



Roman Pichler

Scrum

Agiles Projektmanagement erfolgreich einsetzen

dpunkt.verlag



Roman Pichler arbeitet als unabhängiger Berater, Trainer und Coach mit den Schwerpunkten Lean und Scrum. Seine Kunden schätzen seine langjährige und vielseitige Erfahrung in der Anwendung von Scrum. Diese beinhaltet die Einführung von Scrum bei Start-ups und bei großen globalen Unternehmen. Mehr Informationen finden Sie unter www.romanpichler.com.

Roman Pichler

Scrum – Agiles Projektmanagement erfolgreich einsetzen



dpunkt.verlag

Roman Pichler
roman.pichler@romanpichler.com

Lektorat: Christa Preisendanz
Copy-Editing: Annette Schwarz, Ditzingen
Satz und Herstellung: Frank Heidt
Umschlaggestaltung: Helmut Kraus, www.exclam.de
Druck und Bindung: Koninklijke Wöhrmann B.V., Zutphen, Niederlande

Bibliografische Information Der Deutschen Bibliothek
Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie;
detaillierte bibliografische Daten sind im Internet über <<http://dnb.ddb.de>> abrufbar.

ISBN:
Buch 978-3-89864-478-5
PDF 978-3-89864-849-3

1. Auflage 2008
Copyright © 2008 [dpunkt.verlag](http://www.dpunkt-verlag.de) GmbH
Ringstraße 19
69115 Heidelberg

Die vorliegende Publikation ist urheberrechtlich geschützt. Alle Rechte vorbehalten. Die Verwendung der Texte und Abbildungen, auch auszugsweise, ist ohne die schriftliche Zustimmung des Verlags urheberrechtswidrig und daher strafbar. Dies gilt insbesondere für die Vervielfältigung, Übersetzung oder die Verwendung in elektronischen Systemen.

Es wird darauf hingewiesen, dass die im Buch verwendeten Soft- und Hardware-Bezeichnungen sowie Markennamen und Produktbezeichnungen der jeweiligen Firmen im Allgemeinen warenzeichen-, marken- oder patentrechtlichem Schutz unterliegen.

Alle Angaben und Programme in diesem Buch wurden mit größter Sorgfalt kontrolliert. Weder Autor noch Verlag können jedoch für Schäden haftbar gemacht werden, die in Zusammenhang mit der Verwendung dieses Buches stehen.

5 4 3 2 1 0

Für Melissa

Geleitwort

The only question regarding Roman Pichler's new book on Scrum is, »where is the English version?« I've known and worked with Roman for years with Scrum, so the book is full of practical advice. This book not only faithfully documents Scrum, it also provides a state of the art view of the most current thinking about using Scrum. More information about maintaining the Product Backlog, planning and managing releases, the retrospective, and people management reflect sound practices to know and employ. Since the use of Scrum depends on common sense, these are often presented severally. This book is a solid addition to the compendium of books to aid the Scrum practitioner and ScrumMaster.

*Ken Schwaber
Scrum Evangelist and Author
Boston, August 2007*

Inhaltsverzeichnis

1	Einleitung	1
1.1	Was ist Scrum?	1
1.1.1	Agiles Managementframework	1
1.1.2	Empirischer Prozess	2
1.1.3	Kein Wundermittel, sondern harte Arbeit	2
1.1.4	Scrum und schlankes Management	3
1.2	Warum Scrum?	3
1.2.1	Probleme frühzeitig erkennen, Handlungsspielraum sichern	3
1.3	Warum dieses Buch?	5
1.4	Mehr Informationen zu Scrum	6
1.5	Danke	6
2	Scrum im Überblick	7
3	Die Rollen	9
3.1	Product Owner	9
3.1.1	Die Aufgaben des Product Owner	10
3.1.2	Der Product Owner als Chief Engineer	12
3.1.3	Häufiger Fehler: Product Owner nicht verfügbar oder nicht bevollmächtigt	12
3.2	Team.	13
3.2.1	Individuen und Interaktionen.	13
3.2.2	Teameigenschaften.	13
3.2.3	Teamprozesse: Einer für alle, alle für einen	17
3.2.4	Teamnormen und Standards	18
3.2.5	Visueller Arbeitsplatz.	19

