

Dipl.-Math. Dipl.-Inf. Wolfhart Grote

Agile Softwareentwicklung nach Maß

Klassische und agile Entwicklungsmethoden kombinieren

Erlangen, November 2008: Immer wieder gibt es neue Methoden in der Softwareentwicklung – trotzdem laufen viele Projekte nicht nach Plan. Seit einigen Jahren gibt es neue Ansätze mit verschiedenen agilen Methoden, darunter auch Scrum. Mit der richtigen Auswahl und Kombination von Methoden kann die Erfolgsquote von Softwareentwicklungsprojekten gesteigert werden.

Böse Zungen behaupten, auch in der Softwarebranche gelte die Naturkonstante Pi. Softwareentwicklungsprojekte würden demzufolge die Planwerte für Dauer und Kosten meist um den Faktor 3,14 überschreiten. Das ist sicherlich übertrieben. Doch Tatsache ist, dass sehr viele Softwareentwicklungsprojekte nicht nach Plan verlaufen und manche ihr Ziel nie erreichen. Man denke nur an das spektakuläre Scheitern des Projekts „Toll Collect“. Die Suche nach verbesserten Entwicklungsmethoden ist daher sinnvoll und notwendig.

In den vergangenen 40 Jahren hat es immer wieder neue Methoden und Modelle der Softwareentwicklung gegeben. Einige Merkmale sind all diesen Methoden gemein:

- Strenge Regeln mit ausführlichen Prozessbeschreibungen
- Eine (oft weit) vorausschauende Planung
- Die Möglichkeit einer Anpassung der Methoden über ein so genanntes „Tailoring“
- Risikominimierung beim Auftraggeber durch Vereinbarung von Festpreisen und Festterminen

Agile Methoden: Flexibilität und Geschwindigkeit

Das häufige Scheitern von Softwareentwicklungen führte um die Jahrtausendwende zu Überlegungen, ob nicht der grundsätzliche Ansatz falsch sei. Man suchte die Ursache für das Scheitern nicht bei den handelnden Personen oder bei der unzureichenden Anwendung der Methoden, sondern bei den Methoden selbst. Statt mehr und bessere Regelwerke aufzubauen, versuchte man, mit viel weniger auszukommen. Ziel war es, alle Prozesse flexibel und schnell zu machen und so agil zu werden. Dies zog sich durch alle Bereiche, von der Anforderungsdefinition über die Programmierung bis zum Projektmanagement und der Organisation selbst.

Es entstanden sehr unterschiedliche Ansätze mit dem Ziel, diese Agilität zu erreichen. Im Jahr 2001 wurden alle Lösungsansätze erstmals gebündelt und im so genannten „Agilen Manifest“ [1] zusammengefasst und veröffentlicht. Darin werden Prioritäten neu gesetzt:

- Menschen und Zusammenarbeit vor Prozessen und Werkzeugen
- Lauffähige Software vor umfangreicher Dokumentation
- Zusammenarbeit mit Auftraggebern vor Vertragsverhandlungen
- Reagieren auf Änderungen vor sturem Befolgen eines Plans

Was sind nun aber die Hauptmerkmale agiler Methoden?

- Sie sind eher ergebnisorientiert als prozessorientiert.
- Sie verfolgen das Ziel, so wenig wie möglich, aber so viel wie nötig zu erstellen.

- Miteinander reden zeichnet die Zusammenarbeit eher aus als gegeneinander schreiben.
- Der Auftraggeber wird als intern beteiligter Projektpartner mit in die Verantwortung genommen und bleibt kein Außenstehender. Diese Voraussetzung ist zwingend erforderlich und sicherlich oft schwer zu erreichen.
- Statt großer Kontrollinstrumente setzt man eher auf Vertrauen.
- Die Planung ist eher kurzfristig als langfristig. Dahinter steht die Grundüberlegung, dass in komplexen Neuanwendungen eine langfristige Planung ohnehin nur schwer bis gar nicht möglich ist.
- Ob, und wenn ja wie, diese Methoden bei großen und sehr großen Projekten (viele hundert Mannjahre) erfolgreich sind, lässt sich kaum sagen, da bisher überwiegend kleinere Projekte mit diesen Methoden abgewickelt wurden.

So wie sich bei der Erstellung des Agilen Manifests Vertreter sehr unterschiedlicher Einzelrichtungen zusammengefunden haben, so gibt es auch heute sehr unterschiedliche Ausprägungen der agilen Methoden [2] [3]. Es gibt also nicht die eine agile Methode, sondern viele. Eine dieser vielen Methoden soll hier kurz skizziert werden, und zwar Scrum.

