



□ Dr. Thorsten Keuler

(thorsten.keuler@iese.fraunhofer.de)  
 arbeitet am Fraunhofer Institut für Experimentelles Software Engineering (IESE)  
 und verantwortet dort Forschungs- und Projektarbeiten im Bereich Software und  
 Systemarchitekturen. Sein Interesse gilt insbesondere der Skalierbarkeit von agilen  
 Entwicklungsmethoden im Kontext verteilter Entwicklung komplexer Systeme.

## objektspektrum themenspecial: agility

### Agilität – alles Auslegungssache?

Mittlerweile sind viele Unternehmen dazu übergegangen, ihre Software „agil“ zu entwickeln. Die damit verbundenen Erwartungen sind vor allem ein frühes Kundenfeedback sowie eine kürzere „Time-to-Market“ durch entsprechend schnelle Release-Zyklen. Hinter den Kulissen finden sich jedoch oftmals Probleme, sodass die erwarteten Vorteile agiler Softwareentwicklung nicht oder nur teilweise zum Tragen kommen. Gründe dafür liegen selten an technischen Hürden, sondern oftmals in einer zu oberflächlichen Interpretation der Philosophie von agiler Softwareentwicklung und den damit zusammenhängenden Kernzielen: Planung und Analyse auf ein sinnvolles Maß zu reduzieren und zugleich unnötige Arbeiten zu vermeiden. Was sinnvoll und was unnötig ist, stellt dabei eine der wesentlichen Herausforderungen dar.

#### Das Agile Manifest – mehr als eine Ausrede

Rückblick: Zu Anfang des Jahrtausends ist das Agile Manifest als Reaktion auf die damals eher schwergewichtigen Entwicklungsprozesse entstanden. Um einen Gegenpol zu den etablierten Praktiken der industriellen Softwareentwicklung zu errichten, wurden im Agilen Manifest eine Reihe von Werten eingeführt. Hauptaussage des Manifests war dabei die Rückbesinnung auf das Wesentliche: Die effiziente Produktion von (guter) Software.

Dieses Wertesystem des Agilen Manifests spielt seit jeher bei der Einführung von agilen Methoden eine wichtige Rolle. So ist ein oft beobachtetes Phänomen bei der Umstellung auf agile Vorgehensweisen eine bestenfalls fragwürdige Adaption des agilen Wertesystems. Einstellungen wie „Wir sind agil, wir dokumentieren nicht“ wer-

den kurzerhand mit dem Agilen Manifest begründet: „Working software over comprehensive documentation“. Ein weiterer Klassiker ist die Interpretation von „Responding to change over following a plan“: „Wir sind agil, wir planen nicht“.

#### Freiheit = Chaos?

Wird die Softwareentwicklung umgestellt auf agile Methoden, so hat dies in der Praxis insbesondere für das Entwicklungsteam mehr Freiheiten zur Folge. Dabei liefert das Agile Manifest mit Grundsätzen wie „Individuals and Interactions over Processes and Tools“ alle notwendigen Argumente für entsprechende Freiheiten in der Entwicklung.

War man bislang mit eindeutigen Spezifikationsdokumenten und klaren Übergabepunkten in ein Entwicklungskorsett gezwängt, fühlen sich viele Entwickler wie

befreit, ja teilweise sogar enthusiastisch, wenn auf agile Entwicklung umgestellt wird. Nikolay Entin berichtet in seinem Artikel über Praxiserfahrungen bei der Umstellung auf Scrum.

Ist die erste Euphorie infolge der Umstellung verflogen, zeigt sich schnell, dass die Herausforderungen sowohl an die „neuen“ Rollen, wie „Product Owner“ oder „Scrum Master“, als auch an die klassischen Entwickler durchaus enorm sind. Auf der einen Seite findet man einen wesentlichen Grund darin, dass die Informationen aus den mittlerweile nicht mehr existenten Dokumenten durch Kommunikation und entsprechender Werkzeugunterstützung kompensiert werden müssen.

Sven Hubert und Stefan Rauch zeigen in ihrem Beitrag, wie eine integrierte Werkzeugkette mit gemeinsamer, zentraler Datenbasis aussehen kann. Dr. Hans-

Joachim Pross und Werner Schoepe beschreiben zum Thema Werkzeuge die Entstehung der Jazz-Plattform bei IBM, welche die Basis für die teambasierte Softwareentwicklung Rational Team Concert war.

