

Einsendeaufgabe Scrum

Stefan Berger

Inhaltsverzeichnis

1	Einleitung	3
2	Geschichte	3
3	Personen	4
3.1	Ken Schwaber	4
3.2	Jeff Sutherland	4
3.3	Hiroataka Takeuchi	5
3.4	Ikujirō Nonaka	5
3.5	Kent Beck	6
3.6	Mike Beedle	6
4	Manifest für Agile Softwareentwicklung	7
5	Rollen	8
5.1	Product Owner	8
5.2	Entwicklungsteam	8
5.3	Scrum Master	8
6	Artefakte	8
6.1	Product Backlog	8
6.2	Sprint Backlog	9
6.3	Inkrement	9
7	Fazit	9
	Literaturverzeichnis	10
	Abbildungsverzeichnis	11

Zusammenfassung

Dieses Dokument beschreibt Projektmanagement mittels der Scrum-Technologie bzw. des Scrum-Frameworks. Es gibt einen Überblick über die Entstehungsgeschichte von Scrum und die Personen, die an der Entwicklung beteiligt waren und sind. Es werden die wichtigsten Bestandteile von Scrum (Manifest für Agile Softwareentwicklung, Rollen der Teammitglieder und Artefakte) beschrieben und deren Nutzen erklärt. Im Fazit werden die drei Alleinstellungsmerkmale von Scrum – leichtgewichtig, leicht zu verstehen und schwer zu meistern – verdeutlicht.

1 Einleitung

Scrum ist ein Projektmanagement-Framework. Es existiert seit etwas mehr als zehn Jahren und ist aus Strategien hervorgegangen, die in Asien und den USA erfolgreich angewendet wurden. Im Unterschied zu anderen Projektmanagementstrategien akzeptiert es, dass Softwareentwicklungsprojekte unvorhersehbaren Änderungen unterliegen.

Im ersten Abschnitt werden wir uns mit der Entstehungsgeschichte von Scrum beschäftigen. Die Erfinder und Erhalter von Scrum hatten konkrete Vorstellungen, die mit Scrum verwirklicht und Problemstellungen, die gelöst werden. Im zweiten Abschnitt geht es um genau diese Personen und deren Beiträge zu Scrum. Im Manifest für Agile Softwareentwicklung sind die Grundregeln für den Erfolg bei der Anwendung von Scrum festgehalten, die im dritten Abschnitt behandelt werden. Der vierte Abschnitt beschäftigt sich mit den verschiedenen Rollen, die Scrum für die Teammitglieder vorsieht – Product Owner, Scrum Master und Entwickler. Im fünften Abschnitt beschäftigen wir uns mit den verschiedenen Artefakten des Frameworks.

2 Geschichte

Vor Scrum gab es bereits andere agile Projektmanagement-Ansätze. Elemente des Extreme Programming (XP) [1] fließen in das Scrum-Framework ein. XP ist ebenfalls leichtgewichtig und sieht die Entwicklung in Iterationen vor. Continuous Integration ist ein weiteres Konzept von XP, das oft zusammen mit Scrum angewendet wird. Das Manifest für Agile Softwareentwicklung bezeichnet kontinuierliche Auslieferung von Software als die höchste Priorität [2].

Verschiedene andere Projektmanagement-Ansätze waren denen von Scrum sehr ähnlich. Hirotaka Takeuchi und Ikujiro Nonaka verglichen als erstes den Prozess mit Rugby [3, S. 1]. Wie in dem Sport der Ball zwischen den Spielern hin- und hergeworfen wird, um das Team vorwärts zu bewegen, sollen sich die Projektteammitglieder gegenseitig zuarbeiten, um das Projekt voranzubringen. Einige Charakteristika des Ansatzes finden sich in Scrum wieder, etwa die „eingebaute Instabilität“ und selbstorganisierte Projektteams [3, S. 4].

James O. Coplien berichtet in „Borland Software Craftsmanship: A New Look at Process, Quality and Productivity“ von einem Projekt bei Borland, in dem den Teammitgliedern auf kreative Weise Rollen zugewiesen wurden [4, S. 3]. Es wurde großen Wert auf Meetings gelegt, und es konnte schnell auf Änderungen an der Architektur reagiert werden [4, S. 7].

1997 veröffentlichte Ken Schwaber das Paper „SCRUM Development Process“ [5]. Das Manifest für Agile Softwareentwicklung wurde 2001 von zunächst 17 Personen unterschrieben, unter anderem von Ken Schwaber, Jeff Sutherland und Mike Beedle. 2002 folgte das Buch „Agile Software Development With Scrum“, von Ken Schwaber und Mike Beedle.