Scrum: Im Gedränge

Der englische Begriff Scrum bedeutet wörtlich „Gedränge“. Er wird im Englischen vor allem in der Sportart Rugby verwendet und bezeichnet dort das Drängeln beider Mannschaften, die in Ballbesitz kommen wollen.

Bei Scrum als agiler Softwareentwicklungsmethode [4] sollen die Beteiligten in ähnlicher Weise hart, aber fair um die Lösung des Softwareproblems ringen. Wer sind diese Beteiligten? Scrum kennt als Kerngruppe des Projekts drei Rollen von Verantwortlichen: Den „Product Owner“, das Team und den „Scrum Master“.

Der „Product Owner“ ist der Kunde oder dessen Vertreter. Er entscheidet über die Anforderungen, die das System erfüllen soll. Das Team hingegen ist eine sich selbst organisierende Mannschaft von Entwicklern, die die Entwicklung der Software durchführt. Der „Scrum Master“ schließlich ist ein Moderator, der die Einhaltung des Scrum-Prozesses sicherstellt und zwischen dem Product Owner und dem Team vermittelt. Der „Scrum Master“ soll wie ein Schäferhund arbeiten. Er hält die Herde (das Team) zusammen und hält die Wölfe (Außenstehende) fern.

Zu den Außenstehenden gehören als weitere am Projekt Beteiligte zum Beispiel das Management, andere Kundenvertreter und Vertreter aus Marketing und Vertrieb:

- Das Management legt zwar Rahmenbedingungen fest, darf aber nicht direkt ins Projektgeschehen eingreifen.
- Andere Kundenvertreter dürfen ihre Wünsche nur indirekt über den „Product Owner“ einbringen.
- Vertreter des Marketings und des Vertriebs dürfen ihre Wünsche nur indirekt über den „Product Owner“ einbringen.

Ein wesentliches Merkmal der Methode „Scrum“, die Ende der 1990er Jahre von Ken Schwaber [5] erstmals beschrieben wurde, ist eine klare Trennung zwischen Verantwortlichen (den drei oben genannten Rollen) und den Beteiligten (allen Außenstehenden). Die Interaktion zwischen den beiden Gruppen erfolgt nach klar festgelegten Regeln.

Die klare Trennung und die Interaktion beider Gruppen nach festgelegten Regeln ermöglichen den Entwicklern ein ungestörtes Arbeiten und kanalisieren Änderungswünsche der Außenste-

henden. Sie ermöglichen aber gleichzeitig den Außenstehenden, Einfluss auf das Projekt zu nehmen, allerdings nur zu bestimmten Zeitpunkten.

Der Herzschlag des Projekts

Ein Scrum-Projekt läuft in einem festen periodischen Zeittakt ab. Es wird in so genannte Sprints unterteilt, auch Iterationen genannt, die eine feste Dauer haben. Üblich sind dreißig Arbeitstage. Aber auch kürzere Perioden von zehn oder fünfzehn Arbeitstagen werden verwendet.

