



# **Seminar Software Engineering, Agile vs. klassische Methoden der Software-Entwicklung**

## **Scrum**

eine agile Methode zur Software Entwicklung

Seminararbeit  
Wintersemester 2003/2004

Raffael Schweitzer  
s0070710

Institut für Informatik  
der Universität Zürich  
Prof. Dr. Martin Glinz

Betreuer: Christian Seybold

Präsentation am 18. November 2003



## Inhaltsverzeichnis

<b>1</b>	<b>Einleitung .....</b>	<b>3</b>
<b>2</b>	<b>Was ist Scrum?.....</b>	<b>4</b>
<b>3</b>	<b>Geschichte von Scrum .....</b>	<b>5</b>
<b>4</b>	<b>Funktionsweise von Scrum.....</b>	<b>6</b>
4.1	Ablauf.....	6
4.2	Elemente.....	7
4.2.1	Scrum Master.....	7
4.2.2	Product Backlog und Product Owner .....	7
4.2.3	Scrum Team.....	8
4.2.4	Daily Scrum Meeting .....	8
4.2.5	Sprint Planning Meeting, Sprint Goal, Sprint Backlog .....	9
4.2.6	Sprint.....	10
4.2.7	Sprint Review Meeting .....	11
<b>5</b>	<b>Einsatzbereiche für Scrum .....</b>	<b>11</b>
5.1	Einsatz in neuem oder bestehenden Projekt.....	11
5.2	Ein grosses oder mehrere voneinander abhängige Projekte .....	12
5.3	Scrum und XP.....	12
<b>6</b>	<b>Erfolgsfaktoren .....</b>	<b>13</b>
6.1	Die menschliche Komponente.....	13
6.1.1	Scrum Master.....	13
6.1.2	Kunde.....	14
6.1.3	Scrum Team.....	14
6.1.4	Scrum Grundwerte .....	14
6.2	Grenzen von Scrum.....	14
<b>7</b>	<b>Zusammenfassung und Ausblick.....</b>	<b>15</b>
	<b>Abbildungsverzeichnis.....</b>	<b>16</b>
	<b>Quellenverzeichnis .....</b>	<b>16</b>



## 1 Einleitung

*„In today's fast-paced, fiercely competitive world of commercial new product development, speed and flexibility are essential. Companies are increasingly realizing that the old, sequential approach to developing new products simply won't get the job done. Instead, companies in Japan and the United States are using a holistic method; as in rugby, the ball gets passed within the team as it moves as a unit up the field.“*

H.Takeuchi, I.Nonaka [TAKEUCHI86]

In letzter Zeit machen verschiedene neue Methoden zur Software Entwicklung von sich reden, welche unter dem Begriff ‚agile Software Entwicklungsmethoden‘ zusammengefasst werden können. Klassische Methoden zur Software Entwicklung gehen meist von einem Ansatz aus, der dem Vorgehen eines Ingenieurs entspricht. Dabei haben sich im Einsatz verschiedene Nachteile herauskristallisiert. Dies sind unter anderem:

- Grosser Aufwand in der Planungsphase nötig
- Anpassungen während der Realisierung sind nur mit grossem Aufwand möglich, dadurch schlechte Umsetzung der Anforderungen bei sich schnell änderndem Umfeld
- Oftmals Behandlung der Mitarbeiter als Produktionsfaktor

Die agilen Methoden auf der anderen Seite versuchen genau diesen Problemen entgegenzuwirken. Ihre grundlegenden Eigenschaften sind dabei meist die folgenden:

- Inkrementelles Arbeiten in sehr kurzen Zyklen
- Ständige Anpassung an neue Anforderungen
- Besondere Berücksichtigung der menschlichen Komponente

Im Folgenden wird nun Scrum, eine dieser neuen agilen Methoden, vorgestellt und Vor- und Nachteile diskutiert.

Es ist darauf hinzuweisen, dass eine Vielzahl von weiteren neuen Methoden vorhanden ist, welche ebenfalls als agil zu bezeichnen sind. Wichtige Beispiele sind Extreme Programming (XP)<sup>1</sup> und Dynamic System Development Method (DSDM)<sup>2</sup>.

---

<sup>1</sup> <http://www.extremeprogramming.org>

<sup>2</sup> <http://www.dsdm.org>



## 2 Was ist Scrum?

Scrum ist ein iterativer Prozess zur Entwicklung von Software in einem chaotischen, sich schnell ändernden Umfeld. Dabei besteht Scrum aus einer Serie von 30 Tage dauernden Iterationen, den sogenannten Sprints. Nach jedem Sprint liegt das Produkt in einer neuen, lauffähigen Version vor, welche dem Kunden ausgeliefert werden könnte.

Zwischen zwei Sprints werden die Fortschritte, welche im letzten Sprint gemacht wurden, durch Management und Kunden überprüft und daraufhin bei Bedarf die Anforderungen an das Produkt neu gesetzt. Daraufhin kann der nächste Sprint gestartet werden.

Es handelt sich dabei um ein empirisches Prozessmodell, bei welchem der Prozess aufgrund von verschiedenen Kontrollen gesteuert wird.

Während den Kontrollphasen zwischen den Sprints überprüft das Management und der Kunde die Ergebnisse und führt gegebenenfalls Korrekturen durch. Dann überlässt es die Verantwortung für eine weitere Sprint Phase dem Scrum Team, welches versucht, die Anforderungen möglichst gut umzusetzen. Das während dem Sprint vom Team Erreichte wird dann wieder vom Management überprüft, worauf der Zyklus von neuem beginnt.

Während den Sprints werden Statusdokumente erstellt und regelmässige Treffen abgehalten, welche über den Stand der Arbeiten Aufschluss geben.

#### **scrum or scrum-mage:**

a rugby play in which the forwards of each side come together in a tight formation and struggle to gain possession of the ball when it is tossed in among them<sup>3</sup>

Im Gegensatz zu klassischen Entwicklungsmethoden legt Scrum den Fokus stark auf die einzelnen Personen und das Team. Diese tragen sehr viel Verantwortung, welche ihnen vom Management delegiert wird. Durch diese Betonung der menschlichen Komponente der Software Entwicklung werden grosse Produktivitätssteigerungen erreicht.

Durch die Vorgehensweise von Scrum wird ausserdem Kommunikation und Kooperation zwischen den Entwicklern untereinander und auch dem Management oder dem Kunden verbessert und versucht, die gegensätzlichen Interessen der verschiedenen Gruppen unter einen Hut zu bringen. Einer der Grundsätze von Scrum ist dabei, dass der Entwicklungsprozess von sämtlichen Störungen und Hindernissen von aussen abgeschirmt werden muss, um eine möglichst grosse Produktivität zu erreichen.

