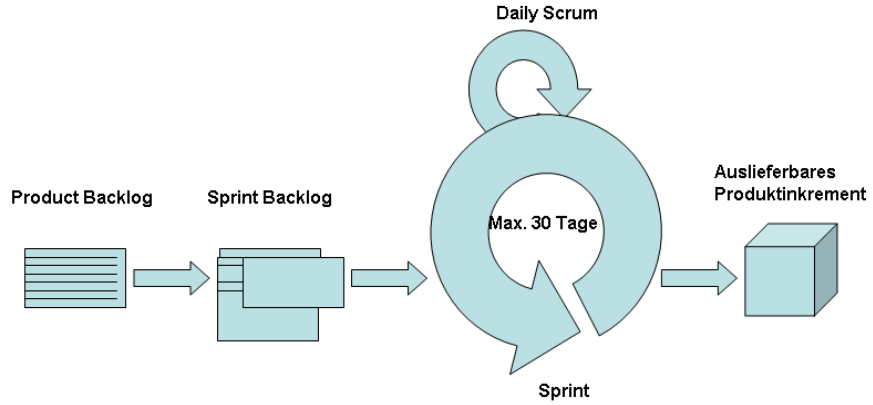

Abb. 1: Scrum im Überblick

Artefakte und Meetings von Scrum

Product Backlog

Das Product Backlog enthält die Anforderungen und Arbeitsergebnisse, die für das Projektziel umgesetzt oder erreicht werden müssen.

Sprint Backlog

Das Sprint Backlog enthält das Sprint-Ziel und alle Aktivitäten für die Umsetzung der Anforderungen, zu denen sich das Team während der Sprint Planungssitzung verpflichtet hat. Das Team erstellt in der Sprint Planungssitzung das Sprint Backlog. Jede Anforderung eines Arbeitspakets wird auf einzelne Aktivitäten heruntergebrochen und mit jeweils zuständigem Bearbeiter im Sprint Backlog geführt. Dazu eignet sich eine Stellwand, hier kann das Sprint Backlog mit Karteikarten dokumentiert werden Das Team aktualisiert im Sprint Backlog die Aktivitäten und schätzt die Restaufwande.

Prio	Anforderung	Zu erledigen	In Arbeit	Erledigt
1	Anforderung A	Aktivität A3 Aktivität A4	Aktivität A1 (Hugo) Aktivität A2 (Paul)	Aktivität 1
2	Anforderung C	Aktivität C1 Aktivität C2		
3				

Abb. 2: Sprint Backlog

Sprint Burndown Chart

Der Sprint Burndown Bericht ist eine graphische Darstellung und zeigt die Verringerung des Restaufwands je Tag für den Sprint. Täglich aktualisiert der ScrumMaster oder das Team den Bericht mit den geschätzten Restaufwänden. Im Idealfall fällt die Kurve kontinuierlich und ist am Ende des Sprints bei Null. Durch diese Darstellungsform lässt sich leicht erkennen, ob der ursprünglich geschätzte Aufwand eingehalten werden kann.

Sprint

In einem Sprint (max. 30 Tage) erstellt das Team aus den Anforderungen ein Produktinkrement. Der Sprint startet mit einer Sprint Planungssitzung. Das Team beschließt in dieser Sitzung welche Anforderungen, untergliedert in Aktivitäten in einem Sprint umgesetzt werden können. Nach der Sprint Planungssitzung startet die Entwicklung für das Produktinkrement. Während des Sprints findet täglich, zur gleichen Zeit und am selben Ort ein Daily Scrum statt. Im Daily Scrum beantworten die Teammitglieder die Fragen:

- Welche Aktivität habe ich seit dem letzten Daily Scrum fertig gestellt?

- Welche Aktivität werde ich bis zum nächsten Daily Scrum bearbeiten?
- Werde ich an der Ausführung meiner Aktivität behindert?

Ist der Sprint beendet findet das Sprint-Review statt. Im Sprint-Review überprüft der Product Owner die Ergebnisse des Sprints. Das Team stellt das Produktinkrement vor, der Product Owner prüft, ob das Ergebnis seinen Anforderungen entspricht. Änderungen aus dem Review werden im Product Backlog dokumentiert. Im Anschluss an das Sprint Review findet die Sprint Retrospektive statt. Die Sprint Retrospektive soll die Zusammenarbeit des Teams und die Abläufe verbessern.

