

## Der dornige Weg zum agilen Team, Teil 7: Abnahmen

# Verhinderer des Projekterfolges?

■ VON JIRI LUNDAK

Sie sind ein Relikt aus den Tagen des Wasserfall-Prozesses. Sie wurden dort zur Beurteilung des Erreichens von Meilensteinen verwendet und sollten dem Kunden die Sicherheit geben, dass sich ein Projekt auf sicherem Weg zum Erfolg befindet. Die Rede ist von so genannten Abnahmen. Der Kunde nahm ein Zwischenergebnis – oft nach mehrmaligem Hin und Her – ab, und erst nach diesem Meilenstein konnte mit dem nächsten Schritt im Entwicklungsprozess weitergefahren werden. Kunden schätzen diese Vorgehensweise. Aber ist es mit agilem Vorgehen vereinbar?

Haben wir schon einmal beobachtet, wie sich das Zauberwort „Abnahme“ auf die an einem Projekt beteiligten Personen auswirkt? Beim Kunden bricht oft Panik aus. „Was nicht in der abgenommenen Spezifikation drin ist, das werde ich nicht bekommen!“, überlegt er. Da denkt er sich alles Mögliche und Unmögliche aus, was am besten auch noch mit hineinkommt. Oder betrachten wir die Abnahme des implementierten Systems nach 20 Monaten Entwicklungszeit: Erst zu diesem Zeitpunkt geht dem Kunden im traditionell geführten Projekt auf, dass nun der letzte Moment ist, um gegen die Implementierung Einspruch zu erheben, und das wird in der Regel dann auch getan. Es wird all das Testen nachgeholt, das in den Monaten zuvor versäumt wurde, da das System ja noch nicht brauchbar und einsetzbar war. Das Resultat: Hunderte von neuen Anforderungen und Fehlern werden entdeckt, und es dauert noch Monate, bis das bereits „fertige“ System produktiv eingesetzt wird.

Und auf der Seite des Entwicklungsteams? Es wird gewartet, bis die Spezifikation abgenommen ist. Vorher etwas zu programmieren, würde ja bedeuten, vielleicht in die falsche Richtung zu laufen.

Selbstverständlich sitzt man nicht tatenlos herum, sondern feilt am Framework oder anderen Komponenten, die man höchstwahrscheinlich im Projekt einsetzen wird.

## Koexistenz

Im Vorgehen nach dem fast schon klassischen „iterativen Wasserfall“ [1] wirken sich Abnahmen meist verzögernd auf das Projekt aus, wie in Abbildung 1 dargestellt. Abnahmen, z.B. einer Spezifikation, führen oft zu unverhofften Verzögerungen, besonders in größeren Projekten, wo ganze Gremien darüber beraten, ob nun alle Details stimmen. Da oft nicht der Kunde selbst die Spezifika-

tion schreibt, sondern ein Analytiker des Dienstleisters, werden solche Dokumente meist nicht partizipativ erstellt [2], sondern in vielen Runden geschrieben und korrigiert und besprochen und wieder korrigiert, und das Ganze natürlich nur mit gelegentlichen Sitzungen und mit viel E-Mail-Verkehr. Das Problem dieses Vorgehens ist weitgehend bekannt: Die beteiligten Parteien versuchen, alles im Voraus durchzudenken, dies jedoch ohne die leiseste Ahnung, ob das System so überhaupt realisierbar ist oder nicht. Und solange man nicht Hand an die implementierte Software legen kann, kann auch niemand so richtig beurteilen, ob die Spezifikation tatsächlich vollständig ist.

## Der dornige Weg zum agilen Team – die Serie

- Warum agile Teams so schwer zu schaffen und am Leben zu erhalten sind
- Was ist kein agiles Team?
- Warum müssen Builds immer brechen?
- Selbstorganisation – Chaos oder Wunderwaffe?
- Was soll heißen: „Ich bin fertig“?
- Metriken: Ballast oder Notwendigkeit?
- Abnahmen: Verhinderer des Projekterfolges?
- Brauchen agile Teams agile Kunden?
- Gedeihen agile Teams nur in agilen Unternehmen?
- Agiles Vorgehen als rotes Tuch
- Nur die Besten sind für das agile Team gut genug
- Cross-funktionale Teams – eine Illusion?
- Wo bleibt der Manager im agilen Team?
- Agile Teams skalieren nicht!
- Wie erhalte ich ein agiles Team am Leben?

