

AGILES MANIFEST VIII: EINEN PLAN VERFOLGEN

Wenn es eine „rote Liste bedrohter agiler Balancen“ gäbe, würde die Abwägung, „uns ist die Reaktion auf Änderungen wichtiger, als einen Plan zu verfolgen“, wohl den Spitzenplatz einnehmen. Mit keiner anderen Abwägung tun sich Teams und Management im Tagesgeschäft schwerer und nirgendwo sonst findet man – selbst bei manchem erfahrenen Berater und auch Trainer – noch einen solch starken Hang zu eher unagilem Vorgehen. Dabei hat sich das Thema agile Planung in den letzten zehn Jahren zu einem der mächtigsten Werkzeuge entwickelt, wenn man es richtig einsetzt.

Vor Kurzem setzte die Managementrunde eines Kunden eine „bessere Vorhersage-Treue“ auf den ersten Platz der Ziele für die Einführung agiler Techniken in ihrem Unternehmen. Vor zehn Jahren, als das Agile Manifest geschrieben wurde, hätte mich diese Priorisierung zu der Diskussion veranlasst, ob ein agiles Vorgehen für das Unternehmen überhaupt geeignet sei. Heute zählt dieses Ziel zu den eher leichteren Übungen für agile Teams – für mich ein Zeichen für den großen Fortschritt, der bei dem Thema in den letzten zehn Jahren gemacht wurde.

Als das Manifest geschrieben wurde, waren Pläne noch weitgehend ein Synonym für komplexe Ressourcen-Auslastungspläne, die als Gantt-Diagramme die Festplatten der Projektleiter belegten. Wenige Monate zuvor war das Buch „Planning Extreme Programming“ von Kent Beck und Martin Fowler erschienen (vgl. [Bec00]), das erstmals alternative Verfahren detaillierter beschrieb. Standard waren diese Verfahren damals noch keineswegs.

Klar war aber, dass die traditionellen Planungswerkzeuge wie Gantt-Diagramme mit den damals noch üblichen zwei- bis vierwöchigen Planungsrhythmen unvereinbar waren: Der Aufwand zur Erstellung war zu groß und die Ausgangslage auf Basis von User-Stories zu unsicher. Die Pläne waren obsolet, bevor sie aufgehängt waren. Nicht selten wurde daher die Werteabwägung als willkommene Befreiung von der Bürde des Planens gesehen – als Freibrief, einfach gar nicht mehr zu planen, denn schließlich würden Änderungen die Planung ohnehin sehr bald über den Haufen werfen.

In seinem 1999 erschienen Buch „Adaptive Software Development“ ging Jim Highsmith sogar so weit, den Begriff der Planung völlig abzuschaffen (vgl. [Hig00]): Das Team spekuliere zu Beginn der Iteration über deren möglichen Verlauf, kollaboriere während der Iteration und lerne dann aus der Reflexion (siehe Abbildung 1). Auch wenn sich Highsmiths Wortwahl nicht

durchgesetzt hat – das Verständnis über die Rolle agiler Planung blieb erhalten. Allerdings sind unsere Spekulationen mittlerweile um einiges besser geworden.

Stand der Kunst ist heute die *velocity*-basierte Planung aus dem eXtreme Programming. Diese folgt der Idee, die Kent Beck als „Yesterday’s Weather“ bezeichnet hat: Wenn man annimmt, dass man in der nächsten Iteration etwa gleich viel schafft wie in den letzten Iterationen, wird man kaum völlig daneben liegen. Mike Cohn hat den Ansatz 2005 noch ein wenig verfeinert und in der Scrum-Community populär gemacht, von wo aus er die derzeitige Verbreitung fand (vgl. [Coh06]). Kombiniert wird die Planung mit dem seit den 70er Jahren systematisch erforschten Potenzial selbst-organisierter Teams und immer mehr auch mit abstrakten Schätzverfahren, die der Gefahr der sich selbst erfüllenden Prophezeiung üblicher Zeitschätzungen entgegen wirken. Heute erlaubt diese Kombination bei eingespielten Teams auch mittel- bis langfristige Planung mit erstaunlicher Aussagekraft [Col11].

