Ausgabe 02/2013
www.agilereview.de

agile review

Das Kundenmagazin von it-agile

Das Chaos beenden!

LeanKanban Central Europe

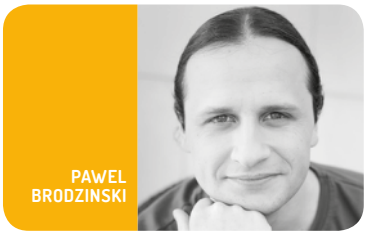
HAMBURG
NOV 4.-5. 2013

Die Konferenz zu Lean und Kanban!

Lernen Sie mehr über

- **R** Risikomanagement
- **F** Lean Forecasting
- **L** Leadership
- **E** Erfahrungsberichte
- **S** Kanban at Scale
- **C** Change Management

TREFFEN SIE



Hamburg, 4.-5. November 2013 • www.leankanban.eu

editorial

Agilität ist längst den Kinderschuhen entwachsen. Das merken wir unter anderem daran, dass wir verstärkt Anfragen nach Schulung und Beratung bekommen, die weit über die reine Softwareentwicklung hinausgehen oder gar nichts mehr damit zu tun haben: Werbeagenturen, ein Werk, das Farben herstellt, oder ein Hersteller von Pumpen sind da nur einige Beispiele. Also war es naheliegend, Agilität außerhalb der IT zum Titelthema dieser Ausgabe zu machen. Dazu passte es dann auch gut, dass unser Moneypenny-Team seine eigenen Erfahrungen mit dem Einsatz von Agilität in der Verwaltung reflektiert und als Artikel verarbeitet hat. Was dabei herausgekommen ist, können Sie ab Seite 22 lesen.

Ein zweiter Trend, der sich abzeichnet, ist die Ausweitung von Agilität und Lean über die reine Softwarentwicklung und -auslieferung hinaus in die Upstream-Prozesse – das so genannte „Fuzzy Front End". Auch hieran lässt sich ablesen, dass agiles Vorgehen immer erwachsener wird. Der Artikel zu Portfolio-Kanban ab Seite 10 beschreibt, wie diese Reise häufig abläuft, welches Potenzial im Produktmanagement schlummert und welche Hürden hier zu nehmen sind.

Schätzungen sind für viele Entwicklungsteams ein großer Schmerz! Sie verursachen oft große Aufwände, bringen wenig Spaß und sind am Ende doch fast nie korrekt. Wie kann man diesen Schmerz lindern? Lässt sich ein pragmatischerer Umgang mit Schätzungen finden? Oder können wir sogar ganz auf Schätzungen verzichten, so wie es die #noestimates-Bewegung propagiert? Diesen und weiteren Fragen geht Stefan Roock in seinem Artikel nach (Seite 38).

Nachdem wir in der letzten Ausgabe über unsere Erfahrungen mit der Entwicklung der iPhone-App für mobile.international berichtet haben, stellen wir diesmal vor, wie die mobile Website von WindFinder.com zustande kam (Seite 46), die beweist, dass es nicht immer eine native Smartphone-App sein muss, wenn man mit dem richtigen Handwerkszeug arbeitet.

Außerdem erwartet Sie in dieser Ausgabe wieder ein Interview (diesmal mit dem Lean-Experten Stephen Parry, Seite 32) sowie ein Artikel zum Wert von kurzen Feedbackschleifen, insbesondere beim Testen (Seite 4). Und Sie erfahren, was es mit der Rolle des „Active Observers" auf sich hat (Seite 50), welche Neuerungen es seit August am Scrum-Framework gibt (Seite 18) und welche Bücher Sie sich einmal genauer ansehen sollten (Seite 30). ■

Viel Spaß beim Lesen wünscht Ihnen

Dr. Arne Roock, Chefredakteur der agile review

INHALT

Was will uns das sagen?

