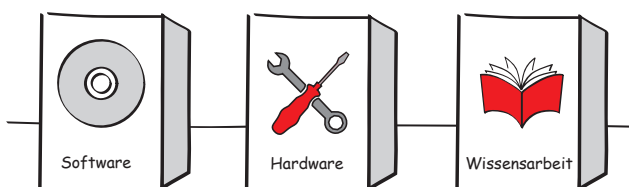


Scrum Guide Update: Klein, aber fein Die Neuerungen im Scrum Guide 2017

Im November 2017 haben Jeff Sutherland und Ken Schwaber die aktuelle Version des Scrum Guide vorgestellt. Im Mittelpunkt steht dabei die Feststellung, dass Scrum ebenso jenseits der Grenzen der IT gut und nützlich ist. Zudem wurden Aspekte von Scrum klarer beschrieben: die Rolle des Scrum Masters, das Daily Scrum und die jeweiligen Zeitvorgaben für die Scrum-Meetings. In diesem Artikel stellen wir die Änderungen vor und ordnen sie ein. Von Urs Reupke und Henning Wolf

Scrum jenseits der IT? Für viele ist das eine überraschende Feststellung. Heißt es nicht „Manifest für Agile Softwareentwicklung“, und hat nicht Scrum selbst dort seinen Ursprung? Ja und nochmal ja. Der neue Scrum Guide möchte dieses Missverständnis ausräumen und macht gleich am Anfang klar: Scrum funktioniert anderswo ebenfalls. Der Scrum Guide ist die zentrale Beschreibung von Scrum, die die beiden Scrum-Erfinder Jeff Sutherland und Ken Schwaber gemeinsam betreuen und regelmäßig aktualisieren. So geschehen zuletzt im November 2017 als sie die Beschreibung mit einer großen Klarstellung und vielen kleinen Verfeinerungen überarbeitet haben.



Scrum dient der Entwicklung komplexer Produkte

Die große Klarstellung ist also: Scrum eignet sich generell für die Entwicklung komplexer Produkte.

Um das zu belegen, führt der Scrum Guide im neuen Kapitel „Anwendungen von Scrum“ verschiedene Einsatzszenarien auf und geht aus der Vogelperspektive auf die besonderen Vorteile von Scrum und die Kerngedanken dahinter ein. Die Liste reicht von verschiedenen IT-Szenarien bis hin zu Arbeit in Schul- und Regierungsprojekten, sie deckt Software genauso ab wie Hardware und Wissensarbeit. Die Autoren betonen die Vorteile von Scrum im Umgang mit Komplexität, und stellen heraus, dass

Scrum skaliert funktioniert, wenn man nur die Teams und ihre Zusammenarbeit in den Mittelpunkt stellt. Im Wortlaut heißt es:

Scrum wurde ursprünglich dazu entwickelt, um Produkte zu managen und zu entwickeln. Seit den frühen 1990er Jahren wird Scrum weltweit ausgiebig genutzt zur:

1. *Erforschung und Identifizierung rentabler Märkte, Technologien und Produktfähigkeiten;*
2. *Entwicklung von Produkten und Verbesserungen;*
3. *Auslieferung von Produkten und Verbesserungen, auch vielfach pro Tag;*
4. *Entwicklung und Aufrechterhaltung von Cloud-Umgebungen (online, secure, on-demand) und anderer Produktumgebungen; sowie*
5. *Erhaltung und Erneuerung von Produkten.*

