

SMART DEVELOPMENT NEWS

DER NEWSLETTER für R&D Excellence

Ausgabe 7 • 2017 • www.co-improve.com

SCRUM in der Hardwareentwicklung Mehr Schein als Sein?

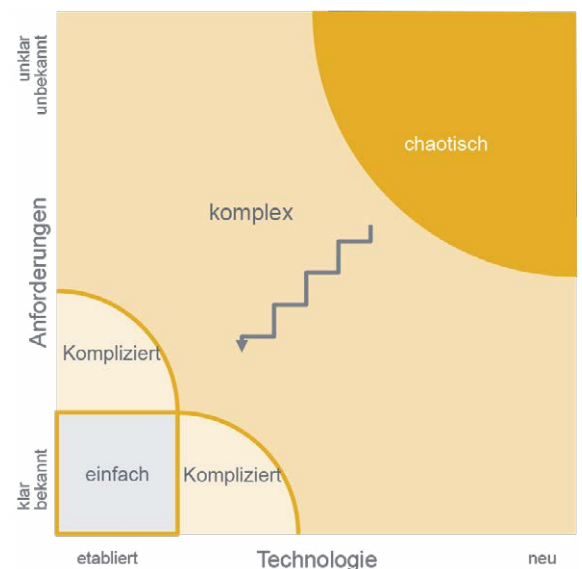
von Dirk Meißner und Dr. Jörk Hebenstreit

Alle sprechen von agiler Entwicklung und SCRUM. Die Potenziale sollen enorm sein: Kurze Reaktionszeiten, bessere Erfüllung der Kundenanforderungen, höhere Entwicklungsqualität, verlässliche Terminezusagen, kürzere Durchlaufzeit und viel motiviertere Projektteams. Aber wie soll das, was in der Softwareentwicklung seit Jahren hervorragend funktioniert, auf die Hardwareentwicklung übertragen werden? Schließlich können hier nicht einfach vollständige Produktinkremente nach kurzer Zeit an den Kunden ausgeliefert werden.

Was ist SCRUM?

Agiler werden, agiles Management und ähnlich Begriffe sind derzeit in aller Munde. Aber was versteht man darunter und was genau ist SCRUM?

Mit 58% ist SCRUM die am häufigsten genutzte agile Methodik in der Produktentwicklung und die Zahl ihrer Nutzer hat sich in den letzten zehn Jahren mehr als verdreifacht. Das sind Ergebnisse einer Umfrage von Version One veröffentlicht im *10th annual survey, Stage of Agile™*, 2015. Allerdings sind die allermeisten der Teilnehmer dieser Umfragen im Bereich der Software-, IT- und IT-nahen Entwicklung tätig. Nur ein sehr kleiner Teil der Unternehmen sind aus Branchen, die Hardwareprodukte entwickeln. Aber auch in diesen Branchen steigt die Zahl der Anwender agiler Methoden wie SCRUM oder Kanban. Dieser Trend ist auch in den großen Unternehmen der Automobilindustrie angekommen und führt dort zu erheblichen Veränderungen der Unternehmenskultur. Auf dem von CO-Improve organisierten Veranstaltungen CAP, Club der Agilen Hardwarepiloten, treffen sich Manager aus



hardwareentwickelnde Firmen, die SCRUM erfolgreich anwenden, um Ideen und praxiserprobte Erfahrungen auszutauschen. [LINK](#)

SCRUM basiert auf Lean und agilen Prinzipien, die im Agilen Manifest beschrieben sind. Es ist ein Framework zum Lösen komplexer Probleme. Immer wenn Unsicherheit besteht über das WAS der Kunden oder der Auftraggeber braucht und/oder das WIE es realisiert werden kann, ist ein iteratives Vorgehen sinnvoll und zielführend. Es wird in einem empirischen Prozess, in kurzen Zyklen, daran gearbeitet, das WAS besser zu definieren und dabei das WIE reifen zu lassen. Als Korrektiv dient das regelmäßige Feedback des Kunden, der nach jedem Zyklus (Sprint) in einem Review die Gelegenheit erhält, die erreichten Arbeitsergebnisse zu bewerten.