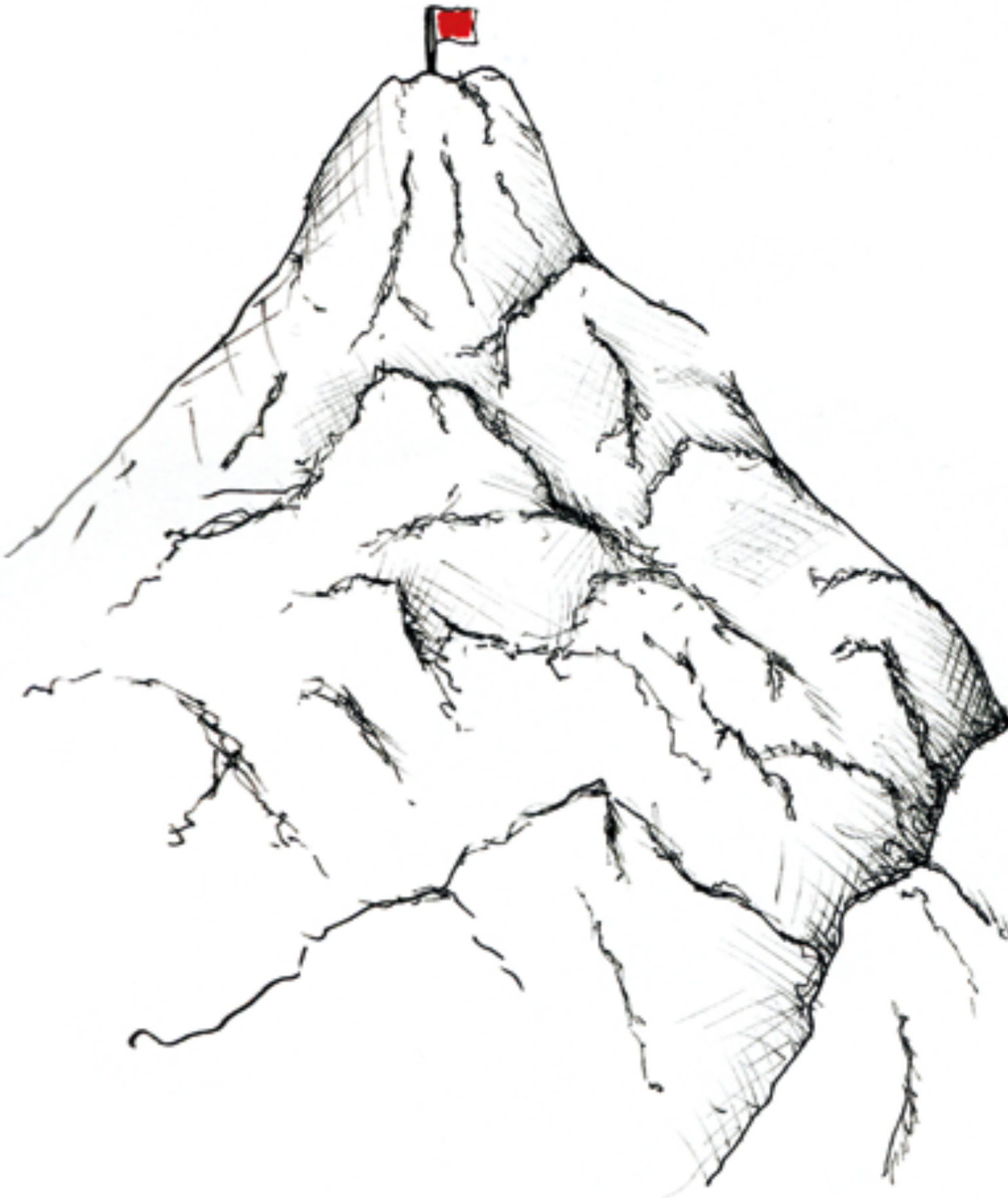
Von Feedbackschleifen und solchen, die es gerne wären.

