Artikel

Von der schwierigen Scrum-Einführung zum ScrumAnd

Scrum ohne Wenn und Aber?

von Andrea Herrmann

Zu den vielen Scrum-Erfolgsgeschichten gesellen sich auch Berichte über Einführungsschwierigkeiten sowie über das Bedürfnis, Scrum anzupassen oder zu ergänzen. Das ist im Grunde nicht sehr verwunderlich, da Scrum einen sehr streng durchgetakteten Arbeitsrhythmus vorgibt, der vielleicht nicht überall passt. Außerdem macht es viel weniger Vorgaben als andere Vorgehensmodelle, eben nur die allernötigsten. In diesem Artikel geht es um praktische Einführungsschwierigkeiten, um die beliebtesten und unbeliebtesten Scrum-Praktiken und häufig praktizierte Anpassungen. Letztlich ist Scrum vielleicht vor allem ein idealer Ausgangspunkt für die Entwicklung eines firmen- oder teamspezifischen Vorgehensmodells, weil Scrum erstens mit einem Mindestsatz an Managementtechniken startet und zweitens die regelmäßige Überprüfung und Verbesserung der Arbeitsweise beinhaltet.

Schwierigkeiten bei der Einführung von Scrum

Obwohl die dokumentierte Geschichte von Scrum mehr Erfolgs- als Misserfolgsmeldungen umfasst, gibt es auch anderslautende Berichte über Herausforderungen bei der Einführung und von einzelnen Mitarbeitern, die sich ausklinken, Firmen, die Scrum wieder aufgeben. Vermutlich gibt es in den Publikationen einen „publication bias“: Erfolgsgeschichten werden eher veröffentlicht als Misserfolge.

Inzwischen gibt es aber nicht nur unter der Hand im vertraulichen Zweiergespräch Berichte über wenig erfolgreiche Scrum-Einführungen, sondern auch in der Fachliteratur kommen sie immer wieder vor.

Grundsätzlich treten beim Übergang zu Scrum dieselben Schwierigkeiten und Schmerzen auf, wie sie generell bei der Einführung neuer Arbeitsweisen häufig auftreten: Nicht alle Betroffenen stehen hinter der Änderung und halten sich an die Regeln, es fehlt die Erfahrung mit den Methoden oder die neue Arbeitsweise passt nicht zur Firmenkultur. Darum gibt es ja die Disziplin des Change-Managements, und die Einführung von Scrum muss auf jeden Fall durch ein Change-Managementkonzept begleitet werden. Dazu gehört die Unterstützung der Umstellung durch das Management und das Ins-Boot-Holen der Mitarbeiter, die Definition von Vorgehensweisen und die Bereitstellung von Werkzeugen, die Schulung der Betroffenen und die Unterstützung durch Change-Champions und Experten (beispielsweise einen zertifizierten Scrum Master).

Die am häufigsten genannten Schwierigkeiten bei der Scrum-Einführung – abgesehen von den oben genannten Change-Managementproblemen – lauten (z. B. bei [Mar08], [Haj11] [Kra14]):

Kunde: Scrum betrifft nicht nur das Scrum-Team und das Management innerhalb der Firma, sondern auch den Kunden. Passt Scrum nicht zu dessen Vorgaben, dann muss irgendein Kompromiss gefunden werden, beispielsweise ein ScrumAnd: Man macht Scrum und erstellt noch zusätzliche Dokumente oder führt zusätzliche Aktivitäten aus, um diese Vorgaben zu erfüllen.

Aufwandsschätzung: Die Aufwandsschätzung wird zwar genauer, wenn man kleine statt große Arbeitspakete schätzt, und doch verschätzen sich auch Scrum-Teams. Somit muss am Ende der Timebox Arbeit in den nächsten Sprint verschoben werden, das Sprint-Ziel wurde nicht erreicht. Wenn man Scrum richtig anwendet, führt allerdings das Lernen zu immer besseren Prognosen.

