

Seminar Softwareentwicklung in der Wissenschaft

Agile Programmierung - Theorie II

SCRUM

Arne Brenneisen

Universität Hamburg

Fakultät für Mathematik, Informatik und Naturwissenschaften

Betreuer: Christian Hovy

25.07.2014

Inhalt

1	Einleitung	2
2	Agiles Manifest.....	2
3	Scrum.....	4
4	Aufbau von Scrum.....	5
4.1	Sprints.....	5
4.2	Prozess.....	6
4.3	Rahmen	7
4.3.1	Product Owner	7
4.3.2	Team.....	8
4.3.3	ScrumMaster.....	8
4.3.4	Sprint-Planung	8
4.3.5	Daily Scrum.....	9
4.3.6	Sprint-Review.....	9
4.3.7	Sprint-Retrospektive.....	9
4.3.8	Product Backlog	9
4.3.9	Sprint Backlog.....	10
4.3.10	Burndown-Diagramm	10
5	Scrum in der Praxis	11
5.1	Komplexität in Projekten.....	11
5.2	Vorteile durch Scrum.....	12
5.3	Grenzen und Nachteile.....	13

1 Einleitung

In unserer dynamischen Wirtschaft sind Überraschungen mittlerweile der Normalfall. So können sich Rahmenbedingungen oder sogar die Projektziele während eines Projekts ändern und Anforderungen instabil oder vorab nicht vollständig beschrieben sein. Es verwundert nicht, dass flexible, agile Projektmanagementmethoden, allen voran Scrum, immer populärer werden und aus den Unternehmen nicht mehr wegzudenken sind.^{1 2}

Aus diesem Kontext heraus, soll diese Arbeit die Projektmanagementmethode Scrum vorstellen und dem Leser die verschiedenen Rollen, Artefakte und Regeln erklären und abschließend eine Übersicht von Vor- und Nachteilen bieten.

2 Agiles Manifest

Um zu verstehen wie die Methode Scrum funktioniert ist es zunächst wichtig auf Agile Manifest einzugehen.

Das Agile Manifest ist eine Reaktion der Entwickler auf die sog. „schwergewichtigen“ und „dokumentenlastigen“ Prozess- bzw. Vorgehensmodelle, wie den Unified Process (UP) oder das „V-Modell“.³ Es definiert vier Leitsätze für eine erfolgreiche und effiziente Software-Entwicklung und zielt darauf ab, den Fokus bei der Softwareentwicklung mehr auf die eigentliche Arbeit zu legen und Software-Lösungen zu designen und zu implementieren.⁴

¹ Vgl. Dominik Maximini, Scrum - Einführung in der Unternehmenspraxis, 2013, Seite IX.

² Vgl. http://www.gpm-ipma.de/fileadmin/user_upload/Know-How/studien/PM_2_14_S40.pdf

³ Vgl. Eckhart Hanser, Agile Prozesse: Von XP über Scrum bis MAP, 2010, Seite 9.

⁴ Edb.

Individuen und Interaktionen sind wichtiger als
Prozesse und Werkzeuge

Funktionierende Software ist wichtiger als
umfangreiche Dokumentation

Zusammenarbeit mit Projektbetroffenen
ist wichtiger als Vertragsverhandlungen

Reaktion auf Änderungen ist wichtiger als
Verfolgung eines festgelegten Plans

Abbildung 1: Agiles Manifest

Entgegen der ersten Vermutung, bedeuten diese Leitsätze nicht, dass grundsätzlich auf Werkzeuge, Dokumentation, Vertragsverhandlungen oder der Verfolgung eines Plans verzichtet werden soll, sondern geben vielmehr eine Vorgabe zur Priorisierung der einzelnen Bestandteile. „Das Agile Manifest stellt mit seinen vier Leitsätzen eine Art Randbedingungen der leichtgewichtigen Software-Entwicklung dar“ und soll agilen Methoden als Vorgabe dienen, diese agilen Leitsätze umzusetzen um ein flexibles Umfeld zu schaffen. ⁵

⁵ Vgl. Eckhart Hanser, Agile Prozesse: Von XP über Scrum bis MAP, 2010, Seite 9.