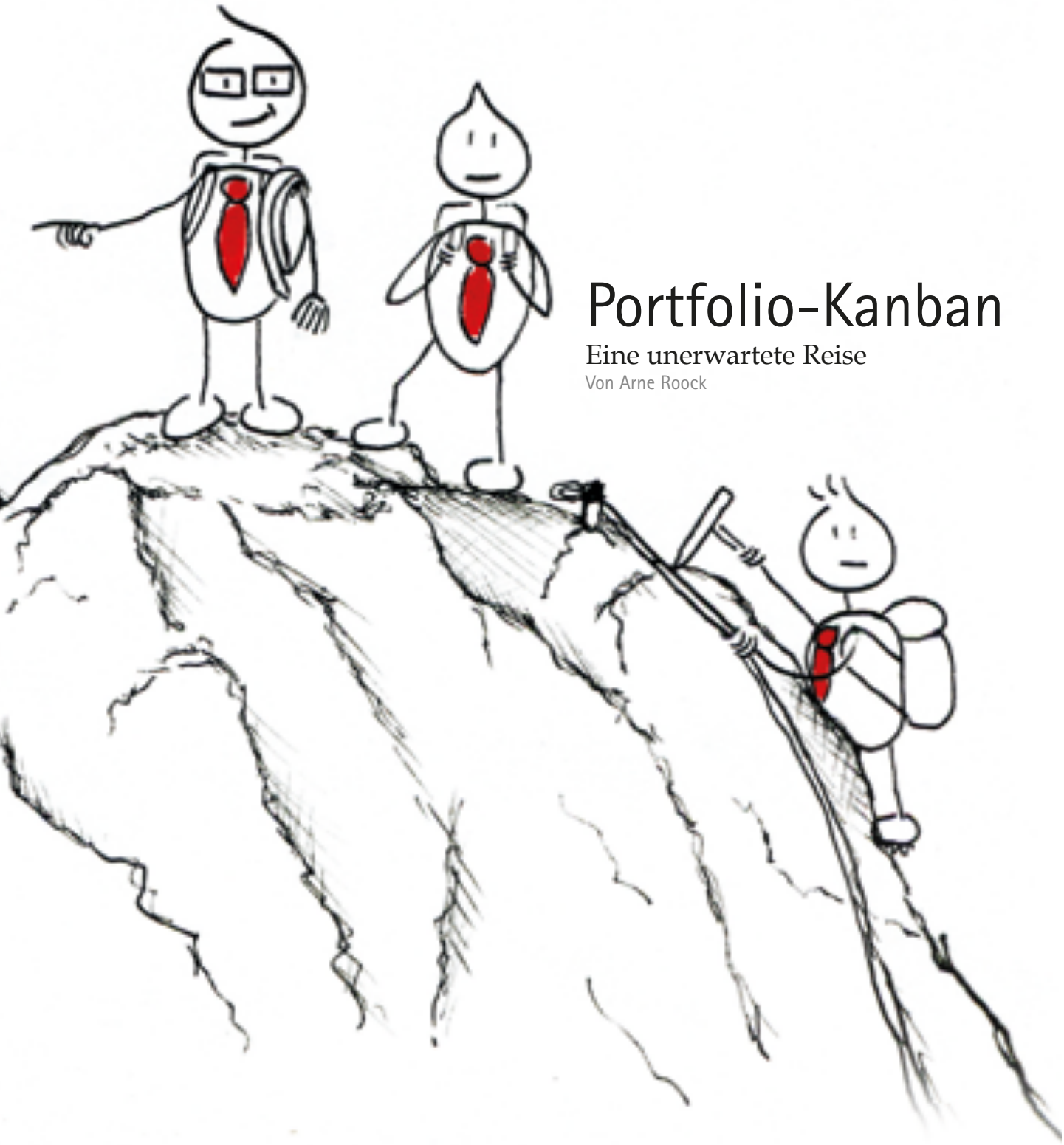
Von Markus Gärtner

„Warum brauchen unsere Tester eigentlich so lange?“ „Testen ist das Bottleneck.“ „Wann sind unsere Tester endlich fertig?“ Kennen Sie solche Aussagen? Mit agiler Softwareentwicklung wird eine derartige Wahrnehmung meistens verstärkt, weil auf der einen Seite das Team regelmäßig ausliefern soll und auf der anderen Seite das Unternehmen viel mehr Transparenz in seinem Entwicklungsprozess erhält. In kurzen Zyklen soll funktionsfähige Software geliefert werden – und das am besten mehrmals täglich.

Die obigen Aussagen lassen sich aber auch anders ausdrücken: „Warum haben wir so viele Risiken nicht entschärft, die die Tester jetzt überprüfen müssen?“ „Wieso müssen wir den engen Flaschenhals der Tester dafür benutzen, um herauszufinden, was in der Flasche drin ist?“ „Wie kann ich meinen Testern dabei helfen, schneller zu werden?“ Der wesentliche Unterschied zwischen den ersten Formulierungen und denen in diesem Absatz liegt darin, dass mit den ersten Formulierungen stets ein impliziter Wunsch einherging. Alle drei Aussagen implizieren, dass die Software schneller getestet werden kann – ohne zu sagen, wie das denn gehen soll.

Um regelmäßig ausliefern zu können, benötigen wir in der Softwareentwicklung Feedbackzyklen. Testaktivitäten schließen lediglich einen Feedbackzyklus. Wenn das Testen lange dauert, bedeutet das zunächst mal, dass wir in der Entwicklung zu viele Risiken aufgebaut haben, die wir jetzt evaluieren und ggf. entschärfen müssen.