Product Owner: Der Product Owner formuliert unklare Anforderungen oder ist nicht kompetent, um alle Arten von Anforderungen zu liefern.

Besprechungsdisziplin: Die Scrum-Events folgen einer klar definierten Tagesordnung und behandeln nur die vorab definierten Fragen. Trotzdem kommt es auch in Scrum-Meetings vor, dass die Diskussion nicht im Zeitrahmen bleibt und auch nicht beim Thema. Dieses Problem bestand vermutlich schon vor der Scrum-Einführung und wird durch Scrum erst richtig sichtbar.

Backlog-Verwaltung: Der Aufwand für die Verwaltung des Backlogs ist hoch. Darum wird das Backlog nicht so ordentlich gepflegt, wie es sein sollte. Damit fehlt dem Team jedoch eine verlässliche Informationsgrundlage für wichtige Entscheidungen.

Rollen, Selbstverständnis und Karrierewege: Die drei Scrum-Rollen Development Team (kurz Team), Scrum Product Owner und Scrum Master wirbeln nicht nur die Rollenbezeichnungen innerhalb eines Projektteams durcheinander, sondern auch die Bezeichnungen der Positionen innerhalb der Firma. Außerdem erweitern sie das Aufgabenspektrum von Mitarbeitern sowie die geforderten Qualifikationen. Es verändert oder beseitigt eventuell sogar bisher existierende Karrierewege, zum Beispiel vom Junior zum Senior. Wie macht man eigentlich Karriere innerhalb von Scrum?

Spezialisten: Obwohl Scrum auf generalistisch qualifizierte Mitarbeiter hofft, die flexibel alle anstehenden Aufgaben übernehmen können, führt die technologische Komplexität in der Praxis doch oft dazu, dass Entwickler zu Experten werden, die sich mit einer bestimmten Technologie besonders auskennen und eventuell sogar nur mit dieser Technologie arbeiten. Diese ziehen ihrer eigenen Wahrnehmung nach wenig Nutzen aus dem Daily Stand-up und bleiben darum fern.

Mikromanagement: Vorgesetzte nutzen die durch Scrum erzeugte Transparenz über den Arbeitsfortschritt und die kleinteilige Planung dazu, ihre Mitarbeiter im Detail zu überwachen und in die Selbstorganisation des Teams einzugreifen. Dann wird aus dem Pull-Prinzip ganz schnell wieder Push, aus dem „sustainable pace“ der alte Stress. Vor lauter Prozessen und Werkzeugen kommen dann das Zwischenmenschliche und die Kommunikation doch zu kurz. Insbesondere werden Fehler, die jemand macht, schonungslos offenbar.

Kulturelle Änderung: Es werden zwar die Praktiken ausgeführt, aber die Denkweise in den Köpfen ist noch nicht agil. Das genügt nicht. Der Wandel muss wegführen vom streng hierarchisch organisierten Projekt, bei dem der Projektleiter für alles verantwortlich ist. Das Scrum-Team organisiert sich selbst mit allen Vor- und Nachteilen: Entscheidungen werden gemeinsam getroffen und sind verbindlich, wer eine Aufgabe übernimmt, ist dafür verantwortlich.

Alle diese Schwierigkeiten verhindern nicht unbedingt die Scrum-Einführung, führen jedoch zu Frust und müssen gezielt abgefangen werden.

Geht der Schmerz vorüber oder passt Scrum wirklich nicht?

Die Einführung neuer Arbeitsweisen wie Scrum verursacht immer gewisse Schmerzen, vor allem wenn es einen großen Unterschied zur bisher gelebten Vorgehensweise gibt. Das bedeutet noch nicht, dass Scrum nicht passt. Nach einer gewissen Eingewöhnungsphase beginnt das Team dann vielleicht doch noch damit, produktiv und effizient zu werden, und sich dabei wohlzufühlen. Um das herauszufinden, darf man nicht zu früh aufgeben.

Woher aber weiß man schon vorher, ob sich die Mühe lohnen wird? Oder wie weiß man schon vorher, dass Scrum in diesem Umfeld nicht funktionieren wird?