Scrum in der Produktwartung

Dieser Abschnitt beschreibt die Veränderung eines Wartungsprojektes, das nach den Regeln von Scrum vorgeht.
Die Situation im Team für die Produktwartung einer Applikation: Das Wartungsteam hat die Aufgaben Fehler in der Produktion zu beheben, den Fachbereich (der diese Applikation nutzt) zu unterstützen, Änderungen/Erweiterungen umzusetzen und die Applikation zu überwachen. Diese betreute Applikation erstellt in der ersten Woche des Monats Geschäftsdaten für das Reporting, aggregiert diese Daten und verteilt diese Daten an weitere Applikationen. In der ersten Woche eines jeden Monats müssen verschiedene Prozesse abgearbeitet werden, um die Daten für das Reporting vorzubereiten oder Daten zu aggregieren. Die Daten werden dann an andere Applikationen geliefert. In dieser Zeit müssen alle Jobs, die die Programme des Systems steuern, rechtzeitig ineinandergreifen, um fristgerecht die Daten zu liefern bzw. für die Qualitätskontrolle und das Reporting bereitzustellen.
Fehler in der Datenaufbereitung oder im Ablauf verursachen im Fachbereich, im Wartungs-Team und bei den Anwendern viel Unruhe. Wie bei jedem Softwareprodukt bleibt die Umgebung nicht beständig. Das Umfeld in dem die Applikation eingebunden ist verändert sich laufend durch neue gesetzliche Anforderungen, Änderungen der Schnittstellen, Erweiterungen der Oberflächen, geänderte Abläufe, neue Datenbankversionen, etc.
Diese Anforderungen müssen vom Produktmanager koordiniert und mit dem Fachbereich abgestimmt werden. Sind die Anforderungen mit allen Beteiligten aufgenommen, fließen diese in den Wartungs-Projektplan ein. Je nach Umfang der Anforderung ist schnell ein Projektplan erstellt, der über eine längere Laufzeit Kapazitäten bindet. Es kam häufig vor, dass zum Zeitpunkt der Planfertigstellung schon neue Anforderungen warteten und damit den aufgestellten Plan hinfällig werden ließ. In diesem Wartungsprojekt hatte man dem Fachbereich feste Terminzusagen für die Entwicklung der Anforderungen gemacht. Natürlich wurden auch die Wartungsaufgaben in der Planung berücksichtigt. Ein Release, mit Änderungen oder Erweiterungen für die Applikation, wurde zu diesem Zeitpunkt alle 3 - 6 Monate ausgeliefert, je nachdem wie weit man mit der Entwicklung gekommen war.

In den nächsten Monaten (nach dem Erstellen des Projektplans) stellte sich heraus, dass der Projektplan und damit die Terminzusagen an die Fachabteilungen nie eingehalten werden konnte, weil Wartungsaufgaben mehr Zeit erforderten als geplant. In der ersten Woche im Monat war das Team mit der „Produktion“ der Geschäftsdaten beschäftigt. War die Produktion erfolgreich, konnten die Aufgaben aus dem Projektplan abgearbeitet werden. Parallel zur Produktion und den Tätigkeiten aus dem Projektplan mussten telefonische Anfragen des Fachbereiches beantwortet werden. Die Mitarbeiter waren dann meistens mit mehreren Aufgaben parallel beschäftigt. Die Abstimmung im Team fand jede Wochen statt, zwischen dem Produktmanager, dem Team und dem Fachbereich nur alle 3 bis 4 Wochen.

In den Teammeetings wurde regelmäßig festgestellt welche Aufgaben aus dem Projektplan nicht abgearbeitet werden konnten und damit welche Funktionalität im nächsten Release fehlen würde. Von der Seite des Produktmanagers konnte nur noch festgestellt werden wie sich der Projektplan verschiebt

bzw. welche Anforderungen im Projektplan verschoben werden mussten, um wichtige Termine zu halten. Die ausgelieferten Releases enthielten immer kleinere Fehler die zu weiteren Verzögerungen im Projektplan führten.

