

# agile review

Das Kundenmagazin von it-agile

## *Lean Startup Schluss mit Spekulieren!*



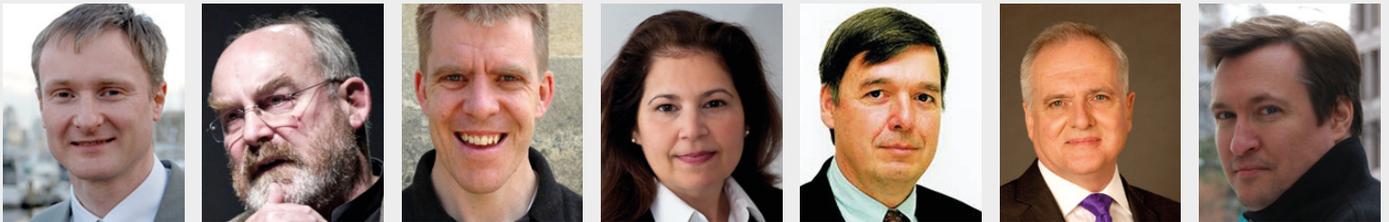


# lean kanban

CENTRAL EUROPE

VIENNA, OCT 22-23 2012

## THE LEADING EVENT FOR LEAN AND KANBAN IN CENTRAL EUROPE



- David J. Anderson - Father of Software Kanban
- Dave Snowden - Founder and chief scientific officer of Cognitive Edge
  - Benjamin Mitchell - Expert on Double Loop Learning
  - Dominica DeGrandis - Expert on Kanban for DevOps
  - Don Reinertsen - Expert on Lean Product Development
- Stephen Parry - Lean Service Thought Leader and Organisational Architect
- Jim Benson - Father of Personal Kanban, Brickell Key Award Winner 2012

[www.lean-kanban.eu](http://www.lean-kanban.eu)

presented by:



LEANability  
consulting & training



STOP STARTING... START FINISHING!



## editorial

Kaum ein anderes Thema hat in den letzten Monaten so viel Aufmerksamkeit auf sich gezogen wie der Lean-Startup-Ansatz von Eric Ries. Grund genug für uns, Lean Startup zum Titelthema dieser agile review zu machen und etwas genauer hinter die Schlagworte zu schauen. Wie praktikabel sind Ansätze wie Build-Measure-Learn, Pivot und Validated Learning wirklich? Und welche Stolpersteine lauern beim praktischen Einsatz? Sieben unserer Kollegen haben im Rahmen des Startup March ein eigenes kleines Lean Startup gegründet und diese Konzepte ausprobiert. Über ihre Erfahrungen können Sie auf Seite 20 lesen.

Letztlich geht es bei Lean Startup darum, in einem Umfeld extremer Ungewissheit ein erfolgreiches Produkt oder Dienstleistung zu entwickeln. Das zugrunde liegende Problem ist also Komplexität. Interessanterweise gibt es auch ganz andere Konzepte, die sich mit genau diesem Problem beschäftigen. Sie sind über 100 Jahre alt und stammen von der Preußischen Armee! Was Clausewitz und von Moltke zum Thema Komplexität zu sagen hatten und wie uns ihre Erkenntnisse heutzutage helfen können, erfahren sie im Interview mit dem Strategie-Experten Stephen Bungay auf Seite 32.

Natürlich gibt es auch Neuigkeiten bei „klassischen“ agilen Themen wie Scrum und Kanban. Auf Seite 28 hat unser Certified Scrum Trainer Stefan Roock 13 Dinge zusammengestellt, die Sie einmal ausprobieren sollten, wenn Sie das Gefühl haben, dass Scrum schon ganz gut bei Ihnen läuft und sie nach weiteren Inspirationen suchen. Seit März läuft das Akkreditierungs-Programm für Kanban-Schulungen der Lean-Kanban University. Was genau das ist, erfahren Sie auf Seite 48. Apropos Kanban: Im Mai fand in Boston die alljährliche Konferenz „Lean Software & Systems“ statt, sozusagen die Mutter aller Kanban-Konferenzen. Das Programm war 0wie immer atemberaubend - nicht nur wegen der Keynotes von Harvard- und MIT-Professoren. Die Video-Aufzeichnungen der Vorträge finden Sie auf [www.limitedwipsociety.org](http://www.limitedwipsociety.org) Die Messlatte liegt also extrem hoch, aber wir werden versuchen, im Oktober mit Lean Kanban Central Europe eine ähnlich herausragende Konferenz in Wien zu organisieren ([www.lean-kanban.eu](http://www.lean-kanban.eu)).

Die Frage, wie man Agilität über die Ebene einzelner Teams hinaus skalieren kann und wie eine Agile Organisation aussehen müsste, ist schon seit einigen Jahren immer mal wieder ein Thema. Ende 2011 veranstaltete die AgileAlliance in Dallas einen Workshop zum Thema Agile Enterprise Adoption, in der genau dieser Frage nachgegangen wurde. Unser Münchener Geschäftsführer Jens Coldewey war einer der Teilnehmer und fasst ab Seite 4 die Ergebnisse zusammen.

Darüber hinaus können Sie in dieser Ausgabe der agile review etwas über Qualität und Agilität lesen, mehr über leichtgewichtige Anforderungen erfahren, sich Anregungen für Ihre Standup Meetings holen und sich ordentlich gruseln, wenn es um Teambildung im Zombieland geht. Und natürlich haben wir auch in dieser Ausgabe wieder Literaturtipps und Veranstaltungshinweise für Sie zusammengestellt. ■

Viel Spaß beim Lesen wünscht Ihnen

Dr. Arne Roock, Chefredakteur der agile review

# INHALT



4 Was heißt hier „Agil“?  
Kennzeichen agiler Organisationen

Die Skalierung agiler Methoden über einzelne Teams hinaus wird ein immer wichtigeres Thema. Jens Coldewey geht der Frage nach, wodurch sich eine agile Organisation auszeichnet und welche Management-Strukturen angemessen sind.

