

Agil, Lean und Open

Dominik Rose

InterFace AG

Rhein.Ruhr (Duisburg)

Schlüsselworte

Agil, Lean, Open, SW-Entwicklung, Business Agilität

Einleitung

Technologie-Trends kommen und gehen, genauso wie Modeerscheinungen im Projektmanagement. Zählen agile Methoden wie Scrum und Kanban dazu? Wenn nein, warum umranken sie immer noch so viele Mythen und Vorurteile? Basierend auf konkreten Projekterfahrungen erläutere ich, warum die Eigenschaften Agil, Lean und Open für jeden Software-Entwickler und Projektmanager essentiell sind, ganz unabhängig vom konkreten Vorgehensmodell. Gleichzeitig möchte ich dafür werben, jede Art von Dogmatismus zu vermeiden. Vielmehr sollten wir uns darauf zu konzentrieren, wie wir unseren Anwendern und Kunden am besten helfen können.

Agile SW-Entwicklung – Im Mainstream angekommen

Februar 2001: Ein Treffen von 17 Software-Experten verändert die Welt der Softwareentwicklung. Geprägt durch die jahrelange Erfahrung der Beteiligten entsteht ein Manifest, das Leitplanken setzt für einen Paradigmenwechsel. Statt Softwareentwicklung als einen streng linearen Prozess anzusehen, der ingenieurmäßig über Jahre im Voraus planbar ist, erkennt das Manifest die Nichtlinearität in klassischen Projekten durch folgende Werte an:

- *Menschen und Interaktionen sind wichtiger als Prozesse und Werkzeuge*
- *Funktionierende Software ist wichtiger als umfassende Dokumentation*
- *Zusammenarbeit mit dem Kunden ist wichtiger als die ursprünglich formulierten Leistungsbeschreibungen*
- *Eingehen auf Veränderungen ist wichtiger als Festhalten an einem Plan*

Rund um diese Werte ranken bis heute zahlreiche Mythen, z.B. „in agilen Projekten wird nicht dokumentiert“. Daher ist es immer wichtig, dass die Werte nur eine Priorität vorgeben und nichts darüber aussagen, dass die Punkte auf der rechten Seite unwichtig sind.

Daneben stellt das agile Manifest 12 Prinzipien, die auf Basis der Werte Leitsätze für die agile Arbeit bilden. Dazu gehören die frühe und kontinuierliche Auslieferung von Software, die hohe Wertschätzung von technischer Qualität sowie die zentrale Bedeutung von Einfachheit als Design-Richtlinie. Insbesondere finden sich dort auch die Begriffe Selbstorganisation des Teams und Selbstreflektion, die die Rolle des Software-Entwicklers vom reinen Befehlsbefolger zum mitdenkenden und individuell handelnden Wissensarbeiter heben.

2014: Agile Vorgehensweisen sind weit verbreitet. Laut [1] nutzen 88% der befragten Organisationen aktiv agile Methoden, darunter mehr als die Hälfte in der Mehrheit ihrer Projekte. [2] spricht davon, dass nur noch 22% der Unternehmen durchgängig klassische Methoden anwenden, der Rest geht in verschiedenen Ausprägungsstufen agil vor. Definitiv Erfolgsmerkmale. Trotzdem bin ich der Überzeugung, dass ein Umdenken erforderlich ist, vor allem in der agilen Community selber. Die folgenden Absätze erläutern den dahinterliegenden Gedankengang.

Agilität ist kein Selbstzweck

Zunächst glaube ich, dass es Zeit ist für etwas Selbstkritik:

„Every specialisation has its own jargon, and IT is no different – but many times it seems that techies love to co-opt regular English words and give them new meanings. Not only does this practice lead to confusion in conversation with non-techies, but even the techies often lose sight of the difference between their geek-context definition and the real-world definition that normal people use“ [3].