3 Personen

3.1 Ken Schwaber

Ken Schwaber hat den Scrum-Prozess zusammen mit Jeff Sutherland in den frühen 1990er Jahren mitentwickelt, um Organisationen bei der Bewältigung komplexer Entwicklungsprojekte zu helfen. Als einer der Unterzeichner des Manifests für Agile Softwareentwicklung in 2001 gründete er anschließend die Agile Alliance und die Scrum Alliance. Die Scrum Alliance verließ er kürzlich, um Scrum.org zu gründen.

Ken Schwaber hat 30 Jahre Erfahrung in der Softwareindustrie und hat drei Bücher über Scrum geschrieben: „Agile Software Development with Scrum“, „Agile Project Management with Scrum“ und „The Enterprise and Scrum“. Er lebt in Lexington, Massachusetts. ([6], Übersetzung des Autors)



Abbildung 1: Ken Schwaber

3.2 Jeff Sutherland

Jeff Sutherland ist Mitgestalter von Scrum und ein führender Experte auf dem Gebiet der Entwicklung des Frameworks, um die Anforderungen heutiger Unternehmen zu erfüllen. Die Methodik, die er 1993 entwickelte und 1995 mit Ken Schwaber formalisierte, wurde seitdem von der überwiegenden Mehrheit der weltweiten Softwareentwicklungsunternehmen eingeführt. Aber Jeff Sutherland erkannte, dass die Vorteile von Scrum nicht auf Software- und Produktentwicklung beschränkt sind. Er passte diese erfolgreiche Strategie für mehrere andere Industrien an, einschließlich der Finanz-, Gesundheits-, Bildungs- und Telekommunikationsindustrie.

Als Geschäftsführer von Scrum Inc. und Senior Advisor und Agile Coach für OpenView Venture Partners erstellt Jeff Sutherland die Vision für Erfolg mit Scrum. Er teilt weiterhin Best Practices mit Organisationen auf der ganzen Welt und hat ausführlich über Scrum-Regeln und -Methoden geschrieben. Mit einem tiefen Verständnis von Geschäftsprozessen – gesammelt in den Jahren als CTO/CEO von elf verschiedenen Softwarebetrieben – besitzt Jeff Sutherland die Fähigkeit zu beschreiben, was die Vorteile von Scrum auf hoher organisatorischer Ebene sind und was es braucht, um hyperproduktive Teams zu erstellen. ([7], Übersetzung des Autors)



Abbildung 2: Jeff Sutherland

3.3 Hirotaka Takeuchi

Hirotaka Takeuchi war Assistant Professor an der Harvard Business School von 1976 bis 1983, wo er im M.B.A.-Programm Kurse in Marketing und Einzelhandel gab. Er fing 1983 als Associate Professor an der Hitotsubashi Universität Tokyo an und wurde 1987 Professor. 1989 und 1990 war er als Gastprofessor wieder an der Harvard Business School und lehrte dort „Konkurrenz und Strategie“. ([8], Übersetzung des Autors)



Abbildung 3: Hirotaka Takeuchi

3.4 Ikujiro Nonaka

Ikujiro Nonaka erhielt 1958 seinen BA in Politikwissenschaften von der Waseda-Universität und 1968 bzw. 1972 den MBA und den Dokortitel in Business Administration von der University of California, Berkeley. [...]

Professor Nonakas vorwiegendes Forschungsinteresse ist es, die Theorie des wissensbasierten Managements von Unternehmen, Gemeinschaften, öffentlicher Verwaltung und Nationen zu begründen und zu verbreiten, um die laufende, nachhaltige Wissensschöpfung und Innovation zu fördern. Als Teil seiner Arbeit hat er vergleichende Forschung mit Führungskräften und an wissensschöpfenden Prozessen in Unternehmen und Organisationen und von Führungskräften auf der ganzen Welt durchgeführt. [...]

„The Knowledge-Creating Company“ und „Enabling Knowledge Creation“ erhielten 1996 beziehungsweise 2000 den „Best Book of the Year“ award in Business und Management von der Association of American Publishers, Inc. ([9], Übersetzung des Autors)



Abbildung 4: Ikujiro Nonaka

4 Manifest für Agile Softwareentwicklung

Das Manifest für Agile Softwareentwicklung beschreibt zuerst 4 Prioritäten der Wertschätzung bei der Softwareentwicklung:

*Individuen und Interaktionen mehr als Prozesse und Werkzeuge
Funktionierende Software mehr als umfassende Dokumentation
Zusammenarbeit mit dem Kunden mehr als Vertragsverhandlung
Reagieren auf Veränderung mehr als das Befolgen eines Plans*

Es legt weiterhin zwölf Prinzipien fest:

Unsere höchste Priorität ist es, den Kunden durch frühe und kontinuierliche Auslieferung wertvoller Software zufrieden zu stellen.