- 3.3 ScrumMaster. 19
 - 3.3.1 Die Aufgaben des ScrumMaster. 20
 - 3.3.2 Der ideale ScrumMaster. 21
 - 3.3.3 Wer ernennt den ScrumMaster? 22
 - 3.3.4 ScrumMaster und Mitarbeiterführung. 22
 - 3.3.5 Der Wandel der ScrumMaster-Rolle. 23
- 3.4 Die Rolle des Projektleiters 23
 - 3.4.1 Scrum und traditionelle Projektmanagementaufgaben 23
 - 3.4.2 Häufiger Fehler: Der ScrumMaster als Projektleiter. 24
- 4 Anforderungen 25**
- 4.1 Klassisches Requirements Engineering und Anforderungsbeschreibung in Scrum. 25
 - 4.1.1 Herkömmliche Verfahren zur Anforderungsbeschreibung 25
 - 4.1.2 Beschreibung der Anforderungen in Scrum 27
- 4.2 Das Product Backlog. 27
 - 4.2.1 Das Product Backlog ist ein lebendes Dokument 28
 - 4.2.2 Die Einträge sind priorisiert. 28
 - 4.2.3 Die Einträge weisen einen unterschiedlichen Detaillierungsgrad auf 28
 - 4.2.4 Die Einträge sind abgeschätzt 28
 - 4.2.5 Die Form des Product Backlog. 29
- 4.3 Das Produktkonzept 29
 - 4.3.1 Von der Produktidee zum Product Backlog 29
 - 4.3.2 Qualitative Marktforschung 31
 - 4.3.3 Nutzen des Produktkonzepts 31
 - 4.3.4 Kurz und knapp. 32
- 4.4 Inkrementelle Innovation 32
 - 4.4.1 Begriffsklärung 32
 - 4.4.2 Vorgehensweise 33
 - 4.4.3 Vorteile 33
- 4.5 Das Auffüllen des Product Backlog. 34
 - 4.5.1 Umfang und Vollständigkeit 34
 - 4.5.2 Unterschiedliche Detaillierungsstufen. 35
 - 4.5.3 Arbeiten mit Themen. 36
 - 4.5.4 Anwendung 36
- 4.6 Der Anforderungsworkshop 37

4.7	Die Priorisierung des Product Backlog	38
4.7.1	Gründe für die Priorisierung	38
4.7.2	Kriterien für die Priorisierung	39
4.7.3	Das Kano-Modell zur Bestimmung von Nutzen	39
4.7.4	Identifizieren und behandeln von Risiken	41
4.7.5	Kostenbestimmung	42
4.7.6	Die Wert-Risiko-Matrix	43
4.7.7	Die MuSCoW-Priorisierung	43
4.8	Merkmale guter Anforderungen	44
4.8.1	Unabhängig	45
4.8.2	Verhandelbar	45
4.8.3	Nützlich	45
4.8.4	Schätzbar	45
4.8.5	Klein	46
4.8.6	Testbar	46
4.9	Benutzergeschichten im Product Backlog	46
4.9.1	Benutzergeschichten im Überblick	46
4.9.2	Nutzen	47
4.9.3	Grenzen	48
4.10	Anwendungsfälle im Product Backlog	48
5	Releasemanagement	49
5.1	Die Releaseplanung im Überblick	49
5.1.1	Releaseplan? Wir arbeiten doch mit Sprints!	49
5.1.2	Keine Überstunden und keine Qualitätskompromisse	50
5.2	Planungsebenen in Scrum	51
5.2.1	Die Releaseplanung	51
5.2.2	Die Sprint-Planung	51
5.2.3	Die Planung des Arbeitstags	52
5.2.4	Die Beziehung von Releaseplan und Sprint Backlog	52
5.3	Projektsteuerungsgrößen und Projekterfolg	52
5.3.1	Steuerungsgrößen	52
5.3.2	Kriterien für den Projekterfolg richtig kommunizieren	53
5.4	Releasestrategie	54
5.4.1	Wertschöpfung in Scrum	54
5.4.2	Explorationssprints	55

