



□ Dr. Thorsten Keuler

(thorsten.keuler@iese.fraunhofer.de)

arbeitet am Fraunhofer Institut für Experimentelles Software Engineering (IESE) und leitet dort die Abteilung „Embedded Software Developments“ mit Schwerpunkten im Bereich Software- und Systemarchitekturen. Sein Interesse gilt insbesondere der Skalierbarkeit von agilen Entwicklungsmethoden im Kontext verteilter Entwicklung komplexer Systeme.

objektspektrum themenspecial: agility

Die agile Adoleszenz

Im Universum der agilen Softwareentwicklung gibt es seit geraumer Zeit eine ganze Reihe von Weiterentwicklungen und Anpassungen, um die positiven Effekte agiler Softwareentwicklung auch in komplexeren Umgebungen erzielen zu können. Stichworte, die hier immer wieder fallen, sind „Agile@Scale“ und „Disciplined Agile Delivery“. Das heißt nicht nur, dass agile Methoden verbessert und optimiert werden, sondern vor allem eins: Den „Hype“-Status agiler Entwicklung kann man getrost zu den Akten legen, agile Entwicklung hat Einzug in die Praxis gehalten. Nichtsdestotrotz scheint die agile Softwareentwicklung mitten in einer Phase der Selbstfindung zu stecken. In der industriellen Praxis trifft man bei agilen Teams teilweise auf Verhaltensweisen, wie man sie sonst nur von pubertierenden Teenagern kennt: Alles wird in Frage gestellt, man testet die gesetzten Grenzen aus, man muss herausfinden, wie weit man gehen kann, testen, wie strikt die Regeln wirklich definiert sind, usw. In einem Satz: Ein Hauch von Revolution liegt in der Luft. Auch wenn es hier und da unterschiedliche Auffassungen in Bezug auf gewisse agile Philosophien gibt, so ist man sich doch nach wie vor in einem Punkt einig: Nichts ist so beständig wie die Änderung.

Kampf gegen die Zeit

Bei Änderungen kommt dem Backlog eine zentrale Rolle zu, da hier der Einstiegspunkt in die Anforderungswelt definiert ist. Was passieren kann, wenn man das Backlog nicht gewissenhaft pflegt, diskutieren **Silvio Wandfluh** und **Joscha Jenni** in ihrem Beitrag „Interview mit dem Product Backlog“.

Als Produktverantwortlicher oder auch als Entwickler würde man sich vielleicht wünschen, dass sich die Software immer sofort entsprechend anpasst und dass alle Änderungen qualitätsgesichert umgesetzt sind. Agile Softwareentwicklung versucht diesem Wunsch möglichst nahezukommen.

Dadurch, dass in kurzen Zyklen gearbeitet wird, entsteht der Anschein einer

kontinuierlichen Änderbarkeit. Gerade wenn es darum geht, auf kurzfristige Änderungen ein Nutzer-Feedback zu bekommen, werden häufig Software-Prototypen eingesetzt.

Software-Prototypen liefern zwar schnelle Ergebnisse, aber anspruchsvolle, hochqualitative Produktentwicklung erfordert darüber hinaus eine gewisse Kontinuität „hinter den Kulissen“. **Dr. Dirk Muthig** illustriert in seinem Artikel, „Solide Produktweiterentwicklung im Windschatten agiler Vorausentwicklung“, wie man Prototypen so einsetzen kann, dass man schnelle Ergebnisse im Sinne des Kunden-Feedbacks bekommt und gleichzeitig den trügerischen Anschein einer produktiv einsetzbaren Softwarelösung gar nicht erst aufkommen lässt.

Im Angesicht einer „Allzeit-Bereit (für Änderungen)“-Erwartungshaltung entsteht in der Realität aber schnell ein gewisses Spannungsfeld. Als Entwickler fragt man sich oft zu Recht: Worauf kann man sich festlegen? Was wird sich wohl wieder ändern? Wer sagt mir, dass ich nicht zu viel oder zu wenig geplant habe?