Scrum wird verwendet, um Software, Hardware, Embedded Software, Netzwerke von interagierenden Funktionen und autonome Fahrzeuge zu entwickeln. Scrum wird außerdem in Schulen, Regierungs- und Marketingprojekten genutzt, zur Verwaltung von Organisationen und der Entwicklung von fast allem, was wir in unserem täglichen Leben als Einzelpersonen und als Gesellschaften verwenden. Scrum bewährt sich täglich im Umgang mit Komplexität, da Technologie-, Markt- und Umweltkomplexitäten und deren Interaktionen rapide zugenommen haben. Scrum hat sich bei iterativem und inkrementellem Wissenstransfer als besonders effektiv erwiesen. Scrum wird heute häufig für Produkte, Dienstleistungen und das Management der übergeordneten Organisation verwendet. Der Kern von Scrum ist ein kleines Team von Menschen. Dieses Team ist flexibel und anpassungsfähig. Diese Stärken wirken weiter zwischen mehreren und vielen Teams und sogar Netzwerken von Teams, die die Arbeit und Ergebnisse von Tausenden von Menschen entwickeln, ausliefern, betreiben und erhalten. Sie arbeiten ►

über durchdachte Entwicklungsarchitekturen und Zielumgebungen für Auslieferungen zusammen. Wenn die Wörter „entwickeln“ und „Entwicklung“ im Scrum Guide verwendet werden, beziehen sie sich auf komplexe Arbeit, wie z. B. die oben beschriebenen Typen.

Das meint Henning

Es hat bisher niemand irgendwen davon abgehalten, Scrum außerhalb von IT zu verwenden. Die Klarstellung hilft trotzdem.

Schwaber und Sutherland als Väter von Scrum betonen, dass die Großväter von Scrum Nonaka und Takeuchi sind, die in ihrem Paper „The New New Product Development Game“ von 1986 ihre Untersuchungen aus Firmen beschrieben haben, die Produkte herstellen (und durch die Bank weg keine Softwareprodukte).

Schwaber hat bei der Vorstellung der Änderungen schön gesagt, wie er sich den Zweck von Scrum vorstellt: „Jemand mit einer Vision, ein fähiges Team, der Wunsch etwas Nützliches herzustellen. Dann schnell herausfinden und lernen!“

Das meint Urs

Produktentwicklung statt Softwareentwicklung – das könnte ebenfalls für das Agile Manifest gelten. Komplexe Produkte können in einer schnelllebigen Umgebung nur mit kurzen Feedbackzyklen und häufiger Auslieferung mit hoher Erfolgswahrscheinlichkeit entwickelt werden. Diesen Anspruch Scrums auf Allgemeingültigkeit endlich klarzustellen, ist lange überfällig.

Dass mit diesem Kapitel endlich Skalierung in den Scrum Guide Einzug findet, begrüße ich. Scrum in großen Projekten ist mittlerweile für zu viele Firmen zu wichtig, um es als unbeantwortete Frage oder als nachgeschobenen Gedanken abzuhandeln.

Eine Fußnote mit Verweis auf erprobte und erfolgreiche Skalierungssysteme wie LeSS oder Nexus fehlt hier zur Abrundung.

Trotz dieser guten Aspekte macht das Kapitel auf mich eher den Eindruck eines sehr öffentlichen Blogposts, denn es trägt durch die geänderte Flughöhe nichts zum Anspruch des Scrum Guide bei, die Regel für den Umgang mit Scrum festzulegen.

Der Scrum Master vertritt Scrum entsprechend des Scrum Guides

Die zweite Änderung bezieht sich auf den Scrum Master und seine Beziehung zum Scrum Guide. Es wird klargestellt, dass sein Auftrag ist, Scrum entsprechend des Scrum Guides zu fördern.

Der Scrum Guide wörtlich:

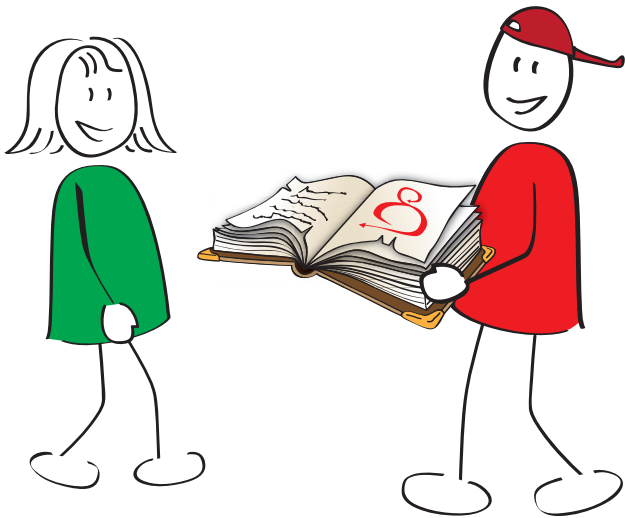
Der Scrum Master ist dafür verantwortlich, Scrum entsprechend des Scrum Guides zu fördern und zu unterstützen. Scrum Master tun dies, indem sie allen Beteiligten helfen, die Scrum Theorie, Praktiken, Regeln und Werte zu verstehen. Der Scrum Master ist ein „Servant Leader“ für das Scrum-Team.

Der Scrum Master hilft denjenigen, die kein Teil des Scrum-Teams sind, zu verstehen, welche ihrer Interaktionen mit dem Team sich hilfreich auswirken und welche nicht. Der Scrum Master hilft dabei, die Zusammenarbeit so zu optimieren, dass der durch das Scrum-Team generierte Wert maximiert wird.

Dabei sind alle Aspekte von Scrum und alle Teilnehmer gleichermaßen wichtig, es geht also keinesfalls nur um die Praktiken oder nur um die Arbeit mit dem Entwicklungsteam, sondern ebenso um die Werte und die Beteiligten außerhalb des Scrumteams.

Das meint Henning

Ich war zunächst etwas erschrocken ob des Anspruchs der Konformität zum Scrum Guide. Andererseits erlebe ich in Praxisprojekten viele unnötige und störende Anpassungen und habe das Gefühl, dass Scrum, so wie



es gemeint ist, gar keine Chance gegeben wird. Bei der Vorstellung des Scrum Guide 2017 gab es noch eine Ergänzung von Schwaber und Sutherland: Das alles macht der Scrum Master im Rahmen der Möglichkeiten, die die Kultur der Organisation vorgeben, und im Rahmen seiner Organisationsfähigkeiten, politischen Fähigkeiten und seiner Geduld. Hier finde ich es eher schade, dass die politischen Fähigkeiten ins Spiel gebracht werden. Aber das mag naiv sein, dass ich es immer gut fand, dass Agilität mit Transparenz und Offenheit eher antipolitisch ist.

Das meint Urs

Diese Änderung stellt einen hohen Anspruch an den Scrum Master, was die Konformität mit den Regeln des Scrum Guide betrifft. Ich hätte dort lieber gelesen, dass er Scrum „im Sinne“ des Scrum Guide vertreten soll – mehr Geist, weniger Regeln. Das aber ist die Perspektive eines erfahrenen Anwenders. Für Anfänger andererseits, eine der größten Zielgruppen des Scrum Guide, ist so eine Einschränkung wichtig, denn so ist klar: Ein Scrum Master bist du nur, wenn du Scrum so vertrittst, wie es hier beschrieben ist, und nicht, wenn du dir mit deinen Kollegen zusammen eine eigene Variante ausgedacht hast. Mit mehr Erfahrung erkennt man dann, wo Abweichungen sinnvoll sind, wo nicht, und worauf man dabei achten muss. Der Hinweis, dass der Scrum Master mit allen Teilnehmern arbeitet, also insbesondere mit den Stakeholdern des Scrum Teams, ist mir willkommen, denn diese Perspektive vernachlässigen

gerade frische Scrum Master.
Insgesamt begrüße ich die Klarstellung also.

