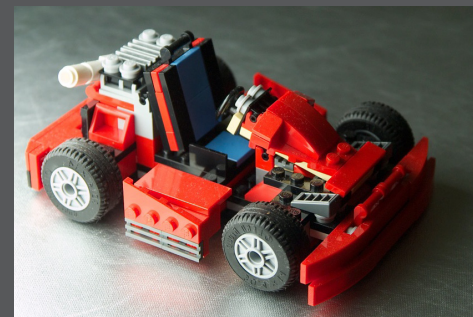
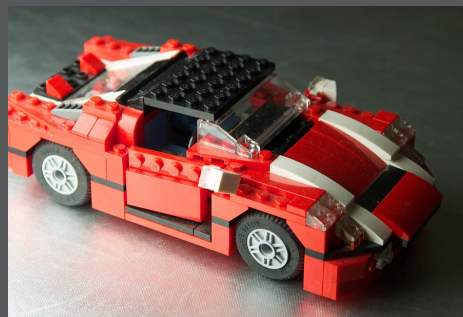


agile review

Das Kundenmagazin von it-agile

*Veränderung
willkommen!*





editorial

2011 war eines der turbulentesten Jahre in unserer Geschichte: Erstmals mussten wir in größerem Umfang Kundenanfragen ablehnen, weil wir ausgebucht waren. Das fiel uns schwer, und oft mussten wir der Versuchung widerstehen, noch eine Schulung mehr „irgendwie mit reinzuquetschen“. Aber natürlich gilt das Lean-Prinzip „Limit work to capacity“, auch für uns selbst, wenn wir eine nachhaltige Arbeitsbelastung für unsere Mitarbeiter gewährleisten und die Qualität unserer Dienstleistungen hoch halten möchten. Wir bitten alle betroffenen Kunden um Verständnis.

Das Interesse an Scrum nimmt weiter zu, und wir freuen uns, dass wir als einer der ersten Anbieter Deutschlands nun auch die Ausbildung zum Certified Scrum Developer auf Java und .NET-Plattformen anbieten können. Außerdem haben wir mit Henning Wolf seit Oktober einen zweiten zertifizierten Scrum-Trainer (CST) und können in Zukunft mehr Zertifizierungsschulungen und flexiblere Termine anbieten.

Immer wieder bekommen wir auch Anfragen von Kunden, die bereits vor geraumer Zeit Scrum eingeführt haben, ohne dass sich die erhofften Vorteile eingestellt haben. Wir finden dann oft dysfunktionale Adaptionen von Scrum vor, die häufig auf fehlende oder - noch schlimmer - schlechte Beratung zurückzuführen sind. Nicht immer ist das so extrem wie bei dem Berater, der den Kunden empfohlen hatte, sich Barfuß auf die Storykarten zu stellen, „um besser in Kontakt mit der Story zu kommen“. Wie auch harmlos scheinende Anpassungen an Scrum zu Problemen führen können, beschreibt Stefan Roock in seinem Beitrag „ScrumBut“ ab Seite 44. Und ab Seite 22 berichtet er von seinen Erfahrungen bei der Umstellung großer Organisationen auf Scrum.

Doch die agile Welt dreht sich nicht nur um Scrum. Im Oktober haben wir die erste Konferenz „Lean Kanban Central Europe“ in München ausgerichtet und konnten dafür viele internationale Vordenker gewinnen: von David Anderson und Karl Scotland über John Seddon bis hin zu Kent Beck - was uns gleich die Möglichkeit gab, unseren schon traditionellen zweitägigen Hackathon mit Kent Beck durchzuführen. Das gestiegene Interesse an Kanban zeigt sich auch in unserem Auftragsportfolio: Wir durften gleich eine ganze Reihe neuer Kunden auf ihrer „Reise in ein agiles Land“ begleiten, um Markus Andrezak von mobile.de zu zitieren. Immer wieder entstehen dabei auch Kombinationen von Kanban und Scrum, wie sie Arne Roock ab Seite 12 erläutert.