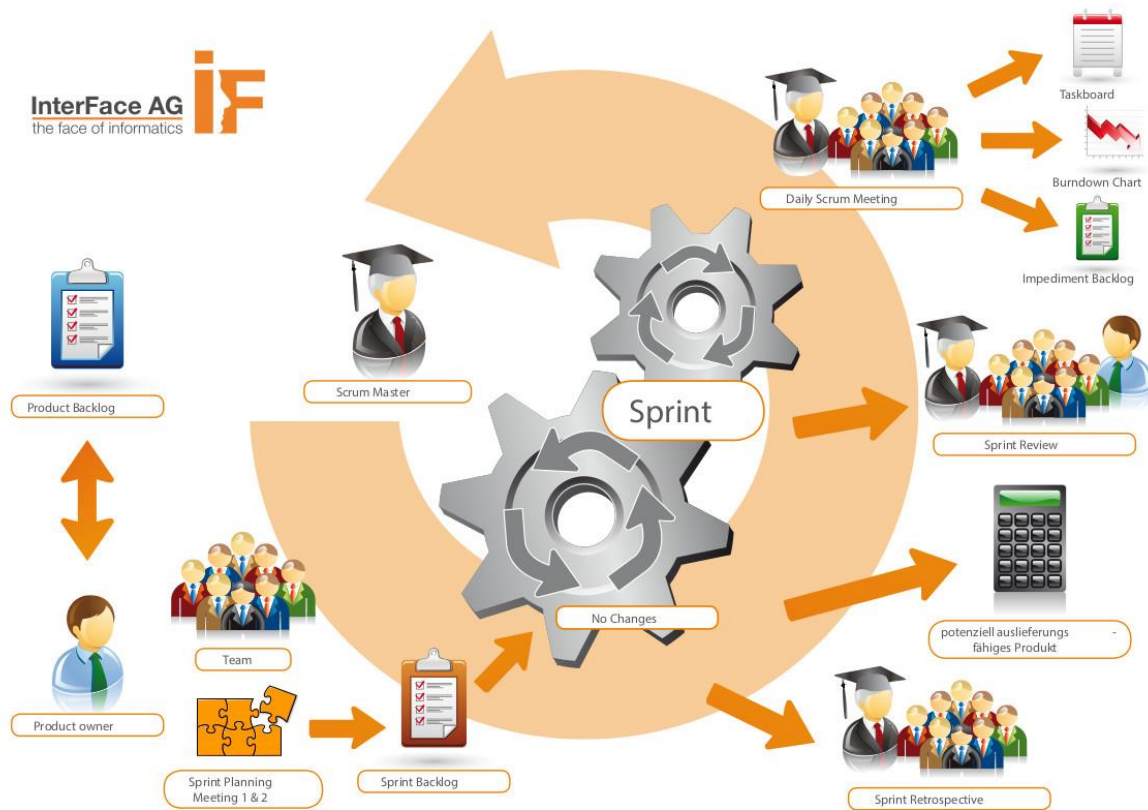
Innerhalb der „agilen Welt“ nehme ich seit einiger Zeit den Trend wahr, in Schwarz und Weiß zu denken: Entweder man ist agil, oder man ist es nicht. Wenn das Management nicht agil ist, dann kann man das nicht ändern, sie verstehen es halt nicht. Oftmals wird dabei eins vergessen: Organisationen dienen einem Zweck, und dazu gehört (in den meisten Fällen) Geld zu verdienen. Besser zu sein als die Konkurrenz. Das ist nicht einzige Daseinsberechtigung eines Unternehmens, aber es wäre naiv es zu missachten. Und die Welt, in der sich die meisten Unternehmen befinden, ändert sich immer schneller. Technologie und Globalisierung sind zwei Hauptgründe dafür.

Es gibt auch auf Management-Ebene schlaue Menschen. Seit einiger Zeit prägt sich daher der Begriff „Business Agilität“. Dieser geht viel weiter als die Anwendung agiler Methoden in der Softwareentwicklung. Vielmehr geht es darum, als Unternehmen flexibel auf den sich ändernden Markt reagieren zu können. Was passiert, wenn man das nicht schafft, zeigen Beispiele wie Kodak, Nokia oder Microsoft.

Warum erzähle ich das auf einer technischen Konferenz? Weil ich der Überzeugung bin, dass die agil denkenden Softwareentwickler eine große Chance haben. Auf der einen Seite, weil IT immer wichtiger wird in Unternehmen, auf der anderen Seite, weil sich viele der erwähnten Prinzipien und Werte sehr gut auf Unternehmensebene übertragen lassen.

Die agilen Werte dürfen nicht in der SW-Entwicklung aufhören

Um diese Chance zu nutzen, hilft ein tieferer Blick in die Entwicklung agiler Methoden. Für viele Menschen ist der Begriff „agile Softwareentwicklung“ eng verbunden mit Scrum.



Scrum wurde 2001 von Ken Schwaber entwickelt und ist ein Projektmanagementframework. Durch Umsetzung von agilen Prinzipien wie Selbstorganisation, Kundenorientierung und Eingehen auf Veränderungen, gleichzeitig aber Sicherstellung der benötigten Stabilität innerhalb einer Iteration (eines sogenannten Sprints) schafft es einen guten Rahmen, um in Softwareentwicklungsprojekten mit hoher Unsicherheit konsequent agil vorgehen zu können. Nicht weniger, aber auch nicht mehr.