Portfolio-Kanban

Eine unerwartete Reise

Von Arne Roock

Wenn Organisationen mit Kanban beginnen, verfolgen sie damit in der Regel gewisse Ziele, insbesondere kürzere Durchlaufzeiten, höhere Qualität oder weniger Stress. Weil es bei mittleren und großen Unternehmen zu Beginn fast nie möglich ist, die gesamte Wertschöpfungskette zu visualisieren und zu verbessern, startet man mit einem kleinen Ausschnitt und definiert die Grenzen seines Kanban-Systems. Sehr häufig handelt es sich bei diesem Ausschnitt um die Softwareentwicklung mit mehr oder weniger Testanteilen. Ein einfaches Kanban-Board sieht dann beispielsweise so aus:

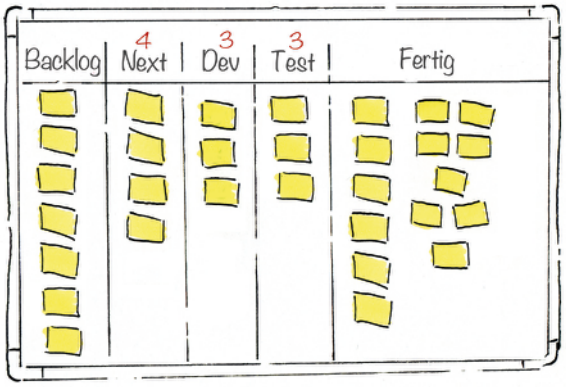


Abb. 1: Ein einfaches Kanban-Board für ein Entwicklungssystem

Das Team visualisiert also seinen Workflow, die Aufgaben sowie Blockaden. Außerdem werden die wichtigsten Prozess-Regeln explizit gemacht und die beiden Aktivitätsspalten „Entwicklung“ und „Test“ mit WiP-Limits (WiP=Work in Progress) versehen. Und siehe da: In der Regel kann man schon nach kurzer Zeit beobachten, dass die gewünschten Effekte tatsächlich eingetreten sind: Kürzere Durchlaufzeiten, höhere Qualität, weniger Stress. Dies hört sich nach Voodoo an, lässt sich aber einfach erklären: Durch die WiP-Limits werden nämlich jetzt die Queues (Warteschlangen) im System kontrolliert und reduziert. Denn jetzt arbeiten die Teammitglieder an weniger Aufgaben gleichzeitig, die Qualität der jeweiligen Aufgabe erhöht sich und vor allem entstehen weniger Wartezeiten, weil eben nicht zwischen so vielen Aufgaben wie vorher hin- und hergewechselt wird. Oder noch schlimmer: Es wurde vielleicht nur ziwschen einigen Aufgaben hin- und hergewechselt, während andere ewig unerledigt am Board hingen. Vor der Einführung von Kanban gibt es in der Regel eine große Menge an Aufgaben, die darauf warten, dass aktiv wieder an ihnen gearbeitet wird: Etwas ist fertig entwickelt, aber es ist kein Tester verfügbar. Oder es ist eine Blockade aufgetreten (z. B. weil eine Information fehlt), und so wurde die Aufgabe beiseite gelegt, um mit einer neuen Aufgaben zu beginnen. Diese Queues führen dazu, dass das System ins Ungleichgewicht gerät, weil die Arbeit im System nicht mehr zur Leistungsfähigkeit des Systems passt.

Sobald wir anfangen, den Flow zu optimieren (anstatt die Auslastung, wie es in klassischen Projekten üblich ist), sind die langen Queues in unserem System das erste, was wir angehen sollten. Und das ist relativ einfach: Das Werk-

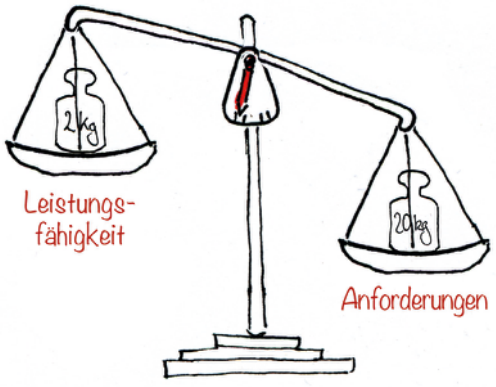


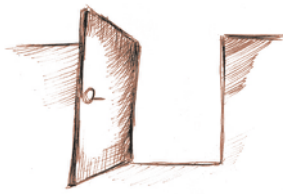
Abb. 2: Flow entsteht immer dann, wenn sich die beiden Schalen „Anforderungen“ und „Leistungsfähigkeit“ in der Waage befinden. Allerdings sieht das Bild bei der Einführung von Kanban so gut wie immer so aus, dass die Anforderungen schwerer wiegen.