Scrum ist also eigentlich kein Software Entwicklungsprozess, sondern ein sogenannt leichter Management Prozess, da es keine Praktiken zur Software Entwicklung enthält. Vielmehr umschliesst es die bestehenden Praktiken auf der Management Ebene. Aus diesem Grund ist es auch möglich, eigentliche agile Software Entwicklungsmethoden, wie zum Beispiel Extreme Programming, mit Scrum zu kombinieren.

<sup>3</sup> Merriam Webster Dictionary, <http://www.webster.com>



## 3 Geschichte von Scrum

Scrum entstand durch vereinte Bemühungen von Advanced Development Methods [ADM] und VMARK Software (VMARK). Beide Firmen entwickeln objektorientierte Software. Sowohl ADM als auch VMARK machten die Erfahrung, dass objektorientierte und komponentenbasierte Entwicklung durch den definierten Ansatz der klassischen Entwicklungsprozesse vielfach eher behindert als gefördert werden.

1995 wollte nun ADM die internen Entwicklungsprozesse optimieren. Dazu wurden Wissenschaftler der DuPont Experimental Station, welche Forschung im Bereich der Prozessautomation für biochemische Prozesse betreiben, mit der Analyse einiger der verbreiteten Software Entwicklungsmethoden beauftragt.

Die Wissenschaftler kamen zum Schluss, dass die meisten der Entwicklungsmethoden davon ausgehen, dass Software in wiederholbaren, klar definierten und vorhersagbaren Prozessen entsteht. Software Entwicklungsprozesse sind jedoch meist nicht vorhersagbar und nicht wiederholbar. Es handelt sich um empirische Prozesse. Sie waren höchst erstaunt darüber, dass die meisten Methoden einen definierten Ansatz wählen, obwohl dieser unter diesen Umständen nicht sehr erfolgreich sein kann.

Diese Erkenntnis führte dazu, dass ADM und VMARK begannen, eine eigene Methode zu entwickeln, welche auf dem empirischen Ansatz beruhte. Hauptsächlich handelte es sich dabei um Jeff Sutherland (VMARK) und Ken Schwaber (ADM), welche gemeinschaftlich Scrum entwickelten. Ausserdem wurden wiederum die Wissenschaftler der DuPont Experimental Station einbezogen.

Sutherland und Schwaber benutzten hauptsächlich Ideen von Ikujiro Nonaka und Hirotaka Takeuchi, welche diese erstmals 1986 in einem Artikel ausformuliert und in ihrem 1995 erschienenen Buch „The Knowledge Creating Company“ verfeinert hatten. [TAKEUCHI86] und [TAKEUCHI95] Takeuchi und Nonaka waren auch die ersten, welche den Begriff ‚Scrum‘ benutzten.

Eine weitere Person, welche an der Entwicklung von Scrum stark beteiligt war, ist Mike Beedle. Er war einer der ersten, die Scrum einsetzten, und brachte verschiedene Vorschläge in Scrum ein. Ausserdem zeichnet er für die Kombination von Scrum mit Extreme Programming verantwortlich (Siehe Kapitel 5.2).

1996 wurde Scrum erstmals anlässlich der OOPSLA Konferenz<sup>4</sup> vorgestellt. Seither ist die Methode von über 50 Unternehmen erfolgreich für die Entwicklung objektorientierter Software eingesetzt worden.

Mit ihrem Buch zu Scrum haben Ken Schwaber und Mike Beedle nun 2001 eine umfassende Beschreibung von Scrum geschaffen. [SCRUM01]

Die Begründer von Scrum engagieren sich ausserdem in der Agile Alliance, welche Vertreter der verschiedenen agilen Methoden vereint.<sup>5</sup>

---

<sup>4</sup> The Eleventh Annual ACM Conference on Object-Oriented Programming Systems, Languages and Applications, <http://www.acm.org/sigplan/oopsla/oopsla96/oopsla96.html>, siehe auch [PLOP98]

<sup>5</sup> Agile Alliance, <http://www.agilealliance.org>



## 4 Funktionsweise von Scrum

In diesem Abschnitt wird die Funktionsweise von Scrum beschrieben. Dazu wird zuerst der grundlegende Ablauf eines mit Scrum durchgeführten Projektes besprochen, um einen Überblick über die Bestandteile von Scrum zu erhalten. Daraufhin werden die einzelnen Elemente detailliert erklärt.

### 4.1 Ablauf

Die Aufgaben, welche während dem Projekt erledigt werden sollen, sind im **Product Backlog** aufgeführt. Dieser entspricht einer Liste aller geforderten Funktionalitäten und Änderungen. Ein Scrum Projekt wird in Iterationszyklen von 30 Tagen aufgeteilt, den sogenannten **Sprints**. Zu Beginn jedes Sprints findet das **Sprint Planning Meeting** statt, in welchem der **Product Owner** die Elemente des Product Backlogs priorisiert und das **Scrum Team** die Aufgaben für den bevorstehenden Sprint festlegt. Diese entsprechen dem **Sprint Backlog**. Während dem Sprint trifft sich das Scrum Team täglich zum **Daily Scrum Meeting**, welches vom **Scrum Master** geleitet wird. Am Ende des Sprints demonstriert das Team die neu geschaffene Funktionalität im **Sprint Review Meeting**.