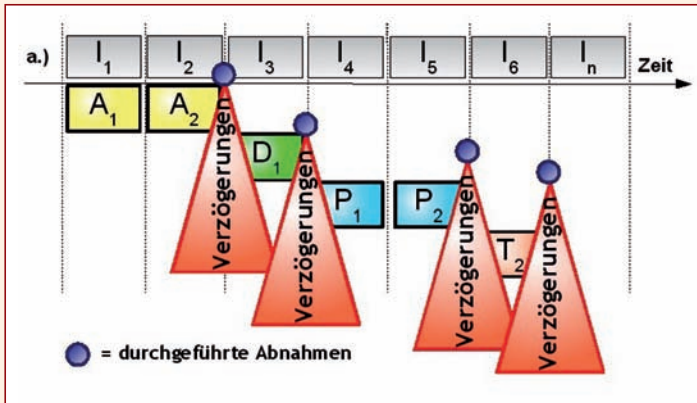


Abb. 1: Abnahmen im „iterativen Wasserfall“

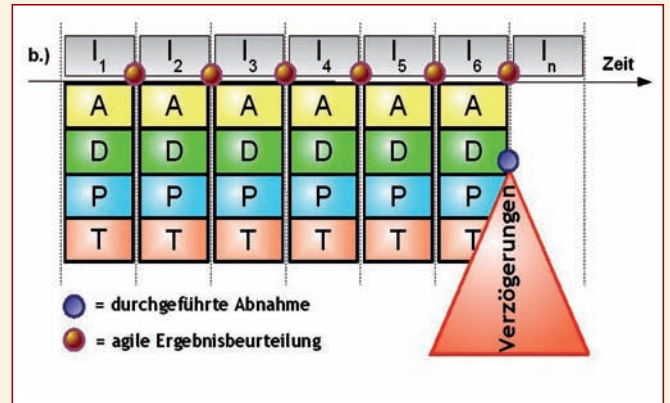


Abb. 2: „Abnahmen“ in einem agilen Projekt

Hier wirken Abnahmen wie Mauern zwischen den verschiedenen am Projekt beteiligten Personengruppen (den Kundenvertretern, den Entwicklern, den Testern etc.). Die Tendenz geht in Richtung gegeneinander statt miteinander. Änderungen in den Anforderungen zum Zeitpunkt der Implementierungsphase sind nur durch schwerfälliges Change-Management in den Prozess einzuspeisen, oft wieder begleitet von Sitzungen, Diskus-

sionen und „Hin-und-her-Geschiebe“ der Verantwortung.

### Kollaboration

Im agilen Projekt darf es ein solches Nebeneinander nicht geben. Eigentlich sollte Zusammenarbeit groß geschrieben sein [3]. Wie aus Abbildung 2 ersichtlich, gibt es aber auch im agilen Projekt so etwas wie Abnahmen. Ich bezeichne diese jedoch lieber als Ergebnisbeurteilung. Der

Kunde beurteilt bei jedem Iterationsende die gelieferten und gezeigten Ergebnisse. Als Grundlage für die Beurteilung hat er das fertig gestellte Inkrement des Systems und nicht nur Papier.

In den meisten größeren (auch agilen) Projekten gibt es immer noch zum Schluss so etwas wie eine Endabnahme. Zu diesem Zeitpunkt möchte der Kunde oft ein System oder Teilsystem in seiner Ganzheit testen und komplexe Gesamtabläufe überprüfen.

Selbst wenn angeblich agil vorgegangen wird, droht hier der rote Keil der Verzögerungen. Warum kann dies der Fall sein?

Eines der Hauptprobleme dabei ist: Die Ergebnisbeurteilung ist zu flau! Was meine ich damit? Ein erlebtes Beispiel mag dies etwas illustrieren:

Das Team präsentiert am Ende der Iteration ein Zwischenergebnis. Weil es sich aber um ein Großprojekt mit komplexen Geschäftsabläufen und großer Datenmigration vom Host-Rechner handelt, die Applikation erst, wenn sie vollständig funktionsfähig ist, beim Kunden produktiv eingesetzt werden darf und das Deployment deshalb monatlich „nur“ in eine Test-Umgebung des Kunden stattfindet, nimmt der Kunde die Ergebnisbeurteilung nicht so ernst. Er bekommt zwar vom Team etwas präsentiert, aber er testet die Software nicht auf Herz und Nieren mit konkreten Geschäftsfällen, sondern gibt sich damit zufrieden, dass er einen oberflächlichen Eindruck gewinnt, ob die Benutzeroberfläche ungefähr die Geschäftsabläufe unterstützt, die er sich vorgestellt.