Schon immer bauen erfahrene Agilisten auf Überlegungen aus der Theorie komplexer Systeme. In den letzten Jahren hat sich hier mit der „Human Systems Dynamic“ ein neuer Zweig entwickelt, der besonders geeignet ist, beim Management organisatorischen Wandels zu helfen. Glenda Eoyang, die dieses Thema entscheidend vorangebracht hat, schildert den Ansatz in ihrem Gastbeitrag ab Seite 61.

Schließlich ist das Wachstum der vergangenen Jahre an uns nicht ganz spurlos vorüber gegangen: Unsere Räume plätzen aus allen Nähten. Seit Oktober haben wir nun in München unser neues Loft in der Rosenheimer Straße bezogen, das uns noch Raum für Zuwachs lässt. Sie sind herzlich auf einen Kaffee bei uns eingeladen. Für 2012 stehen dann auch in Hamburg größere Büroräume auf der Agenda - und eine neue Kaffeemaschine. ■

Viel Spaß beim Lesen wünschen Ihnen

Henning Wolf, Geschäftsführer von it-agile, Hamburg

Jens Coldewey, Geschäftsführer von it-agile, München

INHALT

4

Agiles Testen

In klassischen Vorhaben werden Tester meist erst spät mit eingebunden. Markus Gärtner beschreibt, wie eine sinnvolle Zusammenarbeit von Entwicklern und Testern in agilen Vorhaben aussehen kann.

12

Kanban & Scrum

Kanban und Scrum stellen keine Alternativen dar, sondern lassen sich auf verschiedene Weisen kombinieren. Der Artikel stellt einige Möglichkeiten vor.

20

Scrum-Tandem

Das Scrum-Tandem stellt eine Möglichkeit dar, Scrum sicher im Unternehmen einzuführen.

23

Impressum

24

Enterprise Transition

Wenn ein ganzes Unternehmen auf Scrum umgestellt werden soll, liegt es nahe, diese Umstellung selbst auch mit Scrum zu organisieren.

28

Interview mit Kent Beck



34

To tool or not to tool

In diesem fiktiven Streitgespräch werden Argumente pro und contra elektronische Tools gegenübergestellt.



Viele Teams kennen die „Merge-Hölle“ aus eigener Erfahrung. Feature Flags stellen eine elegante Alternative dar.

Feature Flags 38

„Wir machen Scrum, aber...“ Hinter diesem Satz verbergen sich individuelle Anpassungen am Scrum-Framework. Stefan Rook beschreiben, warum diese mit Vorsicht zu genießen sind.

Scrum But(t) 44



Buchtipps 50

Widersprechen sich Agilität und langfristige Planung? Nein! Aber eine langfristige agile Planung sieht anders aus, als wir es aus klassischen Entwicklungsvorhaben gewohnt sind.

Langfristige Planung 52



Human System Dynamics 61

Schulungsbeschreibungen und Autoren 64



Agiles Testen – Ein Pfad zur Teamunterstützung

Tester werden in Agilen Projekten oft erst später eingebunden. Dabei stauen sich unter Umständen viele technische Schulden an, die nach mehreren Iterationen nur mit erhöhtem Mehraufwand zu korrigieren sind. Doch Tester können einen echten Mehrwert liefern, wenn es gelingt, sie bereits bei der Entwicklung der Software mit einzubinden. Mit Akzeptanztestgetriebener Entwicklung (ATDD) und Explorativem Testen (ET) stehen Teams zwei sich ergänzende Ansätze zur Verfügung, die ihnen langfristig dabei helfen, funktionierende Software mit einer hohen Qualität auszuliefern und dabei ihre Tester nutzbringend einzubinden.

Von Markus Gärtner

Scrum und Kanban - Muss ich mich für eins von beiden entscheiden?

Von Arne Roock

ist, die sog. „Low hanging Fruits“ von Scrum zu ernten (z.B. erhöhte Transparenz und bessere Kommunikation), aber es nicht gelingt, wirkliche organisatorische Änderungen durchzusetzen - vor allem das Zulassen und Fördern der Selbstorganisation und damit die Abkehr vom Command-and-Control-Management.

Was ist Kanban?