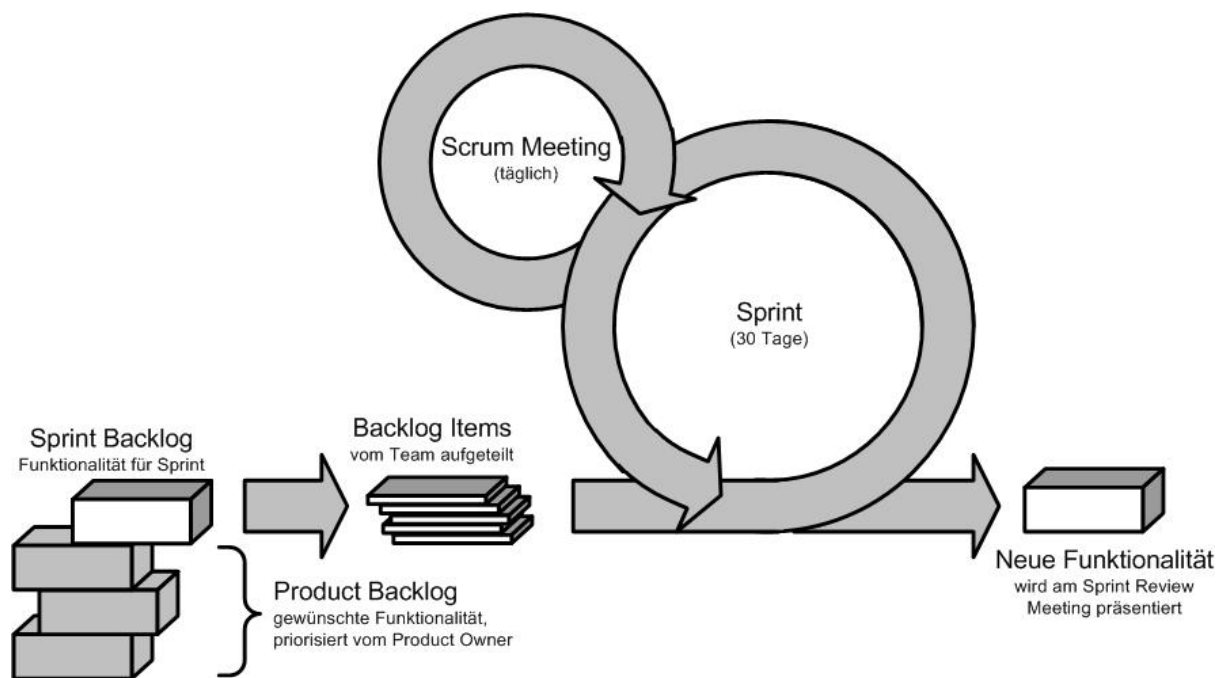


Abbildung 1 Funktionsweise von Scrum



## 4.2 Elemente

### 4.2.1 Scrum Master

Der Scrum Master ist die neue Management Rolle, welche mit Scrum eingeführt wird. Er ist die wichtigste Person im Scrum Prozess und muss eine herausragende Persönlichkeit sein, da er permanent die Initiative ergreifen und seine Aufgaben beharrlich verfolgen muss.

Der Scrum Master ist verantwortlich dafür, dass die Grundsätze von Scrum im Projekt verwirklicht werden. Ausserdem ist er die Schnittstelle zwischen Management und dem Scrum Team. Dabei muss er versuchen, das Team während eines Sprints möglichst von allen Einflüssen von aussen abzusichern.

Bei Beginn eines Projektes unterstützt er das Management und den Kunden bei der Bestimmung eines geeigneten Product Owners. Zusammen mit diesem und dem Scrum Team legt der Scrum Master dann jeweils den Backlog für den nächsten Sprint fest und plant daraufhin mit dem Team den Ablauf des Sprints.

Der Scrum Master leitet die täglichen Scrum Meetings und fungiert als Coach für das Scrum Team. Dabei vergleicht er fortwährend die gemachten Fortschritte mit den Erwartungen und greift allenfalls bei Problemen ein. Zusätzlich ist er während des Sprints verantwortlich für die möglichst vollständige Beseitigung aller Hindernisse, von welchen ihm die Teammitglieder berichten.

Der Scrum Master trifft die Entscheidungen, welche das Team nicht selbst treffen will, sofort an den Meetings. Dies ist wichtig, da so das Team in der Lage ist, weiterzuarbeiten. Falsche Entscheidungen könnten später immer noch revidiert werden.

Die Ausbildung zum Scrum Master ist zertifiziert. Die zertifizierten Scrum Master sind dabei in einer Web Community zusammengeschlossen. [ALLIANCE]

### 4.2.2 Product Backlog und Product Owner

Beim Product Backlog handelt es sich um eine der Dringlichkeit nach geordnete Liste aller Features, Funktionalitäten, Technologien, Erweiterungen und Fehlerkorrekturen, welche ins Produkt eingebaut werden sollen. Die Product Backlog Liste kann von jedem, der am Projekt beteiligt ist, eingesehen werden. Ausserdem kann jeder die Aufnahme einer neuen Anforderung in die Liste anregen. Dies könnte zum Beispiel durch das Marketing, die Verkaufsabteilung, die Entwicklungsabteilung oder den Support geschehen.

Solche Anforderungswünsche können auch sehr vage sein. Je stärker ein Element in der Product Backlog Liste priorisiert wird, desto detaillierter sollte es ausformuliert sein.

Die Liste ist äusserst dynamisch, da sich die Anforderungen an das Produkt während seiner Entwicklung meist verändern. Zu Beginn des Projekts muss sie nicht vollständig sein, sollte jedoch bereits genügend Elemente für einen ersten Sprint enthalten.

Wichtigkeit	Nr.	Beschreibung	Aufw.	Von
<b>Sehr Hoch</b>				
	1	<b>Datenbank Versionierung beenden</b>	16	KH
		<b>Lizenzierung hinzufügen</b>	-	
	2	User Lizenzierung	16	TG
	3	Demo / Evaluation Lizenzierung	12	TG
	4	<b>Unterstützte Dateiformate erweitern</b>	160	TG
	..	..	..	..
<b>Hoch</b>				
		<b>Namensgebung UI vereinheitlichen</b>		
	7	Im Hauptprogramm	24	KH
	8	Im Import Modul	24	AM
	9	<b>Abfrage Tool</b>	260	T&M
	..	..	..	..
<b>Mittel</b>				
	..	..	..	..

Abbildung 2 Beispiel Product Backlog



## Seminar Software Engineering

### Agile vs. klassische Methoden der Software-Entwicklung, WS 03/04

#### Scrum

Dozent: Prof. Dr. Martin Glinz  
Betreuender Assistent: Christian Seybold

Der Product Backlog enthält ausserdem auch Probleme, welche vor der Erledigung bestimmter Elemente gelöst werden müssen. Während das Projekt fortschreitet, werden diese Probleme sukzessive in normale Elemente, also reguläre Anforderungen an das Produkt oder einzusetzende Technologien, umgewandelt.

Für ein Scrum Projekt wird ausserdem ein Product Owner bestimmt. Diese Person verwaltet die Product Backlog Liste. Jeder, der dem Produkt eine neue Anforderung hinzufügen will, muss sich an den Product Owner wenden, welcher über die Aufnahme entscheidet und dann die Priorisierung innerhalb der Liste vornimmt.

Der Product Owner ist insofern wichtig, damit das Scrum Team nicht von allen Seiten mit Wünschen bedrängt wird und die Ziele für das Produkt nur an einer zentralen Stelle verwaltet werden, so dass nur ein einziger, immer aktueller Product Backlog für das Projekt vorhanden ist.