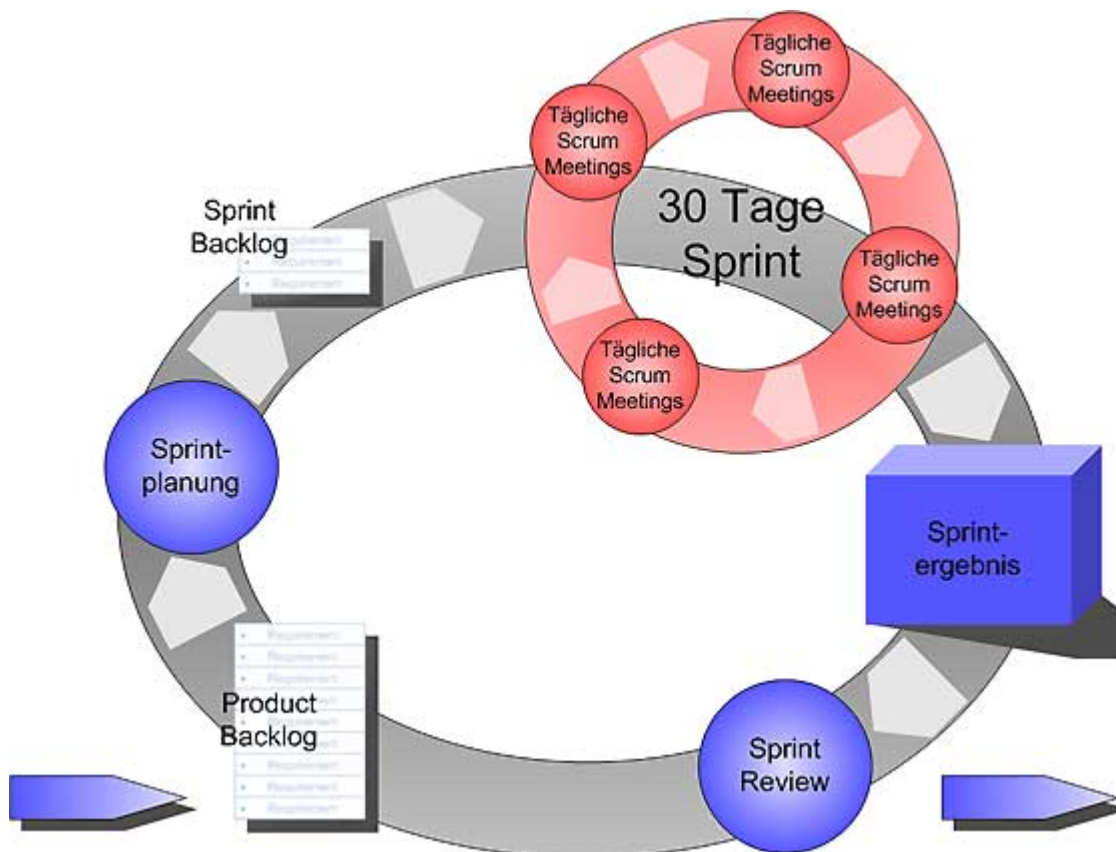


Abb. 1: Bei Scrum verläuft die Entwicklung iterativ in Sprints.

Der „Product Owner“ stellt zu Beginn alle seine Anforderungen in einem so genannten „Product Backlog“ zusammen. Dieses muss am Anfang noch nicht vollständig sein, es kann im Verlauf des Projekts immer wieder geändert oder ergänzt werden. Der „Product Owner“ versieht alle Anforderungen mit einer Priorität, in der er sie realisiert haben möchte.

Am Anfang eines Sprints wird die Sprintplanung durchgeführt. Dabei nehmen die Entwickler die Anforderungen mit der höchsten Priorität aus dem Product Backlog, analysieren sie grob und schätzen ab, wie viel sie davon im nächsten Sprint verwirklichen können. Dieser Teil des Product Backlog wird zum so genannten „Sprint Backlog“. Das Team verpflichtet sich mit einem formalen „Commitment“ dazu, dieses vollständig zu realisieren. Zum Detailablauf während eines Sprints mit täglichen „Scrum Meetings“ sei hier aus Platzgründen auf die Literatur verwiesen [5].

Am Ende des Sprints legt das Team dann das Sprintergebnis vor – ein Stück lauffähige, getestete und fertig dokumentierte Software. In einem „Sprint Review“ nimmt der Product Owner dieses Ergebnis ab. Er hat dann die Möglichkeit, im Product Backlog Änderungen, zum Beispiel in der Vorrangigkeit, vorzunehmen. Anschließend wird aus diesem Product Backlog erneut der vorrangigste Teil entnommen, und ein neuer Sprint beginnt.

Der alte Zielkonflikt

Die klassischen und die agilen Methoden unterscheiden sich unter anderem darin, wie sie die Zielgrößen des Projektmanagements bei Engpässen erreichen. Zu diesen Zielgrößen gehören üblicherweise Qualität, Termine und die Kosten.