Ursprünglich war Kanban eine Methode zur Produktionssteuerung, die bei Toyota entwickelt wurde und in der Automobilherstellung längst zum Standard gehört. Wenn heute allerdings von Kanban im Zusammenhang mit der IT die Rede ist (oft auch „Software-Kanban“ genannt), dann ist damit eine Methode des „Evolutionären Change Managements“ gemeint, die David Anderson bei den Unternehmen Microsoft und Corbis entwickelt hat und die sich seit 2009 rapide um die ganze Welt verbreitet.

Nach dem Motto „Nur was wir sehen können, können wir auch verändern“ beruht Kanban auf der Visualisierung der Wertschöpfungskette und der umzusetzenden Aufgaben. Dafür wird ein „Kanban-Board“ erstellt, auf dem die verschiedenen Prozessschritte, die eine Aufgabe bis zur Fertigstellung durchläuft, als Spalten abgebildet sind. Als Board kann ein Whiteboard, eine Pinnwand oder auch ein elektronisches Tool dienen. Die Aufgaben werden als so genannte „Tickets“ dargestellt. Dies können z. B. Haftnotizen oder Karteikarten sein, auf denen die wichtigsten Informationen festgehalten sind.

Zum Kern von Kanban gehört die Limitierung der parallelen Arbeit („Work in Progress“, kurz WIP) und die Etablierung eines Pull-Systems: Es darf sich nur eine

begrenzte Anzahl von Aufgaben im System befinden. Und erst wenn eine alte Aufgabe fertig gestellt wurde, beginnen wir mit einer neuen Aufgabe.

Ein wesentlicher Bestandteil von Kanban, vermutlich der wichtigste, ist „Kaizen“: Es soll eine Kultur der kontinuierlichen Verbesserung etabliert werden, in der das Team permanent daran arbeitet, den Prozess so zu verbessern, dass immer schneller ein immer größerer Geschäftswert für den Kunden ausgeliefert wird (wobei auch interne Kunden wie Fachabteilungen, andere Teams usw. gemeint sein können).

Scrum und Kanban kombinieren

Bei der Frage nach dem Zusammenhang zwischen Scrum und Kanban kursieren allerlei Halbwahrheiten und arg vereinfachte Darstellungen, die häufig auf eine Gegenüberstellung der Unterschiede verkürzt werden und so ein „Entweder-Oder“ nahelegen. Tatsächlich haben Teams und Organisationen auf der ganzen Welt inzwischen gezeigt, dass sich Scrum und Kanban auf vielfältige Weise sinnvoll miteinander kombinieren lassen.

Gemeinsamkeiten von Scrum und Kanban

Der Grund dafür liegt darin, dass Scrum und Kanban durch eine gemeinsame Denkweise geprägt sind, die viele elementare Prinzipien aus dem Lean Thinking und der agilen Softwareentwicklung enthält. So ist für beide Methoden grundlegend, dass die Teams fokussiert daran arbeiten, schnell und häufig Produkte mit einem hohen Geschäftswert für den Kunden auszuliefern. Um dies zu erreichen, ist es notwendig, „mit leichtem Gepäck“ zu rei- ▶



lean kanban

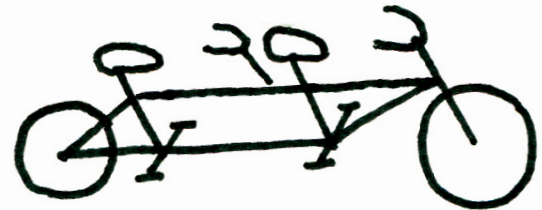
CENTRAL EUROPE

Wien, Oktober 2012

*The leading event on
Lean & Kanban
in Central Europe*

www.lean-kanban-conference.de





Teams mit Scrum infizieren – auf dem Tandem geht's leichter

„Scrum is Simple. Doing Scrum is Hard“ Jim York

von Stefan Rook, Sebastian Sanitz und Henning Wolf

Scrum-Einführung „by the book“

Der Standardansatz für die Scrum-Einführung sieht in etwa so aus: Man sucht ein Pilotprojekt aus, in dem man Scrum ausprobiert. Dazu wählt man ein Projekt aus, das wichtig ist (Erfolg und auch Misserfolg sollen deutlich sichtbar werden). Man sollte aber auch nicht sein Unternehmen auf den Projekterfolg verwetten. Es kann ja durchaus passieren, dass das Projekt scheitert.