Der Product Owner führt in Zusammenarbeit mit dem Team für jedes Element der Liste eine grobe Aufwandschätzung durch. Aufgrund dieser Schätzungen kann das Team dann bestimmen, welche Elemente es im nächsten Sprint erledigen wird.

#### 4.2.3 Scrum Team

Ein Scrum Team umfasst mindestens fünf und allerhöchstens acht Personen. Dies ist wichtig, um eine möglichst grosse Produktivität zu gewährleisten. Werden weniger Personen eingesetzt, so findet zu wenig Informationsaustausch statt. Bei mehr als acht Team Mitgliedern besteht hingegen grosse Wahrscheinlichkeit, dass die Kontrollmechanismen von Scrum, wie das tägliche Scrum Meeting, versagen und zudem zuviel Komplexität entsteht.

Während dem Sprint Planning Meeting erklärt sich das Scrum Team bereit, im nächsten Sprint eine bestimmte Menge von Elementen des Product Backlogs zu erledigen. Wie das Team dieses Ziel erreicht, ist ihm selbst überlassen. Es soll sich dafür selbst organisieren. Während des Sprints kann es im Rahmen der durch das Unternehmen vorgegebenen Richtlinien alle Entscheidungen selbst fällen und erwarten, dass auftretende externe Hindernisse durch den Scrum Master beseitigt werden.

Ein Scrum Team sollte so zusammengesetzt sein, dass seine Mitglieder in ihrer Summe alle Fähigkeiten mitbringen, welche zur Erreichung der für diesen Sprint gesetzten Ziele notwendig sind. So können die Mitglieder zum Beispiel Analysten, Designer, Programmierer oder auch Qualitätsingenieure sein. Sinnvoll ist aber, dass mindestens ein erfahrener Ingenieur Mitglied des Teams ist. Wie das Team die Aufgaben im Einzelnen aufteilt, ist ihm selbst überlassen. An einem Projekt können auch mehrere Scrum Teams beteiligt sein. Dies wird in Abschnitt 5.2 eingehend erläutert.

#### 4.2.4 Daily Scrum Meeting

Das tägliche Scrum Meeting ist das Herzstück von Scrum. Alle Mitglieder des Scrum Teams treffen sich dazu einmal täglich für 15 Minuten jeweils zur selben Zeit und am selben Ort. Das Scrum Meeting wird vom Scrum Master geleitet, der dafür sorgt, dass die Regeln eingehalten werden und dass das Treffen nicht zu lange dauert.

Während des Treffens stellt der Scrum Master jedem Teammitglied die folgenden drei Fragen:

- Was hast du seit dem letzten Treffen erledigt?
- Was wirst du von jetzt bis zum nächsten Treffen erledigen?
- Was hat dich bei deiner Arbeit behindert?





## Seminar Software Engineering

Agile vs. klassische Methoden der Software-Entwicklung, WS 03/04

### Scrum

Dozent: Prof. Dr. Martin Glinz  
Betreuender Assistent: Christian Seybold

Durch die Beantwortung dieser Fragen werden verschiedene Ziele erreicht: Einerseits wird die Kommunikation unter den Teammitgliedern und das Wissen des Einzelnen über das Gesamtprojekt stark verbessert, da jeder weiss, was der andere tut. Andererseits werden Probleme und Hindernisse aufgezeigt, welche das Projekt verlangsamen. Das Treffen hilft ausserdem bei der schnellen Entscheidungsfindung. Die Teilnehmer des Treffens berichten dabei den Status ihrer Arbeit primär nicht dem Scrum Master oder einem Vorgesetzten, sondern den anderen Teammitgliedern

Von den Teilnehmenden gemeldete Hindernisse werden durch den Scrum Master nach Möglichkeit beseitigt. Sei dies persönlich oder durch Delegation an eine zuständige Stelle. Haben Teammitglieder weitere Ideen, wie die Produktivität verbessert werden könnte, oder werden sie zum Beispiel vom Management mit anderen Aufgaben belastet, so bringen sie dies ebenfalls als Hindernisse in ihrer Arbeit vor.

Ein Scrum Meeting ist keine Diskussionsrunde. Es werden während dem Treffen keine Probleme gelöst oder Designfragen besprochen. Bestehen über das Treffen hinausgehende Probleme, Unklarheiten oder zu erörternde Designfragen, so wird während dem Scrum Meeting ein Treffen vereinbart, um die entsprechenden Fragen zu besprechen.

Die täglichen Scrum Meetings sind ausserdem eine effektive Methode, um immer darüber informiert zu sein, wo das Team momentan steht und ob es erfolgreich ist. Auch Personen, welche nicht zum Team gehören, können an Scrum Meetings teilnehmen, um sich zu informieren. Dabei müssen sie sich aber an bestimmte Regeln halten. Ken Schwaber stellt in seinem Buch dazu die ‚Chickens and Pigs‘ Regel auf, welche auf dem nebenstehenden Witz basiert. [SCRUM01]

#### **Chickens and pigs**

A chicken and a pig are together when the chicken says, „Let's start a restaurant!“ The pig thinks it over and says, „What would we call this restaurant?“ The chicken says, „Ham n' Eggs!“ The pig says, „No, thanks. I'd be committed, but you'd only be involved!“

Die Teammitglieder entsprechen dabei den Schweinen und die übrigen Teilnehmer des Meetings sind Hühner. Die Hühner müssen sich am Rand, rund um den Kreis der Scrum Teammitglieder aufhalten und dürfen nur zuhören, aber nicht sprechen.

#### 4.2.5 Sprint Planning Meeting, Sprint Goal, Sprint Backlog

Das Sprint Planning Meeting besteht eigentlich aus zwei Teilen. Als erstes findet das Treffen des Scrum Teams mit Scrum Master und Product Owner sowie interessierten Kundenvertretern statt. Während dieses Treffens erläutert der Product Owner dem Team die am höchsten priorisierten Elemente des Product Backlog, so dass sich das Team darüber klar werden kann, welche Elemente es im nächsten Sprint erledigen kann. Der Product Owner bestimmt nun mit dem Team das Sprint Goal, ein Ziel, welches definiert, was im Sprint erreicht werden soll. Gegen dieses Ziel wird am Ende des Sprints im Sprint Review Meeting geprüft, ob der Sprint erfolgreich war.

Im zweiten Teil des Treffens legt nun das Team mit Unterstützung des Product Owners fest, welche Elemente des Product Backlogs im Sprint erledigt werden sollen. Die Elemente werden dann in einzelne Aufgaben von 4 bis 16 Stunden Dauer aufgeteilt. Diese Aufgaben entsprechen detaillierten Teilstücken zur Umwandlung des Product Backlogs zu Software.