Zu diesem Zeitpunkt gab es im Unternehmen ein Team, das Scrum nutzte und seit einigen Monaten erfolgreich für die Softwareentwicklung einsetzte. Das Team war in den benachbarten Büros und so kam es zu einem Informationsaustausch. Da der Teamleiter und das Wartungs-Team mit der bestehenden Situation unzufrieden waren suchten alle nach einem anderen Vorgehen. Nach einigen Diskussionen schien Scrum ein geeignetes Vorgehen zu sein. Wir mussten jetzt noch den Produktmanager von Scrum überzeugen. Dies gelang, da er für sich auch schon festgestellt hatte, dass viele Aufgaben aus seinem Projektplan ins Leere laufen, da die Aufgaben nach einiger Zeit hinfällig sind und von anderen Themen überholt werden.
Unser Produktmanager sah in dem geänderten vorgehen mit Scrum auch seine Interessen und die vom Fachbereich besser vertreten. Die Chance, nach jedem Sprint ein fertiges Stück Software zu erhalten und Änderungen in die Entwicklung einzubringen, schien ihm ein guter Ansatz.
Schnell hatten wir die Rollen besetzt - wo wir dann später merkten, dass wir hier noch einiges lernen mussten - der Produktmanager war Product Owner, unser Teamleiter der ScrumMaster, das Team behielt seine Aufgaben bei. Mit dieser Rollenverteilung starteten wir dann. Der Product Owner erstellte das Product Backlog und priorisierte die Anforderungen. Das Team schätzte den Aufwand und wählte die Anforderungen für das Sprint Backlog. Der erste Sprint sollte nur 3 Wochen dauern, da man noch einige Zeit für Wartungsaufgaben eingeplant hatte, die man nicht im Sprint umsetzen wollte. Mit dem Product Owner hatten wir definiert was „fertig“ bedeutet und zur Abnahme einer Anforderung führt.

Der erste Sprint
Unser Sprint startete mit einer Timebox von 3 Wochen. Wir griffen uns die Aktivitäten für den Sprint und starteten mit der Entwicklung. Das Daily Scrum raubte uns sehr viel Zeit. 15 Minuten sollte es dauern 1 – 1 1/2 Stunden benötigten wir. Hier musste sich das Team auf die wesentlichen Informationen des Daily Scrum konzentrieren. Nach 8 Tagen im Sprint merkten wir, dass alle Schätzungen der Aufgaben nicht eingehalten wurden und im Sprint nicht mehr zu schaffen waren. Der Sprint wurde abgebrochen. Nach dem missglückten Start wurden uns erste Fehler im Sprint bewusst. Wir verharrten in den alten Strukturen. Die Teammitglieder beschäftigten sich weiter mit dem Support. Die parallelen Aufgaben Support der User und das Bearbeiten einer Sprint-Aktivität führten zu Verzögerungen. Das Springen zwischen den Aufgaben verbrauchte viel Zeit. Hier war unser erster Ansatz, die Unterstützung der Anwender auf den nächsten Sprint in eine Sprint-Aktivität zu verschieben. Der Vorteil des ersten Sprints war, dass wir mit Hilfe des Burndown Charts frühzeitig wussten, das wir kaum voran kamen.

In den weiteren Sprints mussten wir an der Verfügbarkeit unseres Product Owners arbeiten. Dieser war nicht nur mit unserem Produkt beschäftigt, sondern auch mit weiteren Aufgaben, so dass Fragen zu Sprint Anforderungen nicht kurzfristig beantwortet werden konnten.
Die Sprintlänge, von 3 Wochen, war für die Anwender und Produkt Owner nicht akzeptabel. Unser Team konnte sich 3 Wochen, während des Sprints (der nicht gestört werden darf) um keine Unterstützung der Anwender kümmern. Als Folge reduzierten wir die Sprintlänge auf eine Woche. Der Vorteil war, dass der Product Owner nun wöchentlich die Anforderungen im Produkt Backlog ändern konnte. Jetzt bestand die Möglichkeit die Produktionsfehler oder Anwenderunterstützung wöchentlich in den Sprint zu bringen.
Die nächsten Sprints verliefen besser, das Team konnte sich mehr auf die eigentlichen Aufgaben konzentrieren. Wir hatten bewusst wenige Anforderungen in den Sprint genommen, um die Funktionen mit automatisierten Testfällen auszuliefern, so konnten wir die vorhandene Funktionalität

prüfen und Fehler rechtzeitig erkennen. Dieses sehr „langsame“ Vorgehen machte sich bei den nächsten Releases auch bemerkbar. Der Releasezyklus hatte sich von 3 – 6 Monate auf 4 Wochen verringert. Die neuen Releases und Funktionalitäten waren stabiler als die vorhergehenden Auslieferungen. Dies schaffte uns in den nächsten Produktionsphasen, (die erste Woche im Monat) mehr Freiraum. Die monatliche Produktion lief stabiler. Was sich zu diesem Zeitpunkt auch bemerkbar machte war die Wissensverteilung. Jeder im Team hatte einen Überblick über die Aufgaben im Team und die auftretenden Schwierigkeiten bei der Umsetzung. Das Durchdringen des vorhandenen Source Code für Änderungen bzw. Erweiterungen wurde im gesamten Team besser. Das Burndown Chart und das Sprint Backlog mit den Aufgaben, die in Bearbeitung waren, war für uns eine große Hilfe. Die Darstellung durch die Charts machte klar, auf welche Aufgaben man sich konzentrieren musste.

