

„VIELE, DIE IN SCRUM-TEAMS ARBEITEN, WISSEN NICHT, WIE MAN TESTGETRIEBEN ENTWICKELT“

Interview mit Ken Schwaber über den Zustand von Scrum und seine neue Initiative scrum.org.

■ Ken, in den späten 80ern haben Sie zusammen mit Jeff Sutherland Scrum entwickelt. Zwanzig Jahre später benutzen laut einer Forrester-Studie mehr Firmen agile Methoden als den klassischen Wasserfall-Ansatz. 84 % dieser Firmen wiederum wendet Scrum an. Was halten Sie von diesen Zahlen?

■ Jeff Sutherland hat eine Folgestudie durchgeführt und dabei herausgefunden, dass etwas weniger als 50 % der Organisationen, die von sich sagen, sie praktizieren Scrum, tatsächlich auch iterative und inkrementelle Entwicklungstechniken nutzen. Dabei sind aber Iterationen und kurze Zyklen essenziell für Agilität. Erst die inkrementelle Entwicklung liefert die notwendige Transparenz, um einen Prozess intelligent zu inspizieren und anzupassen. Also musste ich erst einmal tief durchatmen. Viele Organisationen sagen, sie nutzen Scrum. Die meisten sind auch tatsächlich besser als vorher. Natürlich sind sie besser, als sie es wären, wenn sie in ihrer Softwareentwicklung das Wasserfall-Modell verfolgen würden. Aber: Scrum war als Framework konzipiert, als ein Werkzeug, mit dem eine Organisation Agilität und Professionalität erreichen kann. Wenn ein Team oder eine Organisation Scrum anwendet, dann werden die Hindernisse auf dem Weg zur Exzellenz sichtbar. Die Organisation priorisiert diese Hindernisse und beseitigt sie. Schritt für Schritt entsteht so eine bessere Entwicklungsorganisation. Das zumindest sind die Theorie und der Plan. Aber unglücklicherweise ist Scrum auch eine Modeerscheinung geworden. Obwohl wir mit Absicht die Nomenklatur wenig attraktiv gestaltet hatten (Scrum, Chicken, Daily Scrum, ...), um die Wahrscheinlichkeit klein

zu halten, dass daraus eine Modewelle wird, sind abstruserweise genau diese bizarren Scrum-Begriffe attraktiv geworden und haben die Anzahl der Leute, die behaupten, damit zu arbeiten, in die Höhe getrieben.

Im letzten Jahr wurde mir klar, dass wir einen Wendepunkt erreicht haben. Ich war mit meinem Rennrad unterwegs. Ein anderer Fahrer zog heran und wir kamen ins Gespräch: plaudern, schnaufen, plaudern, schnaufen. Der Kollege sagte, er sei Softwareentwickler und wollte wissen, wie ich meinen Lebensunterhalt bestreite. Ich antwortete ihm, dass auch ich Entwickler sei. Er sagte: „Cool, hast du schon mal was von Scrum gehört?“ Scrum hatte mein Radfahren erreicht.

■ Über die Unzulänglichkeiten in der Anwendung wollen wir gleich reden – ich komme darauf zurück. Scrum hat also innerhalb einer Dekade die Welt der Projektsteuerung erobert. Wenn Sie die Softwareindustrie von heute betrachten, was sind für Sie die wichtigsten Auswirkungen von Scrum?

■ Die größte Wirkung, die Scrum erzielt hat, besteht darin, dass wir nun endlich einen Prozess zur Entwicklung komplexer Produkte nutzen, der dafür auch geeignet ist. Plangetriebene¹⁾ Projektmanagement-Techniken, wie der Wasserfall, sind hervorragend für großvolumige, einfache, wohlbekanntere Probleme geeignet. Empirische, adaptive Techniken für das Produkt-

¹⁾ Anmerkung Matthias Grund: Ken Schwaber benutzt zur Kennzeichnung dieser plangetriebenen Prozesse den Begriff „predictive“, für das es im Deutschen keine prägnante und treffende Übersetzung gibt. Im weiteren Verlauf habe ich „predictive“ mit „plangetrieben“ übersetzt.