Die Summe dieser Aufgaben entspricht dem Sprint Backlog, der während dem Sprint abzuarbeiten ist. Der Sprint Backlog wird dabei laufend angepasst, wenn zum Beispiel Aufwandschätzungen korrigiert, neue Aufgaben entdeckt oder Aufgaben gestrichen werden.

Wie der Product Backlog, ist auch der Sprint Backlog öffentlich und kann von Aussenstehenden als Statusinformation eingesehen werden.



#### 4.2.6 Sprint

Während dem Sprint versucht das Scrum Team, das gesetzte Sprint Goal zu erreichen. Auf welche Art und Weise es dies tut, ist grösstenteils ihm selbst überlassen. Es passt sich dabei den Umständen, der eingesetzten Technologie und den organisatorischen Rahmenbedingungen an. Ein Sprint dauert in der Regel 30 Tage. Einzige vorgeschriebene Fixpunkte dabei sind das tägliche Scrum Meeting und der Sprint Backlog, der immer auf dem aktuellen Stand gehalten werden muss. Am Ende des Sprints muss zudem eine neue Produktversion vorliegen, welche die während des Sprints geschaffene Funktionalität beinhaltet. Sinnvoll ist dabei auch, jeden Tag eine lauffähige Version des Produkts, einen Daily Build, zu erstellen und mit diesem Regressionstests durchzuführen.

Während des Sprints gibt das Management die Zügel vollständig aus der Hand und überlässt das Team seiner eigenen Verantwortung. Das Risiko hält sich dabei in Grenzen, da ein Sprint auf 30 Tage limitiert ist. Gesetzt den Fall, dass das Team während eines Sprints tatsächlich kein nützliches Ergebnis produziert, so ist doch festzuhalten, dass es in dieser Zeit mit grosser Wahrscheinlichkeit viele wichtige Erfahrungen gesammelt hat, welche für die weitere Entwicklung von Nutzen sind.

Dokumentationen erstellt das Team nur, falls diese während des Sprints hilfreich sind. Benutzerdokumentationen für den Kunden sind in jedem Fall zu erstellen.

Niemand kann von aussen Ziel und Umfang des Sprints verändern oder dem Team Vorgehensweisen vorschreiben, während der Sprint im Gange ist. Von den vier Projektvariablen sind Zeit, Kosten und Qualität während eines Sprints fixiert. Nur der Umfang kann verändert werden, wobei jedoch das Sprint Goal unangetastet bleiben muss.

So kann der Fall eintreten, dass sich das Team bei der Zuweisung von Elementen aus dem Product Backlog während des Sprint Planning Meetings grob verschätzt hat. Dann ist es sinnvoll, dass das Scrum Team mit dem Product Owner über eine Erweiterung beziehungsweise Verringerung der für den Sprint geplanten Aufgaben verhandelt. Dabei sollte das Sprint Goal, wie bereits erwähnt, nicht verändert werden. Am Ende des Sprints kann das Management dann am Sprint Review Meeting entsprechend der weniger beziehungsweise zusätzlich produzierten Funktionalität dem Product Backlog Elemente hinzufügen oder überflüssige entfernen.

Das Scrum Team hat ausserdem jederzeit die Möglichkeit, einen Sprint abzubrechen. Dies kann zum Beispiel sinnvoll sein, wenn das Sprint Goal obsolet wird oder zu viele Hindernisse auftauchen, welche das Erreichen des Sprint Goals in jedem Fall verunmöglichen. Dann muss ein neues Sprint Planning Meeting angesetzt werden, um ein neues Sprint Goal festzulegen.

Der Abbruch eines Sprints ist jedoch sehr selten, da einerseits ein Sprint nur 30 Tage dauert, also eine überschaubare Zeitspanne umfasst, andererseits die Hemmschwelle für einen Abbruch meist hoch ist.

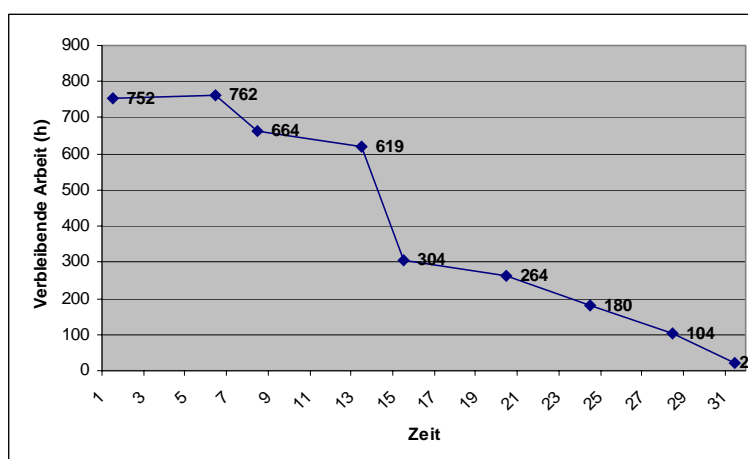


Abbildung 3 Beispiel für den Verlauf eines Sprints

#### Abbildung 3

Bei diesem Sprint wurden dem Sprint Backlog zu viele Elemente zugewiesen. Deshalb ist mit dem Product Owner entschieden worden, den Umfang der Aufgaben zu verringern. Das führte zum starken Abfall der verbleibenden Arbeitsstunden zwischen dem 13. und dem 15. Tag. Nach der Reduktion der Aufgaben verlief der Sprint ohne grosse Probleme und wurde erfolgreich abgeschlossen.



#### 4.2.7 Sprint Review Meeting

Mit dem Sprint Review Meeting wird am Ende des Sprints die neu erstellte Produktversion dem Management, den Kunden, den Usern und dem Product Owner vorgestellt. Das Treffen wird vom Scrum Master geleitet und das Produkt wird durch das Team präsentiert. Dabei wird versucht, möglichst alle Aspekte der neuen Produktversion aufzuzeigen, wie neue Funktionalität, Architektur oder Design. Ausserdem wird über den Verlauf des Sprints berichtet.

Die Resultate des Sprints werden so dem im Sprint Planning Meeting beschlossenen Sprint Goal und dem für den Sprint zugewiesenen Product Backlog gegenübergestellt. Aufgrund dieser Standortbestimmung können nun die Zuhörer über das weitere Vorgehen bezüglich des Projekts entscheiden. Dies wird vielfach zur Änderung von Elementen des Product Backlogs führen.