Auf Basis der Erkenntnisse des Pilotprojektes entscheidet man dann, ob man ernsthaft mit der Scrum-Einführung ins Unternehmen beginnen will (Go/No-Go-Entscheidung). Wenn man sich dafür entscheidet, stellt man schrittweise Projekt für Projekt bzw. Team für Team auf Scrum um. So können Projekte aus den Erfahrungen der vorher gestarteten Projekte profitieren und insgesamt werden Risiken gesenkt. Die folgende Abbildung visualisiert das Vorgehen.

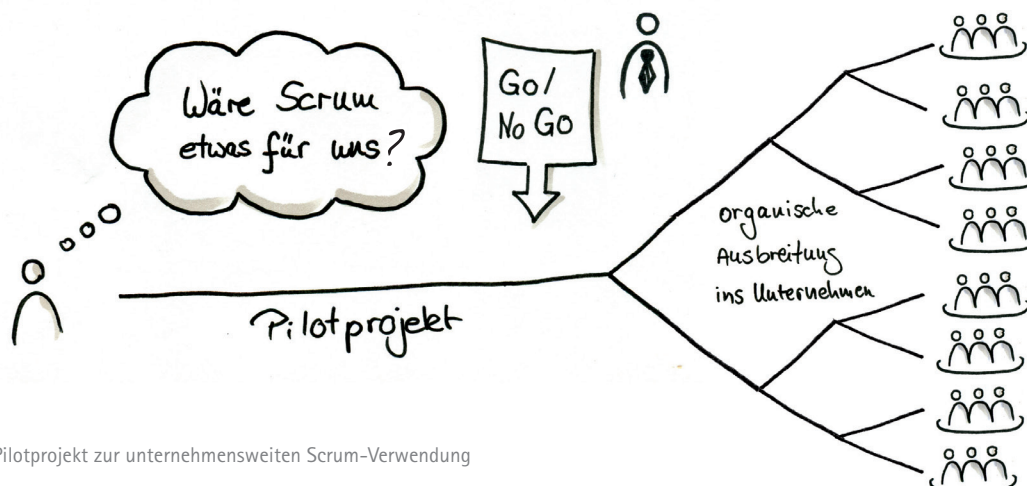


Abb. 1: Vom Pilotprojekt zur unternehmensweiten Scrum-Verwendung

Enterprise Transition mit Scrum

Von Stefan Roock

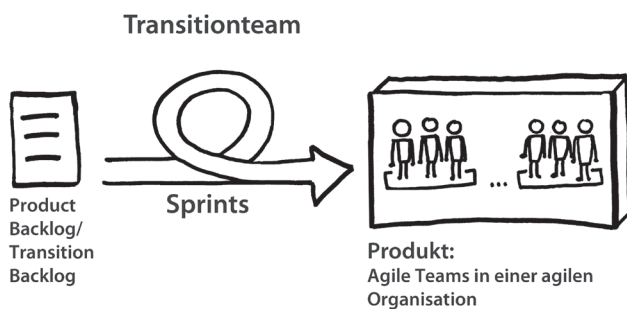


Abb. 1: Das Transitionsteam kümmert sich darum, dass agile Teams und eine agile Organisationsstruktur entstehen

kann man sich ggf. später noch vorknöpfen.

Die Produktvision wird durch das Produkt umgesetzt. Das mögliche Produkt des Transitionsteams variiert von einer Sammlung effektiver Scrum-Teams (relativ einfach zu liefern) bis hin zum komplett agilen Unternehmen (sehr anspruchsvoll).

Nur wenn klar ist, was die Produktvision und das zu erstellende Produkt ist und woran man eine erfolgreiche Lieferung dieses Produktes erkennt, kann das Transitionsteam effektiv auf dieses Ziel hinarbeiten.

Die Rollen

Zuerst stellt sich die Frage, wie die Rollen im Transitionsteam besetzt werden sollten. Der Product Owner sollte, wie sonst auch in Scrum üblich, das Team durch eine Produktvision führen. Er sollte für den Erfolg der Scrum-Einführung verantwortlich sein und in diesem Rahmen

eigenverantwortlich agieren können. Er muss bevollmächtigt (empowered) sein.