Ken Schwaber

ist einer der einflussreichsten Ideengeber der agilen Softwareentwicklung. Er ist Mitgründer der Agile Alliance, Mitinitiator des Agilen Manifests und Gründer der Scrum Alliance. Als Begründer von Scrum (zusammen mit Jeff Sutherland) hat er Softwaregeschichte geschrieben.



Matthias Grund

ist Vorstand der andrena objects ag. Sein besonderes Interesse gilt der Unterfütterung von „Scrum Best Practices“ des Software-Engineering. Zusammen mit Joseph Pelrine hat die Firma andrena 2003 die ersten deutschsprachigen Scrum-Kurse angeboten.

management, wie Scrum, sind für komplexe Situationen gedacht, in denen das Nichtwissen größer ist als das Wissen. Die wirklich nützlichen und erst recht die bahnbrechenden Anwendungen werden in solchen komplexen Entwicklungssituationen erstellt. Häufig handelt es sich um komplexe Produkte, manchmal sind diese entscheidend für den Unternehmenserfolg oder gar in ihrer Anwendung überlebenskritisch.

Wenn ich mit einem Auto fahre, das zum Großteil von Computern gesteuert wird, wenn meine Ärzte Kernspintomographien analysieren, die von Computern erfasst und dargestellt werden, wenn medizinische Apparate in meinen Körper eingebaut werden, dann bin ich heilfroh darüber, dass diese Produkte zunehmend in agilen und adaptiven Prozessen wie Scrum entwickelt werden. Der Grad der Unsicherheit und die Intransparenz in plangetriebenen Prozessen machen jede angebliche Qualitätsstufe zweifelhaft.

Ich dachte, empirische Prozesssteuertechniken seien relativ neu im fortgeschrittenen Produktmanagement, bis ich eine E-Mail von einem Manager aus der Halbleiter-

Industrie bekam, der in seinem Engineering-Bereich agile Methoden eingeführt hatte, wie sie in den 50er Jahren in der Lockheed Skunkworks²⁾ Gruppe etabliert wurden.

Jeff und ich wussten immer, dass Scrum nicht neu ist. Stattdessen war Scrum eine empirische Kombination von Best Practices in ein Framework mit verbindlichen Regeln und einer festgelegten Terminologie. Dennoch war es sehr befriedigend, zu erfahren dass die Praktiken, die wir uns geborgt hatten, bereits vor so langer Zeit, nämlich in den 50er Jahren, zur Entwicklung komplexer und anspruchsvoller Produkte, die ihrer Zeit weit voraus waren, genutzt wurden.

Ich wusste, dass meine Disziplin und mein Berufsstand, die Softwareentwicklung, durch den unangemessenen Gebrauch solcher plangetriebener Methoden wie dem Wasserfall immer weiter in die falsche Richtung getrieben wurden. Gleichzeitig wurde und wird unsere Welt immer komplexer. Während die professionelle Praxis in der Softwareentwicklung degenerierte, wuchs auf der anderen Seite die Notwendigkeit, eben diese Praxis weiter zu entwickeln. Das wurde geradezu essenziell für unsere große komplexe Gesellschaft. Ich weiß, dass das Ablegen alter Gewohnheiten und das Aneignen neuer und besserer Gewohnheiten für viele schwierig sind. Doch es führt kein Weg daran vorbei und der persönliche und der geschäftliche Gewinn sind hoch.