11 Impressum

12 Buchtipps



14 Agile Anforderungen

Wie sehen Ihre Anforderungen aus? Markus Gärtner beschreibt drei verschiedene Wege, um Anforderungen flexibel und leichtgewichtig zusammenzustellen.

20 Lean Startup

Lean Startup ist in aller Munde. Aber was ist wirklich dran am Hype? Wie nützlich sind die Ideen von Eric Ries in der praktischen Anwendung? Sieben Kollegen wollten es wissen und haben ein eigenes kleines Lean Startup „gegründet“. Was dabei herausgekommen ist, können Sie in diesem Erfahrungsbericht lesen.

28 Try this!  
13 Scrum-Tipps

Auch wenn Sie schon seit Jahren Scrum einsetzen, finden Sie hier mit Sicherheit noch die eine oder andere Anregung – auch wenn sie etwas ungewöhnlich klingen mag.



Jeder will Qualität. Aber was genau ist das eigentlich? Und wie stellt man Qualität her? Dieser Artikel bringt Licht ins Dunkel und beschreibt, welche Rolle Qualität im Zusammenhang mit Scrum und Kanban spielt.

Kanban hat sein Nischendasein längst verlassen und wird von immer mehr Unternehmen erfolgreich eingesetzt. Um Kanban weiter zu professionalisieren, wurde ein Akkreditierungsprogramm für Kanban ins Leben gerufen.

Viele Teams nutzen nur einen Bruchteil des Potenzials, das ihnen Standup Meetings bieten. Dieser Artikel beschreibt, warum es um mehr geht als nur Status-Updates.



Erfahren Sie mehr über die acht Entwickler und Berater von it-agile, die Artikel zu dieser agile review beigesteuert haben.

## Interview

Steven Bungay über Strategie und die Preußische Armee

32

## Auf der Suche nach dem Qualitäter

40

## Kanban-Akkreditierung

48

## Standup Meetings

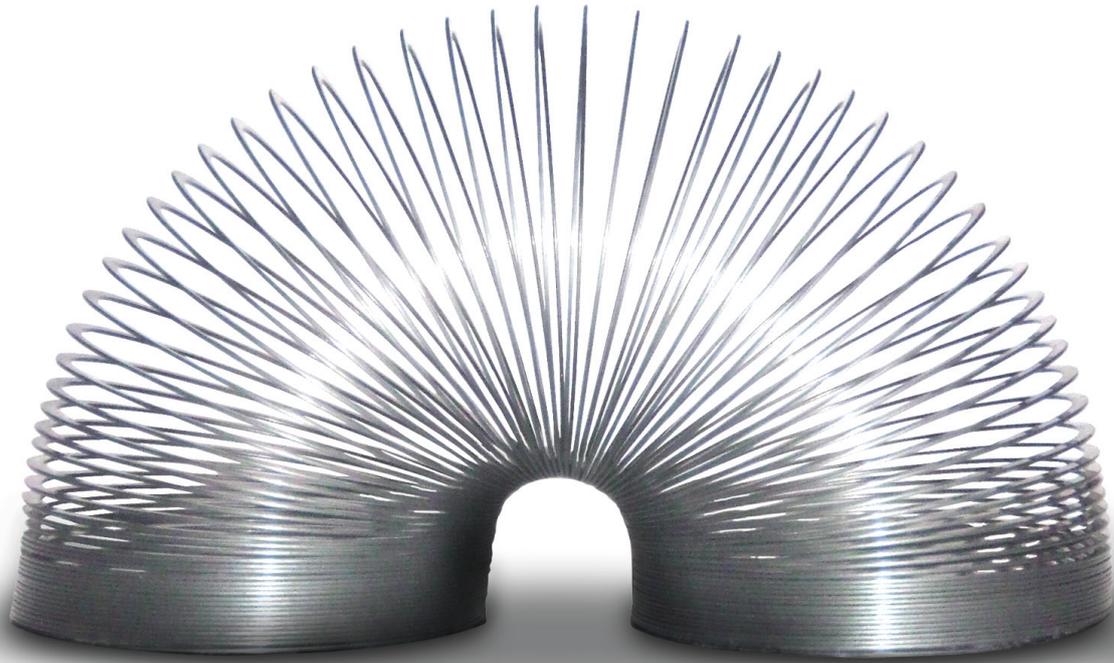
50

## Teambildung im Zombieland

54

## Autoren

56



## Was heißt hier „Agil“?

Wie skaliert man Agilität? Was sind die Kennzeichen agiler Organisationen? Die AgileAlliance hat ein Modell entwickelt, um genau diese Fragen zu beantworten. Von Jens Coldewey

Wer sich mit der Umstellung auf Agilität innerhalb einer Organisation beschäftigt, steht irgendwann vor der Frage, was eigentlich eine agile Organisation ausmacht. Bekannte Versuche, diese Frage zu beantworten, sind zum Beispiel Jeff Sutherlands „Nokia Test“ [Sut09], der den Grad einer Scrum-Einführung anhand von zehn Fragen zu bestimmen versucht, oder auch verschiedene Ansätze, angelehnt an das „Capability Maturity Model“ des SEI ein „Agility Maturity Model“ zu etablieren. Alle diese Versuche scheiterten aber letztlich, weil sie das falsche Problem lösen.