Das SCRUM-Team

Gearbeitet wird in interdisziplinären Teams, die über alle notwendigen Fähigkeiten verfügen sollten, um das gewünschte Projektergebnis zu erreichen. Es gibt drei Rollen im SCRUM. Der Product Owner ist verantwortlich für den Projekterfolg, er gibt Inhalt und Prioritäten vor. Der SCRUM-Master dient als Coach des SCRUM-Teams, seine Leistung manifestiert sich in der Produktivität des Teams. Alle anderen Teammitglieder sind „Entwickler“, im weiteren Sinne also auch Projekteinkäufer, Industrial Engineers, Qualitätsvorausplaner usw. und sind für das Erreichen der vereinbarten Ziele und deren Qualität verantwortlich. In einem Team sollten 3-9 Entwickler arbeiten. Die SCRUM-Teams sind selbstorganisiert und fokussieren sich vollkommen auf die angestrebten Ergebnisse.

Der Product Owner

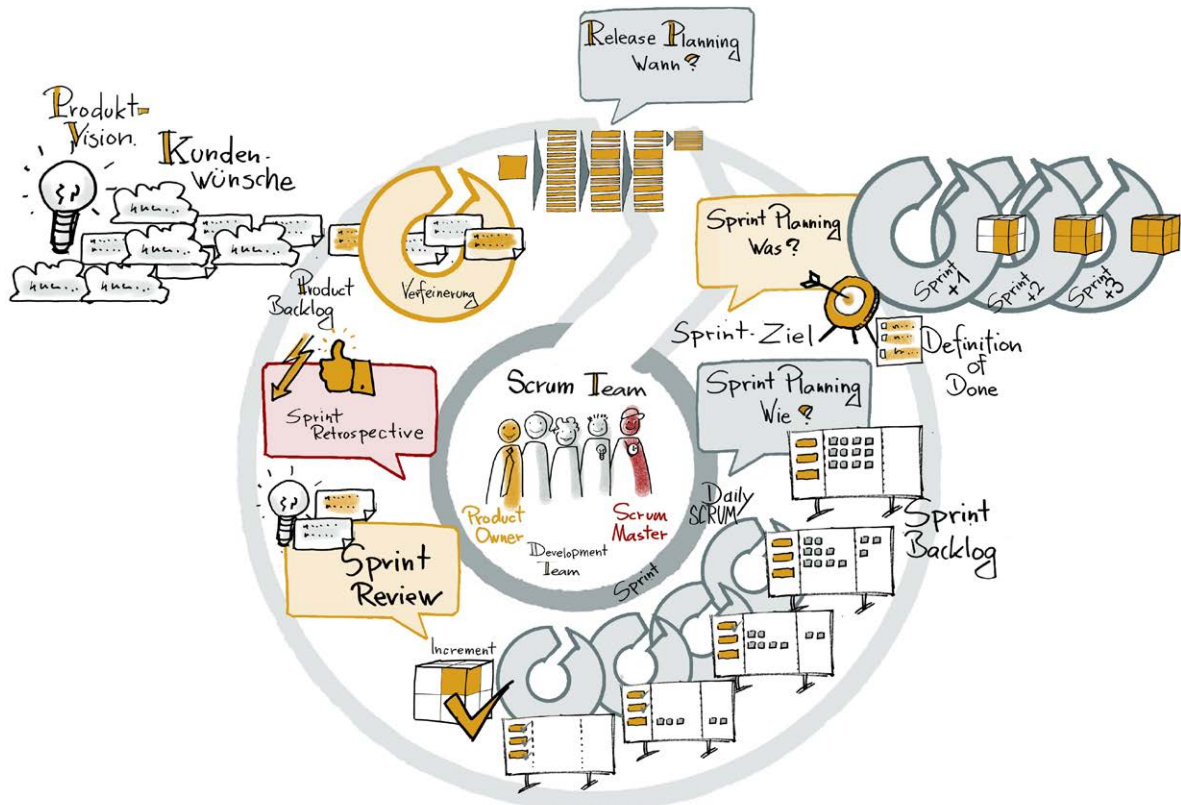
Der Product Owner verantwortet den Product Backlog. Dieses Dokument umfasst die Kundenwünsche. Es ist priorisiert und – im Gegensatz zum klassischen Lastenheft – dynamisch. Es entwickelt sich im Laufe des Projektes weiter, wird konkretisiert, verfeinert, erweitert oder auch verkürzt. Neben dem Product Backlog nutzt der Product Owner die Release-Planung zur Steuerung

des Projektes. Die Abarbeitung des Product Backlogs erfolgt in kurzen Zyklen, den Sprints. Die Sprintdauer kann zwischen einer und vier Wochen betragen, sollte aber immer gleich lang bleiben.