■ Wir haben den Tod des Wasserfalls und die weltweite Verbreitung von Scrum erlebt, das Entstehen einer enormen Community. Scrum scheint zum Mainstream zu werden und zu einem Muss für modernes Projektmanagement. Es wird

²⁾ Anmerkung von Matthias Grund: Lockheed Skunkworks ist eine Abteilung zur Entwicklung fortgeschrittener Kampfflugzeuge. Besonders stolz sind die Amerikaner auf das Spionageflugzeug U-2 und die Überschallflugzeuge „Blackbird“. Der Inhalt der zitierten Mail ist der Folgende: „[...] I am pleased to see that this idea has finally come to practice. I must say, however, that I am chagrined that it is attributed to Ikujiro Nonaka and Hirotaka Takeuchi. This process is the essence of the methodology initiated and promulgated by Kelly Johnson at Lockheed Air Development Center [aka - The Skunk Works] back in the 1950's. As the head of Technology and Engineering for a semiconductor capital equipment manufacturer, I was applying this methodology to managing rapid-response, business critical projects in the mid-1980's. Thank you for your time, Jack G. Ellis, CEO Invictus Integrated Logistics, LLC

immer schwieriger, überhaupt noch Projekte zu finden, die nicht für sich reklamieren, Scrum zu nutzen. Abgesehen von den eben formulierten Zweifeln an der konsequenten Umsetzung agiler Methoden: Ist die Etablierung von Scrum in Unternehmen einfacher geworden?

■ Die Bereitschaft ist enorm gewachsen – die Schwierigkeiten sind aber nicht kleiner geworden. Ich habe Senior Manager von Produkthäusern getroffen, die daran interessiert sind, Scrum einzuführen. Wir diskutieren deren Probleme und dazu gehörten normalerweise:

- Zeitpläne für Releases, die aus dem Ruder geraten.
- Länger andauernde Stabilisierung der Systeme vor einem neuen Release.
- Schwierigkeiten, sich auf Änderungswünsche zwischen den Release-Zyklen einzulassen.
- Abnehmende Qualität.
- Unzufriedene Kunden.
- Erhöhter Planungsbedarf vor Release-Starts.
- Schlechte Stimmung unter den Entwicklern.

Ich frage diese Leute, wie sie erwarten, dass Scrum ihnen hilft. Welche Probleme soll Scrum lösen und wie? Ich bekomme dann zur Antwort, dass sie mich genau deswegen an Bord genommen haben, damit ich ihnen zeige, wie Scrum diese Probleme lösen kann. Aber Scrum ist kostenfrei. Wenn Scrum alle diese Probleme lösen würde, würde ich Millionen von Dollars in Rechnung stellen, weil ich dann zaubern könnte.

Ich diskutiere mit ihnen, dass Scrum nur ein Framework ist, innerhalb dessen sich ihre Produktentwicklung abspielt. Scrum ist ein Werkzeug, das all die Ursachen der genannten Probleme ans Licht bringt oder transparent macht. Jedoch ist die Behebung dieser Probleme und die Durchsetzung von Veränderungen ihre eigene Aufgabe. Ohne grundlegende Veränderungen auf der Ebene der Berufsbilder, der Personen und der Abläufe kommt man diesen Missständen nicht bei. Schlimmer noch: Es gibt keine einfachen Lösungen. Es gibt Muster, die die Richtung anzeigen, aber jede Lösung ist einzigartig und hängt von der jeweiligen Organisation und ihrer Kultur ab.

■ Das entspricht auch unseren Erfahrungen. Lassen Sie uns über Defizite in der praktischen Anwendung von Scrum sprechen. Die haben ihre Ursachen teilweise in einer mangelnden Konsequenz des Managements – nach unserer Erfahrung aber auch in Qualifikationsdefiziten bei der Entwicklung. Im letzten Jahr entstand eine Diskussion über schlaffes Scrum (Flaccid Scrum³⁾. Was ist passiert?

■ Als Jeff und ich Scrum einführten, hatten wir erwartet, dass die von Scrum geschaffene Transparenz zusammen mit der empirische Steuerung eine Befreiung für die Entwickler einleiten würden, ihnen die Freiheit zu professionellem Arbeiten geben würde, die Freiheit, das Richtige zu tun und großartige Qualitätsprodukte zu entwickeln.