Heisse Anforderungsänderungen selbst spät in der Entwicklung willkommen. Agile Prozesse nutzen Veränderungen zum Wettbewerbsvorteil des Kunden.

Liefere funktionierende Software regelmäßig innerhalb weniger Wochen oder Monate und bevorzuge dabei die kürzere Zeitspanne.

Fachexperten und Entwickler müssen während des Projektes täglich zusammenarbeiten.

Errichte Projekte rund um motivierte Individuen. Gib ihnen das Umfeld und die Unterstützung, die sie benötigen und vertraue darauf, dass sie die Aufgabe erledigen.

Die effizienteste und effektivste Methode, Informationen an und innerhalb eines Entwicklungsteams zu übermitteln, ist im Gespräch von Angesicht zu Angesicht.

Funktionierende Software ist das wichtigste Fortschrittsmaß.

Agile Prozesse fördern nachhaltige Entwicklung. Die Auftraggeber, Entwickler und Benutzer sollten ein gleichmäßiges Tempo auf unbegrenzte Zeit halten können.

Ständiges Augenmerk auf technische Exzellenz und gutes Design fördert Agilität.

Einfachheit – die Kunst, die Menge nicht getaner Arbeit zu maximieren – ist essenziell.

Die besten Architekturen, Anforderungen und Entwürfe entstehen durch selbstorganisierte Teams.

In regelmäßigen Abständen reflektiert das Team, wie es effektiver werden kann und passt sein Verhalten entsprechend an.

[2]

5 Rollen

5.1 Product Owner

Der Product Owner ist dafür verantwortlich, den Wert des Produktes zu maximieren, das aus der Arbeit des Entwicklungsteams entsteht. Wie dies geschieht, kann je nach Organisation, Scrum-Team und Einzelpersonen stark variieren. Der Product Owner ist die einzige Person, die für das Management des Product Backlogs verantwortlich ist. Das Product-Backlog-Management umfasst:

- die Product-Backlog-Einträge klar zu formulieren;
- die Einträge im Product Backlog so zu sortieren, dass Ziele und Missionen optimal erreicht werden können;
- den Wert der Arbeit zu optimieren, die das Entwicklungsteam erledigt;

- sicherzustellen, dass das Product Backlog sichtbar, transparent und für alle klar ist sowie zeigt, woran das Scrum-Team als nächstes arbeiten wird; und
- sicherzustellen, dass das Entwicklungsteam die Product-Backlog-Einträge im erforderlichen Maße versteht.

Der Product Owner kann die oben genannten Arbeiten selbst durchführen oder sie durch das Entwicklungsteam erledigen lassen. Der Product Owner bleibt jedoch immer rechenschaftspflichtig. [11, S. 6]

5.2 Entwicklungsteam

Das Entwicklungsteam besteht aus Profis, die am Ende eines jeden Sprints ein fertiges [„Done“] Inkrement übergeben, welches potenziell auslieferbar ist. Im Sprint Review muss ein fertiges Inkrement vorhanden sein. Nur Mitglieder der Entwicklungsteams erstellen das Produktinkrement.

Entwicklungsteams sind von der Organisation so strukturiert und befähigt, dass sie ihre eigene Arbeit selbst organisieren und managen. Die daraus resultierende Synergie optimiert die Gesamteffizienz und -Effektivität des Entwicklungsteams. [11, S. 7]

5.3 Scrum Master

Der Scrum Master ist dafür verantwortlich, Scrum entsprechend des Scrum Guides zu fördern und zu unterstützen. Scrum Master tun dies, indem sie allen Beteiligten helfen, die Scrum- Theorie, Praktiken, Regeln und Werte zu verstehen.