Der Sprint

Der Sprint beginnt mit der Sprint-Planung. Der Product Owner stellt die gewünschten Ergebnisse für diesen Sprint vor und das Team plant deren Realisierung. Dabei wird besprochen, wie das Ergebnis erreicht werden kann und der notwendige Aufwand dafür wird abgeschätzt. Dabei wird nach Prioritäten vorgegangen, solange bis die verfügbare Kapazität des Teams verplant ist. Das Team verständigt sich über die Qualitätskriterien für die Erfüllung der einzelnen Aufgaben und verpflichtet sich auf die Ergebniserreichung.

Jetzt geht's los. Es werden nur Aufgaben bearbeitet, die in der Planung abgestimmt wurden. Das Team ist für die Dauer des Sprints abgeschirmt und vor ad-hoc Aufgaben geschützt. Zur Synchronisierung der Arbeit trifft sich das Team täglich zur gleichen Zeit zu einem Daily. Das sind kurze Meetings, in denen der Fortschritt, mögliche Risiken und Störungen abgestimmt werden. Die Daily finden im Stehen statt, das macht die Abstimmung effizienter.



Am Ende des Sprints führen die Einzelergebnisse zu einem Produktinkrement, das, idealerweise, lieferbar wäre. In der Software ist das häufig der Fall. Für eine Hardwareentwicklung braucht es hier Anpassungen. Der Sprint wird mit einem Review und einer Retrospektive abgeschlossen. Im Review präsentiert der Product Owner seinen Kunden die erreichten Ergebnisse und die Entwickler stehen bereit, diese konkret vorzuführen. Dabei besteht für die Kunden/Stakeholder die Gelegenheit zum Feedback. Das ist das wichtigste Ziel eines Reviews und kann zu einer Korrektur des Product Backlogs führen. Im Anschluss trifft sich das Team zur Retrospektive, um darüber nachzudenken, was gut lief und was im nächsten Sprint die Zusammenarbeit verbessern könnte. Da sowohl das Review als auch die Retrospektive nach jedem Sprint stattfinden, führt dieser empirische Prozess zu einer stetigen Vervollkommnung des Produktes und zur Verbesserung der Teamarbeit.

Damit diese Arbeitsweise eingeführt und dauerhaft zu Erfolgen führen kann, braucht es eine Organisation, die Lean funktioniert und deren Kultur auf Offenheit, Vertrauen, Respekt und Mut basiert. Das Management überlässt den Teams die Selbststeuerung und sorgt für Prioritäten sowie Klarheit der Ziele und Aufgaben.

Die Adaption für Hardware-Projekte

Wenn, wie z. B. bei der Entwicklung von Software die Funktionen in Pakete zerlegt werden können, wird in einem Sprint praktisch der gesamte Entwicklungsprozess für diese Funktionalität durchlaufen. Voraussetzung ist, dass die Pakete klein genug sind, um innerhalb eines Sprints umgesetzt zu werden. Somit wird es möglich, nach jedem Sprint ein fertiges Produkt(inkrement) lieferfähig zu machen.

Allerdings ist das in einer komplexen Hardwareentwicklung heute noch nicht möglich. Obwohl moderne Technologien, wie z.B. 3D-Drucken, zur Verfügung stehen, gelingt es in der Regel nicht, nach jedem Sprint ein lieferfähiges Inkrement abzuschließen. Um die Vorzüge agiler Arbeitsweisen zu nutzen, zerlegt man das Gesamtprodukt in funktionale oder mechanische Komponenten. Als Ziele für die einzelnen Sprints werden sinnvolle Zwischenschritte im Reifeprozess einzelner Komponenten definiert. Das könnten z.B. Simulationen, virtuelle 3D-Modelle, Zeichnungen, 3D-gedruckte Teile usw. sein. Somit werden mehrere Sprints durchlaufen, um eine Komponente tatsächlich fertig zu stellen. Das Feedback

vom Kunden holt man sich dann für diese Zwischenschritte ein. Dabei ist es wichtig, immer etwas Vorführbares verfügbar zu machen. Ansonsten ist ein Feedback nur schwer möglich oder nur unkonkret. Um den Fortschritt im Gesamtprojekt zu planen und zu koordinieren, werden Etappen definiert, nach denen jeweils auf Produktebene neue Funktionalität integriert werden kann. Diese Release-Planung obliegt dem Product Owner in Abstimmung mit dem SCRUM-Team. Sie bildet den zeitlichen Rahmen für den Projektablauf und kann sich auch am existierenden Produktentstehungsprozess (PEP) orientieren.