Wir dachten, dass unsere Kunden begeistert sein würden von den Techniken des iterativen und inkrementellen Entwickelns in kurzen Zyklen, von der Optimierung des betriebswirtschaftlichen Wertes und der Reduzierung von Abfallproduktion. Zu unserem großen Erstaunen waren die meisten dieser Leute in alten Praktiken gefangen. Ja, diese alten Praktiken schienen eine Art Muskelgedächtnis ausgebildet zu haben. Tief in ihren Köpfen steckt drin, dass sie mit dieser alten Praxis – obwohl fehlerhaft und unzulänglich – durch einige Bredouillen laviert sind. Konnten Sie auf die neuen empirischen Techniken in Scrum auf die gleiche Weise zählen?

Eine weit verbreitete Praxis in der Softwareentwicklung ist es, Abstriche an der Qualität zu machen. Dein Chef, dein Kunde, dein Zeitplan verlangt mehr. Du verminderst die Qualität dessen was du lieferst – nicht transparent, sondern im Verborgenen – und du kannst mehr liefern. Du brauchst keine ausgefeilten Werkzeuge und Techniken, du verstehst die Technologie, Du programmierst und testest da, wo die Fehlerwahrscheinlichkeit am höchsten ist. Wenn die Qualitätskontrolle oder die Kunden Fehler finden, reparierst du sie schnell – der Rest bleibt verborgen.

³⁾ Anmerkung von Matthias Grund: Martin Fowler hat mit seinem Blog-Eintrag „Flaccid Scrum“ vom 29. Januar 2010 (<http://martinfowler.com/bliki/FlaccidScrum.html>) die Diskussion eröffnet. Ken Schwaber hat sich im Namen der Scrum Alliance in die Diskussion eingeschaltet (www.scrumalliance.org/resource_download/745).



Die Konsequenz daraus, Geschwindigkeit über Qualität zu setzen, war eine fehlgeleitete Berufsauffassung. Es wurde üblich, sich nicht darum zu scheren, neue Arbeitstechniken zu erlernen, mit denen die Produkte hätten verbessert werden können. Viele der Entwickler, die in Scrum Teams arbeiten sollten, wussten nicht, wie man testgetrieben entwickelt – auf Feature- und auf Unit-Ebene. Sie wussten nicht, wie man automatisierte Testwerkzeuge nutzt, wie man automatisch Schritt für Schritt die Tests vertieft, wie Code und Design zu refaktorisieren sind, wie ein Produkt kontinuierlich gebaut und getestet wird oder wie Schwächen und falsche Annahmen gefunden werden können. Sie waren nicht einmal in der Lage, Kodier- und Review-Standards zu erfüllen. Sie kümmerten sich nie darum, diese Techniken zu lernen, weil sie sowieso nicht die Zeit gehabt hätten, sie anzuwenden. „Runter-Programmieren und Ausliefern“ war die Devise.

Dazu kommt: Aus der Wasserfall-Zeit haben wir isolierte funktionale Wissens-Silos übernommen, in denen jeder ein Experte in irgendetwas ist. Das Management bindet diese Experten zu einem Skill-Bündel, um ein Produkt zu produzieren. Die Leute mit einer bestimmten Fähigkeit waren oft überfordert, mit Leuten mit anderen Fähigkeiten zu kommunizieren und zusammenarbeiten. Wie konnten Sie irgendetwas erzeugen, das einen wirtschaftlichen Wert hatte? Es gab sogar ein Kastensystem, in dem Architekten auf Entwickler herabgeschaut haben, die auf die Qualitätssicherung herabgeschaut haben, die wiederum auf die Handbuch-Schreiber herabgeschaut haben. Dabei überlappten sich die Fähigkeiten bei der konkreten Entwicklungsarbeit beträchtlich. Diese Überschneidungen können aber nur genutzt werden, wenn die Individuen einander wertschätzen.