Wirft man zusätzlich noch verteilte Entwicklung in den Ring, so kommen unweigerlich weitere Fragen auf: Wer hat dieses Feature implementiert? Gibt es bereits einen Service, der das tut, was ich gerade brauche? In diesem Zusammenhang beschreibt **Stefan Glase** mit seinem Artikel „Der Werkzeugkoffer für verteilte agile Teams“, wie man mit persistenten Chatrooms und ausgewählten Tools im Falle von verteilten agilen Teams erfolgreich

sein kann. Daneben zeigt Frank Maar im Beitrag „Der moderne Application Lifecycle“, wie man mit einer integrierten Application Lifecycle Management-Plattform agile Teams unterstützen kann.

Gesunde Anpassung ist eine Frage der Geschwindigkeit

Aus der Evolutionstheorie kennt man vielleicht den Satz „It is not the strongest of the species that survives, nor the most intelligent, but rather the one most adaptable to change.“ Evolution wird dabei als ein Prozess verstanden, der erst über Generationen hinweg Wirkung zeigt. Evolution hat demnach mit Geschwindigkeit so viel gemein, wie der Papst mit der Urknalltheorie. Die Natur passt benötigte Fähigkeiten den Lebensumständen entsprechend an – Dinge, die nicht nützlich sind, werden einfach abgeschafft.

Auch im Bereich der Anpassung und Weiterentwicklung von Software ist Evolution ein gängiger Begriff. Was man dabei jedoch schnell aus den Augen verliert, ist die Tatsache, dass Evolution mit kurzfristigen Anpassungen nicht viel zu tun hat. So hört man beispielsweise vielfach den Satz „Das ist bei uns historisch gewachsen“, evolutionär sozusagen. Bei genauem Hinsehen findet man dann aber schnell heraus, dass die historisch gewachsenen Umstände doch erst vier Wochen alt sind. Die Wahrnehmung, was evolutionär und was eine schnelle Änderung ist, scheint damit immer im Auge des Betrachters zu liegen.

Gerade bei der Umstellung auf agile Entwicklungsmethoden spielt neben den Änderungen im Kontext von Produktanforderungen ein evolutionärer Änderungsprozess eine sehr wichtige Rolle: die Änderung der Unternehmenskultur. Viele Entwickler die eine Umstellung auf agile Vorgehensweisen mitgemacht haben, können bestätigen, dass die schwierigste Änderung die der Unternehmenskultur war.

Thomas Rümmler beschäftigt sich dazu in dem Beitrag „Agilität in der Produktentwicklung: Wie agile Softwareentwicklung und klassisches Produktmanagement zusammenpassen“ mit der Herausforderung, agiles Vorgehen mit strikt durchgeplanten Produktionsprozessen zusammenzubringen. Wie auch immer der Begriff der Evolution verwendet wird, es muss klar sein, dass die Unternehmenskultur nicht von heute auf morgen umgestellt werden kann. Man sollte sich immer bewusst sein: Wenn man zu schnell zu viel

ändert, dann führt das meist nur zu Problemen.

Kulturwandel kommt von innen

Sich neuen Gegebenheiten anzupassen, ist nicht nur sinnvoll, sondern teilweise sogar überlebenswichtig. Dies gilt insbesondere für den Bereich der Unternehmenskultur. Warum ist das wichtig? Die Agilität einer Entwicklungsmannschaft kann nur nachhaltig sein, wenn eine entsprechende Unternehmenskultur etabliert ist. Kultur bedeutet hier, dass alle die agilen Werte verinnerlicht haben. Wichtig ist, dass alles aus innerer Überzeugung kommt, dass die Werte authentisch gelebt werden.