3 Scrum

Die agile Projektmanagementmethode Scrum wurde in frühen 1990er Jahren von Ken Schwaber und Jeff Sutherland entwickelt und ist heute der De-facto-Standard in der agilen Softwareentwicklung.⁶ Schwaber und Sutherland „erkannten, dass kleine Entwicklungsteams mit Teammitgliedern, die im Wesentlichen alle Skills haben, effektiver und schneller bei höherer Qualität Softwareapplikationen liefern als große Teams.“⁷ „Da Scrum im Gegensatz zu eXtreme Programming das Management und das Business anspricht, setzte es sich als De-facto-Standard der agilen Softwareentwicklung durch“ und wird in vielen Unternehmen verwendet.⁸

Scrum folgt der Idee des Lean Managements, des „schlanken“ Managements was sich unter anderem in der flachen Hierarchie widerspiegelt, insbesondere durch das gewollte Fehlen eines Projektmanagers und den damit einhergehenden selbst-organisierenden Teams. Ein Produkt wird ähnlich wie beim eXtreme Programming (XP) iterativ in Abschnitten von üblicherweise monatlichen *Sprints* entwickelt wird.⁹ Alle Anforderungen sind im *Product Backlog* festgehalten. Den Kern des Vorgehensmodells bilden verschiedene Rollen, Artefakte (wie z.B. der *Product Backlog*) und mehrere Arten von regelmäßigen Meetings mit bestimmten Zwecken. Die wenigen Regeln sollen ein agiles Umfeld schaffen. Besonders auffallend sind dabei vor allem die durch Scrum festgelegten Rollen *Produkt Owner*, *Team* und der *ScrumMaster*, welche im nächsten Kapitel näher beschrieben werden sollen. Fragen zu der Zusammenarbeit und Teamstruktur stehen allerdings nicht im Vordergrund, sondern vielmehr der Ablauf des Projekts an sich selbst.¹⁰

⁶ Vgl. Boris Gloger, Informatik-Spektrum (Volume 33, Issue 2), April 2010, Seite 1.

⁷ Edb.

⁸ Edb.

⁹ Vgl. Eckhart Hanser, Agile Prozesse: Von XP über Scrum bis MAP, 2010, Seite 61.

¹⁰ Edb.

4 Aufbau von Scrum

Nachdem im vorangegangenen Kapitel ausführlich beschrieben wurde was Scrum ist, wird im folgenden Abschnitt die Funktionsweise von Scrum fokussiert. Dabei wird im Einzelnen auf die Themen *Sprints*, *Prozess* und den *Rahmen* von Scrum eingegangen.

4.1 Sprints

Ein Sprint ist im *Rahmen* der Methode Scrum eine feste vordefinierte Zeitspanne. Der *Sprint* verfügt somit über eine feste Iterationslänge und einen festen Endtermin. In der Regel beträgt die typische Sprintdauer 2-4 Wochen. Die konstante Dauer von *Sprints* ermöglicht einen besseren Rhythmus. Ziel eines jeden *Sprints* ist es nach dessen Beendigung einen Teil des Produktes fertig gestellt zu haben. *Sprints* sind analog zu den Iterationen des eXtreme Programming aufgebaut. Jeder *Sprint* umfasst jeweils wieder den Entwurf, die Implementierung der Elemente und einen abschließenden Test nach Beendigung des *Sprints*. Dieses Vorgehen wird für jeden einzelnen *Sprint* wiederholt und ist essenziell, um die Qualität des Endproduktes sicherzustellen, das sich zusammen aus der Beendigung verschiedener *Sprints* ergibt. Ein wesentlicher Faktor ist neben der Planung der drei oben genannten Punkte zudem, dass das *Team*, das an dem *Sprint* arbeitet, während der Bearbeitung des *Sprints* nicht gestört werden sollte. In der *Sprint-Planungssitzung* werden die Ziele pro *Sprint* definiert. Um neben den Zielen auch die Anforderungen an die *Sprints* nachhalten zu können, werden die Anforderungen in einem *Sprint Backlog* festgehalten.¹¹

¹¹ Vgl. Eckhart Hanser, Agile Prozesse: Von XP über Scrum bis MAP, 2010, Seite 69ff.