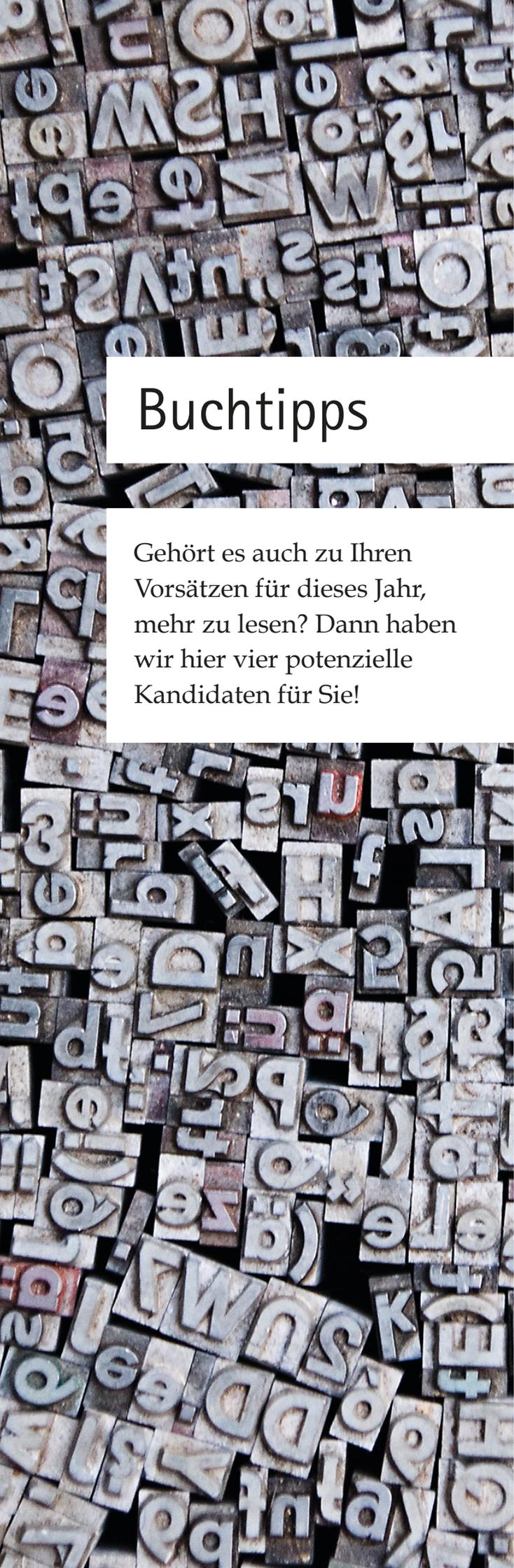
### Das falsche Problem

Diese Ansätze konzentrieren sich auf die An- oder Abwesenheit bestimmter Praktiken. So fragt zum Beispiel der Nokia Test in seiner neuesten Fassung nach der Iterationslänge. Die höchste Punktzahl erhalten Teams, die mit festen Iterationen von 4 oder weniger Wochen arbeiten, Null Punkte jene Teams, die keine Iterationen haben. Nun sind Iterationen beziehungsweise Sprints fester Länge ein zentrales Element von Scrum, aber vier Wochen werden

von vielen Teams bereits als extrem lang angesehen, der Stand der Kunst liegt mittlerweile eher bei ein bis zwei Wochen. Und sind Teams, die mit Kanban arbeiten und einen „Continuous Flow“ anstreben, deshalb weniger agil? Ist nicht die Selbstorganisation des Teams in Scrum wichtiger als die Sprintlänge?

Die Definition über Praktiken funktioniert aus mehreren Gründen nicht: Zum einen stellen Praktiken nur eine Momentaufnahme der derzeitigen Möglichkeiten dar. Waren zur Gründerzeit der agilen Verfahren zwei- bis vierwöchige Iterationen noch die technische Spitze, stellen Firmen wie Flickr oder ebay heute mit „Continuous Deployment“ dutzende neuer Versionen pro Tag in Produktion. Hier Grenzen zu ziehen, ist immer willkürlich und oft veraltet, bevor man sich auf die Grenze geeinigt hat.

Zum zweiten werden dadurch Teile des Prozesses festgelegt. Das erscheint zumindest merkwürdig für eine Bewegung, die als ersten Leitsatz hat „Uns sind Individuen und Interaktionen wichtiger als Prozesse und Werkzeuge“. Zwar verwenden agile Teams viel Aufwand dafür, ihren Prozess im Rahmen von Retrospektiven zu pflegen und zu optimieren, aber gerade das bedeutet ja auch ständige



# Buchtipps

Gehört es auch zu Ihren Vorsätzen für dieses Jahr, mehr zu lesen? Dann haben wir hier vier potenzielle Kandidaten für Sie!