Das Entwicklungsteam hat die Aufgabe, das Produkt (also die agile Organisation) herzustellen, und das bedeutet, die Organisation merklich zu verändern.

Das alles deutet darauf hin, dass die Product-Owner-Rolle von einer hochrangigen Führungskraft wahrgenommen werden sollte (z.B. CEO, Vorstand oder direkt darunter). Das Entwicklungsteam sollte interdisziplinär besetzt sein und ebenfalls hochrangige Führungskräfte (z. B. Abteilungsleiter) zu seinen Mitgliedern zählen. Durch die Einbindung der Führungskräfte bekommt das Transitionsteam die Autorität im Unternehmen, um für die notwendigen Änderungen zu sorgen.

Der ScrumMaster muss ausreichend Selbstbewusstsein und Autorität haben, dass er es schafft, die diversen Führungskräfte zu einer produktiven Zusammenarbeit zu bewegen. Und die ScrumMaster-Rolle im Transitionsteam ist in vielerlei Hinsicht noch anspruchsvoller als in regulären Scrum-Teams. Zum einen ist es nicht trivial, die Führungskräfte dazu zu bringen, dem Scrum-Framework zu folgen. Außerdem besteht bei den Führungskräften die Besonderheit, dass sie nicht Fulltime im Scrum-Team mitarbeiten können. Sie haben ihre bisherige Führungstätigkeit parallel weiterhin auszuführen. Allzu leicht versinken sie im Tagesgeschäft und arbeiten nicht mehr aktiv im Transitionsteam mit. Der ScrumMaster hat die Aufgabe, ihre kontinuierliche Mitarbeit zu gewährleisten. ►



Wenn man keine Kunden hat,
keine Einnahmen und keinen Code,
dann ist Continuous Deployment
ein Selbstgänger!

Bernd Schiffer spricht mit Kent Beck und seiner Frau Cynthia über
Responsive Design, Software G-Forces und Ziegenzucht.



DAYS GERMANY ALL ABOUT AGILE

22.-23. November 2012
in Hamburg

**KONFERENZ ZU
AGILER SOFTWAREENTWICKLUNG
& EXTREME PROGRAMMING**

■ www.xpdays.de

Agil
Erfahrungen
Kanban
Kontakte
Lean
OpenSpace
Retrospektiven
Scrum
Soft Skills
TDD
XP
Spaß

& Vorträge, Tutorials
Diskussionsrunden
Community Day



andrena
OBJECTS

To tool or not to tool

Über die Vor- und Nachteile von elektronischen und physikalischen Tools.

Meike Mertsch, Stefan Roock



Feature Flags

Bessere Software durch Verzicht auf Branches

Von Alex Beppe

lität verfügen: dem Mutter-Branch und den jüngeren Geschwister-Branched. Eine zähe Arbeit – und eine risikoreiche.

Denn jeder Merge birgt die Gefahr struktureller und semantischer Konflikte. Ändert man sowohl im Mutter-Branch als auch in einem Tochter-Branch an der gleichen Stelle, können Merge-Werkzeuge die beiden Veränderungen unter Umständen nicht selbständig zusammenführen. Handelt es sich hierbei lediglich um formale Probleme, zum Beispiel Import-Statements, ist die manuelle Arbeit vor allem lästig. In statisch typisierten Sprachen bewahrt der Compiler vor vielen Patzern. Auch Unit-Tests leisten gute Dienste. Anders sieht es aus, wenn sich die Bedeutung des Codes verändert hat. Da helfen weder Compiler, noch Unit-Tests noch ausführbare Schlüssel-Beispiele.

Feature-Branched

Doch auch an anderer Stelle lauert Gefahr. Nämlich dort, wo Weiterentwicklungen unabhängig voneinander, das heißt in Geschwister-Branched, passieren. Man führt die Änderungen dem Haupt-Branch erst zu, wenn die Entwicklung des Features abgeschlossen ist. Warum man das tun sollte, könnten Sie fragen – und als Antwort

bekommen, dass dann die Entwicklungen ungestört voneinander passieren könnten (für ein Beispiel vgl. [2]). Oder auch, dass ein Team nur so sicherstellen könne, dass es zum Sprint-Ende ein auslieferbares Inkrement bereitstellt.