Neuerungen im Scrum-Guide 2013

Jeff Sutherland und Ken Schwaber, die Väter von Scrum, haben den Scrum-Guide im Juli 2013 erneut angepasst. Hier die wichtigsten Änderungen seit der letzten Aktualisierung 2011. Von Henning Wolf

1. Transparenz der Artefakte

Der Guide enthält jetzt einen neuen Abschnitt zum Thema Transparenz der Artefakte. Scrum beruht auf Transparenz. Entscheidungen zur Wertoptimierung oder zum Risikomanagement basieren auf dem wahrnehmbaren Zustand der Artefakte. Je mehr Transparenz hergestellt wird, desto fundierter wird damit die Basis der Entscheidungen. Je geringer die Transparenz wird, umso fehlerhafter werden die Entscheidungen, umso geringer der Nutzen und umso höher möglicherweise die Risiken.



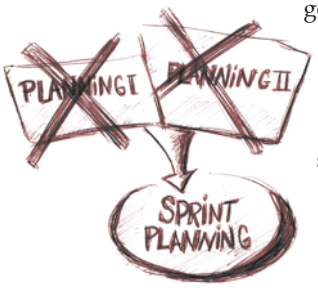
2. Sprint-Planning ist jetzt ein (in Zahlen: 1) Meeting

Bisher unterschied der Scrum-Guide zwischen Sprint-Planning I und II. Stattdessen ist die Sprint-Planung jetzt ein Meeting mit zwei Themen:

a) Was können wir schaffen in diesem Sprint?

b) Wie wird die ausgewählte Arbeit erledigt?

Nachdem das Entwicklungsteam eine Vorhersage der im Sprint realistisch zu erledigenden Product-Backlog-Einträge vorgenommen hat, erstellt das Scrum-Team (also Product Owner, Entwicklungsteam und ScrumMaster gemeinsam) ein Sprint-Ziel. Das Sprint-Ziel stellt den Zusammenhang der Arbeit des Entwicklungsteams dar, den es bei unterschiedlichen Initiativen ohne gemeinsames Ziel nicht geben würde. Man beachte die formale Aufnahme des Sprint-Ziels in den Scrum-Guide.



3. Product-Backlog-Einträge werden verfeinert bis zum Zustand „Ready"

Das Product Backlog wird eher verfeinert als gepflegt (Schwaber und Sutherland schreiben von „Refinement" statt „Grooming"). Verfeinerte Product-Backlog-Einträge sind transparent, ausreichend verstanden und feingranular genug, um in die Sprint-Planung einzugehen und für den Sprint selektiert zu werden. Product-Backlog-Einträge mit dieser Transparenz nennen wir „Ready". „Ready" und „Done" sind zwei Zustände, welche die Transparenz verstärken.

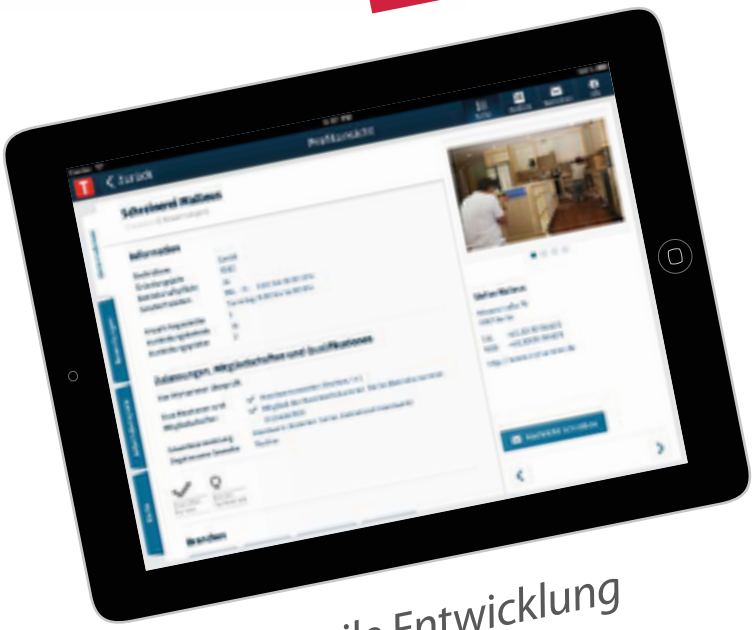
4. Scrum Meetings und Time-Boxing

Scrum schreibt einen Satz an Meetings vor, um Regelmäßigkeit herzustellen und den Bedarf an zusätzlichen nicht in Scrum definierten Meetings zu minimieren. Alle Meetings sind zeitbeschränkt (time-boxed) in dem Sinne, dass sie eine Maximallänge haben. Ein Sprint stellt ein umgebendes Meeting dar mit einer festen Länge, die weder gekürzt noch verlängert werden kann. Alle anderen Meetings können beendet werden, wenn ihr Zweck erreicht ist; dies stellt sicher, dass angemessen Zeit verwendet wird, aber keine Verschwendung im Prozess herrscht.

it-agile

Entwicklung, Beratung, Schulung

Was? it-agile macht auch **Entwicklung?**



Agile Entwicklung mit bester Flexibilität, Time-to-Market und Qualität – z. B. die iPad-App von MyHammer.