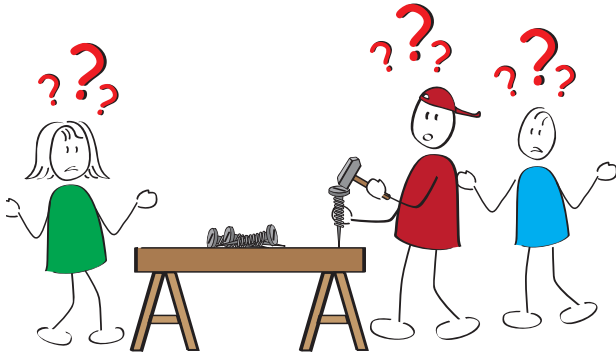
Mangelndes Produktverständnis ist ein Impediment

Die dritte Verfeinerung bezieht sich auf die Unterstützung des Product Owners durch den Scrum Master. Sutherland und Schwaber stellen klar, dass der Scrum Master helfen soll, das Verständnis für das Produkt bei allen Mitgliedern des Scrum Teams zu fördern – also ebenfalls sein eigenes. Im Wortlaut (ergänzt in der Aufzählung „Der Dienst des Scrum Masters für den Product Owner“):

Sicherstellen, dass Ziele, Umfang und Produktdomäne von allen im Scrum Team so gut wie möglich verstanden werden; das bedeutet natürlich nicht, dass er selbst dafür sorgen muss, sondern weist nur darauf hin, dass mangelndes Verständnis ein Impediment sein kann.

Das meint Urs

Ich habe es noch vor Augen: Nach einem ausführlichen Gespräch über agile Entwicklungsmethoden zögert der Abteilungsleiter des Fachbereichs für einen Moment sagt dann, leicht ungläubig: „Das heißt ja, Herr Reupke, die Entwickler müssen unsere Arbeit verstehen!“ So ist es. Eine Gruppe von Programmierern allein ist kein Entwicklungsteam im Sinne von Scrum. Damit das Entwicklungsteam gemeinsam mit dem Product Owner und dem Scrum Master ein Produkt schafft, das die Benutzer begeistert, müssen sie verstehen, was die Benutzer davon brauchen und in deren Sinne mitdenken können – und das setzt ein Mindestmaß an Domänen- und Produktwissen voraus. Dieser Gedanke gilt, das macht die Ergänzung klar, gleichermaßen für den Scrum Master. „Ich bin nur für den Prozess da“ habe ich von mehr als einem Scrum Master gehört. Das stimmt zwar, ist aber zu kurz gedacht, denn die Suche nach dem idealen Entwicklungsprozess geht Hand in Hand mit der Weiterentwicklung



des Produkts und den Wünschen der Stakeholder. Produktverständnis ist also unerlässlich. Das spricht einmal mehr für einen festangestellten Scrum Master, denn nur für den gibt es großen Anreiz, seine Umgebung inklusive der Produktdomäne wirklich kennenzulernen.

Das meint Henning

Ich schließe mich Urs' Meinung an.

Im Daily Scrum wird der Fortschritt Richtung Sprint-Ziel sichergestellt

Im Abschnitt über das Daily Scrum betonen die Autoren den Zweck des Meetings. Es geht darum, Zusammenarbeit und Leistung in Hinblick auf die Erreichung des Sprint-Ziels zu optimieren. Wie das Meeting aufgebaut ist, ist zweitrangig, solange dieses Ziel im Mittelpunkt steht.

Wörtlich heißt es zu Beginn:

Das Daily Scrum ist eine Timebox von 15 Minuten für das Entwicklungsteam. Das Daily Scrum findet an jedem Tag des Sprints statt. Das Entwicklungsteam plant dabei die Arbeit für die nächsten 24 Stunden. Es überprüft die Arbeitsergebnisse seit

dem letzten Daily Scrum und prognostiziert die im Sprint bevorstehende Arbeit, um die Zusammenarbeit und Leistung des Teams zu optimieren. Um die Komplexität zu reduzieren, wird das Daily Scrum an jedem Tag zur gleichen Uhrzeit am gleichen Ort abgehalten.

Und im weiteren Verlauf:

Die Struktur des Ereignisses wird vom Entwicklungsteam festgelegt und kann auf unterschiedliche Weise durchgeführt werden, sofern die Erreichung des Sprint-Ziels im Fokus steht. Einige Entwicklungsteams verwenden Fragen, andere konzentrieren sich eher auf Diskussionen.