Leider ist meine Erfahrung bei der Einführung von Scrum nicht nur positiv. Bei mehreren Unternehmen ist der Begriff mittlerweile mit negativen Assoziationen geprägt, aus zwei Gründen:

Zum einen erzeugt Scrum Transparenz. Typische Artefakte wie eine Retrospektive oder ein Burndown-Chart haben das Ziel, konsequent über den Projektfortschritt zu reflektieren. Allerdings bedeutet Reflexion auch, sich konstruktiv mit Problemen auseinanderzusetzen und sich darauf aufbauend zu ändern. Wenn das nicht klappt, ist es sehr einfach, die Schuld anderen zu geben, z.B. der mit hohem Aufwand eingeführten Methode.

Zum zweiten hat Scrum hohe Einführungshürden. Um Scrum konsequent zu leben, darf man nicht in der SW-Entwicklung aufhören. Vielmehr bedeutet es Veränderung in den Fachbereichen, in der Budgetierung und auch im Betrieb. Ich habe einige Projekte scheitern sehen, weil diese notwendigen Veränderungen missachtet wurden.

Kommen wir aber zurück zum schönen Begriff „Business Agilität“ und der Rolle der Softwareentwickler. Helfen uns dogmatische Diskussionen über Scrum und andere Methoden wirklich weiter? Oder sollten wir uns nicht vielmehr darauf konzentrieren, agile Werte in unser Umfeld hereinzubringen?

Konzentriert euch auf Menschen und Interaktionen statt auf Prozesse und Werkzeuge

Ein gutes Beispiel ist ein Großprojekt, in dem ich vor ca. 2 Jahren beteiligt war. Ca. 200 Beteiligte aus bestimmt 10 verschiedenen Firmen, als ich dazukam, waren schon mehr als zwei Jahre vorbei und der Projektfortschritt war objektiv als kritisch zu betrachten. Der Projektmanager vom Kunden war sehr versiert in klassischem Projektmanagement und darauf bedacht, jeden Entwickler mit seinem Arbeitspakt zu mikromanagen. Durch konsequente Schaffung von Transparenz und Selbstorganisation haben wir es in einem kleinen Team geschafft, trotzdem agil vorzugehen und in einem schwierigerem Umfeld Ergebnisse zu schaffen. Gerade die geschaffene Transparenz beeindruckte dabei.

Ich bin überzeugt, dass das der richtige Weg ist. Und eigentlich steht es ja schon im agilen Manifest: „Menschen und Interaktionen sind wichtiger als Prozesse und Werkzeuge“. Statt also dogmatische Diskussionen über die Anwendung von Scrum oder Test Driven Development zu führen, sollten wir uns darauf konzentrieren, wie wir dem Kunden oder unserer Firma wirklich weiterhelfen können.

Lernt von anderen

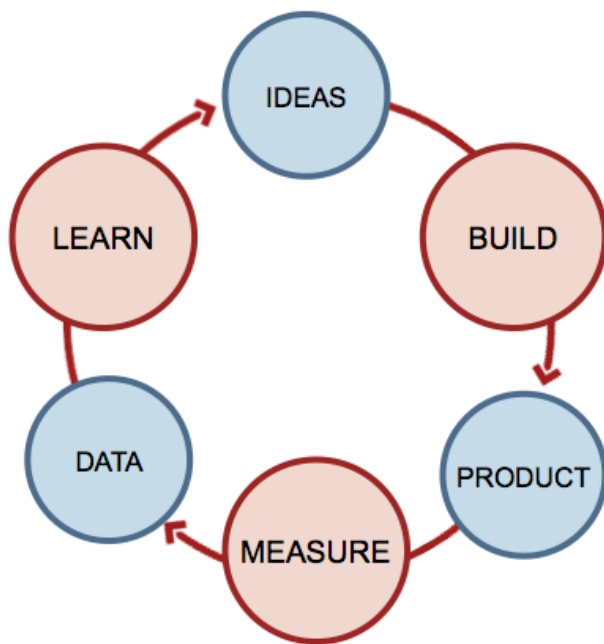
Das klingt alles recht abstrakt. Daher hilft ein Blick über den Tellerrand weiter.

Ein hervorragendes Beispiel ist Toyota: Mitte der 90er Jahre half Toyota, die Automobilindustrie zu revolutionieren. In der Folge etablierte sich Toyota als weltweite Marke. Viel wichtiger ist aber, was darüber hinaus passierte. Nicht nur beeinflusste Toyota die westeuropäischen Konkurrenten. Vielmehr entwickelte sich ein komplett neuer Managementansatz, der stabile Prozesse und konsequentes Qualitätsbewusstsein verband. Unter dem Schlagwort „Lean Management“ oder „Lean Thinking“ erhielt diese Denkweise Eingang in fast alle Branchen.