MyHammer

Agile Entwickler von it-agile

www.it-agile.de

coders@it-agile.de

Die IT-Welt ist nicht genug!

Eine Moneypenny-Kollegin und ein agiler Berater beschreiben hier die agilen Veränderungen, die wir in den letzten 18 Monaten in unserem Verwaltungsteam durchgeführt haben. Von Nadine Lammers und Urs Reupke

Herbst 2011 – Ich brauche Veränderung

Nach gut vier Jahren im gleichen Job (Assistenz / Verwaltung / Einkauf), in dem ich viel gelernt und wenig gelitten habe, suche ich nach einer neuen Herausforderung. Ich sehe für mich in der Firma keine Entwicklungsmöglichkeit mehr, die mich reizt. Ich brauche Veränderung. In dem Moment, in dem mir das so richtig bewusst wird, ist die Kündigung beschlossene Sache und es gibt kein Zurück mehr. Und schon laufen mir zwei Jobs über den Weg. Einer in Form von Simone, die ich auf dem Flur treffe; sie arbeitet als Moneypenny bei it-agile. Und der zweite über den Kontakt einer Freundin. Letzterer: Persönliche Assistenz und gute Fee in einer Coaching-/Beratungsfirma. Das bedeutet ein spannendes Umfeld und viel persönlicher Einsatz, was mir liegen würde, aber ich wäre wieder Einzelkämpferin, also lehne ich ab.

Simone und it-agile kenne ich schon ein wenig, weil unsere Räume auf demselben Büroflur liegen und ich schon so manchen Konferenzmaterial-Transport per Palette für sie organisiert habe. Die agilen Kollegen sind in meinem Umfeld schon immer als etwas andersartig aufgefallen: „Das sind die mit den bunten Bällen und den vielen Zetteln an den Wänden, die immer so viel lachen im Büro." Andersartigkeit reizt mich, it-agile will mich, also geht's los. Dass ich mit meinem Wechsel drei Bürotüren weiter eine derart andere Arbeitswelt betreten würde, habe ich dabei nicht erwartet.

1. Dezember – Mein erster Tag

Mein erster Tag besteht aus einem Moneypenny-Workshop in einem Co-Working-Space.
Neben mir sind drei weitere Moneypennies aus Hamburg

Das sollten Sie lesen.

JUnit-Profiwissen: Effizientes Arbeiten mit der Standardbibliothek für automatische Tests in Java

Michael Tamm (2013), dpunkt.verlag, 300 Seiten

Ich erinnere mich noch genau, es war direkt nach einem Kundenauftrag zu TDD und Refactoring. Ein paar Entwickler kannten sich mit xUnit-Frameworks und entsprechenden Mocking-Bibliotheken bereits aus, andere nicht. Ich hatte keine wirklich griffige Referenz für den Einstieg in JUnit. Es gab zwar schon allerhand gute Bücher zu dem Thema, aber ich hätte eigentlich eine Kombination aus Büchern empfehlen müssen, um alle Bereiche abzudecken, die sinnvoll waren. Kurz danach bekam ich das Manuskript zu JUnit-Profiwissen von Michael Tamm. Als ich das Inhaltsverzeichnis überflog, merkte ich, wie wichtig dieses Buch für den gerade beendeten Kundenauftrag gewesen wäre. Denn es deckt inhaltlich JUnit 3 und 4 ab. Für beide Versionen gibt es einen Einstieg am Anfang. Ich persönlich würde für ein neues Projekt derzeit ausschließlich JUnit 4 verwenden, allerdings gibt es noch allerhand alte Projekte, die auf JUnit 3 aufbauen. In Bezug auf JUnit 4 werden nicht nur die Grundlagen, sondern auch fortgeschrittene Themen behandelt. Dazu gehören beispielsweise @Categories und @Theories. Zum ersten Mal tiefer verstanden habe ich allerdings die JUnit @Rules, die seit ein paar Jahren Einzug in JUnit gefunden haben. Mit ihnen lassen sich beispielsweise datengetriebene Tests, Vorbereitung eines Spring-Test-Kontexts und das Aufräumen der Datenbank anschließend miteinander kombinieren.