Wahrscheinlicher wird durch dieses Vorgehen, dass die Features, die später zu Ende entwickelt werden, nur noch schwer in den Haupt-Branch zu integrieren sind. Die Änderungen in Feature-Branched sind im Schnitt größer als die Einzeländerungen in Wartungsbranches. Umso höher ist das Risiko struktureller und semantischer Konflikte.

Kunden-Branched

Vergessen wir nicht Melchior und Balthazar, unsere Kunden. Mit ihren Wünschen sind sie in der Lage, eine explosive Situation anzuzünden. Denn manchmal haben sie verschiedene Wünsche. Nehmen wir als Beispiel eine

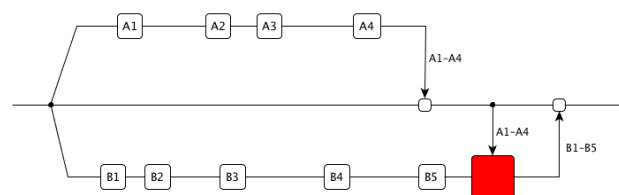


Abbildung 2: Potentieller Merge-Konflikt mit zwei Feature-Branched



Scrum But(t)

Natürlich führt nicht jede Änderung der Scrum-Regeln direkt in die Hölle. Allerdings ist es ratsam, erst einmal „Scrum in Reinform“ auszuprobieren, bevor man eigene Anpassungen vornimmt.

Von Stefan Roock

Karate für Vierjährige

Als mein 4-jähriger Sohn Mika neulich unseren Kleiderschrank durchwühlte, fand er meine alte Karate-Ausrüstung, die ich vor 6 Jahren eingemottet hatte. (Ich habe es immerhin bis zum Orange-Gurt gebracht, musste dann aber aufhören, weil ich durch die viele Reiserei nicht mehr regelmäßig zum Training gehen konnte). Mika war sofort begeistert: Er band sich gleich zwei Gurte um (doppelt hält besser), dafür aber nur einen Faustschützer, und ich musste ihm zeigen, wie eine richtige Karate-Stellung aussieht.

Da stand er nun auf dem Bett: Im Schlafanzug, mit einem gelben und einem orangenen Gurt um den Bauch, in einer merkwürdigen Haltung und rief immer wieder „Ich bin ein Karate-Kämpfer!“ Und das war er ja auch...irgendwie. Immerhin wies er



Buchtipps

Welche Bücher sollte man lesen, wenn es darum geht, ein agiles Management aufzubauen, erfolgreiche Test-Strategien einzuführen und schnell innovative Produkte auf den Markt zu bringen?

Management 3.0. Leading Agile Developers, Developing Agile Leaders.

Jurgen Appelo (2011), Addison-Wesley. 464 Seiten.

Schon während der Entstehung des Buchs konnten Besucher der Seite noop.nl vorab viele Teile daraus lesen. Als Management 3.0 dann schließlich in die Läden kam, war es schon gut vermarktet, aber für mich stellte sich die Frage, ob es sich auch zu lesen lohnt. Jürgen Appelo spricht in Management 3.0 vor allem Linienmanager an, da sie in Publikationen zur agilen Softwareentwicklung häufig viel zu kurz kommen. Er erklärt, wie agile Softwareentwicklung funktioniert und was diese Art der Arbeit für Manager bedeutet, in der sich die Menschen auf Selbstorganisation stürzen und ihre Manager oft als größtes Hindernis sehen.