Warum schätze ich agile Planung dennoch als gefährdet ein? Gefahren kommen aus zwei Richtungen:

- Zum einen führen immer wieder Gewohnheiten aus dem traditionellen Management die Planungen der Teams ad absurdum.
- Zum anderen gibt es auch innerhalb der Teams Praktiken, die – falsch verstanden – dazu führen, dass die Teams doch eher einen Plan abarbeiten, als auf Änderungen zu reagieren.

Unagile Manager

Aber der Reihe nach: Viele Manager sind es gewohnt, Termine vorzugeben: „Ich habe dem Kunden zugesagt, dass wir morgen eine Lösung für diesen Fehler ausliefern“, ist eine typische Aussage. Gegen eine solche Aussage ist nichts einzuwenden, wenn die Fehlerursache bereits identifiziert und behoben und



Jens Coldewey

[E-Mail: jens.coldewey@it-agile.de]

ist Münchner Geschäftsführer der it-agile GmbH und Chefredakteur von OBJEKTspektrum. Agile Ansätze praktiziert er seit 1998.

das Ergebnis fertig getestet ist. Aber so ist die Aussage in den wenigsten Fällen gemeint. In der Regel lautet die Übersetzung: „Ich habe das dem Kunden zugesagt, weil das ja nicht so schwierig sein kann, deshalb strengt euch mal ein bisschen an, und wenn es nicht klappt, müsst ihr halt die Nacht durcharbeiten, damit ich mich beim Kunden nicht lächerlich mache oder vielleicht sogar Vertragsstrafen fällig werden.“

Was diese Manager nicht zur Kenntnis nehmen: Sie können nicht entscheiden, dass ein Fehler bis zu einem bestimmten Termin behoben ist! Nach keinem Verfahren dieser Welt kann man das entscheiden – ebenso wie kein Meteorologe entscheiden kann, wie morgen das Wetter wird. Selbst Bombendrohungen an den deutschen Wetterdienst bewirken meist kein besseres Wetter.

Man kann natürlich sehr wohl entscheiden, dass ein Team alle Kraft darauf verwenden soll, einen Fehler zu beheben. Dadurch fehlt aber die Kraft, sich anderen Themen zu widmen, die daher entsprechend später fertig gestellt werden. In der agilen Planung nennt man diese Entscheidung „Priorisierung“ – und sie ist das zentrale Steuerinstrument in agilen Vorhaben. Es ist nicht sinnvoll, erst eine Priorisierung vorzugeben und dann zu versuchen, daran vorbei doch noch mehr zu bekommen. Das ist in etwa so, als wolle man sein Haushaltsgeld aufstocken, indem man aus der eigenen Geldbörse Geld stiehlt. Es verschlechtert die Vorhersagbarkeit und führt häufig genug dazu, dass man weder das bekommt, was man ursprünglich oben auf der Prioritätenliste hatte, noch das neu hinein gepresste.

Natürlich kann man unzufrieden sein mit der Produktivität des Teams – und agile Planungsverfahren verschaffen schonungslos Offenheit über die eigene Produktivität. Die Vorstellung, ich könne die Produktivität einer ohnehin überlasteten „Maschine“ namens Team dadurch erhöhen, dass ich immer mehr vorne hinein presse, ist aber bestenfalls naiv. Stattdessen muss man in diesem Fall herausfinden, wo die wichtigsten ▶

Produktivitätsbremsen liegen, und diese Stück für Stück ausräumen. Das agile Werkzeug dafür ist die ernst gemeinte Retrospektive. Ironischerweise stellt sich bei den meisten Teams mit mangelnder Produktivität dann übermäßiger Druck als eine der wichtigsten Produktivitätsbremsen heraus: aktueller Druck, aber auch die Kollateralschäden des Drucks der letzten Jahre. Die Bereitschaft, sich selbst als wichtigen Teil des Problems zu sehen und das eigene Verhalten entsprechend zu verändern, ist aber nicht allen Managern gegeben.