Gleichzeitig bewahren die JUnit-Rules aber auch eine Kapselung der einzelnen Problemlösungen, die man dann auch mehrfach wiederverwenden kann. Das Buch behandelt darüber hinaus Mocking-Bibliotheken wie JMock und Mockito. Es werden zwar nicht alle Aspekte vorgestellt, aber mit den Grundlagen kann eigentlich jeder von jetzt auf gleich mit einem fokussierten Test loslegen. Außerdem geht das Buch noch auf fließende Assertions mit Hilfe von entsprechenden Bibliotheken wie Hamcrest und FEST ein. Abgerundet wird das Buch durch Hinweise zu wartbaren Unit-Tests und wie man nicht nur funktionale Tests mit Hilfe von JUnit erstellen kann. Durch die Fülle an abgedeckten Themen halte ich das Buch deshalb für ein Must-read für jeden Entwickler.

Markus Gärtner

„Für mich heißt Lean vor allem: Menschen, Menschen, Menschen“

Stephen Parry spricht mit Doreen Timm über Lean, den Aufbau von Organisationen, die am Kunden orientiert sind, und über das, woran Manager gemessen werden sollten.

Der Titel deines Buches lautet „Sense and Respond“ – was genau ist damit gemeint?

Der Titel stellt eine Aufforderung an Organisationen dar, sich viel eingehender mit den Bedürfnissen ihrer Kunden zu beschäftigen. Es reicht nicht aus zu wissen, was die Wünsche der Kunde bezüglich des Produkts oder der Dienstleistung sind, sondern wir müssen auch verstehen, welcher tatsächliche Zweck erfüllt werden soll und welches Ziel der Kunde verfolgt. Warum das wichtig ist? Traditionell arbeiten Organisationen nach dem Motto „Ich produziere etwas und vermarkte es“; man identifiziert sich mit dem Produkt oder dem Service, den man anbietet, und betreibt dann großen Aufwand, um den Kunden zu überzeugen, dass er dieses Produkt unbedingt braucht. Demgegenüber stehen Organisationen, die sich nach dem Prinzip Erkennen und Reagieren (Sense and Respond) ausrichten: Sie definieren sich über den Wert, den sie für ihre Kunden schaffen, und sind bereit, ständig mit neuen innovativen Produkten und Dienstleistungen auf veränderte Anforderungen zu reagieren.

In dem herkömmlichen Modell – ich produziere etwas und vermarkte es dann – übernimmt der Hersteller die Führung und glaubt an das Prinzip von Angebot und Nachfrage, was schlicht überholt ist. Mit Sense and Respond übernimmt der Kunde die Führung, es sollte also viel mehr in Richtung „Nachfrage und Angebot“ gedacht werden. Der Organisationsaufbau muss entsprechend angepasst werden – Mitarbeiter müssen sich ganz und gar in den Kunden hineinversetzen und so viel wie möglich über dessen Umfeld und dessen Anliegen lernen. Die Führungskräfte müssen auf diese Erkenntnisse reagieren, indem sie ihre Mitarbeiter in die Lage versetzen, das Geschäft und das Produkt ständig in diesem Sinne zu verbessern.

Was sind deiner Meinung nach die größten Probleme, die bei der Umwandlung hin zu einem kundenorientierten Unternehmen auftreten?

Das ist einfach: ein System aus Messen und Belohnung. Selbst wenn man den Nachweis und den Business Case für eine Veränderung eines Unternehmens präsentiert, wird nichts passieren, solange man nicht auch das System aus Messen und Belohnung verändert.

Die meisten Belohnungssysteme zielen auf die effiziente Verarbeitung großer Mengen an Arbeit ab. Die meisten Messsysteme konzentrieren sich auf Quantität, wie beispielsweise die Verkaufszahlen, die Zahl der beseitigten Fehler oder die Anzahl der Codezeilen anstatt auf die Auswirkungen auf den Kunden. Das Problem ist die Messung dieser sogenannten Effizienz, die irrtümlicherweise für Effektivität, also für geschaffenen Kundenwert, gehalten wird. Wenn man das Belohnungssystem anpasst ▶

Darf's ein bisschen mehr sein? (Ver-)Schätzen in der Softwareentwicklung

Die Diskussion um das Thema Schätzen im agilen Kontext ist vielschichtig. Diese Vielschichtigtkeit wird noch übertroffen vom Dogmatismus, mit dem die Diskussion geführt wird. Die einen meinen, man müsste alles immer mit Story-Points und Planning-Poker schätzen, während die anderen Schätzungen generell für Teufelszeug halten und es einfach ganz bleiben lassen wollen. Dieser Artikel stellt verschiedene Perspektiven dar, unter denen das Thema Schätzen betrachtet werden kann, und macht deutlich, dass das Ob und Wie vom Kontext abhängt. Die Leser erhalten eine Orientierung darüber, welcher Schätzansatz für ihre Situation geeignet ist. von Stefan Roock

addicted to the wind!