4.2 Prozess

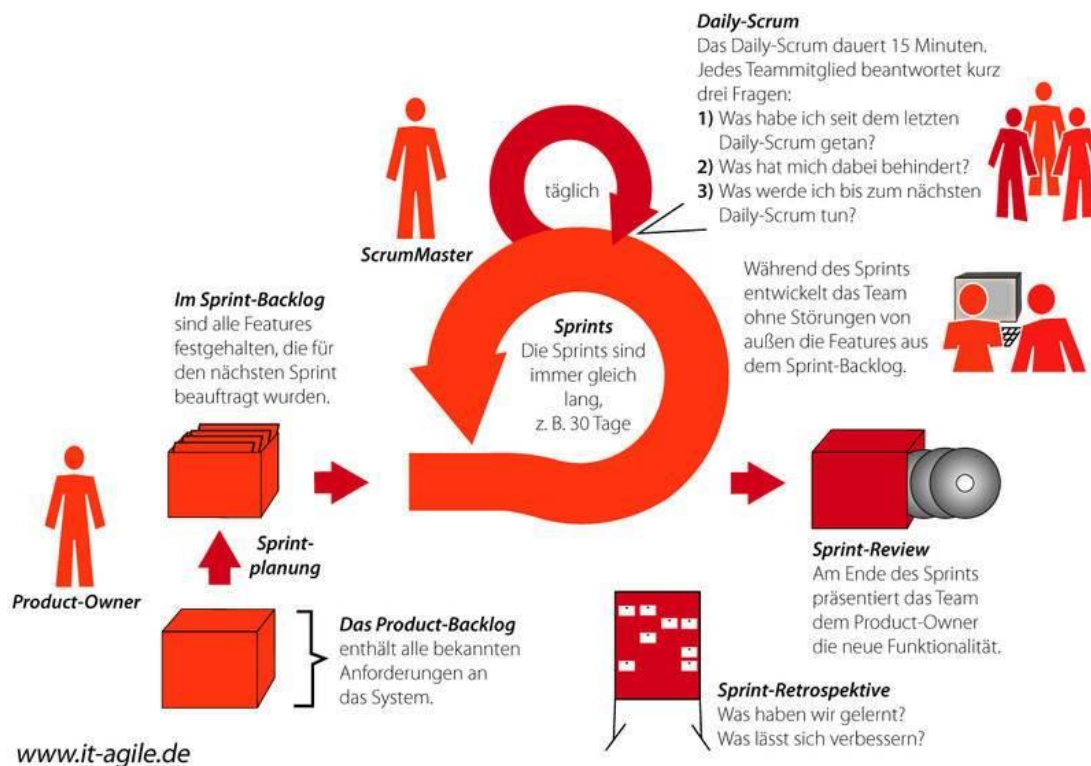


Abbildung 2: Der Prozess¹²

In jedem Scrumprojekt wird eines der Scrumteilnehmer als *ScrumMaster* definiert. Dieser ist der Prozessverantwortliche und u.a. dafür zuständig die *Daily Scrum* einzuberufen. Bei den *Daily Scrums* handelt es sich um tägliche Treffen, die 15 Minuten dauern. In einem *Daily Scrum* beantwortet jedes Teammitglied folgende Fragen:

- Was habe ich seit dem letzten *Daily Scrum* getan?
- Was hat mich dabei behindert?
- Was werde ich bis zum nächsten *Daily Scrum* tun?

Das *Product Backlog* ist Basis des Scrumprojektes. Das *Product Backlog* enthält alle bekannten Anforderungen an das System, die vom *Product Owner* definiert worden sind. Aus dem *Product Backlog* heraus lassen sich dann die einzelnen *Sprints* planen. Diese Planung wiederum ist im *Sprint Backlog* festgehalten, das alle Features enthält, die für den nächsten *Sprint*

¹² Vgl. <http://www.it-agile.de/wissen/methoden/scrum/>

beauftragt wurden. Während des *Sprints* kann dann das *Team* ohne Störungen von außen die Features aus dem *Sprint Backlog* entwickeln. Nach Beendigung des *Sprints* werden die Ergebnisse bzw. die neuen Funktionalitäten des Produkts dem *Product Owner* präsentiert. Dies geschieht in einem *Sprint Review*. Um die Sprintplanung kontinuierlich verbessern zu können, ist die *Sprint Retrospektive* wichtig. Dabei wird nach Beendigung des *Sprints* die Fragen gestellt was gelernt wurde und was sich beim nächsten *Sprint* verbessern lassen kann.¹³