Unagile Teams

Manche Teams brauchen allerdings nicht unbedingt Manager, die das Planungsverfahren nicht verstanden haben, um die Balance des agilen Manifests auszuhebeln – manche Teams schaffen das auch ganz allein. Jeder Hebel benötigt einen Ansatzpunkt. In diesem Falle ist es das in Scrum propagierte *Commitment* des Teams während der Sprint-Planung – oder, um genauer zu sein, eine Fehlinterpretation dieses Commitments.

Um das Commitment in Scrum zu verstehen, ist es hilfreich, sich Scrums Ursprünge in der Produktentwicklung in Erinnerung zu rufen. Häufig werden den Kunden dabei für die nächsten Releases bestimmte Features versprochen und die Kundenzufriedenheit hängt davon ab, ob diese Versprechen eingehalten werden. Die Idee des Commitments ist es nun, dass sich das Team darauf einswört, diese Kundenerwartungen zu erfüllen, ähnlich wie sich ein Rugby-Team darauf einswört, ein Spiel zu gewinnen. Wie selbst Gelegenheits-sportler allerdings bestätigen können, ist das keine Garantie, dass das Spiel auch wirklich gewonnen wird. Auch der ehrgeizigste Trainer wird selbst bei den höchsten bezahlten Profis einsehen müssen, dass diese nicht jedes Spiel gewinnen können – schließlich braucht jedes gewonnene Spiel auch einen Verlierer. Und noch in einem anderen Aspekt trägt die Sportmetapher: Das Team wird auf das Ziel eingeschworen, nicht auf einzelne Spielzüge, auch wenn gute Trainer mit den Spielern durchaus im Vorfeld Spielzüge diskutieren.

Das Pendant in Scrum zum Sieg ist das Sprint-Ziel, das Gegenstück zu den Spielzügen sind die User-Stories und die einzelnen Aufgaben. Das Sprint-Ziel ist also ein Versprechen an die Anwender, dass mit der Software dieses Sprints eines ihrer Probleme behoben sein wird. Das Sprint-Ziel besteht nicht darin, eine bestimmte

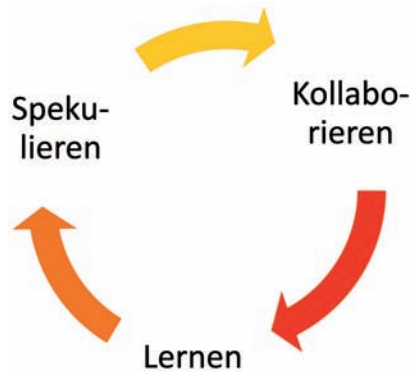


Abb. 1: Jim Highsmith verzichtete in seinem „Adaptive Software Development“ vollständig auf den Begriff „Planung“ und verwendete stattdessen den Ausdruck „Spekulation“. Die Terminologie trifft den Kern zwar besser, konnte sich aber dennoch nicht durchsetzen.

Anzahl von User-Stories umzusetzen. Der Unterschied ist entscheidend für die Balance zwischen Offenheit für Änderungen und Planverfolgung:

Nehmen wir an, das Sprint-Ziel wäre es, die Online-Bestellung von Büchern zu ermöglichen. In der Sprint-Planung hat man die Anbindung an das Bestellsystem des Großhändlers geschätzt und in die Sprint-Planung aufgenommen. Während der Realisierung stellt sich diese Anbindung nun als deutlich aufwändiger heraus, als das Team angenommen hatte. Preisfrage: Welches Team hat sein Commitment erfüllt? Das Team, das 50 % Überstunden macht, die Anbindung mit Ach und Krach zusammenschustert und mit zwei schweren Fehlern in Produktion stellt, die dazu führen, dass jede zehnte Bestellung verloren geht? Oder das Team, das nach Rücksprache mit dem *Product Owner* kurzerhand die Bestellseite in eine PDF-Datei rendert und diese über einen Fax-Server an den Großhändler schickt, auch wenn ursprünglich andere Storys geplant waren?