5.5	Schätzen und Planen	57
5.5.1	Einführung.	57
5.5.2	Aufwandsbestimmung	58
5.5.3	Schätzklausur und Planungspoker	60
5.5.4	Die optimale Sprint-Länge	61
5.5.5	Die Entwicklungsgeschwindigkeit	63
5.5.6	Der Releaseplan.	67
5.6	Verfolgen des Projektfortschritts	69
5.6.1	Einführung.	69
5.6.2	Der Release-Burndown-Bericht	70
5.6.3	Entwicklungsgeschwindigkeitsbericht	72
5.6.4	Themenpark	73
5.7	Optimieren des Projektfortschritts	74
5.7.1	Vorausschauende Planung	74
5.7.2	Minimieren von Variation	75
5.7.3	Eliminieren von Verschwendung	75
5.7.4	Vermeiden von Überlastung.	77
5.7.5	Ausgewogener Arbeitsanfall.	78
5.8	Projektmanagementwerkzeuge	79
6	Sprints	81
6.1	Sprints im Überblick	81
6.2	Eigenschaften von Sprints	83
6.2.1	Erstellen eines Produktinkrements	83
6.2.2	Schutz vor Veränderungen	85
6.2.3	Verhältnis von Scrum-Besprechungen und Umsetzungsaktivitäten.	86
6.2.4	Wahl eines geeigneten Wochentags für Sprint-Beginn und -Ende	87
6.3	Vorbereitung der Sprint-Planungssitzung	87
6.3.1	Identifizieren des Sprint-Ziels.	88
6.3.2	Vorbereiten der Anforderungen	89
6.3.3	Identifizieren der Teamkapazität	90
6.3.4	Organisation der Räumlichkeiten	92

6.4	Die Sprint-Planungssitzung	93
6.4.1	Ziel	93
6.4.2	Aufgabenverteilung	93
6.4.3	Die Planungsschritte im Überblick	93
6.4.4	Etablierung eines gemeinsamen Verständnisses des Sprint-Ziels	94
6.4.5	Erzielen eines gemeinsamen Verständnisses der ausgewählten Anforderung	94
6.4.6	Identifizieren und Abschätzen der benötigten Aktivitäten . . .	95
6.4.7	Überprüfen von Kapazität und Leistungsvermögen	97
6.4.8	Abschluss der Planungssitzung	99
6.4.9	Typische Fehler	99
6.5	Das Sprint Backlog	102
6.5.1	Übersicht	102
6.5.2	Aktualisierung	102
6.5.3	Karten und Stellwand	102
6.6	Die Daily-Scrum-Besprechung	104
6.6.1	Zeitpunkt, Dauer und Ort	104
6.6.2	Zielsetzung	104
6.6.3	Teilnehmer	104
6.6.4	Ablauf	105
6.6.5	Vorbereitungsarbeiten	105
6.6.6	Nützliche Techniken	106
6.7	Das Sprint-Review	107
6.7.1	Zeitpunkt, Dauer und Ort	107
6.7.2	Zielsetzung	107
6.7.3	Teilnehmer	107
6.7.4	Ablauf	108
6.7.5	Vorbereitungsarbeiten	109
6.7.6	Nützliche Techniken	109
6.7.7	Typische Fehler	110
6.8	Die Sprint-Retrospektive	111
6.8.1	Zeitpunkt, Dauer und Ort	111
6.8.2	Zielsetzung	111
6.8.3	Teilnehmer	112
6.8.4	Ablauf	112
6.8.5	Vorbereitungsarbeiten	114
6.8.6	Nützliche Techniken	115
6.8.7	Typische Fehler	115

6.9	Frühzeitiges Beenden des Sprint	116
6.10	Verfolgen des Sprint-Fortschritts	117
6.10.1	Der Sprint-Burndown-Bericht	117
6.10.2	Der Hindernisbericht	119
6.10.3	Der Sprint-Endebericht	120
6.11	Optimieren des Sprint-Fortschritts	121
6.11.1	Kontinuierliches Review	121
6.11.2	Keine halben Sachen	121
6.11.3	Überlastungen vorbeugen	123
7	Große und verteilte Projekte	125
7.1	Größe, Verteilung und Zusammenarbeit	125
7.1.1	Große Projekte	125
7.1.2	Verteilte Projekte	126
7.1.3	Integration und Kommunikation	127
7.2	Bevor Sie skalieren oder verteilen	128
7.2.1	Klein anfangen	128
7.2.2	Klare Ziele	128
7.2.3	Brook's Law	128
7.3	Organisches Wachstum	129
7.3.1	Zurück zur Natur	129
7.3.2	Grundlagen schaffen	130
7.3.3	Langsam wachsen	131
7.4	Optionen für die Projektorganisation	133
7.4.1	Product Owner Team	134
7.4.2	Feature- vs. Komponententeams	136
7.4.3	Beispiele für die Organisation großer und verteilter Projekte	141
7.5	Praktiken für große und verteilte Projekte	143
7.5.1	Anforderungsmanagement	143
7.5.2	Multiteamplanung	144
7.5.3	Multiteamkoordination	149
7.5.4	Projektweite Normen	152
7.5.5	Infrastruktur	152
7.5.6	Agile Entwicklungspraktiken	154