Hier ist ein Beispiel, was verwendet werden könnte:

- Was habe ich gestern getan, das dem Entwicklungsteam geholfen hat, das Sprint-Ziel zu erreichen?
- Was werde ich heute erledigen, um dem Entwicklungsteam beim Erreichen des Sprint-Ziels zu helfen?
- Sehe ich irgendein Hindernis, das mich oder das Entwicklungsteam daran hindert, das Sprint-Ziel zu erreichen?

Das meint Henning

Schon in einer der früheren Anpassungen des Scrum Guide verfolgten Schwaber und Sutherland dieses Ziel, indem sie die drei Fragen für das Meeting umformuliert haben:

- Was hast du seit dem letzten Daily Scrum erledigt, was uns dem Sprint-Ziel näher gebracht hat?
 - Was wirst du bis zum nächsten Daily Scrum tun, was uns dem Sprint-Ziel näher bringt?
 - Welche Probleme siehst du bei dir oder anderen, die uns behindern könnten, das Sprint-Ziel zu erreichen?
- Aber die reine Umstellung der Fragen hat noch nicht den erwünschten Effekt erzielt. Deswegen also die Klarstellung: Das Daily Scrum dient dem inspect&adapt für das Entwicklungsteam, um zu überprüfen und ggf. Anpassung vorzunehmen, damit das Sprint-Ziel erreicht wird.



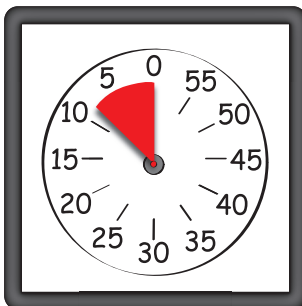
Das meint Urs

Kein Meeting in Scrum ist so schwer gut hinzukriegen wie das Daily Scrum. Zu leicht ist es, in altgewohnte Berichtsform zurückzufallen, zu oft findet sich ein Teammitglied, dass die ganze Veranstaltung für Zeitverschwendung hält. Um so wichtiger, dass Ken und Jeff endlich klarstellen, wozu das Treffen da ist: Zusammenarbeit und Leistung sollen optimiert werden, und zwar nicht auf irgendwas, sondern auf das Sprint-Ziel. Damit wird noch klarer: Ohne Sprint-Ziel geht es nicht. Der Unterschied wird sich nicht nur hier bemerkbar machen.

Das meint Urs

„Kürzer geht immer“ – eine sinnvolle Klarstellung in Zeiten outlookgetriebener Tagesplanung. Der zweite Teil der Botschaft ist „länger keinesfalls“ und richtet sich an die Teams, bei denen die Meetings regelmäßig länger dauern als vorgesehen. Hier steckt Verbesserungspotenzial.

Zeitangaben sind Obergrenzen

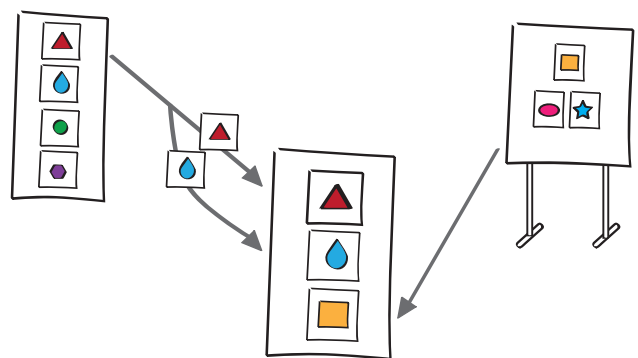


Bei den Timeboxes haben die Autoren klargestellt, dass die Länge nicht voll ausgeschöpft werden muss, indem sie an allen entsprechenden Stellen das Wort „maximal“ ergänzt haben. Die Ausnahme zu dieser Regel ist der Sprint, der

nach wie vor nur in Ausnahmefällen und nur durch den Produkt Owner verkürzt, also abgebrochen werden darf.