Scrum in Reinform passt tatsächlich in vielen Fällen schlecht, beispielsweise

wenn das Team mehr als 10 Personen umfasst,

wenn der Kunde ein Lastenheft vorgibt und einen Festpreis einfordert,

wenn ein innovatives Produkt neu entwickelt werden soll,

wenn für später umfangreiche Dokumentationen benötigt werden,

wenn kein kompetenter Product Owner zur Verfügung steht.

Tatsächlich sind jedoch inzwischen zahlreiche Scrum-Varianten entstanden, die für fast jedes Umfeld die richtige Arbeitsweise anzubieten scheinen. Die Alternative lautet darum nicht nur: „Scrum – ja oder nein?“ Es stehen noch weitere Möglichkeiten zur Verfügung. Manche Scrum-Variante ist dann allerdings auch nicht mehr richtig agil.

Scrum-Varianten

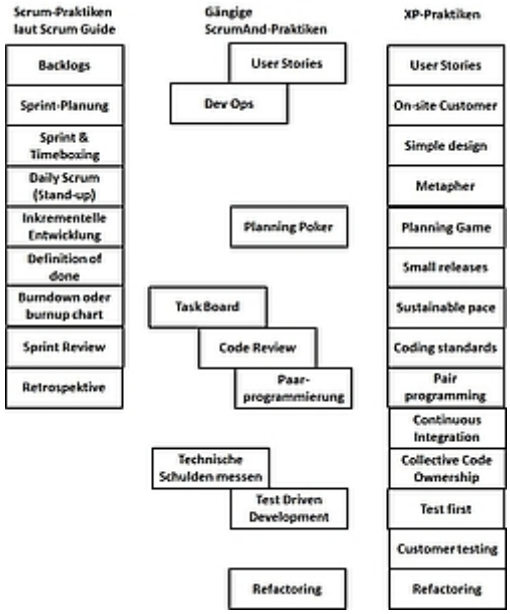

Abb. 1: Die Scrum-Praktiken laut Scrum Guide [Scrum], die XP-Praktiken nach Kent Beck [Beck00] und übliche Praktiken von ScrumAnds (verschiedene Quellen)

Über Extreme Programming (XP) schrieb Kent Beck [Beck00], man müsse alle seine Praktiken anwenden, weil die Nachteile der einen durch andere ausgeglichen werde. Scrum sieht nach dem Leitfaden von Schwaber und Sutherland [Scrum] allerdings ausdrücklich vor, dass in der Retrospektive der Arbeitsprozess angepasst wird. Dabei kann das Team entscheiden, Praktiken wegzulassen, durch andere Praktiken zu ersetzen oder durch weitere zu ergänzen. So entstanden verschiedenste Scrum-Varianten:

ScrumBut: Wir machen Scrum, aber wir lassen diese und jene Praktik weg.

ScrumAnd: Wir machen Scrum und ergänzen es um eine oder mehrere zusätzliche Praktiken.

ScrumBan: Eine Mischung aus Scrum und Kanban.

Water-Scrum-Fall: Hier wird nur die Implementierung mit Scrum durchgeführt, aufbauend auf einem Big Upfront Design (BUD), Machbarkeitsstudie und Festpreis-Budgetierung, und dann abgeschlossen durch eine Qualitätssicherung des Gesamtsystems und einer Big Bang-Einführung.

Large Scale Scrum (LeSS): Ein Vorgehen, bei dem mehrere Scrum-Teams dasselbe Product Backlog verwenden [LeSS].

Scaled Agile Framework (SAFe): Hier arbeiten die Scrum-Teams eingebettet in ein Umfeld von Programm- und Portfolio-Management [SAFe].

Diese Liste ist vermutlich nicht vollständig und wird sich zukünftig auf jeden Fall noch verlängern. Allein dies ist ein Indiz dafür, dass Scrum nicht überall in derselben Reinform passt.