## Tacit and Explicit Knowledge

Harry Collins (2010), University of Chicago Press, 200 Seiten

Immer wieder bin ich erstaunt darüber, aus welchen vielseitigen Bereichen ich für die Softwareentwicklung etwas lernen kann. Als ich im vergangenen Jahr mehrfach die Empfehlung zu einem Buch aus der Erkenntnistheorie erhielt, war ich zunächst etwas skeptisch. Was kann ich denn aus der Philosophie lernen für meinen täglichen Beruf? Harry Collins' „Tacit and Explicit Knowledge“ überraschte mich allerdings positiv. Collins beschreibt darin zwei wesentliche Aspekte: Erstens geht er darauf ein, dass flüchtiges Wissen nicht ohne explizites Wissen existieren kann. Er geht dabei auch auf verschiedene Abstufungen von flüchtigem Wissen ein, die es uns selbst erschweren, Wissen zu transportieren. Was ist beispielsweise der Unterschied zwischen einem Softwareprojekt und australischen Ureinwohnern, die mit einer modernen Spülmaschine konfrontiert werden? Zweitens beschreibt Collins mit seiner Unterscheidung zwischen polymorphen und mimeomorphen Aktionen, warum Testautomatisierung niemals einen fähigen Tester ersetzen kann. Wir können uns daran vielleicht annähern, aber gewisse Entscheidungen werden nach wie vor von einer Person mit einer kritischen Mentalität getätigt werden müssen. Zu guter Letzt kann ich noch für alle eine Entwarnung geben, die glauben, es handle sich um ein trockenes philosophisches Buch. Collins beschreibt sein Modell fundiert mit vielen Beispielen, die beim Lesen das Modell transparent machen.

Markus Gärtner

## Leadership Agility – Five Levels of Mastery for Anticipating and Initiating Change

Bill Joiner, Stephen Josephs (2006), Jossey-Bass, 336 Seiten

„Führungsflexibilität“ müsste man den Titel dieses Buches wohl übersetzen. Es stellt ein fünfstufiges Modell für die Fähigkeit von Führungspersonlichkeiten vor, mit Veränderungen umzugehen – so zumindest das Versprechen des Titels. Die beiden Autoren kommen aus der Organisations-Entwicklung und haben mit Softwaremanagement bisher kaum Berührungspunkte gehabt. Dass das



# Anforderungen als Leichtkost

**„Wenn die Landkarte und das Terrain nicht übereinstimmen, vertraue immer dem Terrain.“** Diktum der Schwedischen Armee [Gau89]

Agile Softwareentwicklung lehrt uns zwar, Flexibilität bei den Anforderungen zu bewahren. Doch wie geht das? Wie können wir sicher stellen, dass wir das Produkt bauen, das wir benötigen? Wie lässt sich das Anforderungsdokument durch leichtgewichtige Ansätze ersetzen, die zudem noch die notwendige Flexibilität besitzen?

Vor allem bekannt in der agilen Softwareentwicklung sind User Stories. Dabei gibt es allerdings ein paar Tücken zu beachten, denn viele Teams fallen hier schnell auf die Nase, wenn sie Funktionen nur für ihren Product Owner implementieren. Spezifikation mit Beispielen geht da noch ein Stück weiter und leitet Spezifikationen aus den Benutzerzielen ab. Durch testbare Beispiele entsteht zudem langfristig eine ausführbare Spezifikation, die einige Teams als lebendiges Dokumentationssystem nutzen. Doch wie gehen Startups mit Anforderungen um, wenn noch nicht einmal das eigene Geschäftsmodell klar ist? Hier kennt der Lean Startup mit seinem Build - Measure - Learn Zyklus einige Antworten.



# Lean Startup – Lessons learned

Von Doreen Timm und Sandra Reupke-Sieroux

Neal, ein junger Mann aus dem Norden, hat eine großartige Produktidee. Die Investoren schütteln alle verneinend die Köpfe, aber Jugend und Leichtsinn lassen ihn an seiner Idee festhalten. Also fängt er an, im Schrebergarten von Freunden zu programmieren – hoch motiviert, aber mit keinem Cent in der Tasche. Nach einem Jahr dann der Durchbruch! Seine Internetidee geht online, und er hat in den ersten Wochen mehr als 500.000 zahlende User. Wahnsinn! Drei Jahre später beschäftigt seine Firma 150 Mitarbeiter, macht 20 Millionen Euro Gewinn pro Jahr und geht kurz darauf an die Börse.

# Try this!

## 13 Dinge, die Sie in Scrum mal ausprobieren sollten Von Stefan Rook

Läuft Scrum bei Ihnen bereits einigermaßen rund? Suchen Sie noch Inspirationen, wie weitere Verbesserungen aussehen könnten? Dann finden Sie hier ein paar Dinge, mit denen Sie experimentieren können.

### Beipackzettel

Die vorgestellten Dinge sind Anregungen. Viele werden nicht auf Anhieb bei Ihnen funktionieren. Einige Dinge führen dazu, dass Sie streng genommen kein Scrum mehr machen.

Aber selbst, wenn etwas nicht funktioniert, trainiert es die Scrum-Muskeln: Experimentieren, Lernen und Verbessern.

## 1 Sprint-Burndown-Chart von Hand zeichnen

Viele Teams erstellen das Sprint-Burndown-Chart elektronisch aus einem Tool heraus. Das kann problematisch sein, weil die Teammitglieder das Burndown-Chart nur sehen, wenn sie sich aktiv darum kümmern. Je unangenehmer der im Burndown-Chart ablesbare Sprint-Fortschritt, desto geringer die Wahrscheinlichkeit, dass sich Teammitglieder aktiv das Chart ansehen. Daran ändert sich wenig, wenn das Burndown-Chart ausgedruckt und an die Wand gehängt wird. Der Ausdruck ist so klein, dass man das Burndown nur lesen kann, wenn man sehr nah heran tritt. Alternativ kann das Team das Sprint-Burndown-Chart von Hand auf einem Flipchart pflegen. Das klappt gut im Daily Scrum: Einfach mit dem ganzen Team die Restaufwände addieren und dann sofort das Chart fortschreiben - es geht übrigens besonders leicht, wenn Tasks sehr klein sind und das Team einfach nur alle Tasks zählt, die in Arbeit oder offen sind.

Als Ergebnis muss sich das Team mit dem Burndown-Chart beschäftigen, auch wenn es unangenehme Wahrheiten zeigt.