Das Sprint-Backlog enthält eine Maßnahme aus der Sprint-Retrospektive

Um die Bedeutung kontinuierlicher Prozessverbesserung für die erfolgreiche Anwendung von Scrum zu unterstreichen, haben die Autoren klargestellt, dass Verbesserungsmaßnahmen Teil des Sprint-Backlogs sind.



Das meint Henning

In gewisser Weise ist es lustig, was alles so unklar sein kann. Da passen die Leute wild alles mögliche an Scrum an, halten sich aber stur an Vorgaben bzgl. der Zeiten, selbst, wenn sie früher fertig sind? Unfassbar, aber vermutlich die Ausnahme. Schön, dass es jetzt explizit ist.

Im Wortlaut (über das Sprint-Backlog):

Um kontinuierliche Verbesserungen zu gewährleisten, umfasst es mindestens eine wichtige Prozessverbesserung, die in der vorherigen Retrospektive identifiziert wurde.

Das meint Henning

Schwaber und Sutherland haben diese Ergänzung vorgenommen, weil sie den Eindruck hatten, dass

Teams zuweilen glauben, sie hätten nichts mehr zu verbessern. Das ist unwahrscheinlich und beruht auf einem Missverständnis, man könnte jemals am Ende der Verbesserungsfahnenstange ankommen. Jeff wollte nach eigener Aussage übrigens genau eine Maßnahme aus jeder Retro für das Sprint Backlog vorschreiben, aber das war Ken zu eng, deswegen ist es jetzt mindestens eine.

Das meint Urs

Eine lang überfällige Verfeinerung! Endlich haben die Maßnahmen aus der Retrospektive einen definierten Platz in Scrum. Als Resultat erwarte ich klarere Kommunikation in der Sprint-Planung, mehr Übersicht und klarere Nachvollziehbarkeit.

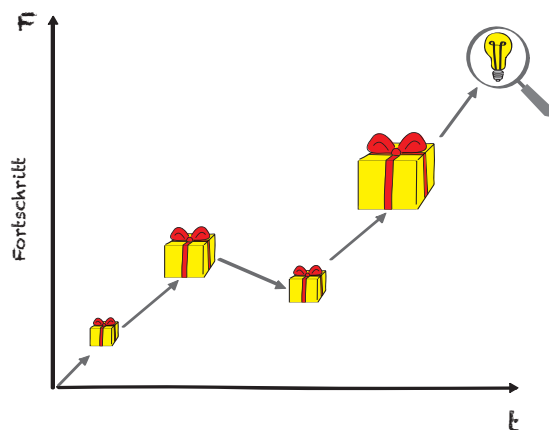
Die Änderung sollte gerade unerfahrenen Teams helfen, ihre Retrospektive nicht lediglich mit Willenserklärungen zu beenden, sondern mit ausführbaren und überprüfbaren Maßnahmen.

Das Inkrement dient der Empirie und fördert die Produktvision

Zum Schluss haben die Autoren noch die Bedeutung des Inkrements als Ergebnis jedes Sprints geschärft.

Wörtlich heißt es im Scrum Guide:

Ein Inkrement ist ein Gegenstand inspizierbarer, fertiger [„Done“] Arbeit, der die Empirie am Ende des Sprints unterstützt. Das Inkrement ist ein Schritt in Richtung einer Vision oder eines Ziels. Das Inkrement muss im Sinne der Definition of Done fertig sein und dem Team dabei helfen, spätestens am Ende des Sprints zu lernen und gleichzeitig der Produktvision näher zu kommen. Damit findet die Produktvision zum ersten Mal Einzug in den Scrum Guide. Jedes Inkrement unterstützt das Scrum-Team so dabei, ein erfolgreiches Produkt zu schaffen.