Wir machten einen Test und führten mit solchen funktionalen Silos, die über Jahre in klassischer Weise zusammengearbeitet hatten, einen kurzen Entwicklungszyklus durch. Wir wollten wissen, welche Ergebnisse in einem Zyklus von einem Monat erreicht werden konnten. Das Experiment zeigte uns: Es funktioniert nicht. Die Mehrzahl der Kollegen hatte keine Vorstellung, wie sie in der Iteration zusammenarbeiten sollten.

Trotzdem hatte ich zunächst ein Problem damit, dass Martin Fowler diese Phäno-

mene „Flaccid Scrum“ nennt. Denn Scrum funktionierte gut. Scrum brachte Schwächen an den Tag: in den professionellen Fähigkeiten, im Einsatz von Werkzeugen, im Einsatz des Prozesses bei vielen unserer derzeitigen Entwickler. Und damit stellte sich die Frage: „Wie bringen wir das in Ordnung?“

■ Wir haben die Erfahrung gemacht, dass viele Kollegen Scrum als ein Allheilmittel betrachten, mit dem sich alle Probleme von selbst lösen. In Wirklichkeit aber sind wir noch lange nicht am Ziel. Und auch Sie treibt ja eine gewisse Unzufriedenheit mit dem Zustand von Scrum um. Sie haben im letzten Jahr scrum.org gegründet. Jeff Sutherland hat sich Ihnen angeschlossen. Was ist die Mission dieses neuen gemeinsamen Unterfangens, 20 Jahre später? Was ist die Aufgabe von scrum.org?

■ Ich gründete 2009 scrum.org, um meine Vision zu verfolgen, die professionelle Praxis der Softwareentwicklung zu verbessern: Unsere Kunden sollen uns gerne für sie arbeiten sehen, wir sollen unsere Arbeit wieder mögen und unsere Kinder sollen nicht zögern, in unsere Fußspuren zu treten und sich für unserem Beruf zu entscheiden. Die Scrum Alliance wurde zu dieser Zeit sehr stark darauf ausgerichtet, Umsatz zu generieren, was oft im Konflikt mit meinen Zielen stand. In scrum.org kann ich diese Ziele konsequent verfolgen – gemeinsam mit wohl gewählten Mitstreitern.

■ Mit scrum.org haben Sie verschiedene Programme gestartet. Können Sie deren Ziele beschreiben?

■ Ich habe ein Remake des bekannten CSM-Programms (*Certified Scrum Master*), das ich vor über zehn Jahren entwickelt hatte, gestartet. Dieser Neustart benutzt Kursmaterial, das erneuert wurde und das darauf fokussiert, wie Scrum angewendet wird (ein Programm auf mittlerem Level). Wir kontrollieren und unterstützen die Trainer dieses Programms. Die Teilnehmer durchlaufen Prüfungen, die von der IT-Industrie anerkannt werden, und werden bei Erfolg zertifiziert. Das neue Programm heißt „Professional Scrum Master“. Es basiert, kurz gesagt, auf dem gesammelten Wissen über Scrum und dem Kursmaterial

„Scrum-In-Depth“. Umfangreiche Information stellen wir auf www.scrum.org zur Verfügung.

Ich habe auch das „Professional Scrum Developer“ Programm gestartet. Innerhalb dessen trainieren wir Entwickler in einem Scrum-Team. Das Training umfasst die Zusammenarbeit in einem sich selbst organisierenden, cross-funktionalen Team mit dem Ziel, ein fertiges⁴⁾ Inkrement innerhalb eines Sprints zu liefern. Wir arbeiten auf einem definierten Technologie-Stack (.NET, Java oder Java Eclipse) mit modernen Software-Engineering-Praktiken. Wir haben das Programm im Dezember angekündigt und die ersten Trainingseinheiten werden im April 2010 angeboten. Das ist ein Zertifizierungsprogramm, d. h. zertifiziert wird, wer eine Prüfung besteht. Das Ziel des PSD-Programms ist die von mir angesprochenen Mission von scrum.org, also die professionelle Praxis der Softwareentwicklung zu verbessern.