Im Zentrum von Lean Thinking stehen drei Thesen:

1. Eliminate waste, eigentlich eine inkonsequente Übersetzung des japanischen Begriffes „muda“. „muda“ steht nicht für bedingungslose Kosteneffizienz, z.B. durch Entlassung von Personal, sondern für Verschwendung, die in der Zusammenarbeit mehrerer Menschen auftaucht. Zum Beispiel überflüssige Arbeit an Ideen, die ein Kunde nicht braucht, Dokumentation die niemand liest, oder Meetings, die zu keinen Ergebnissen führen.
2. Continuous improvement, also das Streben nach Verbesserung. Damit einher geht Raum für Reflexion und für Verbesserung, frei nach der schönen Geschichte „Ich habe keine Zeit, meine Axt zu schärfen, ich muss Bäume fällen“.
3. Involve everyone. Ähnlich wie in der oben genannten Wandlung des Softwareentwicklers geht es darum, jeden am Verbesserungsprozess zu beteiligen, nicht nur das Management. Vielmehr wissen Fließbandarbeiter oft viel besser, wie Qualität verbessert werden kann.

Bemerkenswert ist, dass sich diese Ideen seit kurzer Zeit auch in der Start-Up Szene wiederfinden. Hinter dem Begriff „Lean Startup“ [4] versteckt sich eine Methode, die zwei klassische Formen von „muda“ in Gründungen adressiert – die zeitintensive Erstellung eines detaillierten Business Plans, der verzweifelt versucht, in die Zukunft zu sehen, sowie das kostenintensive Erstellen eines Produktes, für das kein Markt existiert. Es ist kein Zufall, dass agile Softwareentwicklung hier einen Platz als Methode in der „Build“-Phase findet.



Das letzte Beispiel, was ich bringen möchte, ist die Open Source Bewegung. Auch davon kann die agile Welt eine Menge lernen. Bemerkenswert finde ich vor allem die Verbindung von wirtschaftlichem Interesse der Firmen und technologischem Interesse der Entwickler. Auf wirtschaftlicher Ebene steht der Begriff open dafür, konsequent zwischen Firmen zu teilen. Nicht, weil Firmen auf einmal altruistisch geworden sind. Sondern weil es rein wirtschaftliche Interessen dafür gibt, z.B. Synergieeffekte durch gemeinsame Entwicklung und Standards, das gemeinsame Erschließen eines neuen Marktes (z.B. Android) oder die Positionierung als technischer Vorreiter, die für die Reputation bei Kunden und Bewerbern Nutzen bringt.

Auf der technischen Ebene bringt es den beteiligten Entwicklern nicht nur persönlichen Erfolg, sondern vor allem Lerneffekte durch die offene Zusammenarbeit. Natürlich spielt auch eine Rolle, durch seinen Zeitaufwand etwas Relevantes beitragen zu können.

Agil, Lean und Open

Hauptbotschaft, die ich heute loswerden möchte, sind zwei Appelle:

- Liebe Entwickler, nutzt die Chance, durch ein höheres Verständnis von der Welt um euch herum die agilen Ideen noch konsequenter in die Firmen hereinzutragen
- Liebe Manager, lernt von der agilen Softwareentwicklung und von Beispielen wie Toyota und bietet damit die Plattform dafür, die Business Agilität eures Unternehmens zu erhöhen, ohne die Menschen zu vergessen.

Ich bin der Überzeugung, dass sich die Werte und Prinzipien von Agil, Lean und Open auf drei verschiedenen Ebenen kombinieren lassen:

- Auf Organisationsebene durch die Kombination von Flexibilität, Kosteneffizienz und wirtschaftlicher Offenheit, z.B. für sinnvolle Kollaboration mit anderen Unternehmen
- Auf Managementebene durch eine Kultur des Lernens und der Reflexion, durch ein hohes Qualitätsbewusstsein und eine hohe Transparenz

- Auf SW-Entwicklungsebene durch die Fokussierung auf Werte und Prinzipien, aber auch auf technische Exzellenz, anstatt auf die dogmatische Diskussion über die richtige oder falsche Methode.

Referenzen

- [1] 8th Annual State Of Agile Survey (2013) - <http://www.versionone.com/pdf/2013-state-of-agile-survey.pdf>
- [2] Status Quo Agile (2012) - <http://www.komus.de/fileadmin/downloads/public/2012-07-Status-Quo-Agile-Studien-Hilights.pdf>
- [3] Jason Bloomberg – The Agile Architecture Revolution (2013)
- [4] Eric Ries – Lean Startup (2011)

Kontaktadresse:

Dominik Rose
InterFace AG
Am Mühlenberg 19
D-47051 Duisburg

Telefon: +49 (0) 151 52743916
Fax: +49 (0) 89 6104985
E-Mail: dominik.rose@interface-ag.de
Internet: www.interface-ag.de