- 7.6 Tipps für verteilte Projekte 154
 - 7.6.1 Verteilung entlang der Teamgrenzen 154
 - 7.6.2 Product Owner und ScrumMaster pro Team
und Standort 154
 - 7.6.3 Schrittweises Verteilen 156
 - 7.6.4 Regelmäßiger Austausch der Projektmitglieder
vor Ort 157

- 8 Unternehmensweite Einführung von Scrum 159**
- 8.1 Unternehmenswandel und Scrum 159
 - 8.1.1 Gründe für den Wandel 159
 - 8.1.2 Tragweite und Dauer des Wandels 160
 - 8.1.3 Merkmale des Wandels 161
- 8.2 Einführungsphasen 162
 - 8.2.1 Pilotphase 162
 - 8.2.2 Etablierungsphase 163
- 8.3 Praktiken zur Einführung von Scrum 164
 - 8.3.1 Bewusstsein schaffen 164
 - 8.3.2 Die Geschäftsleitung geht mit gutem Beispiel voran. 165
 - 8.3.3 Die Einführung von Scrum als Scrum-Projekt managen 165
 - 8.3.4 Eine glaubhafte Vision entwickeln 166
 - 8.3.5 Oft und richtig kommunizieren 166
 - 8.3.6 Mitarbeiter bevollmächtigen 167
 - 8.3.7 Veränderungen schrittweise vornehmen 167
 - 8.3.8 Nach Perfektion streben. 168

Anhang

Glossar 169

Literaturverzeichnis 173

Index 179

1 Einleitung

1.1 Was ist Scrum?

1.1.1 Agiles Managementframework

Scrum [skrʌm] ist ein agiles Managementframework zur Entwicklung von Software, das aus wenigen klaren Regeln besteht. Diese beinhalten die Anwendung der drei Rollen Product Owner, Team und ScrumMaster, die Verwendung eines priorisierten Product Backlog sowie das Erstellen von Produktinkrementen innerhalb kurzer Arbeitszyklen, die Sprints genannt werden. Scrum lässt sich auf alle Arten der Softwareentwicklung anwenden: Software als eigenständiges Produkt und Software als Bestandteil eines Produkts, Software als unternehmensinterne Lösung oder Software, die im Auftrag eines Kunden entwickelt wird; Software, die neu entwickelt, und Software, die gewartet wird.

Als agiles Framework verkörpert Scrum die Werte des Agilen Manifests [Beck et al. 2001]. Dieses stellt den Menschen in den Mittelpunkt der Softwareentwicklung (*individuals and interactions, collaboration*). Schließlich entsteht Software nur durch die Interaktion und Kollaboration von Menschen. Scrum ist nicht technologie- oder toolorientiert, sondern fordert und fördert die enge Zusammenarbeit der Beteiligten. Das Agile Manifest formuliert außerdem die Optimierung von Kundenzufriedenheit und Wertschöpfung als Ziel der Softwareentwicklung (*working software, collaboration, responding to change*). Für kommerzielle Softwareprojekte zählt letztendlich, ob die wirtschaftlichen Ziele des Projekts erreicht wurden, ohne dabei Raubbau an den Mitarbeitern oder zukünftigen Softwareversionen und damit Kundenzufriedenheit und Wertschöpfung zu treiben. In Scrum ist der Product Owner für die Erreichung der wirtschaftlichen Ziele des Projekts verantwortlich und steuert dieses durch das priorisierte Product Backlog und den Releaseplan.