4.3 Rahmen

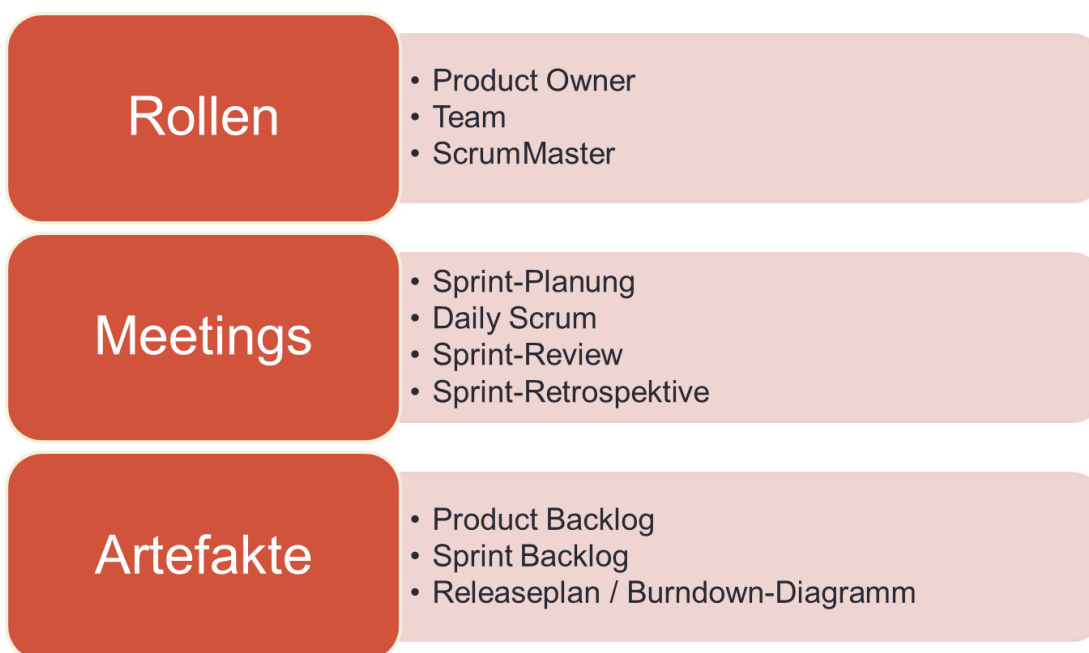


Abbildung 3: Der Rahmen

Den *Rahmen* von Scrum bilden verschiedene Rollen, Meetings und Artefakte, auf die in den nächsten Abschnitten eingegangen wird.

4.3.1 Product Owner

Der *Product Owner* ist der Produktverantwortliche und übernimmt die Sicht des Kunden innerhalb eines Scrumprojektes. Er steuert die Software-Entwicklung, erfasst alle Anforderungen, priorisiert die Produkt-Features und beschreibt diese in Form des *Product Backlogs*.

¹³ Vgl. Eckhart Hanser, Agile Prozesse: Von XP über Scrum bis MAP, 2010, Seite 68f.

Seine Aufgabe ist es, Auslieferungsdatum und Inhalt eines Projekts zu bestimmen und die Verantwortung für das finanzielle Ergebnis des Projekts zu übernehmen. Er passt die Features und Prioritäten nach Bedarf für jeden *Sprint* neu an und akzeptiert oder weist Arbeitsergebnisse zurück.¹⁴

4.3.2 Team

Das *Team* entwickelt das Produkt. Dabei ist die fehlende Teamleitung und die damit einhergehende Selbstorganisation besonders auffällig. Ein *Team* besteht typischerweise aus 5-9 Personen, die funktionsübergreifend zusammenarbeiten. Die Zusammensetzung ist entscheidend. Die Mitgliedschaft kann sich nur zwischen *Sprints* verändern und ist üblicherweise eine Vollzeitmitgliedschaft, bei der keines der Teammitglieder zusätzlich in anderen Scrumprojekten beschäftigt ist.¹⁵

4.3.3 ScrumMaster

Der *ScrumMaster* ist der Prozessverantwortliche. Seine Aufgabe ist es, das *Team* zu coachen und den Prozess zu etablieren. Er repräsentiert das Management gegenüber dem Projekt und ist verantwortlich für die Einhaltung der Regeln. Dabei beseitigt er Hindernisse und schützt das *Team* vor Störungen. Der *ScrumMaster* stellt sicher, dass das *Team* produktiv ist und unterstützt die Zusammenarbeit zwischen allen Rollen.¹⁶

4.3.4 Sprint-Planung

Das *Sprint-Planungs-Meeting* findet vor jedem *Sprint* statt. Dabei wird der *Product Backlog* analysiert und ausgewertet. Der *Product Owner* legt das neue *Sprint* Ziel fest und entscheidet gemeinsam mit dem *Team*, wie man es erreichen kann. Aus dem *Product Backlog* wird anschließend der *Sprint Backlog* und ein aktualisierter Releaseplan gemäß dem neuen Ziel erstellt. Dabei muss der zeitliche Aufwand für alle Features aus dem *Sprint Backlog* geschätzt werden.¹⁷

¹⁴ Vgl. Eckhart Hanser, Agile Prozesse: Von XP über Scrum bis MAP, 2010, Seite 62.