Die Reißleine

Bei gravierenden Produktionsfehlern, die sich nicht als Sprint-Aktivität bearbeiten ließen wurde der Sprint abgebrochen. Dann wurden alle Kräfte des Teams gebündelt um diese Produktionsfehler in kürzester Zeit zu beheben oder auch um die Änderungen, möglichst ohne Fehler, in die Produktion zu bringen.

Nach diesen Produktionsfehlern erfolgte ein Sprint Review. Dabei wurden der Arbeits- und Projektfortschritt begutachtet. In der Sprint Retrospektive wurde die Frage beantwortet „Wie können wir den Fehler beim nächsten mal vermeiden“?

Warum funktioniert Scrum in der Produktwartung

In der Produktwartung sind viele Aufgaben und Schwierigkeiten analog der Softwareentwicklung. Dazu kommen folgende Herausforderungen (SW02):

- Programm-Verstehen
- fehlende Information
- vorhandene Software ist oft redundant, inkonsistent und fehlerhaft
- das Ergebnis des Wartungs-Falles geht unmittelbar in den Produktivbetrieb. Im Wartungs-Extremfall in den laufenden Betrieb.
- Unterschätzung des Änderungsaufwands bei Anwender und Hersteller
    - Kostendruck
    - Zeitdruck,
    - mangelnde Fortbildung

Die genannten und weitere Aufgabenstellungen in der Produktwartung, lassen sich mit Scrum bewältigen. Das Framework legt großen Wert darauf, dass die entwickelte Software stets gute Qualität besitzt. Ohne passende Qualität lassen sich keine Produktinkremente ausliefern. Es dürfen keine Anforderungen implementiert werden, die nicht getestet und dokumentiert sind oder fehlerhafte Arbeitsergebnisse liefern. Es sollten am Ende eines Sprints keine teilweise fertig gestellten Arbeitsergebnisse vorliegen.

Mit der Priorisierung der Anforderungen wird Software rechtzeitig ausgeliefert. Durch die Zusammenarbeit, Selbstorganisation des Teams, vermeiden von Fehlleistungen und Konzentration auf die wichtigsten Anforderungen steigert Scrum die Produktivität (SC01).

Das Sprint-Review macht transparent, in welchem Ausmaß das Team das Sprint-Ziel erreicht hat. Mit Metriken wie Testprotokollen, Testabdeckung, Refaktorisierungspotenzial und Kodierrichtlinien kann die Qualität der Software visualisiert und das Review unterstützt werden.

Die Scrum Teams einigen sich auf Normen für Architekturprinzipien, Kodierrichtlinien, eingesetzte Werkzeuge und bekommen mit dem Daily Scrum den Informationsaustausch über Hindernisse und oder Erfahrungen der anderen Teammitglieder.

Das Sprint Backlog und Sprint Burndown Chart visualisieren den Fortschritt innerhalb des Sprints und ermöglichen es gegebenenfalls Gegenmaßnahmen zu ergreifen.

Zusammenfassung

Die klaren Regeln bei Scrum, konsequent und diszipliniert angewendet, schaffen Transparenz, sichern die Qualität und sorgen für einen Wissensaustausch. Diese Attribute sind ideal für die Produktwartung.

Literatur

SW01	Christoph Bommer, Markus Spindler, Volkert Barr: Software Wartung, dPunkt.verlag 2008
SC09	Roman Pichler: Scrum Agiles Projektmanagement erfolgreich einsetzen, dPunkt.verlag 2009
SC02	Ralf Wirdemann: Scrum mit User Stories, Hanser 2009
AG01	Esther Derby, Diany Larsen: Agile Retrospectives making good teams great, Prgmatic Programmer 2009
SW02	Dr. Markus Pizka: Vorlesungsskript Software Wartung, Technische Universität München 2004
SW03	Harald Gall; Vorlesungsskript Software Wartung und Evolution, Universiät Zürich 2010
SW04	David Lorge Parnas: Software Aging. In: International Conference on Software Engineering. IEEE Computer Society Press, Sorrento, Italy 1994

Kontaktadresse:

Martin Heilemann
Lynx-Consulting GmbH
Johanniskirchplatz 6
D-33615 Bielefeld

Telefon: +49 (0) 521-5247-0
Fax: +49 (0) 521-5247-250
E-Mail martin.heilemann@lynx.de
Internet: www.lynx.de