Das Agile Manifest ist hier eindeutig: Die Erkenntnisse während der Implementierung sind eine Änderung der Annahmen. Darauf adäquat zu reagieren, um das eigentliche Ziel dennoch zu erreichen, ist wichtiger, als den ursprünglichen Plan einzuhalten.

Leider (miss-)verstehen viele Teams und auch manche Coaches und Trainer das Commitment in Scrum als eine Verpflichtung des Teams, die zugesagten User-Stories auch wirklich umzusetzen – anders formuliert, den Plan einzuhalten. Sie würden eher dem ersten Team zugutehalten, sein Commitment eingelöst zu haben. Überstunden und schlechte Qualität sind in der Regel die Folge. Dabei

hätte ein Blick in das Agile Manifest geholfen, mit weniger Aktionismus die wirtschaftlich bessere Lösung zu finden, indem man sich neuen Erkenntnissen anpasst, statt die ursprünglichen Pläne auf Biegen und Brechen zu verfolgen.

Warum ist das Verfolgen von Plänen so attraktiv, dass manche Teams nicht nur ihr Privatleben, sondern auch gleich noch den Erfolg ihrer Arbeit dafür opfern? Das hat zum einen mit der Einbettung der Vorhaben in komplexe Landschaften zu tun. Die Entwicklung findet nicht auf der grünen Wiese statt, sondern andere Teams warten auf die Ergebnisse und stellen ihre Arbeit wiederum auf sie ein. Aber nicht jede Abhängigkeit muss unbedingt in eine Planungsabhängigkeit münden: Manchmal kann das Pull-Prinzip, das auch Kanban zu Grunde liegt, Abhängigkeiten deutlich entschärfen. Und immer kann man die agile Tugend der Transparenz verfolgen und klare Auskünfte darüber erteilen, wie verlässlich die Planung eigentlich ist (vgl. [Col11]).

Häufiger haben die Probleme aber mit überkommenen Managementstrukturen zu tun: Wo Manager im „besten“ Kommando-Kontroll-Stil glauben, mit einer Powerpoint-Präsentation oder eben einem Projektplan könnten sie die Realität formen, halten sie irgendwann die Planung für die Realität und sehen Abweichungen als Angriff auf ihre Macht, als Insubordination, gegen die es einzuschreiten gilt. Solches Denken abzubauen, muss integraler Bestandteil jeder agilen Einführungsstrategie sein und vom oberen Management vorgelebt werden. Gutes Management formt nicht mit Gewalt Wunschrealitäten, sondern schafft die Teams und die Strukturen, die hervorragende Leistungen ermöglichen. In einer sich schnell verändernden Welt ist es dabei wichtiger, auf Änderungen zu reagieren, also das Richtige zu tun, als das zu tun, was man vor einiger Zeit für richtig gehalten hat, also den Plan zu verfolgen. ■

Literatur

- [Bec01] K. Beck, M. Fowler, Planning Extreme Programming, Addison-Wesley 2001
- [Col11] J. Coldewey, Vorhersage des Unvorhersagbaren: Langfristige Planung in agilen Vorhaben; in: OBJEKTSpektrum 1/2011
- [Coh06] M. Cohn, Agile Estimating and Planning, Prentice Hall 2006
- [Hig00] J.A. Highsmith, Adaptive Software Development – A Collaborative Approach to Managing Complex Systems, Dorset House 2000