Das Sprint Review Meeting ist sehr informell gehalten. Es soll sich primär um eine Arbeitssitzung handeln, bei dem Fragen gestellt, Beobachtungen weitergegeben oder Vorschläge gemacht werden können.

Anschliessend an das Sprint Review Meeting und die Korrektur der Anforderungen an das Produkt kann ein neuer Sprint mit einem Sprint Planning Meeting initiiert werden. Ist der Kunde jedoch mit dem entstandenen Produkt zufrieden oder ist die zur Verfügung stehende Zeit oder die bewilligten Sachmittel aufgebraucht, kann die Entwicklung an dieser Stelle auch beendet werden.

## 5 Einsatzbereiche für Scrum

### 5.1 Einsatz in neuem oder bestehenden Projekt

Scrum eignet sich sowohl für den Einsatz bei neuen Projekten als auch zum ‚Retten‘ von festgefahrenen Projekten, welche mit herkömmlichen, klassischen Methoden begonnen wurden.

Wird ein Projekt von Anfang an mit Scrum durchgeführt, so muss zuerst ein genügend grosser Product Backlog vorhanden sein. Ausserdem sollte ein System Framework für die Entwicklung des Systems in der gewählten Technologie erstellt werden. In einem ersten Sprint wird dann die Entwicklungsumgebung für das Projekt aufgesetzt sowie eine erste Schlüsselfunktionalität des Produkts implementiert. Diese erste erstellte Funktionalität ist wichtig, damit der Kunde merkt, dass es nun ernst gilt, und er sich im Projekt engagieren muss, damit seine Anforderungen an das Produkt umgesetzt werden.

Wird Scrum in einem laufenden, festgefahrenen Projekt eingeführt, so hat das Projektteam meist entweder bisher noch gar keine oder seit langer Zeit keine funktionierenden Programmteile geliefert. Der Kunde rechnet in einem solchen Fall oftmals nicht mehr mit einem erfolgreichen Abschluss des Projektes. Darum geht es nun als erstes darum, das Vertrauen des Kunden in das Projektteam und das des Projektteams in sich selbst wiederherzustellen. Es ist deshalb sinnvoll, in einem ersten Sprint irgendeine Funktionalität des Produkts zu implementieren, welche in 30 Tagen geschaffen werden kann. So wird bewiesen, dass das Team Ergebnisse liefern kann. Dann kann meist in einem Sprint Planning Meeting mit der regulären Entwicklung weitergefahren werden.

Während der weiteren Durchführung des Projekts sollten in beiden Szenarien dann die Entwicklungspraktiken des Teams sukzessive optimiert beziehungsweise an die Vorgehensweise von Scrum angepasst werden.



#### 5.2 Ein grosses oder mehrere voneinander abhängige Projekte

Wie in Abschnitt 4.2.3 bereits erwähnt, setzt sich eine Scrum Team typischerweise aus fünf bis acht Personen zusammen. Dies limitiert die Grösse eines mit Scrum durchführbaren Projekts beträchtlich. Für grössere Projekte können jedoch mehrere Scrum Teams gleichzeitig eingesetzt werden. Diese starten einen Sprint alle zur selben Zeit und präsentieren ihre Ergebnisse am Sprint Review Meeting.

Zur Koordination der einzelnen Teams untereinander finden jeweils ein- bis zweimal wöchentlich ein Treffen mit je einem Mitglied von allen Teams statt, ein sogenanntes ‚Scrum of Scrums‘. Diese Struktur kann sich auch über mehrere Ebenen fortsetzen. Jeff Sutherland, einem der Mitbegründer von Scrum ist es auf diese Weise gelungen, erfolgreich mit Scrum ein Projekt mit über 800 Mitarbeitern durchzuführen.

Eine ähnliche Ausgangslage liegt vor, wenn in einem Unternehmen mehrere Applikationen gleichzeitig entwickelt werden, welche jeweils Komponenten der anderen wiederverwenden sollen. Scrum schlägt in diesem Fall vor, dass zuerst nur mit einer einzigen Applikation gestartet wird. Sind bei dieser einmal genügend stabile Komponenten vorhanden, welche wiederverwendet werden können, so sind diese auszugliedern und einem neuen Scrum Team, dem Shared Resources Team, zu übergeben, welches neu die Pflege dieser Komponenten übernimmt.

Nun kann nacheinander mit der Entwicklung der weiteren Applikationen begonnen werden, welche Komponenten des Shared Resources Teams verwenden oder ihm ihrerseits weitere Komponenten zur Pflege übergeben können.

Bei der Entwicklung der Applikationen sollte darauf geachtet werden, dass zum Beispiel eine Schichten-Architektur mit Komponenten verwendet wird, um Wiederverwendung zu vereinfachen. Dabei ist jedoch festzuhalten, dass Wiederverwendung für die Scrum Teams nur zweite Priorität hat. Ihr Ziel wird immer noch durch das Sprint Goal bestimmt.

Die oben beschriebene Vorgehensweise kann leicht für ein grosses Projekt mit mehreren Scrum Teams übernommen werden, welche alle an derselben Applikation arbeiten. Dies wird möglich, indem den einzelnen Teams Elemente des Backlogs dermassen zugewiesen werden, dass diese alle auf denselben Grundobjekten beruhen, dann aber jeweils einzelne Teilaspekte des Gesamtprodukts behandeln. Ein Beispiel dafür wäre das Grundobjekt ‚Person‘. Teams könnten sich nun mit den Aspekten ‚Angestellter‘, ‚Adressdaten‘ oder ‚Mitarbeitervergünstigungen‘ befassen. Auf diese Weise kann die Problemstellung oftmals auf die der Wiederverwendung reduziert werden.

#### 5.3 Scrum und XP

Da Scrum primär die Management Ebene der Software Entwicklung betrachtet, kann es leicht mit anderen Methoden kombiniert werden, welche auf die Entwicklungspraktiken fokussieren. Dies wurde vor allem mit Extreme Programming (XP) versucht.

Dabei zeigte sich, dass sich die beiden Methoden sehr gut ergänzen. Einzig beim Planning Game (XP) respektive Sprint Planning Meeting (Scrum) liegt eine Überschneidung vor. Diese ist jedoch nicht gravierend. Eine Kombinationsmöglichkeit ist, dass User Stories von XP als Backlog Elemente für den Product Backlog verwendet werden.