Scrum und Rugby

Der Begriff Scrum stammt aus dem Rugby und wird auf Deutsch als »Gedränge« übersetzt. Vereinfacht lässt sich der Spielzug folgendermaßen beschreiben: Jeweils acht Spieler der beiden Mannschaften formen das Gedränge. Beide Spielergruppen stehen sich eng umschlungen und nach vorne gebeugt gegenüber. Die vordersten drei Spieler verkeilen Kopf und Schultern. Alle Spieler drücken nun nach vorne. Ein Spieler außerhalb des Gedränges, der sog. Gedrängehalb der ballführenden Mannschaft, wirft den Ball seitlich in das Gedränge. Seine Mitspieler im Gedränge müssen den Ball mit den Füßen nach hinten schieben. Erst wenn der Ball das Gedränge verlassen hat, darf er wieder aufgenommen und ein Angriff eingeleitet werden. Das Gedränge ist ein komplizierter Spielzug, der sorgsam einstudiert und orchestriert werden muss. Er verlangt eine disziplinierte Teamarbeit.

1.1.2 Empirischer Prozess

Scrum ist ein empirischer Prozess. Arbeitsweise und Produkt werden regelmäßig begutachtet und angepasst (sog. *inspect and adapt*). Am Ende eines jeden Sprint beurteilt der Product Owner die Angemessenheit der erzielten Ergebnisse, und das Team reflektiert über seine Zusammenarbeit und die Anwendung des Prozesses. So lernt das Projekt von Sprint zu Sprint dazu und kann sich kontinuierlich verbessern. Scrum ist keine herkömmliche Methodologie und keine Komplettlösung. Scrum schreibt nicht detailliert vor, was wann zu tun ist, sondern fördert die Kreativität der Mitarbeiter. Daher beinhaltet Scrum auch keine Verfahrensanweisungen oder Templates: Soweit diese hilfreich sind, müssen Sie sie für Ihr Projekt und Ihre Organisation selbst erarbeiten.

1.1.3 Kein Wundermittel, sondern harte Arbeit

Scrum ist kein Wundermittel, das, einmal in eine Organisation eingeführt, quasi von selbst alles besser werden lässt. Im Gegenteil: Oft sind die ersten Sprints für Projektmitarbeiter und Management schwierig. Hindernisse und Probleme treten auf und müssen beseitigt werden, um zielgerichtet weiterarbeiten zu können. Dabei müssen alle Beteiligten nicht nur die neuen Spielregeln erlernen, sondern auch alte Angewohnheiten ablegen. Diese beinhalten das Zuweisen von Aufgaben an Mitarbeiter und das Erstellen qualitativ minderwertiger Software. Das erfolgreiche Anwenden von Scrum ist also ein Lernprozess, der Zeit und Geduld benötigt und neben den Teammitgliedern, dem ScrumMaster und dem Product Owner auch das Management betrifft. Seien Sie dabei auf der Hut: Oft ist es verlockend, nicht die eigenen Arbeitspraktiken, sondern Scrum zu ändern.

1.1.4 Scrum und schlankes Management

Die Geburtsstunde von Scrum fällt in das Jahr 1993: In diesem Jahr wurde das erste Scrum-Projekt durchgeführt [Sutherland 2004]. Beeinflusst wurde Scrum von Anfang an von neuen, innovativen Wegen in der Produktentwicklung, wie sie insbesondere von japanischen Unternehmen pilotiert wurden [Takeuchi&Nonaka 1986]. Diese neue Form des Managements und der Produktentwicklung wird heute als »schlank« (*lean*) bezeichnet [Womack&Jones 1996]. Besonders Toyota hat bei der Entwicklung schlanker Entwicklungsprozesse eine führende Rolle eingenommen [Liker 2003]. Ich verweise in diesem Buch auf erprobte Vorgehensweisen der schlanken Produktentwicklung [Morgan&Liker 2006, Ward 2007] und der schlanken Softwareentwicklung [Poppendieck 2003], wo diese Scrum-Praktiken erklären oder sinnvoll ergänzen. Scrum ist übrigens ein schlanker Prozess, der ein Ziehsystem (*pull*) einsetzt und Überlastungen systematisch vermeidet.