Das meint Henning

Bei der Vorstellung der Änderungen haben Ken Schwaber und Jeff Sutherland klargestellt, dass bereits während des Sprints geliefert werden darf, wenn z. B. Continuous Deployment praktiziert wird. In diesem Zusammenhang haben sie darauf verwiesen, dass das Review eben kein Kontroll- oder Abnahme-Meeting ist oder sowas wie ein Quality-Gate, ob das shippable increment ausgeliefert wird oder nicht.

Es geht im Sprint-Review darum, gemeinsam mit Stakeholdern darüber zu kooperieren, wie wir das bestmögliche Produkt zur Lösung unseres Problems bauen.

Das meint Urs

Fertig muss das Inkrement sein, das ist ein Punkt, den zu viele Teams vernachlässigen. Nur so kriegt der Kunde eine verlässliche Demonstration und gewinnt Vertrauen in die Ergebnisse; nur so kriegt das Team Feedback, auf das es sich verlassen kann; kurzum: Nur so können sie die Vorteile voll ausnutzen, die ihnen Scrum bietet. Vertrauen in die Ergebnisse soll der Kunde gewinnen, also in das Produkt selbst und den Prozess, durch den es entsteht, nicht Vertrauen in einzelne Team-Mitglieder!

Erst, wenn er dieses Vertrauen hat, kann er mit Fehlern

und Missständen zuversichtlich und konstruktiv umgehen und spricht mit dem Scrum-Team als Einheit, anstatt Einzelne um Heldentaten zu bitten oder hinter vorgehaltener Hand Kritik zu üben.

Anders gesagt: Ein fertiges Inkrement und das Vertrauen, das dadurch entsteht, verringern die Politik in der Umgebung und eliminieren damit langfristig eine Quelle von Verschwendung.

Fazit

Die Änderungen in diesem Jahr sind nicht spektakulär, aber sinnvolle Klarstellungen. Ein Produktentwicklungsframework wie Scrum steht immer in einem Spannungsfeld: Zu genau beschrieben ist es für weniger Kontexte geeignet; zu ungenau beschrieben bietet es zu wenig Orientierung. Aus unserer Sicht gelingt Schwaber und Sutherland dieser Spagat gut, und dies trägt dazu bei, dass Scrum der bisher am weitesten verbreitete agile Ansatz ist.

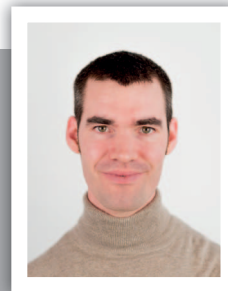
Uns hat besonders ein Spruch von Ken Schwaber bei der Vorstellung gefallen: „Scrum ist nichts für Leute, die nicht denken können!“

Wir würden ergänzen: Scrum ist nichts für Leute, die nicht denken wollen! ■

Links

Deutsche Übersetzung des Scrum Guide 2017 [PDF]:
<http://scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-German.pdf>

Übersicht über die Änderungen auf unserer Website:
<https://www.it-agile.de/wissen/einstieg-und-ueberblick/scrum/scrum-guide-2017>



URS
REUPKE

Urs Reupke ist Management-Berater bei it-agile. In 15 Jahren Erfahrung mit Agilität hat er gelernt, dass Miteinander, Prozess und Technik Hand in Hand gehen müssen, damit das Ergebnis stimmt. Heute gibt er diese Erfahrung weiter und hilft Unternehmen aller Größen, elegante Lösungen für Ihre strukturellen Probleme zu finden.



HENNING
WOLF

Henning Wolf arbeitet bei it-agile als Trainer und Coach für agile Softwareentwicklung sowie als Geschäftsführer. Er verfügt über praktische Erfahrung mit agilen Methoden seit 1999, hat mehrere hundert Personen geschult und ist Buchautor. Neben Scrum gelten seine besonderen Interessen Lean-Startup-Ideen und der Frage, wie man Organisationen verändern muss, um den höchsten Nutzen aus agiler Softwareentwicklung ziehen zu können.

Die agile review bequem im Abo!

ab 20 € pro Jahr (3 Ausgaben)



Hier Abo bestellen: www.agilereview.de