■ Die Scrum Alliance hat kürzlich das Programm „Certified Scrum Developer“ angekündigt. Dort klingt vieles ähnlich. Wie unterscheidet sich Ihr Programm von dem Programm der Scrum Alliance?

■ Das Programm der Alliance setzt voraus, dass die Teilnehmer entweder eine Scrum-Master-Zertifizierung besitzen oder einen eintägigen Kurs bei einem Scrum-Trainer besuchen. Danach wählt man aus einer Vielzahl von Kursen aus, die von Drittanbietern stammen und von der Scrum Alliance anerkannt wurden. Etwas polemisch formuliert: Für die erfolgreiche Auswahl wird der Teilnehmer dann zertifiziert. Bei scrum.org gibt es genau einen Kurs. Die Materialien sind standardisiert, werden regelmäßig überprüft und auf den neuesten Stand gebracht.

Ein weiterer Unterschied ist, dass scrum.org nur eine Zertifizierung erteilt, wenn die erworbenen Fähigkeiten in einer Prüfung nachgewiesen werden. Außerdem lässt scrum.org seine Kurse ausschließlich von Trainern durchführen, die beständig

⁴⁾ Anmerkung Matthias Grund: Ken Schwaber spricht hier von „done“. Damit ist im Scrum-Kontext gemeint: fertig entwickelt, fertig getestet, fertig dokumentiert – also im Prinzip auslieferbar. Es gibt nur „fertig“ oder „nicht fertig“.

überprüft werden und die in der Industrie ein entsprechendes Ansehen haben. Der Fokus des scrum.org-Programms liegt auf der Befähigung, ein Softwareinkrement im Rahmen eines Scrum-Prozesses zu bauen. Das ist ein komplett neues Programm, das im Verlauf des letzten Jahres mit großer Sorgfalt entwickelt wurde. Das Ziel ist, die „Flaccity“, die Martin Fowler beschrieben hat, und die von vielen bestätigt wurde, zu beseitigen. Beim Programm der Scrum Alliance kann ich keinen Fokus erkennen. Ein Bündel von Kursen anzubieten und die Entwickler diese zu einem sinnvollen Ganzen zusammenstoppeln zu lassen – das gibt es schon heute, ohne klangvollen Namen. Dass dieser Ansatz nicht funktioniert, hat die Praxis bewiesen.

■ Wenn wir uns die Entwicklung von Scrum anschauen, können wir die Jahre

1990 bis 1992 als die Zeit des Entstehens, 1992 bis 2001 als die Zeit der Validierung und Verfeinerung und die Jahre danach als die Zeit der Etablierung als führende Projektsteuerungsmethode ansehen. Was kommt nun? Treten wir in eine neue Phase ein?

■ Unsere Reise hat gerade erst begonnen. Wir erleben den Übergang vom Industriezeitalter in eine kollaborative und emergente Gesellschaft, die aus ihren Fehlern lernt. Scrum ist nur ein Framework für diese Art zu arbeiten und meine Hoffnung ist, dass Scrum verschwindet und seine Techniken zur ganz normalen Alltagspraxis werden.

■ Sie haben viele unterschiedliche Firmen in ihrem Übergang von einer deterministischen zu einer, wie Sie sagen, empirischen Steuerung von Entwicklungsprojekten,

unterstützt. Ich vermute, Sie haben sowohl Erfolge als auch Misserfolge erlebt. Gibt es einen kritischen Erfolgsfaktor für den Erfolg einer Firma mit Scrum?

■ Die Voraussetzung ist das Wissen und Bewusstsein, dass Probleme existieren, und die Erkenntnis, dass diese Probleme mit den Methoden von Scrum angegangen und korrigiert werden können. Die *conditio sine qua non* ist harte Arbeit, Verpflichtung, Sturheit und der Wille zur Verbesserung. Nur Unternehmen, die beides beherzigen, werden sich langfristig im Wettbewerb behaupten können – die anderen werden verschwinden.

Ken, herzlichen Dank für das Gespräch

Das Gespräch führte Matthias Grund auf Englisch. ■