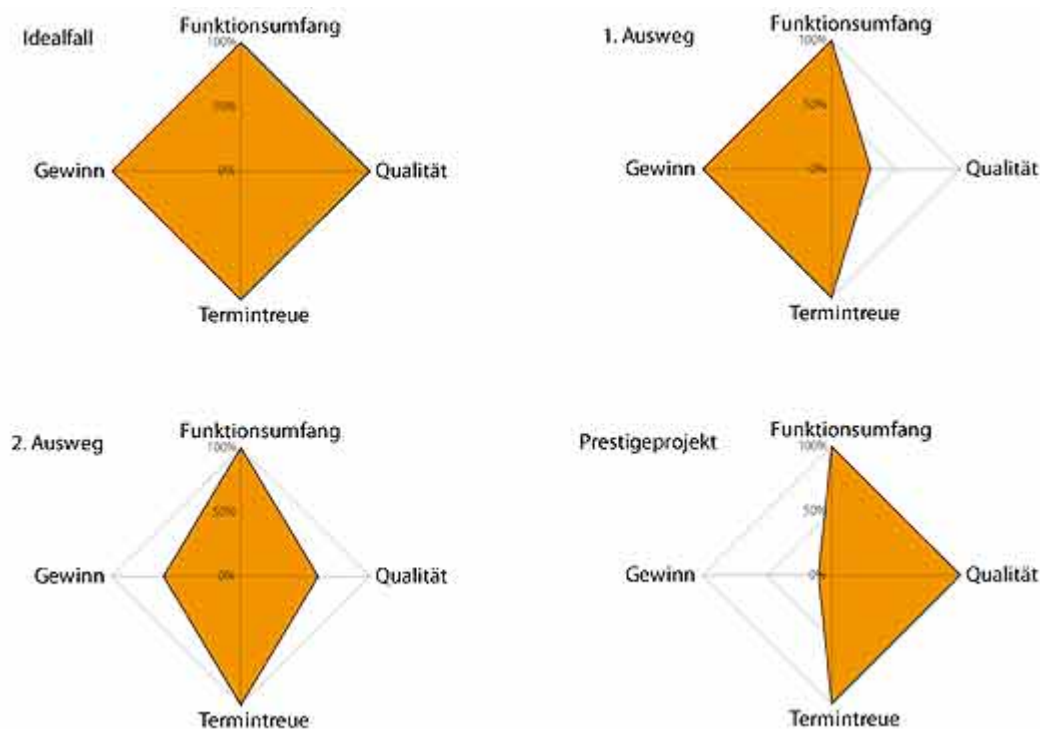


Abb.2: Klassische Methoden stellen im Allgemeinen den Funktionsumfang als Zielgröße nicht in Frage, sondern machen eher Abstriche bei Qualität und Gewinn, um Vertragsstrafen bei nicht eingehaltenen Terminen zu umgehen.

Eine weitere Zielgröße, nämlich der Funktionsumfang, wird im Allgemeinen bei den klassischen Methoden nicht als variabel angesehen. Das Erreichen dieses Ziels wird stattdessen stillschweigend vorausgesetzt. Hier soll der Funktionsumfang aber als vierte Zielgröße zur Verdeutlichung ausdrücklich mit hinzugenommen werden. Damit bei allen Zielgrößen ein großer Wert immer auch ein gutes Ergebnis bedeutet, wird die Zielgröße „Kosten“ durch deren Inversion, nämlich „Gewinn“ ersetzt.

Im Idealfall werden alle Zielgrößen zu 100 Prozent erfüllt. In der Wirklichkeit treten jedoch oft Engpässe auf. Es müssen Kompromisse eingegangen und auf die Vollerfüllung einzelner Zielgrößen verzichtet werden. Oft sieht das dann bei der Anwendung klassischer Methoden so aus, dass an der Qualität gespart wird, weil deren unvollständige Erfüllung zunächst nicht offensichtlich ist. Termintreue und Funktionsumfang werden dagegen meist eingehalten, da sonst häufig Vertragsstrafen drohen. Das Management fordert in der Regel, die Kosten einzu-

halten, das heißt, den geplanten Gewinn zu erreichen – Einbußen bei der Qualität sind das übliche Resultat. Oft muss die reduzierte Qualität dennoch im Nachhinein nachgebessert werden, was wiederum zusätzliche Kosten verursacht und damit den Gewinn trotz allem reduziert.

Ganz anders sieht der Ansatz bei den agilen Methoden aus. Qualität, Termin- und Kosteneinhaltung sind „heilig“. Wenn Abstriche gemacht werden müssen, erfolgen diese beim Funktionsumfang.