Der PEP sollte bei der Einführung von SCRUM im ersten Schritt noch nicht verändert werden. Am Anfang geht es vor allem darum, die Methodik zu erlernen und mit den agilen Arbeitsweisen vertraut zu werden. In der Folge entstehen Ansatzpunkte, wie und wo der Prozess angepasst werden muss, um die neue Arbeitsweise weiter zu befördern. Dies beginnt bei trivialen begrifflichen Anpassungen (z. B. die SCRUM-Rollen), geht über veränderte Synchronisationspunkte in den Phasen des PEP bis hin zu veränderten Vorgaben für Dokumente und Freigaben.

Kompakt

Die Herausforderung

Die für Softwareprojekte seit vielen Jahren etablierten agilen Methoden können nicht 1:1 auf Hardwareprojekte übertragen werden. Deshalb scheuen viele Unternehmen den Einsatz von SCRUM für Hardwareprojekte. Ein Fehler!

Der Lösungsansatz

Geht man zurück auf die Prinzipienebene von SCRUM und findet für deren Umsetzung hardwaregeeignete Lösungen so wird ein erfolgreicher Einsatz möglich.

Der Nutzen

SCRUM steigert nachweislich Effizienz, Entwicklungsqualität und Termintreue. Die wesentlichen Vorteile sind Echtzeittransparenz, bessere interdisziplinäre Zusammenarbeit, Konzentration auf ein zu lieferndes Ergebnis, kleinere Losgrößen und schnellere Korrekturen, regelmäßiges Feedback der Stakeholder und motivierte eigenverantwortliche Teams.

Der Nutzen in Hardware-Projekten

Ein wesentlicher Vorteil von SCRUM liegt in der Echtzeittransparenz der laufenden Aufgaben, der Hindernisse und des Entwicklungsfortschritts. Das verbessert auch die interdisziplinäre Zusammenarbeit und Parallelisierung von Aufgaben. Verbunden mit den täglichen Abstimmungen und sofort getroffenen Entscheidungen kann viel schneller auf Störungen im Projekt reagiert werden. Durch Priorisierung der Aufgaben und die damit erzielte Fokussierung wird schädliches Multitasking reduziert, die Arbeitseffizienz gesteigert und die Termintreue deutlich verbessert.

An die Stelle mehrwöchiger Zyklen für ein Design-Freeze verbunden mit einer enormen Losgröße an Informationen und einer sequentiellen Bearbeitung des Konstruktionsergebnisses durch andere Fachbereiche treten deutlich kürzere Zyklen. Oft stellt sich heraus, dass nachfolgende Fachbereiche gar keinen kompletten Datensatz benötigen. Vielfach ist es ausreichend, sich auf kritische Teilumfänge zu beschränken. Damit werden Fehler schneller entdeckt und korrigiert. Die Entwicklungsqualität zu einem Design-Freeze steigt und Entwicklungszeit wird eingespart.

Die Übergabe eines verifizierten Beta-Prototyps am Ende eines Sprints an den Kunden ist in einer Hardwareentwicklung nicht möglich. Trotzdem besteht eine Vielzahl interner Kunden-Lieferantenbeziehungen in einem Projekt, zu denen Informationen in einer Qualität übergeben werden können, die dem internen Kunden ein wertvolles Feedback ermöglichen. So können Anforderungen noch während der Entwicklung verändert werden, frühzeitige Richtungskorrekturen erfolgen und Over-Engineering vermieden werden. Stakeholder werden viel früher und konkreter informiert als in klassischen Stage-Gate-Abläufen.

Zudem garantiert die strikte Priorisierung der gewünschten Produktfunktionen im Product Backlog durch den Product Owner nach Abstimmung mit den Stakeholdern eine ökonomische Orientierung am Kundenwert sowie eine frühe Bearbeitung der identifizierten Risiken und unterstützt so frühe Richtungsentscheidungen im Projekt.

In agilen Softwareentwicklungsprojekten sind Ressourcen und Termine fix. An der Qualität werden keine Abstriche gemacht. Es wird das zum Zieltermin ausgeliefert, was bis dahin geschafft wurde. Fehlende Funktionen werden in späteren Updates ausgeliefert. In der Hardwareentwicklung ist das nur in Teilen, z. B. für Optionen, möglich.

Das Prinzip kann dennoch übertragen werden. Schaffen wir es nicht rechtzeitig, eine niedrigpriorisierte Funktion wie geplant durch eine neue Technologie zu realisieren, wird dann eben eine bereits existierende Lösung eingesetzt. Der Termin wird trotzdem eingehalten und an der Entwicklungsqualität der Gesamtlösung werden keine Abstriche gemacht.