Eine mobile Webseite für WindFinder.com

Stefan ist begeisterter Surfer. Am liebsten würde er jedes Wochenende sein Material einpacken, an die Ostsee fahren und auf sein Board steigen. Dabei gibt es nur ein Problem: Surfen kann man nur, wenn die Windverhältnisse stimmen. Bei Windstille kommt man nicht voran, bei Sturm wird es zu gefährlich. Und bei 3 Grad Celsius und Regen macht Surfen auch nur bedingt Spaß. Deshalb ist es für Stefan wichtig, vor dem Wochenende immer mal wieder kurz die Windverhältnisse zu checken, um zu entscheiden, ob er einen Surftag einplant oder sich lieber mit seinen Freunden zum Grillen verabreden soll.
Schnell auf verlässliche Wetterdaten zugreifen zu können, ist nicht nur für Surfer wichtig, sondern auch für Angler, Segler, Taucher und viele andere. Aus diesem Grund gibt es WindFinder.com – eine Plattform, die sich immer größerer Beliebtheit erfreut. Neben dem Web-Service bietet WindFinder.com inzwischen auch native Smartphone-Apps für iPhone und Android. Daneben hat Geschäftsführer Jonas Kaufmann jedoch ein weiteres Bedürfnis seiner User identifiziert:

„Wind- und Wetterinformationen gehören zu den Themen, die häufig unterwegs gesucht werden. Wir haben bereits sehr gut verwendbare Apps für iPhone und Android, aber wir

Active Observer

Teams regelmäßig einen Spiegel vorzuhalten und frühzeitig auf mögliche Probleme und Missverständnisse hinzuweisen, kann sehr wertvoll sein. Dafür bietet das Konzept des Active Observers eine nützliche Struktur. Was genau macht ein Active Observer? Und welche Fallstricke gilt es zu vermeiden?

Von Meike Mertsch

Abb. 1: Die Beobachtung selbst

nehmen. In unserem Beispiel könnten wir zu dem Schluss kommen, dass die Person mit den Geschehnissen nicht einverstanden ist, ihr kalt ist oder diese Position schlicht bequem für sie (vgl. Abb. 2).

Der dritte Aspekt ist die Signifikanz, die wir der Beobachtung und Interpretation zuordnen. In einem warmen Raum wären wir verwirrt, wenn Kälte der Grund für die Körperhaltung wäre. Ist der Raum hingegen kalt, wird diese Möglichkeit wahrscheinlicher.

In einem vierten Schritt erfolgt dann eine Reaktion. Wir könnten ein Fenster zumachen oder die Heizung hochdrehen, falls wir die verschränkten Arme als ein Zeichen von Kälte interpretieren (vgl. Abb. 3). Genauso könnte ein Team entweder deshalb schnell auseinanderströmen, weil es in kopflosen Aktionismus verfällt, oder aber, weil jeder genau weiß, was als nächstes zu tun ist. Im ersten Fall würde ich die Beobachtung teilen und eventuell vorschlagen, ▶

XP DAYS GERMANY

XP Days Germany 2013

14.-16. November in Karlsruhe

Die Mitmachkonferenz für
Extreme ProgrammiererInnen

10 Jahre

Offene Konferenz für
Agile Software-Entwicklung
und Extreme Programming
mit Fokus auf Entwicklerthemen.

Vorträge, Hands-On, Pecha Kuchas,
Diskussionsrunden und Community Day

www.xpdays.de

andrena OBJECTS

it-agile

SCRUM-ASSESSMENT

NUTZEN SIE DIESE VORTEILE VON SCRUM?

kürzere Time-To-Market
größere Transparenz
bessere Planbarkeit
innovativere Produkte
höhere Qualität

INHALTE DES SCRUM-ASSESSMENTS

- *Was ist Kontext und Ziel der Scrum-Einführung?*
- *Vorort-Beobachtungen (insbesondere Meetings)*
- *Einzelinterviews*
- *Inspektion der Artefakte (insbesondere Product Backlog)*
- *Ergebnisaufbereitung als Präsentation für Team/Management*

ERGEBNISSE DES SCRUM-ASSESSMENTS

- *Standortbestimmung zu Ihrer Scrum-Implementation*
- *Identifikation von Verbesserungspotential*
- *konkrete Verbesserungsvorschläge*

STEFAN ROOCK

Das Scrum-Assessment dauert je nach konkreter Situation zwei bis fünf Tage. Ich berate Sie gerne.

Stefan Roock
(Senior-Berater)
E-Mail: stefan.roock@it-agile.de
Tel. 0172/429 76 17

it-agile