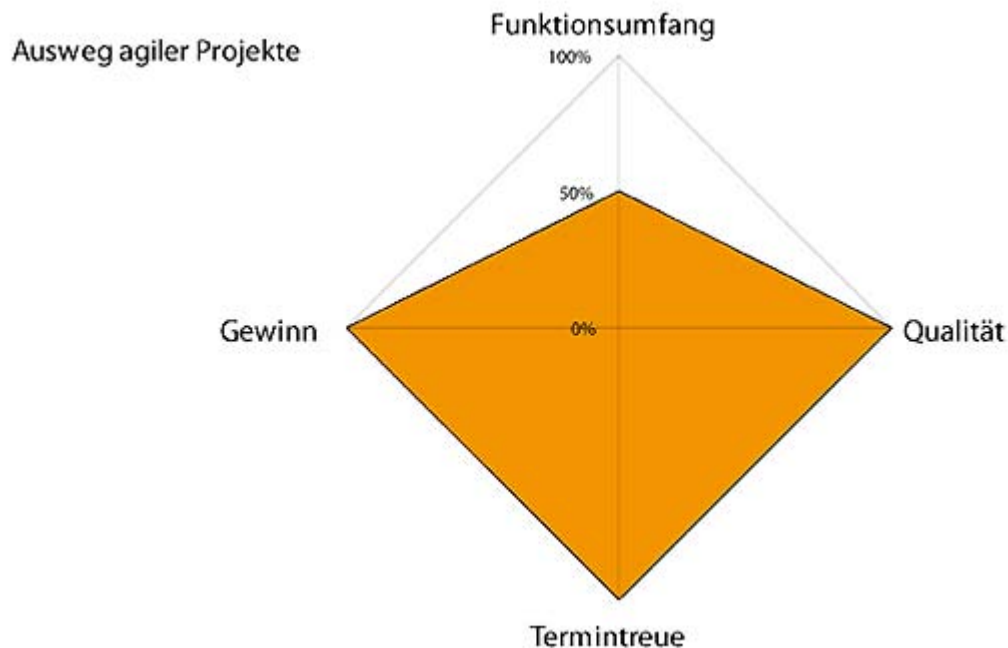


Abb. 3: Agile Methoden schränken bei Schwierigkeiten eher den Funktionsumfang ein, um die anderen Zielgrößen nicht zu beeinträchtigen.

Es wird also immer termingerecht eine Software in vereinbarter Qualität und zu vereinbarten Kosten geliefert, nur hat diese eventuell einen etwas verringerten Funktionsumfang. Bei der Methode Scrum kann der Auftraggeber dabei über die Priorisierung des Product Backlog selbst bestimmen, welche Funktionen zunächst nicht realisiert werden.

Fazit

Agile Methoden sind sicher kein Allheilmittel zur Lösung aller Probleme bei der Softwareentwicklung, auch wenn mancher „Guru“ dies versprechen mag. Es kommt ganz wesentlich auf die Gegebenheiten des einzelnen Softwareentwicklungsprojekts an. Denn es gibt nicht die eine optimale Methode für alle Projekte, sondern immer nur eine passende Methode oder Methodenkombination für ein einzelnes Projekt.

Man stelle sich beispielsweise Software in einem Anlagenprojekt mit festem Preis und festem Termin vor. Um alle vier Zielgrößen zu der Zufriedenheit aller Beteiligten zu erreichen, kann die Grobplanung, also die Aufteilung in mehrere vordefinierte Teillieferungen, nach klassischen Methoden erfolgen. Die Feinsteuerung jedoch erfolgt in Iterationen mit einem festen Zeittakt nach den Regeln von Scrum.

Die Festlegung einer für den Einzelfall maßgeschneiderten Methodenkombination erfordert Erfahrungen in beiden Welten, in jener der klassischen Methoden und der agilen Methoden.

Durch eine solche Kombination kann aber der Softwareentwicklungsprozess deutlich verbessert werden. Die Naturkonstante Pi hat dann als Faktor in der Softwareentwicklung ausgedient.

Links und Literatur

- [1] Das agile Manifest: <http://www.agilemanifesto.org/>
- [2] Forum für agile Methoden: <http://www.agilealliance.org>
- [3] Agility kompakt: P. Hruschka, C. Rupp, G. Starke, 2004
- [4] Forum für Scrum: <http://www.scrumalliance.org>
- [5] K. Schwaber: Agiles Projektmanagement mit Scrum, 2007

Der Autor



Dipl.-Math. Dipl.-Inf. Wolfhart Grote (60) ist Vorstand der Software Ring eG, Geschäftsführer der Firma infolab GmbH und Certified Scrum Master. In den vergangenen 40 Jahren hat er Hunderte von Softwareentwicklungsprojekten aller Größenordnungen begleitet und verantwortet. Dabei hat er so manchen „modischen Trend“ im Bereich der Methoden kommen und gehen sehen. Aus dieser langjährigen Erfahrung heraus übernimmt er mit infolab Beratungs-, Coaching- und Realisierungsaufgaben in Softwareentwicklungsprojekten.

www.infolab.de