1.2 Warum Scrum?

1.2.1 Probleme frühzeitig erkennen, Handlungsspielraum sichern

Softwareentwicklung ist schwierig und herausfordernd. An dieser Tatsache ändert auch Scrum nichts. Denn das Wesen der Softwareentwicklung ist Innovation und Kreativität: Jedes Softwareentwicklungsprojekt befriedigt neue Kundenbedürfnisse. Um Bedürfnisse aufzudecken, zu verstehen und die richtige Lösung zu entwickeln, benötigen wir eine ordentliche Portion Kreativität. Dies fällt vielen Organisationen nicht leicht: Oft scheitern Softwareentwicklungsprojekte oder liefern Ergebnisse, die weder Kunden zufriedenstellen noch die angestrebten wirtschaftlichen Ziele erreichen. Organisationen und Projekte verfallen dabei in einen Teufelskreislauf:

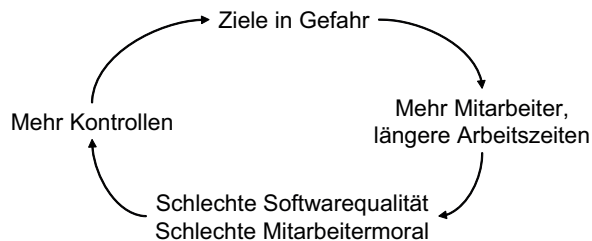


Abb. 1-1 Ein Teufelskreislauf, Quelle: [Ward 2007]

Sind die Ziele in Gefahr, so bitten wir häufig die Projektmitarbeiter länger zu arbeiten und fügen neue Mitarbeiter zum Projekt hinzu. Bedingt durch Hektik und Stress, längere Arbeitszeiten und schlecht eingearbeitete Projektmitglieder sinken

die Softwarequalität und die Moral der Mitarbeiter. Dies veranlasst das Management, mehr Kontrollen einzuführen, die die Entwicklung weiter verlangsamen.

Softwareentwicklungsprojekte weisen eine suboptimale Arbeitsorganisation nicht etwa deswegen auf, weil Management und Mitarbeiter nicht guten Willens sind. Das zentrale Problem traditioneller Vorgehensweisen besteht darin, dass wir frühzeitig versuchen, alle Eventualitäten und Arbeitsdetails zu antizipieren und einzuplanen, um anschließend unseren Plan auszuführen. Gleichzeitig erhalten wir erst spät im Projekt Rückmeldung über den tatsächlichen Fortschritt, meist erst dann, wenn die Software integriert und getestet wird. In Scrum führen wir alle Softwareentwicklungsaktivitäten innerhalb eines Sprint aus. So bekommen wir bereits nach wenigen Wochen Rückmeldung über den Fortschritt und etwaige Probleme und Hindernisse. Die Projektplanung fußt auf dem tatsächlichen Fortschritt des Projekts. Dabei gehen wir vor, wie in Abbildung 1–2 dargestellt.

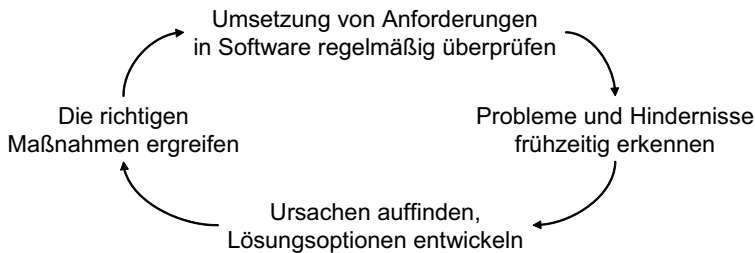


Abb. 1–2 Der Scrum-Kreislauf

Durch die Verwendung von kurzen Arbeitszyklen, an deren Ende ein Mehrwert entstanden sein muss, werden Probleme in Scrum frühzeitig offensichtlich. Wir haben so die Möglichkeit, rechtzeitig die Ursache des Übels aufzufinden, Lösungsoptionen zu entwickeln und die richtigen Maßnahmen zu ergreifen. Das frühe Auffinden von Problemen eröffnet uns einen größeren Handlungsspielraum und Flexibilität. Finden wir Probleme erst spät im Projekt, so sind viele Entscheidungen bereits gefällt und wir meist zur Schadensbegrenzung verdammt. Ursachenanalyse zu betreiben und die richtigen Verbesserungsmaßnahmen zu ergreifen, ist dabei kein Fingerschlecken, sondern harte Arbeit.

Die Mitarbeiter, Kunden und das liebe Geld

Wenn Sie Scrum konsequent einsetzen, so sollten Sie in den nachfolgend aufgeführten Bereichen positive Veränderungen erfahren.