Es gibt verschiedene Versuche, welche Scrum mit XP kombinieren. Ein erster stammt von Ken Schwaber und Martin Fowler, welche ihr Produkt xp@Scrum beispielsweise erfolgreich bei Trans Canada Pipeline Ltd. eingeführt haben.<sup>6</sup>

Die Firma PatientKeeper, in welcher Jeff Sutherland, der andere Mitbegründer von Scrum, tätig ist, bietet ebenfalls einen eigenen Ansatz an. [SUTHERLAND]

Ausserdem hat Mike Beedle ein ähnliches Produkt kreiert, das er XBreed (Crossbreed) nennt und welches zusätzlich zu XP und Scrum noch weitere Elemente enthält. [XBREED]

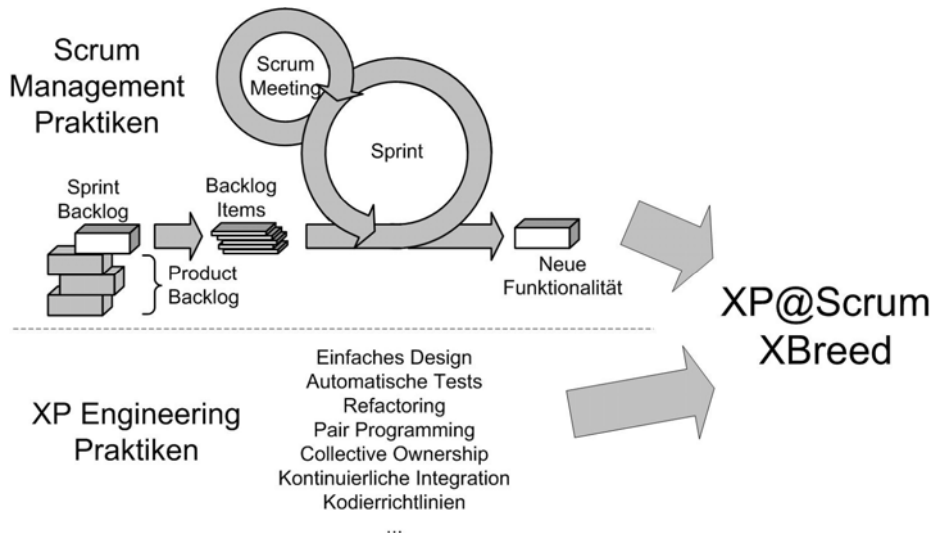


Abbildung 4 Kombination von Scrum und Extreme Programming

## 6 Erfolgsfaktoren

### 6.1 Die menschliche Komponente

Es ist festzuhalten, dass die menschliche Komponente beim Einsatz von Scrum eine äusserst kritische Rolle spielt. Dem einzelnen Mitarbeiter wird durch diesen Ansatz sehr viel Verantwortung übertragen. Dabei kann es auch passieren, dass dies von den Beteiligten ausgenützt wird. Scrum geht jedoch von einem positiven Menschenbild aus, das voraussetzt, dass die Mitarbeiter sich für das Projekt motivieren lassen. Die Mitarbeiter müssen sich dabei ihrer Verantwortung bewusst sein und entsprechend handeln. Ist dies nicht der Fall, so sind Projekte mit Scrum zum Scheitern verurteilt

#### 6.1.1 Scrum Master

Der Scrum Master ist in jedem Scrum Projekt die zentrale Figur. Er ist Motivator und Facilitator, muss das Team von äusseren Einflüssen abschirmen und Hindernisse beseitigen. Mit seinem Einsatz und Durchsetzungsvermögen steht und fällt Scrum. Entscheidend ist vor allem auch, dass er vom Unternehmen die entsprechenden Kompetenzen erhält, um auftretende Hindernisse beseitigen zu können. Dies setzt vom Unternehmen ein grosses Vertrauen in Scrum voraus, da so bestehende unternehmensweite Vorschriften in Frage gestellt werden können.

<sup>6</sup> <http://www.computerworld.com/softwaretopics/software/appdev/story/0,10801,69183,00.html> und [ADM], <http://www.controlchaos.com/XPKane.htm>



## Seminar Software Engineering

### Agile vs. klassische Methoden der Software-Entwicklung, WS 03/04

#### Scrum

Dozent: Prof. Dr. Martin Glinz  
Betreuender Assistent: Christian Seybold

#### 6.1.2 Kunde

Damit ein Scrum Projekt Erfolg haben kann, muss der Kunde sich aktiv am Entwicklungsprozess beteiligen. Die Interaktion des Kunden mit dem Scrum Team und dem Product Owner an den Sprint Review Meetings ist absolut notwendig. Nur so ist das Team in der Lage, auch die gewünschte Software zu liefern. Der Kundenvertreter muss dazu kompetent und auch weisungsbefugt sein.

#### 6.1.3 Scrum Team

Dass die Zusammenarbeit innerhalb des Scrum Teams von entscheidender Bedeutung ist, muss kaum mehr erwähnt werden. Jeder einzelne hat die ihm übertragenen Aufgaben nach bestem Gewissen zu erledigen und die anderen bei ihrer Arbeit zu unterstützen. Dabei muss darauf geachtet werden, dass nicht wieder eine Rückkehr zu den klassischen Rollen in einem Entwicklungsteam stattfindet. In einem Scrum Team sind prinzipiell alle gleichgestellt. Andererseits sollte aber in jedem Team ein Entwickler mit grosser Erfahrung vertreten sein, der in Architekturentscheidungen das Vorgehen in die richtigen Bahnen lenkt und die unerfahrenen Teammitglieder unterstützt.

#### 6.1.4 Scrum Grundwerte

Ken Schwaber und Mike Beedle haben fünf Grundwerte definiert, welche Voraussetzung für den erfolgreichen Einsatz von Scrum sind. [SCRUM01] Diese Grundwerte sind implizit in den oben erwähnten Erfolgsfaktoren enthalten. Sie sind äusserst allgemein gehalten und sicherlich auch ausserhalb von Scrum erstrebenswert. Aus diesem Grund wird an dieser Stelle nur eine Auflistung wiedergegeben:

<b>Commitment</b>	Be willing to commit to a goal. Scrum provides people all the authority they need to meet their commitments.
<b>Focus</b>	Do your job. Focus all of your efforts and skills on doing the work that you've committed to doing. Don't worry about anything else.
<b>Openness</b>	Scrum keeps everything about a project visible to everyone.
<b>Respect</b>	Individuals are shaped by their background and their experiences. It is important to respect the different people who comprise a team.
<b>Courage</b>	Have the courage to commit, to act, to be open, and to expect respect.