Das Buch bietet nicht nur Hintergrundwissen, sondern auch konkrete Ansätze, wie es möglich ist, Teams zu führen, aber ihnen trotzdem die Freiheiten zu geben, sich selbst zu organisieren und vor allem sich weiterzuentwickeln. Jürgen Appelo gibt dem Leser Einblicke in die Gründe, warum Manager ihre Arbeit ebenso trans-

parent machen sollten, wie ihre Teams. Er zeigt, wie man Teams und gleichzeitig die eigene Position stärkt; und wie man geschickt Beschränkungen setzt, in denen sich Teams voll entfalten können und zusammenwachsen. Für nachhaltigen Erfolg im Allgemeinen sind sowohl Disziplin als auch Können wichtig. Daher beschreibt Jürgen Appelo auch wie man Kompetenzen in Teams fördert. Nicht zuletzt beschäftigt sich das Buch mit Mitteln und Wegen, um gute und sinnvolle Kommunikation zu etablieren.

Die Lektüre ist nicht nur unterhaltsam, sondern deckt auch viele Wege auf, mit agilen Entwicklungsteams erfolgreich zu werden. Managern aus der klassischen Softwareentwicklung mögen diese Ansätze ungewohnt vorkommen, aber in einer sich rasant verändernden Branche kann man es sich nicht leisten, auf dem Stand von vorgestern zu bleiben. Und auch für andere Beteiligte lohnt sich das Buch. Viele Inhalte sind auch für Schulungsleiter und Projektverantwortliche anwendbar. Daher kann ich das Buch mit gutem Gewissen weiterempfehlen.

Meike Mertsch



Vorhersage des Unvohersehbaren: Langfristige Planung in agilen Vorhaben

Agile Ansätze sind besonders stark bei Vorhaben mit hoher Änderungsfrequenz. Aber auch hier erwarten Kunden, Fachabteilung und das Unternehmensmanagement oft einen Planungshorizont von mehreren Quartalen. Ein agiles Team kann diese Forderungen nicht mit der Begründung, „das ist nicht Scrum“, zurückweisen, ohne schlechten Service für seine Kunden zu liefern. Andererseits begrenzen agile Verfahren den Planungshorizont bewusst auf eine Iteration oder ein Release, um sich ihre Reaktionsfähigkeit zu erhalten. Aus diesem Dilemma kann man sich mit Vorhersage-Techniken aus anderen Bereichen, wie zum Beispiel der Klimaforschung, helfen. Ich stelle in diesem Artikel ein Modell vor, das valide Zukunftsprognosen in agilen Vorhaben erlaubt.

Von Jens Coldewey

Agile Entwicklung folgt dem Grundsatz, „auf Änderungen zu reagieren ist wichtiger, als einen Plan zu verfolgen“. Dahinter steckt die Erkenntnis, dass Projekte nicht dann erfolgreich sind, wenn sie einen ursprünglichen Projektplan möglichst genau verfolgt haben, sondern wenn sie das Problem des Auftraggebers mit möglichst geringen Investitionen gelöst haben.

Bei den meisten Vorhaben kann man dennoch nicht auf eine mittel- bis langfristige Planung verzichten. Viele der organisatorischen Prozesse, in die ein agiles Vorhaben eingebettet ist, benötigen einerseits einen längeren Planungshorizont, können aber andererseits mit der hohen Geschwindigkeit agiler Teams nicht mithalten. Sich einfach mit der Einstellung „das ist euer Problem“ zurückzulehnen, ist genauso wenig hilfreich, wie ein dogmatisches Festhalten an halbverstandenen Lösungen. In beiden Fäl-

len bekommen die Kunden einfach schlechten Service.

Nun ist es glücklicherweise nicht so, dass agiles Arbeiten überhaupt keine Möglichkeiten böte, langfristige Prognosen aufzustellen. Allerdings sehen diese Prognosen nicht so aus, wie es die meisten Manager und Controller gewohnt sind, sondern folgen der agilen Tugend der Transparenz: Sie machen deutlich, wie sicher oder unsicher die Aussagen sind.

Wenig sinnvoll ist es, einfach Techniken zur Langzeitplanung aus traditionell geführten Projekten zu übernehmen, auch wenn das gerne als „das Beste aus beiden Welten“ verkauft wird. Traditionelle Planungstechniken bauen auf umfangreiche Analysen des kompletten Vorhabens. Eine solche Analyse liegt bei agilen Vorhaben schlichtweg nicht vor – sie wird während des Projekts entwickelt und fließt direkt in die Testfälle und den Code ►



Agilität und Human Systems Dynamics

Glenda Eoyang, Übersetzung Jens Coldewey

Was haben der Arabische Frühling, die europäische Schuldenkrise, der Pressekodex, ein Software-Entwicklungs-Team und eine glückliche Vierjährige gemeinsam? Sie sind alle selbst-organisierende Systeme! Sie alle bestehen aus relativ lose gekoppelten interagierenden Bestandteilen, die systemweite Verhaltensmuster ausprägen.