## 2 Variation der Fragen im Daily Scrum

Die klassischen drei Fragen in Daily Scrum lauten:

- Was habe ich seit dem letzten Daily Scrum gemacht?
- Welche Hindernisse sind sichtbar geworden?
- Was mache ich bis zum nächsten Daily Scrum?

Eine mögliche Variation formuliert die Fragen etwas anders und gruppiert sie. Zunächst beantworten alle Teammitglieder die ersten beiden Fragen, leicht umformuliert:

- Was habe ich seit dem letzten Daily Scrum erledigt?
- Welche Hindernisse sind sichtbar geworden?

Im Zuge dieser beiden ersten Fragen wird das Taskboard und ggf. das Burndown-Chart aktualisiert. So stehen dem Team alle Informationen über den aktuellen Sprint-zustand zur Verfügung, und es kann die dritte Frage als Team beantworten:

- Wie können wir als Team bis zum nächsten Daily Scrum den maximalen Mehrwert schaffen?

## 3 Sprint-Planung ohne Beamer

Häufig präsentiert der Product Owner die Storys für den Sprint mit einem Beamer in der Sprint-Planung. Dadurch wird eine asymmetrische Kommunikationssituation zwischen Product Owner und Entwicklungsteam geschaffen. Lassen Sie doch mal den Beamer weg und machen Sie so aus der Sprint-Planung ein Arbeitsmeeting! Der Product Owner bringt die potenziellen Storys auf Karteikarten mit und hängt sie an eine Pinnwand. Product ►

# Ich befürchte, einige Führungskräfte halten sich für Gott

Arne Roock spricht mit Stephen Bungay über modernes Management, das Preußische Militär und die Spice Girls

Organisationen, ihre Strategien in Handlungen umzusetzen, ist eine Seuche – genau genommen sogar eine internationale Pandemie, die so gut wie jede Firma infiziert hat, die ich kennen lerne.

Und zwischen diesen beiden Phänomenen gibt es eine Verbindung, die sich durch ein bestimmtes Modell erklären lässt, mit dem Organisationen in der Regel betrachtet werden. Dieses Modell wurde für Unternehmen gemacht, wie es sie während der industriellen Revolution gab. Damals standen Maschinen im Mittelpunkt von Organisationen. Die Vorstellung, dass Wirtschaftsunternehmen als Maschinen angesehen werden sollten und Management als eine Wissenschaft, hat sich vermutlich zu Beginn des 20. Jahrhunderts fest etabliert und geht zurück auf die Werke Frederick Winslow Taylors. Taylors Prinzipien verhalfen Unternehmen dazu, sehr effizient zu arbeiten, so lange die Umwelt stabil blieb und sie immer dieselben Dinge taten. Aber heute haben wir es mit einer Umwelt zu tun bekommen, die sich ständig ändert, so dass Ergebnisse nicht mehr vorhersagbar sind. Massenproduktion ist nur ein Baustein des Erfolges, und Effizienz allein reicht

nicht mehr aus, um wettbewerbsfähig zu sein. Darüber hinaus muss man auch effektiv arbeiten. Und man muss in der Lage sein, sich schnell auf Änderungen einzulassen. Das alte Taylor-Modell von Organisationen als Maschinen ist dafür aber nicht nützlich, auch wenn es immer wieder modifiziert wird. Ich bin der Meinung, es gibt ein anderes Modell, das viel hilfreicher ist. In diesem Modell werden Organisationen als Organismen betrachtet, die ganzheitlich geführt werden müssen. Denn wenn wir alle Teile einzeln optimieren, dann optimieren wir eben nicht das Ganze (wie das Taylor-Modell es annimmt). Ganz im Gegenteil: Wir verschlechtern das Ganze!

In dem Modell, das ich favorisiere, werden Menschen nicht als Roboter angesehen, die einfach nur Handlungen ausführen müssen, die von irgendwelchen schlaunen Planern erdacht wurden. Stattdessen ist es entscheidend, dafür zu sorgen, dass Menschen ihre Arbeit an den Resultaten ausrichten, die es zu erreichen gilt und ihnen dann die nötige Freiheit zu geben, damit sie die richtigen Dinge tun können. Dieses Modell, von dem ich spreche, wurde in der Preußischen Armee von Clau- ►

Auf der Suche  
nach dem Qualitäter!



Klar, jeder möchte Qualität in seiner Software. Und zwar immer und möglichst viel. Aber was genau ist eigentlich Qualität? Wie stellt man sie her? Und wer entscheidet, ob eine Software genug Qualität hat oder nicht? Dieser Artikel stellt der klassischen Vorstellung von Qualität eine alternative Sichtweise entgegen, die den Ideen der agilen Softwareentwicklung und des Lean Thinking entstammt. Von Arne Roock und Stefan Roock

Go slow to go fast! So lautet ein Motto aus dem Lean Management. Hinter diesem scheinbaren Paradoxon steckt eine fundamental wichtige Erkenntnis: Wenn wir unseren Fokus nur auf Geschwindigkeit, also effizientes Arbeiten richten (Effizienz = „die Dinge richtig tun“), werden wir uns damit früher oder später Probleme einhandeln, die unsere Geschwindigkeit massiv drosseln. Dieses Problem spiegelt sich auch in der klassischen Definition von Qualität wider, wie wir sie zum Beispiel in der Norm EN ISO 9000:2008 (der gültigen Norm zum Qualitätsmanagement) finden. Der zufolge ist Qualität der „Grad, in dem ein Satz inhärenter Merkmale Anforderungen erfüllt“. Wenn wir also Software in hoher Qualität bauen wollen, müssen wir laut der ISO-Norm nur dafür sorgen, dass wir die Anforderungen an das neue System richtig verstanden haben, um es dann entsprechend dieser Anforderungen zu entwickeln, und zwar möglichst effizient. Dass dieser Ansatz problematisch ist, zeigen Untersuchungen über die Benutzung von Softwaresystemen. Demnach werden 60 bis 70 Prozent der Funktionen selten oder gar nicht verwendet. Ob diese Funktionen gemäß den Anforderungen korrekt implementiert wurden, ist völlig unerheblich. Dass sie überhaupt implementiert wurden, ist das eigentliche Problem.