### 6.2 Grenzen von Scrum

Bezüglich des Einsatzes von Scrum in grösseren, komplexen Projekten konnten leider keine unabhängigen Berichte gefunden werden. Die Anwendbarkeit auf diese Art von Projekten wird aber auch von langjährigen Praktikern, bei allem Lob für Scrum, bezweifelt. [RISING00] Der Ansatz von Scrum scheint eher für überschaubare Projekte bis mittlerer Grösse geeignet.

Scrum setzt voraus, dass sich das zu erstellende Software Produkt in einzelne Funktionen, beziehungsweise Elemente, aufteilen lässt, deren Erstellung wiederum in einzelne Arbeiten aufgeteilt werden kann. Diese Bedingungen sind bei der Entwicklung von objektorientierter Software vielfach gegeben. Ein Einsatz ausserhalb dieses Bereichs ist aber oftmals nicht problemlos möglich.





Da der Erfolg von Scrum, wie im vorherigen Abschnitt erwähnt, teilweise stark von einzelnen Personen wie dem Scrum Master abhängig ist, muss auch überlegt werden, was bei Ausfall einer Schlüsselperson mit dem Scrum Projekt geschieht. Gibt es irgend eine Standardprozedur für einen solchen Fall? Diese Frage konnte im Rahmen dieser Arbeit nicht schlüssig beantwortet werden.

Ein weiteres Thema, welches in den vorhandenen Unterlagen nicht abschliessend behandelt wird, ist die Frage der Dokumentation. Aussagen dazu sind immer sehr allgemein gehalten, so zum Beispiel, dass Dokumentationen nur dann erstellt werden sollen, wenn sie für den Entwicklungsprozess notwendig sind. Ausserdem müssen für die Präsentation der neuen Produktversion am Ende eines Sprints sicherlich ein oder mehrere Übersichtsdiagramme erstellt und bei der Fertigstellung des Produkts dem Kunden eine Benutzerdokumentation abgegeben werden. Weitere Dokumente wären aber gerade bei einem Produkt mit längerer Lebensdauer nützlich, um die Pflegbarkeit der Software zu gewährleisten.

## 7 Zusammenfassung und Ausblick

Bei Scrum handelt es sich um einen leichtgewichtigen Managementprozess, der sich auf Projekte von verschiedenen Grössen anwenden lässt.

Die Grundeigenschaften von Scrum sind:

- Kleine Teams, welche die Kommunikation und den informellen Informationsaustausch fördern
- Anpassungsfähigkeit bezüglich Veränderungen von Technologie und Benutzeranforderungen
- Häufige Erstellung von neuen Produktversionen, welche inspiziert, angepasst, getestet und dokumentiert werden können
- Aufteilung der zu erledigenden Arbeit in kleine, voneinander möglichst unabhängige Teilaufgaben
- Die Möglichkeit, ein Projekt jederzeit als beendet zu erklären, sei dies aus zeitlichen, finanziellen, wettbewerbstechnischen oder anderen Gründen

Die grosse Stärke der Methode liegt in der Einfachheit ihres Ansatzes und ihren klar definierten Regeln. Scrum ist in seinen Grundzügen jedem nach kurzer Zeit verständlich.

Durch die Übertragung von Verantwortung auf jeden einzelnen können grosse Produktivitätssteigerungen erreicht werden. Diese Delegation von Kompetenzen kann aber bei Missbrauch auch zum Scheitern der Methode führen.

Sehr vielversprechend tönt die Kombination von Scrum mit Extreme Programming, die naheliegend scheint.

Scrum wird momentan immer noch stark von seinen Erfindern getragen. Inwieweit sich die Methode nach deren Ausscheiden halten wird, kann heute noch nicht mit Sicherheit beantwortet werden. Durch den simplen Aufbau von Scrum und durch die Organisation der Scrum Master in einer Community mit klar definierten Regeln sind die nötigen Voraussetzungen für eine weitere Verbreitung auf jeden Fall gegeben.



## Seminar Software Engineering

Agile vs. klassische Methoden der Software-Entwicklung, WS 03/04

### Scrum

Dozent: Prof. Dr. Martin Glinz  
Betreuender Assistent: Christian Seybold

## Abbildungsverzeichnis

- Abbildung 1 Funktionsweise von Scrum, nachgebildet gemäss Graphik  
<http://www.controlchaos.com/Scrumo.htm>
- Abbildung 2 Beispiel Product Backlog, nachgebildet gemäss Graphik  
<http://www.mountangoatsoftware.com/scrum/productbacklog.php>
- Abbildung 3 Beispiel für den Verlauf eines Sprints, nachgebildet gemäss Figure 4.8, S.79, in [SCRUM01]
- Abbildung 4 Kombination von Scrum und Extreme Programming, gemäss Graphik  
<http://www.controlchaos.com/xpScrum.htm>

## Quellenverzeichnis

- [SCRUM01] Schwaber, K., Beedle, M.: Agile Software Development with Scrum, Prentice Hall, 2001, <http://www.agilescrum.com>
- [PLOP98] Beedle, M., Devos, M., Sharon, Y., Schwaber, K., Sutherland, J., SCRUM: An extension pattern language for hyperproductive software development, Paper for Conference on Pattern Languages of Programs (PLoP), 1998, <http://www.mikebeedle.com/pub/scrum.pdf>
- [TAKEUCHI86] Takeuchi, H., Nonaka, I., The new new Product Development Game, Harvard Business Review 86116, 1986, [http://harvardbusinessonline.hbsp.harvard.edu/b01/en/common/item\\_detail.jhtml?id=86116](http://harvardbusinessonline.hbsp.harvard.edu/b01/en/common/item_detail.jhtml?id=86116)
- [TAKEUCHI95] Takeuchi, H., Nonaka, I., The Knowledge Creating Company, Oxford University Press, 1995, <http://www.oup-usa.org/isbn/0195092694.html>
- [RISING00] Rising, L., Janoff, N.: The Scrum Software Development Process for Small Teams, IEEE Software, 2000, <http://members.cox.net/risingl1/articles/IEEEScrum.pdf>
- [ADM] Offizielle Webseite zu Scrum von ADM (Ken Schwaber), <http://www.controlchaos.com>
- [ALLIANCE] Webseite der Community der zertifizierten Scrum Master, siehe vor allem auch Artikelsammlung, <http://www.scrumalliance.org>
- [SUTHERLAND] Webseite von Jeff Sutherland, Mitbegründer von Scrum, siehe vor allem SCRUM Log, <http://www.jeffsutherland.com>
- [XBREED] Offizielle Webseite zu X Breed von Mike Beedle, <http://www.xbreed.net>