



□ Dr. Thorsten Keuler

(KeulerThorsten@JohnDeere.com)

Dr. Thorsten Keuler arbeitet seit diesem Jahr bei John Deere und ist dort verantwortlich für Strategien und Maßnahmen zur nachhaltigen Sicherung der Softwarequalität in der Displayentwicklung. Zuvor hat er zehn Jahre am Fraunhofer Institut für Experimentelles Software Engineering (IESE) in Kaiserslautern gearbeitet und dort vorwiegend Industrieprojekte und Schulungen im Bereich System- und Softwarearchitekturen geleitet. Sein Interesse gilt insbesondere der Skalierbarkeit von agilen Entwicklungsmethoden im Kontext verteilter Entwicklung komplexer Systeme.

## objektspektrum themenspecial: agility

### Agilität – Spagat zwischen Zweck und Selbstzweck

Schaut man sich aktuelle Entwicklerkonferenzen an, könnte man schnell den Eindruck bekommen, dass Unternehmen kaum mehr erfolgreich sein können, ohne Software agil zu entwickeln. Es scheint sogar fast so zu sein, dass Entwicklungsleiter in Softwareunternehmen gute Gründe haben müssen, die Entwicklung nicht auf „agil“ umzustellen. In erster Linie sind es oftmals Schwierigkeiten bei der Vermarktung der Produkte, die das Management über Änderungen in der Softwareentwicklung nachdenken lässt. Anders ausgedrückt: Die Softwareentwicklung muss kosteneffektiver werden, damit man sich über den Preis am Markt behaupten kann. Schnell wird übersehen, dass neben dem Preis noch ein anderer Faktor über Marktanteile entscheidet: Innovation.

Wenn man so will, ist die Stellschraube „Preis“ der passive Weg, das Geschäft anzukurbeln. Dabei ist die Strategie über entsprechend hohe „Stückzahlen“ den gewünschten Gewinn zu erzielen. Die Stellschraube „Innovation“ hingegen zielt primär auf Qualität ab. Sie ist der aktive Weg, um sich über Alleinstellungsmerkmale von Mitbewerbern zu differenzieren. Im Artikel „Wie agile Methoden Innovationen in Unternehmen unterstützen“ beschreiben **Alain C. Boss** und **Oliver Gabor**, wie agile Methoden dabei helfen können, in fremde Märkte zu expandieren und die Zukunft des Unternehmens langfristig zu sichern. Geht man der Frage nach, was die Gründe für fehlende Innovationskraft in Unternehmen sind, so stößt man meist auf die folgenden Antworten:

- Die Dauer der Softwareentwicklung bis zur Auslieferung ist zu lang, damit kommen neue Features nicht schnell genug auf den Markt.
- Die Entwickler haben kaum Freiräume, um an Innovationsthemen zu arbeiten, da sie sich voll und ganz auf Change Requests, Integration, Tests, Bug Fixing usw. konzentrieren müssen.

Grundsätzlich bietet agile Softwareentwicklung durchaus das Potenzial, besser und schneller Software entwickeln zu können. Es hat sich in der Praxis jedoch bereits vielfach gezeigt, dass alleine die Tatsache, auf agile Praktiken umzuschwenken, die Erreichung der Geschäftsziele nicht garantieren kann. Im Artikel

„Scrum für Manager: Warum lohnt sich Agilität aus betriebswirtschaftlicher Sicht?“ beschreiben **Thomas Fischer**, **Thomas Schissler** und **Uwe Baumann** die betriebswirtschaftlichen Hintergründe agiler Softwareentwicklung und erläutern, wie man sich auf die Transformation zu mehr Agilität vorbereiten kann.

Die Gefahr liegt potenziell in einer zu oberflächlichen Betrachtung der Auswirkungen bei einer Umstellung auf agile Entwicklung. Zu schnell ist das Management versucht, eine Veränderung hin zu agiler Entwicklung zu erwägen, mit einem bunten Strauß an unrealistischen Erwartungen. Dabei ist gerade den Entscheidungsträgern im Management oftmals nicht bewusst, dass auch sie Änderungen unterworfen sind. Insbesondere die Bereiche

Personal- und Führungsinstrumente sollten auf den Prüfstand gestellt und entsprechend agiler Prinzipien und Wertevorstellungen angepasst werden. Zu diesem Thema beschreibt **André Häusling** im Artikel „Die Relevanz agiler Personal- und Führungsinstrumente: Agile Führung als entscheidende Erfolgskomponente“, inwiefern sich die Rolle der Führungskraft im agilen Kontext verändert und welche Auswirkungen dieses neue Verständnis auf die Umsetzung verschiedener Personal- und Führungsinstrumente hat.