¹⁵ Vgl. Eckhart Hanser, Agile Prozesse: Von XP über Scrum bis MAP, 2010, Seite 63f.

¹⁶ Vgl. Eckhart Hanser, Agile Prozesse: Von XP über Scrum bis MAP, 2010, Seite 65f.

¹⁷ Vgl. Eckhart Hanser, Agile Prozesse: Von XP über Scrum bis MAP, 2010, Seite 70.

4.3.5 Daily Scrum

Das tägliche *Daily Scrum* ist eine Teambesprechung immer zur selben Zeit am selben Ort und hat das Ziel die Arbeit des *Teams* zu koordinieren und Probleme anzusprechen. Während der 15 Minuten wird der *Sprint Backlog* aktualisiert und jedes Teammitglied muss die drei schon bekannten Fragen beantworten. Dabei sind Gäste ausdrücklich erlaubt, haben aber kein Rederecht.¹⁸

4.3.6 Sprint-Review

Am Ende eines *Sprints*, beim *Sprint-Review*, präsentiert das *Team* mit einer Live-Demo auf einem Integrationsrechner das entstandene Software-Inkrement mit allen neuen Features. Das Meeting dauert maximal 4 Stunden und wird vom *ScrumMaster* moderiert. Der *Product Owner* akzeptiert nur vollständige Resultate. Auch hier sind Gäste wie Endanwender ausdrücklich erlaubt.¹⁹

4.3.7 Sprint-Retrospektive

Bei der *Sprint-Retrospektive* werden im Gegensatz zum *Sprint-Review* keine technischen Aspekte besprochen, sondern der Fokus auf die Optimierung des Prozesses und der Zusammenarbeit des *Teams* gelegt. Was hat gut und was hat nicht so gut funktioniert? Das Meeting dauert typischerweise 15–30 Minuten, maximal 3 Stunden und findet nach jedem *Sprint-Review* statt. Der *ScrumMaster* notiert vorgeschlagene Veränderungen und diskutiert diese mit dem *Team*. Alle Veränderungen werden gemeinsam beschlossen und sind Grundlage für den nächsten *Sprint*.²⁰

4.3.8 Product Backlog

Das zentrale Dokument zum Erfassen und Verwalten der Anforderungen ist der *Product Backlog* und muss für das gesamte *Team* deutlich sichtbar sein.

¹⁸ Vgl. Eckhart Hanser, *Agile Prozesse: Von XP über Scrum bis MAP*, 2010, Seite 71.

¹⁹ Edb.

²⁰ Vgl. Eckhart Hanser, *Agile Prozesse: Von XP über Scrum bis MAP*, 2010, Seite 71f.

Hier werden die zu liefernden Funktionalitäten gesammelt. Der *Product Owner* priorisiert die Funktionalitäten zu Beginn jedes *Sprints* neu.²¹

4.3.9 Sprint Backlog

Der *Sprint Backlog* ist die Liste der Aktivitäten eines *Sprints* und wird vor jedem *Sprint* neu erstellt. Alle Anforderungen für das *Sprint* Ziel werden identifiziert, die Aufwände geschätzt und eingetragen. Pro Anforderung soll der Aufwand zwischen 4 - 16 Stunden liegen und muss notfalls weiter aufgeteilt werden. Die *Team*-Mitglieder wählen ihre Tasks selber aus. Dabei kann jedes *Team*-Mitglied Tasks hinzufügen, löschen oder ändern. Die geschätzte restliche Arbeit wird täglich aktualisiert und oft in Form von User Stories an einer Stellwand dargestellt.²²

4.3.10 Burndown-Diagramm

Bei größeren Projekten wird oft ein Releaseplan erstellt und die Abfolge der *Sprints* fest geplant. Das *Burndown-Diagramm* gibt Auskunft über den zeitlichen Rahmen des Projekts und dient zur Überwachung der Planung. Es stellt pro *Sprint* den insgesamt verbleibenden Aufwand dar.²³

²¹ Vgl. Eckhart Hanser, Agile Prozesse: Von XP über Scrum bis MAP, 2010, Seite 73f.