Versuchen wir es also mit einer anderen Sichtweise: Gerald Weinberg schreibt „Quality is value to some person“ [1]. Wie hoch die Qualität unserer Software ist, hat also zunächst einmal gar nichts mit den Anforderungen zu tun. Entscheidend ist viel mehr, ob bestimmte Personen, nennen wir sie Kunden, die Software wertvoll finden. Im besten Fall gibt es natürlich eine Übereinstimmung zwischen beiden Ansätzen: Der Wert für den Kunden wird

geliefert, indem seine definierten Anforderungen in Software überführt werden. Das Problem dabei ist, dass der Kunde in der klassischen Sichtweise nur einmal kurz am Anfang und am Ende des Entwicklungsprozesses beteiligt ist. Und das führt zu einer Reihe von Schwierigkeiten:

- Wir können am Anfang nie sicher sein, ob wir die Bedürfnisse der Kunden richtig verstanden haben. Weil wir den Kunden nach der Anforderungsdefinition aber ziemlich lange nicht zu Gesicht bekommen, können wir unsere Annahmen nicht wirklich validieren. Stattdessen setzen wir sie in Software um und hoffen, dass wir richtig liegen. Das trifft aber leider nicht immer zu.

- Was für unsere Kunden Wert schafft und was nicht, ist keine feste Größe, sondern ändert sich mit der Zeit. Nur weil ein Kunde zu einem bestimmten Zeitpunkt einen bestimmten Satz an Anforderungen für besonders wertvoll hielt, heißt das noch lange nicht, dass dies zu einem späteren Zeitpunkt noch so ist. Der Markt ändert sich, die Konkurrenz schläft nicht, neue technische Chancen ergeben sich und neue Ideen entstehen. Unter diesen Gesichtspunkten wäre es am Besten, wir könnten unseren Kunden möglichst oft die Chance geben, Einfluss auf die Weiterentwicklung des Systems zu nehmen.

- Mit den ersten beiden Punkten hängt die Frage nach der Wichtigkeit der einzelnen Anforderungen eng zusammen. Weil ich nicht genau weiß, welche Teile der Software den größten Wert für den Kunden schaffen, versuche ich natürlich, alles mit der höchsten Priorität umzusetzen. Wenn dann der Zeitplan durcheinander gerät (also eigentlich immer), kann ich nicht vernünftig entscheiden, welche Anforderungen die wichtigsten sind. ►



## „Where there is no standard, there can be no kaizen“

Die Lean-Kanban University ruft ein Akkreditierungsprogramm für Kanban-Schulungen ins Leben. Taiichi Ohno

Kanban hat in den letzten Jahren stark an Popularität gewonnen und führt schon lange kein Nischendasein mehr. Die Bandbreite von Firmen, die Kanban einsetzen, reicht dabei von kleinen Webagenturen (z. B. e-netconsulting) über Startups (Jimdo) und Web-Plattformen (mobile.de) bis hin zu großen Medien (BBC) und internationalen Konzernen wie SAP. Die Videos der Konferenz Lean Kanban Central Europe 2011 zeigen, wie breit das Spektrum der Einsatzgebiete ist: [www.lean-kanban-conference.de](http://www.lean-kanban-conference.de)

Wo viel Licht ist, da ist bekanntlich viel Schatten, und so bleibt es nicht aus, dass immer mehr Gerüchte und Halbwahrheiten über Kanban kursieren - etwa dass Kanban sich nur für Wartung und Systemadministration einsetzen ließe, nicht jedoch für die Produktentwicklung.

### Die Lean-Kanban University

Die Lean-Kanban University (LKU) hat es sich zur Aufgabe gemacht, Lean-Prinzipien und insbesondere Kanban zu professionalisieren und industrielle Standards hierfür zu etablieren und weiterzuentwickeln.

Zu den wichtigsten Aufgaben der LKU gehören im Moment:

- Der Aufbau und die Weiterentwicklung einer Webseite, die zur zentralen Anlaufstelle für den professionellen Einsatz von Kanban werden soll [1]. Zusätzlich bleibt die Limited WIP Society als nicht-kommerzielle Community-Seite erhalten und wird weiter ausgebaut [2]
- Die Ausrichtung und Unterstützung von Konferenzen, die einen Fokus auf Kanban haben, im Herbst 2012 stehen Lean Agile Scotland [3], Lean Kanban France [4], Lean Kanban Central Europe [5] und Lean Kanban Netherlands [6] an.
- Die Etablierung von Standards für hochwertige Kanban-Schulungen sowie die Ausbildung entsprechender Trainer.

Um den dritten - und derzeit zentralen - Punkt gewährleisten zu können, wurde im Februar 2012 offiziell das Akkreditierungsprogramm für Kanban-Schulungen ins Leben gerufen - initiiert und betreut von David Anderson. An diesem weltweiten Programm nehmen inzwischen Unternehmen aus zwölf verschiedenen Ländern in vier Kontinenten teil.