Abbildung 1 zeigt die Liste der Scrum-Praktiken und der XP-Praktiken und dazwischen dargestellt solche, die in ScrumAnds gerne zusätzlich noch eingesetzt werden.

Die beliebtesten und unbeliebtesten Scrum-Praktiken

Scrum ohne Abweichung setzen nur 42 Prozent aller Scrum-Teams ein. Die folgenden Zahlen stammen aus der Umfrage der Agile Alliance [ScrumR] aus dem Jahr 2015 sowie aus einer Umfrage der Firma SwissQ aus dem Jahr 2012, die leider nicht mehr online

steht.

Folgende Scrum-Praktiken werden von Scrum-Teams häufig verwendet:

- 90 Prozent arbeiten iterativ.
- 82 Prozent machen ein Daily Stand-up.
- 81 Prozent führen ein Backlog.
- 81 Prozent der Teams führen eine Retrospektive durch.
- 67 Prozent verwenden ein Burndown Chart.
- 64 Prozent definieren die „definition of done“.

Gerne weggelassen werden folgende Praktiken, die allerdings, wenn man genau hinsieht, auch gar nicht im Scrum Guide stehen:

- 58 Prozent machen Refactoring.
- 36 Prozent betreiben Test-Driven Development.
- 28 Prozent programmieren im Paar.
- 16 Prozent messen technische Schulden.

Gefahren von ScrumAnd und ScrumBut

ScrumAnds sind sehr häufig. Der Scrum Guide [Scrum] sagt beispielsweise nichts von User Stories oder Epics, Task Board, Planning Poker, Code-Reviews oder DevOps, obwohl diese Praktiken in der Scrum Community weit verbreitet eingesetzt werden. Ergänzt man Scrum durch eine zusätzliche Praktik, geht man meist kein großes Risiko ein. Vielleicht hat man ein Dokument erstellt, das dann doch niemand braucht, doch das stellt sich innerhalb einiger Sprints heraus und kann dann korrigiert werden.

ScrumButs jedoch sind gefährlich, weil hier Scrum-Praktiken weggelassen werden. Hierdurch kann ein viel höherer Schaden entstehen als durch das Hinzufügen einer Praktik. In Scrum ist schließlich nichts redundant, kein Event, keine Aktivität und kein Artefakt. Lassen Sie beispielsweise das Burndown Chart weg, dann fehlt es völlig an einer quantitativen Einschätzung des Arbeitsfortschritts während des Sprints.

Ausblick: Wie Sie Ihr eigenes ScrumAnd entwickeln

Scrum muss nicht überall passen, und tut es vermutlich auch nicht. Schließlich gibt es immer noch eine Vielzahl alternativer Vorgehensmodelle, die sich bewährt haben. Vor allem sollte man eine so abrupte Änderung der Arbeitsweise nur dann einführen, wenn alle Beteiligten und Betroffenen Lust darauf haben.

Wenn Sie aber Scrum verwenden möchten, es aber auf Anhieb nicht perfekt läuft, bietet Scrum durch seine kurzen Sprints, das Impediment-Backlog, den Scrum Master als Prozessverantwortlichen und die Retrospektive im Team die Möglichkeit, die Schwierigkeiten gemeinsam zu analysieren, zu verstehen und Gegenmaßnahmen auszuprobieren. Auf diese Weise können Sie die Arbeitsweise im Sprint-Takt inkrementell verbessern.

Als Ausgangspunkt für diese allmähliche Verbesserung der Arbeitsweise können Sie entweder Scrum laut Scrum Guide verwenden oder – viel weniger riskant – ein Water-Scrum-Fall. Starten Sie mit Scrum, dann lassen Sie anfangs testweise sehr viele Aktivitäten und Dokumente weg, die Sie bisher eingesetzt haben. Wenn Sie nach zwei Monaten feststellen, dass Ihnen nun doch ein Feindesign fehlt, müssen Sie dieses mühsam rekonstruieren. Die anfängliche Zeitersparnis wird so ganz schnell durch den Mehraufwand wieder aufgefressen.