In diesem Fall wäre es wohl besser, dass dem Kunden gesagt würde, dass er das Inkrement jeweils zum Ende der Iteration „abnehmen“ muss. Ich habe zu Beginn des Artikels erwähnt, wie das Wort „Abnahme“ auf Kunden in der Regel wirkt. Mit einem engeren Einbeziehen des Kunden *während* der Iteration hätte man erreichen können, dass er nicht am Ende erst erfährt, in welche Richtung sich die Applikation entwickelt, sondern er hätte bereits eine genaue Vorstellung (genauso wie das ganze Team) davon, was am Ende der Iteration zu überprüfen ist, um sicher zu sein, dass die Applikation funktioniert. Dies wäre der Idealfall. Leider werden im Rahmen größerer Projekte verschiedene Argumente ins Feld geführt, warum dies nicht oder nur schwer durchführbar ist.

Eines der Probleme, das der Kunde anführt, betrifft die zusätzliche Belastung seiner Mitarbeiter, die neben ihrer Funktion als Domänenexperten im Projekt auch noch ihren täglichen Aufgaben innerhalb der Kundenorganisation nachkommen müssen. Dies führt dazu, dass diese Mitarbeiter zwar an den monatlichen Demos teilnehmen, aber während des Zeitraumes der Iteration nur sporadisch beigezogen wer-

den können, um Feedback zu geben. Hier ist das Team gefordert, kreative Lösungen zu finden, um den Kunden möglichst häufig ins Projektgeschehen einzubeziehen.

Ein weiteres Problem, das bewältigt werden muss, ist der Drang des Menschen, zu viele Dinge parallel lösen zu wollen. Wenn z.B. ein Geschäftsprozess sehr komplex ist und sehr viel Zeit braucht, bis er vollständig implementiert ist (z.B. mehrere Wochen), dann ist es nicht sinnvoll, diesen Geschäftsprozess einem einzigen Entwickler zuzuteilen, sodass dieser innerhalb einer Iteration nur einen unvollständigen Teil davon implementieren (geschweige denn testen) kann. Parallel dazu arbeiten andere Entwickler an anderen Geschäftsprozessen, die ebenfalls nur bruchstückhaft fertig gestellt werden können. Stattdessen sollten komplexe Abläufe so zerlegt werden, bis sie in eine Iteration passen und vom Team in Zusammenarbeit erledigt werden können. Nur so kann am Ende einer Iteration auch etwas „abgenommen“ werden.

Wir sehen somit eines klar: Im agilen Umfeld sind Abnahmen notwendig, allerdings haben sie eine ganz andere Bedeutung als im gewohnten Wasserfall-Ansatz. Dem Kunden wird die Gelegenheit geboten, ein Stück potenziell produktiv einsetzbarer Applikation zu beurteilen. Helfen wir ihm dabei, diese Gelegenheit richtig zu nutzen. Tun wir dies nicht, müssen wir uns nicht wundern, dass agile Projekte an denselben Problemen kranken wie traditionelle.



**Jiri Lundak** ist mit mehr als 20 Jahren Erfahrung ein Urgestein der Softwareentwicklung. Er arbeitet gegenwärtig als Entwicklungsleiter für Löwenfels Partner in der Schweiz. Er entwickelt als Architekt immer noch aktiv Software, hat aber erkannt, dass die meisten Probleme in Entwicklungsprojekten nicht technischer, sondern vielmehr menschlich-sozialer Natur sind. Als praktizierender Certified ScrumMaster lebt er die Rolle eines Vorkämpfers für agiles Verhalten innerhalb der Firmenorganisation. Kontakt: [jiri.lundak@loewenfels.ch](mailto:jiri.lundak@loewenfels.ch).

### ■ Links & Literatur

- [1] Jiri Lundak: Der iterative Wasserfall, in *Java Magazin* 8.2006
- [2] Scott W. Ambler: Agile Documentation: Strategies for Agile Software Development: [www.agilemodeling.com/essays/agileDocumentation.htm#EffectiveDocumentationHandoffs](http://www.agilemodeling.com/essays/agileDocumentation.htm#EffectiveDocumentationHandoffs)
- [3] Craig Larman, Philippe Kruchten, Kurt Bittner: How to Fail with the Rational Unified Process: Seven Steps to Pain and Suffering (S. 10): [agilealliancebeta.org/system/article/file/941/file.pdf](http://agilealliancebeta.org/system/article/file/941/file.pdf)

Anzeige