## Akkreditierte Kanban-Schulungen

Damit ein Unternehmen akkreditierte Kanban-Schulungen anbieten darf, müssen drei Bedingungen erfüllt sein: a) Das Unternehmen muss akkreditiert sein und sich als verlässlicher und professioneller Schulungsanbieter bewährt haben. b) Das Schulungsmaterial muss akkreditiert sein, also dem Curriculum und den Qualitätsansprüchen der LKU genügen. c) Der jeweilige Trainer muss akkreditiert sein, d.h. er muss über ausreichend Wissen und Erfahrung beim Einsatz von Kanban verfügen und regelmäßig aktiv in der Kanban-Community mitarbeiten. Die Liste der akkreditierten Trainer enthält übrigens internationale Koryphäen wie David Anderson, Karl Scotland, Yuval Yeret, Mattias Skarin, Dominica De Grandis, Olav Maasen, Rodrigo Yoshima und Mike Burrows. Werden alle drei Bedingungen erfüllt, garantiert die LKU, dass es sich um eine professionelle Kanban-Schulung handelt, die ihren aktuellen Standards und Qualitätsansprüchen genügt. Akkreditierte Schulungen sind am LKU-Logo erkennbar.



## Schulungen in Deutschland

it-agile hat als einer der 18 Charter Member das Akkreditierungsprogramm mitentwickelt und ist aktiv am Ausbau beteiligt. Mit Florian Eisenberg und Arne Roock verfügen wir über zwei akkreditierte Trainer, die sich kontinuierlich mit Kanban-Experten aus der ganzen Welt austauschen und sich so weiterbilden. Wir bieten akkreditierte Kanban-Schulungen (1-tägig und 2-tägig) sowohl inhouse als auch als offene Variante an. Die aktuellen Termine finden Sie auf unserer Webseite [www.it-agile.de/kanban](http://www.it-agile.de/kanban)

Haben Sie Fragen zu Kanban-Schulungen? Dann schreiben Sie einfach eine E-Mail an [kanban@it-agile.de](mailto:kanban@it-agile.de) ■

## Links

- [1] [www.leankanbanuniversity.com](http://www.leankanbanuniversity.com)
- [2] [www.limitedwipsociety.org](http://www.limitedwipsociety.org)
- [3] [www.leanagilescotland.com](http://www.leanagilescotland.com)
- [4] [www.lkfr.org](http://www.lkfr.org)
- [5] [www.lean-kanban.eu](http://www.lean-kanban.eu)
- [5] [www.leankanban.nl](http://www.leankanban.nl)

# Standup Meetings – mehr als nur im Kreis stehen und reden

Standup Meetings sind integraler Bestandteil der meisten agilen Methoden und gehören heute in vielen Teams und Organisationen zum Standard. Die Erfahrung zeigt jedoch, dass oft nur ein Bruchteil des Potenzials genutzt wird, das Standups bieten und dass sich schnell Routine breit macht, die Kreativität behindert und Verbesserung erschwert. Von Arne Rook

Als Berater lerne ich viele unterschiedliche Teams und Organisationen kennen. Viele dieser Teams führen schon Standup Meetings durch. In diesem Fall besuche ich das nächste Standup und beobachte aufmerksam, was genau sich dort abspielt. In den meisten Fällen haben die Standups die Funktion eines Status-Meetings – und diese Funktion erfüllen sie in der Regel auch gut. Das bedeutet, nach dem Meeting wissen die Teammitglieder, in welchem Zustand sich die aktuellen Aufgaben befinden, wo es Probleme gibt und welcher Kollege an welcher Aufgabe arbeitet.

## Mehr als Status-Meetings

Tägliche Status-Updates sind wichtig, aber wie Jason Yip in seinem tollen Blog Post „It's Not Just Standing Up: Patterns for Daily Standup Meetings“ beschreibt [1], ist dies nur eine von vielen Aufgaben, die ein gutes Standup erfüllen sollte. Er stellt dar, dass Standups GIFTS sein sollten.

### ■ G für Good Start

Gute Standups sollten das Team mit Energie für den Rest des Tages versorgen – und auf keinen Fall Energie entziehen! Energie entsteht, wenn das Meeting kurz und knackig ist, das Team an einem Strang ziehen lässt (weil es ein gemeinsames Ziel hat) und am besten auch noch Spaß macht.

### ■ I für Improvement

Im Sinne einer Kaizen-Kultur sollten Standups ein weiterer Ort für Verbesserungen sein (zusätzlich zu Retrospektiven). Dabei geht es nicht nur um das Lösen von Problemen, sondern auch darum, Ideen und Tipps & Tricks auszutauschen sowie kleine Experimente zu beschließen und regelmäßig zu bewerten.

### ■ F für Focus

Standups sollten dafür sorgen, dass das Team fokussiert an einem Ziel arbeitet, anstatt einfach nur möglichst viele Dinge zu tun. Insofern ist es oft gar nicht so entscheidend, was genau die Teammitglieder getan haben und was sie vorhaben zu tun, sondern wie das Team gemeinsam seinem Ziel einen Schritt näher kommt.

### ■ T für Team

Standups können und sollten ein Ort sein, an dem nach Hilfe gefragt wird und an dem aktiv Hilfe angeboten wird. Wie kann ich dazu beitragen ein Problem möglichst schnell aus dem Weg zu räumen? Ist es sinnvoll, dass wir zu zweit oder dritt an einer Aufgabe weiterarbeiten, weil diese gerade den größten Nutzen für unser Ziele bietet? Standups können also Ausdruck von funktionierendem Teamwork sein und gleichzeitig dazu beitragen, die Teamarbeit weiter zu verbessern.