Im Water-Scrum-Fall arbeiten Sie im Wesentlichen wie bisher, nur die Implementierung wird im Stile von Scrum organisiert und in Sprints durchgeführt. Dadurch riskieren Sie nicht viel und können ausprobieren, wie sich diese Arbeitsweise in Ihrer Unternehmenskultur bewährt. Nach und nach können dann die der Entwicklung vor- und nachgelagerten Arbeitsphasen in den Sprint-Rhythmus einbezogen werden.

Einladung zu einer Umfrage über die Umsetzung der Agilität in der Praxis

Die Universität Magdeburg (Fakultät für Informatik) führt eine Umfrage zur Verbreitung und Nutzung der agilen Softwareentwicklung in der Softwareindustrie im deutschsprachigen Raum durch. Dabei stellt sie nicht nur die Frage, welche agilen Praktiken in der Praxis angewendet werden, sondern auch, welche agilen Werte und Prinzipien. Untersucht werden soll auch der Einfluss der agilen Praktiken auf die Qualität der Kostenschätzung und der Software.

Die Zielgruppe sind alle, die aktiv in der Softwareentwicklung arbeiten, egal in welcher Rolle – Entwickler, Projektleiter, Berater, Tester usw. Die Beantwortung der Fragen dauert ca. 10 Minuten. Die Umfrage erfolgt anonym. Wenn Sie es wünschen, werden Ihnen die Ergebnisse zugesandt.

Wir freuen uns über Ihre Unterstützung und wünschen Ihnen viel Spaß beim Ausfüllen des Fragebogens unter:

https://www.soscisurvey.de/agile-software-entwicklung/

Literatur & Links

[Beck00] K. Beck, Extreme programming explained: embrace change, Addison-Wesley, 2000

[Haj11] H. Hajjdiab, Al Sh. Taleb, Adopting Agile Software Development: Issues and Challenges, in: Int. J. of Managing Value and Supply Chains (IJMVSC), Vol. 2, No. 3, September 2011, siehe: https://pdfs.semanticscholar.org/b081/d68c98f12808c658829c29e60df11337f31d.pdf

[Kra14] B. Kraft, A. Zöll, Von der Langstrecke zum Sprint – Agile Methoden in traditionellen Unternehmen, in: Tagungsband Projektmanagement und Vorgehensmodelle 2014 – Soziale Aspekte und Standardisierung, S. 35-45, 2014, siehe: https://subs.emis.de/LNI/Proceedings/Proceedings236/35.pdf

[LeSS] Overview: Large Scale Scrum, siehe: https://less.works/

[Mar08] A. Marchenko, P. Abrahamsson, Scrum in a Multiproject Environment: An Ethnographically-Inspired Case Study on the Adoption Challenges, in: Agile Conference 2008, pp. 15-26, siehe: http://artemmarchenko.com/files/Scrum%20in%20Multiproject%20Environment.pdf

[SAFe] http://scaledagileframework.com/

[Scrum] J. Sutherland, K. Schwaber, Der Scrum Guide – Der gültige Leitfaden für Scrum: Die Spielregeln, 2013, siehe: https://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-DE.pdf

[ScrumR] The 2015 State of the Scrum Report, ScrumAlliance, 2015, siehe: https://www.scrumalliance.org/scrum/media/scrumalliancemedia/files%20and%20pdfs/state%20of%20scrum/scrum-alliance-state-of-scrum-2015.pdf

Dr. Andrea Herrmann

ist seit 2012 freiberufliche Trainerin und Beraterin für Software-Engineering mit mehr als 20 Berufsjahren in Praxis und Forschung. Sie berät Firmen bei der scrumlosen, leichtgewichtigen Verbesserung ihrer Vorgehensweisen, insbesondere im Bereich des Requirements Engineering. Sie verfasste mehr als 100 Fachpublikationen, hält regelmäßige Konferenzvorträge und ist offizielle Supporterin des IREB-Board.

E-Mail: herrmann(at)herrmann-ehrlich.de

Bildnachweise:

Herrmann & Ehrlich