Bei Unternehmen, die bereits auf agile Entwicklung umgestellt haben, sind natürlich nicht alle Probleme automatisch gelöst. Neben den extern sichtbaren Symptomen am Markt zeigt sich eine Umstellung auf agile Entwicklung auch oftmals intern als vermeintliche Erschwernis. So ist es beispielsweise nicht ungewöhnlich, dass es zu Verzögerungen im aktuellen Produktrelease kommt. Aufgrund anfänglicher Unsicherheiten kommt es auch oftmals zu mehr Defekten, der Aufwand für Nacharbeiten und Bug-Fixes steigt an, und der Informationsaustausch zwischen den Entwicklerteams ist schlechter als vorher. Die konkrete Ausgestaltung solcher internen Symptome ist vielfältig. In diesem Zusammenhang beschreibt **Harald Müller** im Artikel „Agilität - Allheilmittel oder gezielte Medikation?“ Erkenntnisse, die Entscheidungsträger über die Analogie zur Medizin gewinnen können, und erläutert die Grenzen von Agilität bei der Betrachtung als Allheilmittel.

Allen Symptomen ist in der Regel gemein, dass sie durch eine halbherzige Einführung von agilen Methoden und Praktiken entstanden sind. Wenn beispielsweise nur ein Teil der Wertschöpfungskette auf agile Entwicklung umgestellt wird, dann sind die gewünschten Effizienzgewinne entsprechend gering, da es immer noch Flaschenhälse gibt, die nicht mit einer agilen Vorgehensweise reibungsfrei zusammenspielen. Ziel einer Umstellung sollte

daher immer eine „Ende-zu-Ende“-Agilität sein. Das heißt, man ist als Unternehmen nur so agil wie die langsamste Einheit in der gesamten Wertschöpfungskette. Im Artikel „Agilität zu Ende gedacht – von der Entwicklung bis zum Betrieb“ zeigen **Dr. Hans-Joachim Pross**, **Florian Georg** und **Dr. Henning Sternkicker**, warum nicht nur Entwicklerteams agil sein sollen, sondern auch der Betrieb.

Selbst wenn man nicht alles von heute auf morgen umstellen kann, so sollte man sich stets der „Ende-zu-Ende“-Agilität als Zielstellung bewusst sein. Versucht man bestehende Prozesse schrittweise umzubauen, so empfiehlt es sich, die Prozessschritte zu priorisieren, die den größten Nutzen für einen Teilbereich der Entwicklung versprechen. Dazu illustrieren **Thomas Rümmler** und **Stefan Mieth** im Artikel „Agile Entwicklung mit dem TFS und dem CMMI-Template“ am Beispiel des Team Foundation Servers, wie man zugunsten der Informationsvielfalt das CMMI-Process-Template des TFS verwenden und dabei trotzdem einen agilen Entwicklungsprozess leben kann.

Spätestens, wenn agile Ansätze organisationsweit ausgerollt werden, zeigt sich, wie gut die Mitarbeiter auf die Umstellung vorbereitet wurden und wie tief sich die agilen Denkweisen bereits unterbewusst manifestiert haben. In der Praxis kommt es nämlich immer noch sehr häufig zu Problemen, wenn sich Fragen stellen, wie zum Beispiel „Wie schneide ich am besten meine Teams?“ oder „Wie organisiere und koordiniere ich die Arbeit innerhalb und unter den Teams?“. **Dr. Thorsten Janning** gibt zu diesen und weiteren Fragen pragmatische Antworten in seinem Artikel „Scaling Agility: Wie führt man 100 agile Teams?“ und zeigt zudem die Grenzen von großen Organisationen auf.

Da jeder Entwicklungsprozess mit dem Einsatz und der authentischen Hingabe eines jeden Einzelnen steht oder fällt, gilt dies natürlich auch für die Einführung von

agilen Methoden. Sobald Agilität als Selbstzweck, also rein dogmatisch, interpretiert und gelebt wird, ist der eigentliche Grundgedanke von agiler Softwareentwicklung bereits ad absurdum geführt. Dinge zu tun, ohne sie zu hinterfragen, weil sie „irgendwo stehen“, ist wenig zielführend und sorgt auch nicht dafür, dass sich eine angemessene Kultur entwickeln kann. Auf der anderen Seite sollte man sich davor hüten zu glauben, man sei agil, nur weil man bestimmte Dinge tut, die wiederum „irgendwo stehen“. Man kann beispielsweise so tun, als wüsste man nicht, welche Anforderungen vom Kunden kommen, auch wenn man schon zigfach erfolgreich Dutzende Automodelle auf den Markt gebracht hat. Es kann ja niemand damit rechnen, dass ein Auto neben der reinen Fahrtrichtungsanzeige auch ein „Warnblink-Feature“ braucht. Dadurch entsteht die selbsterfüllende Prophezeiung, dass sich die Anforderungen tatsächlich fast täglich ändern, was ja schließlich auch ein erklärtes Ziel agiler Vorgehensweisen ist: Embrace Change!

Manchmal werden existierende Prozesse und Vorgehensweisen aber auch einfach nur umbenannt, sodass man nun eine „Retrospective“ macht, anstatt einer „Projektnachbesprechung“. Inhalt, Ablauf und Ergebnis dieser Retros haben sich im Vergleich zum etwas langweiliger klingenden Vorgänger leider nicht geändert. Agilität darf nicht zum reinen Selbstzweck verkommen. Agilität ist Mittel zum Zweck, und den Zweck dürfen wir nicht aus den Augen verlieren: Schnellere Auslieferung von innovativer und hochqualitativer Software. ■

Damit wünsche ich Ihnen nun viel Spaß beim Lesen dieses Online Themenspecials!

Ihr

*Dr. Thorsten Keuler*