### ■ S für Status

Jedes Teammitglied sollte zu jeder Zeit wissen, wie der

# Teambuilding im Zombieland

Von Holger Bohlmann

... zuerst gab es nur ein paar Berichte, die niemand glauben wollte. Doch dann überschlugen sich die Ereignisse. Was erst fern war, kam mit der Wucht eines gewaltigen Orkans über uns herein. Es gab kein Entkommen: Die Straßen füllten sich mit den lebenden Toten und ihr Heer wuchs innerhalb weniger Stunden, ohne dass es jemand aufhalten konnte. Monster, die nichts als rasende Wut kannten. Tot, aber doch erschreckend lebendig und hungrig auf das Fleisch der Lebenden. Wenige konnten sich retten, sich verstecken oder einfach nur weglaufen, weg von ihnen. So wie wir, sieben Überlebende, die sich in das Rechenzentrum geflüchtet haben. Das Gebäude ist sicher, die Eingangstür ist massiv und nicht zu überwinden. Keine Fenster im Erdgeschoss, die von den Untoten eingeschlagen werden können. Nun sitzen wir hier, Wasser gibt es genug, Essen und Munition sind knapp. So hat uns das Schicksal zusammengeführt, sieben noch lebende Menschen, die sich nicht wirklich kennen. Was können wir tun? Wie können wir den Alptraum überleben - Können wir ihn ÜBERHAUPT überleben? ...

Jedes Jahr treibt mich wieder ein Filmfest ins Kino. Und was man da zu sehen bekommt, ist nicht wirklich Mainstream-Kino, sondern kommt eher aus der Ecke Action & Suspense. Auch der gute alte Zombie-Streifen darf da nicht fehlen. Spannend bei solchen Filmen ist, dass es nicht nur um das Bewältigen der stetig wachsenden Armee der Untoten geht, sondern auch immer um die bunt zusammengewürfelte Truppe, das Team, welches um das nackte Überleben kämpft.

## Wo bleibt das gemeinsame Ziel?

Manchmal kommt einem das alles irgendwie bekannt vor. Wenn z.B. in dem Horror-Projekt mal wieder düstere Szenarien ausgemalt werden, für den Fall, dass das Projekt scheitert. Wir werden alle sterben! Wie im Film sind solche Aussagen nur begrenzt hilfreich. Aber wie im Film wäre es dann mal an der Zeit, etwas zu unternehmen, beispielsweise mutig einen Ausfall zu wagen. Irgendwo gibt es immer zumindest die Chance einer Rettung. Wenn sich alle zusammenreißen und auf ein Ziel hinarbeiten, die Schrotflinte oder was auch immer in die Hand nehmen und rausgehen, dann - und nur dann - können sie es überhaupt schaffen. Man mag jetzt sagen: Klar, im Film stirbt dabei auch immer die Hälfte des Teams und verwandelt sich in Zombies. Aber das passiert meist genau dann, wenn die Leute irgendetwas Anderes, vor allem etwas Unüberlegtes und nicht Abgesprochenes tun, was nicht dem eigentlichen Ziel dient. Häufig ist es so, dass einzelne Teammitglieder verdeckte Ziele haben, die erst dann offensichtlich werden, wenn es zu spät ist und die falschen Leute eine Waffe in der Hand haben. Ärgerlicherweise bricht genau

dann auch noch eine wilde Horde von Zombies in das sichere Versteck ein. Unter diesem äußeren Druck und der inneren Zerrissenheit zerbricht das Team schnell, genau wie im Projekt! Ausgleichende Gerechtigkeit: In einem Zombie-Film wird das Verfolgen von hinterhältigen Zielen direkt mit einem Biss bestraft. Haps!

Ein ebenfalls spannendes Thema ist die Übermacht der Übernatürlichen. Ich habe einen Film gesehen, bei dem durch einen Unfall außerirdisches Leben auf unseren Planeten kommt, sich verteilt, gut gedeiht und sich eigentlich sehr heimisch fühlt. Unglücklicherweise sind die neuen Lebewesen auf unserem Planeten auch recht groß, man könnte sagen, es sind Monster, was natürlich zu einem großen Haufen an Problemen führt. Die Menschen haben versucht, dieses neue Leben zu bekämpfen, was nicht so gut klappte. In dem Film hatte ich den Eindruck, dass das Team Mexiko, bei dem die „Verseuchung“ anfang, sich irgendwie damit abfindet und langsam merkt, dass man nicht alles großflächig einzäunen kann. Dagegen investiert das Team USA offenbar viel Ressourcen in den Aufbau einer gewaltigen Mauer. Und was stellt sich später heraus? Die Mauer ist vollkommen nutzlos, man kann die Natur nicht aufhalten. Eine einfache Wahrheit, die ignoriert wurde. Passiert auch in Projekten, mit den gleichen Folgen. Viel Zeit und Geld wird in etwas investiert, was man nicht aufhalten oder kontrollieren kann. Der Klassiker: Wenn die zu entwickelnde Software auch noch den letzten komplizierten Prozess abdecken soll. Funktioniert nie. Nachher kehren die Anwender doch zur Natur zurück und benutzen wieder Papier und Telefon, weil es einfach angemessener ist und man damit auch (über)leben kann.



**Reporting  
zum obersten Chef**

**Schnelle pragmatische  
Lösungen für die  
Fachabteilung**

**Die Entwickler wollen  
mehr Selbstorganisation**

**Der Betrieb soll laufen**

**Wie das gehen soll?  
Antworten, Ideen, Anregungen unter  
[www.it-agile.de/agileOrganisation](http://www.it-agile.de/agileOrganisation)**