Der Scrum Master ist ein „Servant Leader“ für das Scrum-Team. Der Scrum Master hilft denjenigen, die kein Teil des Scrum-Teams sind, zu verstehen, welche ihrer Interaktionen mit dem Team sich hilfreich auswirken und welche nicht. Der Scrum Master hilft dabei, die Zusammenarbeit so zu optimieren, dass der durch das Scrum-Team generierte Wert maximiert wird. [11, S. 8]

6 Artefakte

6.1 Product Backlog

Das Product Backlog ist eine geordnete Liste von allem, von dem bekannt ist, dass es im Produkt enthalten sein soll. Es dient als einzige Anforderungsquelle für alle Änderungen am Produkt. Der Product Owner ist für das Product Backlog, seine Inhalte, den Zugriff darauf und die Reihenfolge der Einträge verantwortlich. [11, S. 15]

6.2 Sprint Backlog

Das Sprint Backlog ist die Menge der für den Sprint ausgewählten Product-Backlog-Einträge, ergänzt um einen Plan, um das Produktinkrement zu liefern und das Sprint-Ziel zu erreichen. Das Sprint Backlog ist eine Prognose des Entwicklungsteams darüber, welche Funktionalität im nächsten Inkrement enthalten sein wird, sowie über die erforderliche Arbeit, um diese Funktionalität in einem fertigen [„Done“] Inkrement zu liefern. [11, S. 16]

6.3 Inkrement

Das Inkrement ist das Ergebnis aus allen in einem Sprint fertiggestellten Product-Backlog-Einträgen und dem Resultat der Inkremente aller früheren Sprints. Am Ende eines Sprints muss das neue Inkrement fertig [„Done“] sein; das heißt es muss in einem verwendbaren Zustand sein und die Definition of Done des Teams erfüllen. Ein Inkrement ist ein Gegenstand inspizierbarer, fertiger [„Done“] Arbeit, der die Empirie am Ende des Sprints unterstützt. Das Inkrement ist ein Schritt in Richtung einer Vision oder eines Ziels. Es muss auch dann im einsatzfähigen Zustand sein, wenn der Product Owner es aktuell noch gar nicht ausliefern will. [11, S. 17]

7 Fazit

Scrum ist leichtgewichtig, denn es unterteilt den Prozess der Softwareentwicklung in Iterationen, und das Team plant nur eine Iteration genau voraus. Die Planungen der nächsten Iterationen basieren dann in einem erheblichen Maß auf den Ergebnissen der vorherigen.

Scrum ist außerdem leicht zu verstehen. Das liegt zum einen an der Leichtgewichtigkeit und zum anderen daran, dass keine Vorkenntnisse des Projektmanagements notwendig sind, um die Konzepte zu verstehen und auszuprobieren. Zwar ist es am Anfang vor allem für Softwareentwickler ungewohnt, aber leicht anzuwenden.

Schließlich ist Scrum schwer zu meistern. Es erfordert viel Übung, die Artefakte des Frameworks so anzuwenden, dass die bestmögliche Software oder Software, die gut genug ist [12] entwickelt werden kann.

Literaturverzeichnis

- [1] Kent Beck und Cynthia Andres. *Extreme Programming Explained: Embrace Change (2Nd Edition)*. Addison-Wesley Professional, 2004. ISBN: 0321278658.
- [2] Beck et al. *Manifesto for Agile Software Development*. 2001. URL: <http://www.agilemanifesto.org/>.
- [3] “The New New Product Development Game”. In: *Harvard Business Review* (1986). URL: <https://www.thescrummaster.co.uk/wp-content/uploads/2016/09/The-New-New-Product-Development-Game.pdf>.
- [4] James Coplien. “Borland Software Craftsmanship: A New Look at Process, Quality and Productivity”. In: (Apr. 1994).
- [5] Ken Schwaber. “SCRUM Development Process”. In: *Proceedings of the 10th Annual ACM Conference on Object Oriented Programming Systems, Languages, and Applications (OOPSLA)*. 1995, S. 117–134.
- [6] Ken Schwaber. URL: <https://www.scrumguides.org/ken.html>.
- [7] Jeff Sutherland. URL: <https://www.scrumguides.org/jeff.html>.
- [8] Hirotaka Takeuchi. URL: <https://web.archive.org/web/20071007165216/http://www.ics.hit-u.ac.jp/faculty/detail.php?id=113>.
- [9] Ikujiro Nonaka. URL: http://www.ics.hub.hit-u.ac.jp/faculty/profile/nonaka_ikujiro.html.
- [10] Kent Beck. *Implementation Patterns*. Addison-Wesley Professional, 2006. ISBN: 0321413091.
- [11] Schwaber et al. *The Scrum Guide*. 2001. URL: <http://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-German.pdf>.
- [12] James Bach. *The Challenge of „Good Enough“ Software*. 1996.

Abbildungsverzeichnis

1	Ken Schwaber	4
2	Jeff Sutherland	4
3	Hiroataka Takeuchi	5
4	Ikujiro Nonaka	5
5	Kent Beck	6
6	Mike Beedle	7