Dies erklärt auch, warum eine Änderung der Unternehmenskultur nicht „von extern“ oder „von oben“ herbeigeführt werden kann. Einen zielgerichteten Kulturwandel kann man als Unternehmen daher nur schwer steuern, geschweige denn kontrollieren. Dazu beschreibt Markus Gärtner in seinem Artikel „Firma, verändere dich“ das Konzept des „Veränderungsabdrucks“, welcher die Anpassbarkeit eines Unternehmens über die Zeit beschreibt.

Organisation, Architektur, Kompetenz – die Säulen einer agilen Unternehmenskultur

Auch wenn man als Unternehmen einen Kulturwandel nicht steuern oder kontrollieren kann, so kann man doch dafür sorgen, dass die Umgebung optimal das Entstehen einer Kultur ermöglicht, in der sich die Prinzipien agiler Entwicklung entfalten können. Dabei kann ein fruchtbarer Nährboden für eine Unternehmenskultur nur entstehen, wenn die Säulen *Organisation*, *Architektur* und *Kompetenz* im Unternehmen etabliert sind und in einem gesunden Gleichgewicht zueinanderstehen.

- **Organisation:** Das Management muss eine klar kommunizierte Überzeugung in Bezug auf agile Werte haben. Weniger erfolgreich ist es, wenn nur die Entwicklung auf agile Werte getrimmt wird, das Management aber in „alten Schubladen“ denkt. An dieser Stelle sei auch erwähnt, dass Prinzipien wie „Empower the Team“ auf Organisationsebene umgesetzt werden müssen, damit die „alte“ Organisationsstruktur bei der Entscheidungsfindung nicht zum Flaschenhals wird.
- **Architektur:** Die Softwarearchitektur spielt auch bei der agilen Planung und

Umsetzung von größeren Softwareprojekten eine entscheidende Rolle. Hier lautet die Devise: Planen nach Maß. Balthasar Weitzel und Mathias Schefke berichten in ihrem Artikel „Erosion von Agilität mit Epic-Architekturen erfolgreich bekämpfen“, wie man mithilfe von gut dosierten Architekturaktivitäten auch langfristig die Agilität als Teameigenschaft aufrechterhalten kann. Ohne eine gute Softwarearchitektur wird es schwer, die Ergebnisse von (verteilten) Teams ohne viel Aufwand zu einem konsistenten Gesamtprodukt zu integrieren. In diesem Kontext beleuchtet Jens Coldewey in seinem Beitrag „Architektur auf 3 m²“, wie Architektur in agilen Methoden verstanden und auch missverstanden wird.

- **Kompetenz:** Gerade bei der Einführung agiler Vorgehensweisen wird der Kompetenzfaktor leicht übersehen. Wer redet schon gerne über Kompetenzprofile? Niemand stellt sich gerne selbst die Frage, ob man die geeigneten Skills mitbringt, um den Job, den man schon seit 20 Jahren macht, jetzt auch noch machen kann. Oliver Röhrsheim beschreibt dazu in seinem Artikel „Wesentliche Rahmenbedingungen bei der Einführung agiler Entwicklungsmethoden“ Konzepte, wie man mithilfe eines agilen Moderators die Entwicklungsteams unterstützen kann.

Sind die drei Säulen nicht gut aufeinander abgestimmt oder gibt es ein gravierendes Ungleichgewicht zwischen ihnen, so wird es für Unternehmen sehr schwer bis unmöglich auf Dauer erfolgreich eine agile Kultur zu etablieren und diese auch durchzuhalten. Mein abschließender Appell richtet sich daher an alle Unternehmen, die an einem Kulturwandel im Sinne der agilen Entwicklung interessiert sind: „Sorgt einfach dafür, dass Agilisten in einer Umgebung ‚aufwachsen‘ können, die es ihnen ermöglicht, ihre jugendliche Unbeschwertheit im Alltag der Softwareproduktion zu bewahren.“ ■

Jetzt wünsche ich Ihnen viel Vergnügen beim Lesen dieses Online-Themenspecials!

Ihr

Dr. Thorsten Keuler