Praxisbeispiel Pumpenhersteller

Das Unternehmen ist ein führender Anbieter von Pumpen, Armaturen und zugehörigen Serviceleistungen. In der Business Unit Pumpen entwickeln ca. 200 Produktmanager und Entwickler an drei Standorten Standardprodukte und kundenspezifische Lösungen. Mit der Einführung von klassischen Prozess- und Projektmanagementverbesserungen konnten erste Effizienzsteigerungen erreicht werden. In einem nächsten Schritt sollte in einem Pilotprojekt aufgezeigt werden, dass mit der Einführung von SCRUM in einem Produktentwicklungsprojekt mit Hardware- Elektronik- und Firmware-Entwicklung weitere Effizienzverbesserungen erreichbar sind. Dazu wurden folgende Aktivitäten durchgeführt:

- Top-Management und Stakeholder-Training
- SCRUM-Pilotteam-Training
- Rollenzuordnung und Coaching des internen SCRUM-Masters
- Definition der Product Vision
- Aufbau des Product Backlog
- Planung des ersten Sprints
- Moderation der Daily SCRUMs
- Product Backlog Refinement
- Vorbereitung und Moderation des ersten Reviews
- Moderation der ersten Retrospektive

Nachdem erste Erfolge sehr schnell sichtbar wurden, entschloss sich das Unternehmen, weitere Entwicklungsprojekte ebenfalls agil zu starten.

Die Veränderung wurde von einem Berater in der Rolle eines Agile Coach über 9 Monate und 40 Beratertage begleitet. Parallel wurden interne Mitarbeiter befähigt, die Rolle des SCRUM-Masters zu übernehmen.

Agile Transformation - Ein Selbstläufer?

Bei der Einführung von SCRUM in der Hardwareentwicklung in unterschiedlichen Branchen und Unternehmensgrößen konnte immer wieder erlebt werden, dass die Mitarbeiter in den SCRUM-Teams sehr schnell von der Methodik begeistert sind.

Selbst anfängliche Zweifler lernen schnell die Vorteile für sich zu erschließen und wollen nicht mehr zurück. Durch die tägliche Abstimmung und die räumliche Nähe verändert sich das Kommunikationsverhalten sehr schnell. Anstelle von zahlreichen E-Mails mit großem Verteiler und oft unnötig langen und großen Abstimmungsmeetings, tritt die direkte Kommunikation in der Teamumgebung. Beeindruckend sind verändertes Verhalten und gestiegene Motivation, wenn sich die Teams wirklich selbst organisieren dürfen und sie die volle Verantwortung für das Projekt erhalten. Das wird aber nur in einem Top-down Ansatz erfolgreich sein. Deshalb ist es immer erforderlich, zunächst die beteiligten Manager und Stakeholder zu überzeugen, gewünschte Verhaltensänderungen klar zu machen und ihr Commitment einzufordern.

Um sich mit der Methodik vertraut zu machen, ist es in jedem Fall sinnvoll, mit Pilotprojekten zu starten und diese über einige Sprints mit erfahrenen Coaches zu begleiten.

Ausblick – Scaled SCRUM für Hardware

Sollten Sie sich nun fragen – alles ganz toll, aber wie kann ich meine Projekte auf SCRUM umstellen, für die es mehr als neun Entwickler braucht – dann verweise ich auf den nächsten Newsletter. Dieser beschreibt sinnvolle Skalierungskonzepte mit SCRUM für Hardware. Bleiben Sie gespannt.

Tipps aus der Praxis

- Bereiten Sie das Top-Management auf die Kulturveränderung vor.
- Holen Sie bei den Stakeholdern ein Commitment für die bei der Einführung agiler Methoden erforderliche Verhaltensänderung ein.
- Starten Sie mit einem geeigneten Pilotprojekt (frühe Phase, mittlere Komplexität, ausreichende Dauer, interdisziplinäres Team, Vollzeitmitarbeit, ein Standort).
- Suchen Sie dafür ein motiviertes freiwilliges Projektteam.
- Lassen Sie das Projektteam in der Projektvorbereitung und dem ersten Sprint intensiv von einem externen Agile Coach betreuen.
- Bilden Sie im Pilotprojekt interne SCRUM-Master aus.
- Fahren Sie die externe Unterstützung mit dem Erfahrungszuwachs der internen SCRUM-Master sukzessive zurück.
- Vermarkten Sie intern die Erfolge.