²² Vgl. Eckhart Hanser, Agile Prozesse: Von XP über Scrum bis MAP, 2010, Seite 75f.

²³ Edb.

5 Scrum in der Praxis

„Als Indikation für agile Vorgehensweisen wird über alle Projektarten hinweg ein dynamisches Umfeld genannt“, bei dem sich Rahmenbedingungen ändern können und Anforderungen oft instabil oder unvollständig sind.²⁴ Überraschungen sind mittlerweile der Normalfall und treten durch Annahmen, die sich als unzutreffend erweisen, Entwicklungen in der Umwelt außerhalb eines Projekts aber auch innerhalb des Projekts auf.²⁵

5.1 Komplexität in Projekten

Je unbekannter die Anforderungen und die Technologie in einem Projekt sind, desto chaotischer kann der Projektverlauf werden. Bei größeren Projekten ist es oft schwierig am Anfang alle Anforderungen zu kennen und diese genau zu benennen. Hier zeigt sich oftmals, welche Vorteile Scrum für solche Projekte bietet.²⁶

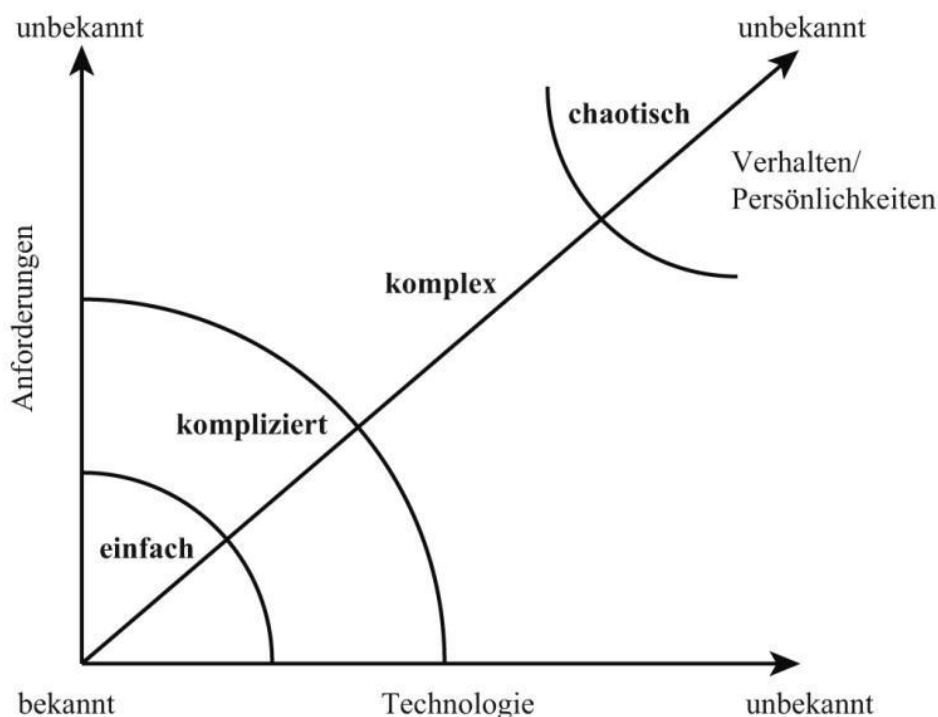


Abbildung 4: Komplexität in Projekten, nach Stacey (2007)²⁷

²⁴ Vgl. <http://www.computerwoche.de/a/warum-scrum-so-erfolgreich-ist,1231935>

²⁵ Edb.

²⁶ Vgl. Dominik Maximini, Scrum - Einführung in der Unternehmenspraxis, 2013, Seite 16.

²⁷ Vgl. Dominik Maximini, Scrum - Einführung in der Unternehmenspraxis, 2013, Seite 17.