Neben den Herausforderungen an Kommunikation und Werkzeugunterstützung fehlt es aufgrund der neu gewonnenen Freiheit an entsprechend konkreten Vorgaben in Bezug auf die Umsetzung von bestimmten Entwicklungsaktivitäten. In diesem Zusammenhang zeigt **Raphael Auf der Maur** in seinem Beitrag, wie sich klassisches Requirements Engineering in Scrum einfügen kann. **Silke Geisen** und **Baris Güldali** stellen ein Konzept von agilen Testtypen und die dabei durchzuführenden Testaktivitäten vor.

### **Ich denke, also bin ich ... „agil“**

Da es sich bei der Umstellung auf agile Entwicklung um Änderungen handelt, die sich in der Regel in neuen Organisationseinheiten und Rollen zeigen, ist der Haupttreiber solcher Änderungen oft das Management. Eine Grundproblematik dabei ist, dass eine erfolgreiche Umstellung auf agile Entwicklung nicht „per Dekret“ erfolgen kann.

Man ist nicht „agil“ durch die Einführung neuer Rollen und die Anwendung von agilen Techniken. Agilität ist eine extern wahrnehmbare Eigenschaft, die man sich

erarbeiten muss. **Dr. Dirk Muthig** beschäftigt sich entsprechend mit der Tatsache, dass Agilität immer bezüglich einer Schnittstelle, beispielsweise zwischen Kunde und der Entwicklungsorganisation, betrachtet werden muss. **Jürgen Pilz** beschreibt in diesem Zusammenhang, wie eine Basis für agile Entwicklung geschaffen werden kann.

### **Agile Werte leben**

Es hat sich gezeigt, dass agile Softwareentwicklung mehr als die Summe von agilen Techniken ist. Um erfolgreich agil zu entwickeln, muss man als Organisation die agilen Werte verinnerlichen. Dies bedarf in erster Linie einer Umstellung des Selbstverständnisses der Rollen in der Entwicklung. Effektive agile Entwicklung erfordert zunächst die Bereitschaft zum Umdenken und vor allem Disziplin.

Agil heißt aber nicht nur Umdenken für die Entwickler: auch das Management muss entsprechend der agilen Werte die Führungsphilosophie auf den Prüfstand stellen. Ungeachtet aller potenziellen Unwägbarkeiten auf den technischen Ebenen ist der Faktor Mensch eine ausschlaggebende und zugleich schwer kalkulierbare Einflussgröße für den erfolgreichen Umstieg auf agile Softwareentwicklung.

Neben dieser Erkenntnis ist es für eine realistische Erwartungshaltung essenziell, zu akzeptieren, dass agile Werte nicht von heute auf morgen verinnerlicht werden

können – sowohl in der Entwicklung als auch im Management. Wie wichtig eine Verankerung des agilen Mindsets in der Praxis sein kann, beschreibt **Christian Popp** in seinem Beitrag.

### **Ich weiß, dass ich nichts weiß**

„Prognosen sind äußerst schwierig, vor allem wenn sie die Zukunft betreffen“. Es sollte mittlerweile jedem klar sein, dass Pläne gemacht werden, damit sie geändert werden können. Kombiniert man dieses Erkenntnis mit der Tatsache, dass Softwareentwicklung nur dann selbstorganisiert funktionieren kann, wenn alle Beteiligten eine gemeinsame Vision des Systems haben, so kommt man vielleicht zu dem Schluss, dass die Wahrheit wie so oft in der Mitte liegt: nämlich zwischen sinnvoller Planung und effizienter Umsetzung.

**Joscha Jenni**, **Maurice Perriard** und **Silvio Wandfluh** zeigen, wie man erfolgreich CMMI und agile Entwicklung kombinieren kann. Aus einem anderen Blickwinkel beleuchtet **Dirk M. Sohn**, warum sich agiles Vorgehen und Personaleinsatzplanung zusammenraufen müssen.

Ich wünsche Ihnen viel Vergnügen beim Lesen dieses Online Themenspecials!

*Dr. Thorsten Keller*