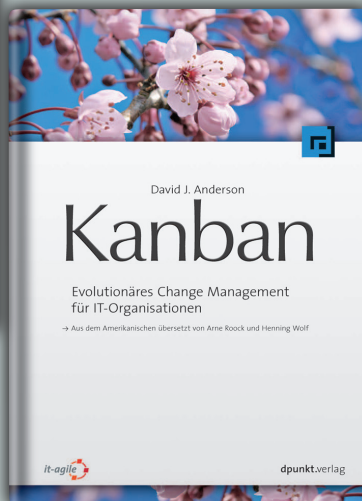
Im Arabischen Frühling bildeten die Bürger die Bestandteile, sie interagierten oft über das Internet, und die „emergenten“ Muster waren soziale und politische Reformen - oder zumindest ihr Versprechen. Für das Kind

sind die Bestandteile die Muster physischen, emotionalen, intellektuellen und sozialen Behagens; das emergente Ganze ist das gesunde Maß an körperlicher und geistiger Entwicklung. Die Bestandteile, Interaktionen und Muster der Software Entwicklung sind bekannt. Und diese Muster können befriedigend und produktiv sein, aber auch frustrierend und zeitverschwendend.

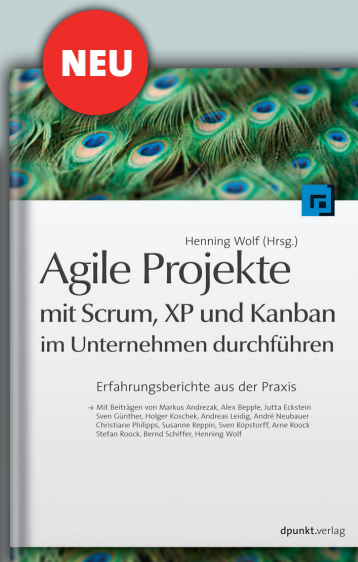
Agile Verfahren sind so entworfen, dass sie möglichst produktive Verhaltensmuster in der Arbeit fördern. Sie sind effektiv, weil sie die Voraussetzungen für einen ►



2011, 216 Seiten, Broschur
2. Auflage
€ 29,90 (D)
ISBN 978-3-89864-701-4



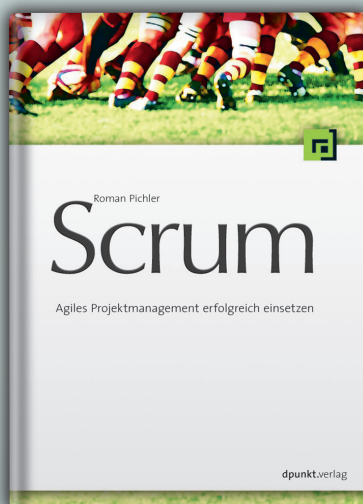
2011, 302 Seiten, Broschur
€ 34,90 (D)
ISBN 978-3-89864-730-4



2012, 222 Seiten, Broschur
€ 34,90 (D)
ISBN 978-3-89864-752-6



2011, 184 Seiten
€ 32,90 (D)
ISBN 978-3-89864-719-9



2008, 200 Seiten
€ 29,00 (D)
ISBN 978-3-89864-478-5



2012, 266 Seiten
2. Auflage
€ 36,90 (D)
ISBN 978-3-89864-790-8



2009, 240 Seiten
€ 36,00 (D)
ISBN 978-3-89864-630-7



2010, 264 Seiten
€ 29,90 (D)
ISBN 978-3-89864-640-6

Die agile review bequem im Abo!

ab 20 € pro Jahr (3 Ausgaben)



Hier Abo bestellen: www.agilereview.de