5.2 Vorteile durch Scrum

Betrachtet man die Situation von Scrum in der Praxis lassen sich die verschiedenen Vorteile erkennen. In klassischen Projekten werden oftmals sämtliche Anforderungen gleichwertig angesehen und wichtige Funktionen erst spät im Projektverlauf und unter Zeitdruck implementiert werden. Das ist bei Scrum nicht so, da die Anforderungen in einem Projekt in der Regel sehr unterschiedlich priorisiert werden und so mehr Arbeitskraft für die wichtigeren Features aufgewendet wird. Solch ein striktes Vorgehen nach Prioritäten führt zu einer höheren Kundenorientierung und besseren Qualität.²⁸

„Hinzu kommt, dass in Festpreissituationen ein besonderes Bedürfnis nach einer finanziellen Obergrenze besteht, sonst wäre das Vertrauen des Auftraggebers in den Auftragnehmer wenig belastbar.“ „Aufgrund der höheren vertrauensbildenden Transparenz und objektiven Fortschrittskontrolle sowie der nach wirtschaftlichen Kriterien priorisierten Herstellungsreihenfolge“ lassen sich solche Situationen in Scrum besonders gut handhaben.²⁹

Bezogen auf das dynamische Umfeld und die Überraschungen in der heutigen Wirtschaft schafft die höhere Flexibilität in Scrumprojekten eine Überraschungsrobustheit.³⁰

Besonders die flache Hierarchie und die Selbstorganisation des Teams führen zu einem besseren Teamwork und insgesamt einer höheren Mitarbeitermotivation.³¹

Ein weiterer Vorteil ist die bessere Verteilung der Arbeitslast speziell am Anfang eines Projekt auf den gesamten Projektzeitraum.³²

Insgesamt lässt sich durch den Fokus auf der Qualität des Produkts und der hohen Mitarbeitermotivation eine höhere Nachhaltigkeit bei Scrumprojekten feststellen.³³

²⁸ Vgl. <http://blog.seibert-media.net/blog/2010/12/03/welche-vorteile-bietet-mir-als-kunde-ein-scrum-projekt/>

²⁹ Vgl. <http://www.computerwoche.de/a/warum-scrum-so-erfolgreich-ist,1231935>

³⁰ Edb.

³¹ Edb.

³² Vgl. <http://blog.seibert-media.net/blog/2010/12/03/welche-vorteile-bietet-mir-als-kunde-ein-scrum-projekt/>

5.3 Grenzen und Nachteile

Verbunden mit den Vorteilen lassen sich darüber hinaus auch verschiedene Risiken und Nachteile erkennen.

So ist es bei Scrum besonders ausschlaggebend, dass die gewonnenen Erkenntnisse auch verwertet werden.³⁴ „Ob Scrum dazu führt, dass Produkte schnell, gut, günstig oder qualitativ hochwertig entwickelt werden, hängt davon ab, was das Scrum-Team mit den gewonnenen Erkenntnissen macht,“ und ob diese bei den einzelnen Sprints verwertet werden.³⁵

Die hohe Flexibilität in Scrum-Projekten hat den Nachteil, dass es problematisch werden kann, wenn ein Projekt engen Zeit- oder Budgetvorgaben unterliegt.³⁶

Besonders durch die kleinen Teams und der Selbstorganisation dieser ist es wichtig, dass diese auch gut funktionieren und zusammenarbeiten können um hinderliche Einflüsse bei der Teamzusammensetzung und resultierende Konflikte zu vermeiden.³⁷

Hinzu kommt ein generell hoher Aufwand für Tests und etwaige juristische Erwägungen, da eine stärkere Unschärfe über die zu erbringende Leistung und deren Abnahmekriterien besteht.³⁸

Die Bedeutung der wenigen Scrum-Elemente ist im Detail nicht immer offensichtlich und verleitet oft besonders Anfänger zu kleinen, gravierenden Fehlentscheidungen.³⁹

³³ Vgl. <http://blog.seibert-media.net/blog/2010/12/03/welche-vorteile-bietet-mir-als-kunde-ein-scrum-projekt/>

³⁴ Vgl. <http://tigertechnik.wordpress.com/2013/02/04/grenzen-und-moeglichkeiten-von-scrum/>

³⁵ Edb.

³⁶ Edb.

³⁷ Edb.

³⁸ Edb.

³⁹ Edb.

6 Zusammenfassung

Scrum ist ein agiles Vorgehensmodell, welches eine höhere Transparenz und Sicherheit bei großer Flexibilität und Überraschungsrobustheit bietet.

Scrum beschreibt ein überschaubares Rahmenwerk mit wenigen Rollen, Artefakten und Meetings und fordert von den Beteiligten Verantwortung zu übernehmen.

Für die erfolgreiche Umsetzung ist die Einhaltung der Regeln unerlässlich.