



# COCKPIT FÜR AGILE PROJEKTE

Wie man ein agiles Projekt mit den richtigen Messwerkzeugen auf Kurs hält.

| von UWE KOBLITZ

Immer wieder laufen Projekte aus dem Ruder, auch agile Projekte bleiben davon nicht verschont. Die Frage ist, wie frühzeitig erkannt werden kann, dass das Projekt vom geplanten Kurs abgekommen ist. Um rechtzeitig auf Fehlentwicklungen aufmerksam zu werden, sind Messungen notwendig, die den Leistungsstand zeigen und Ansätze für Verbesserungen liefern. Zwei Schritte sind dafür nötig:

- **Messen der Leistung:** Dieser Schritt umfasst die Planung, Ausführung und Überwachung der Projektarbeiten. Die einfachste und wichtigste Methode ist die Messung der Fertigstellung von Liefereinheiten des Projekts.
- **Regeln der Leistung:** In diesem Schritt wird die Leistungsmessung aus dem ersten Schritt verwendet, um Probleme transparent zu machen und die Leistung im Projekt zu verbessern.

Dieser Artikel zeigt, wie in einem agilen Projekt schrittweise verschiedene Stufen einer Leistungsmessung eingeführt werden können, um die Projektsteuerung zu verbessern.

## EBENEN DER PLANUNG

Geplant wird in Organisationen auf verschiedenen Ebenen. Auf den obersten drei Ebenen sind das 1. die strategische Planung der Behördenleitung beziehungsweise Geschäftsführung, 2. die Portfolio-Planung der verschiedenen Abteilungen/Geschäftsbereiche und 3. die Programmplanung innerhalb dieser Abteilungen und Bereiche. Die Programme selbst bestehen aus verschiedenen Produktentwicklungen (Projekten), die in mehreren Versionen (Stufen) geplant werden.

Für klassische Projekte gibt es viele etablierte Messmethoden und Cockpits. Für agile Projekte gibt es das bisher nicht. Die Visualisierung beschränkt sich in den meisten Fällen auf einfache Darstellungen des Restaufwands in sogenannten Burndown Charts. Doch aus klassischen Projekten können viele Kenngrößen übertragen werden, denn die Unterschiede zu den agilen „Basis-Kenngrößen und Messmethoden“ sind gar nicht so groß.



Abbildung 1: Planungsebenen in Organisationen

Bei agilen Software-Projekten sind für jede Stufe der Produktentwicklung mehrere Iterationen mit einer typischen Dauer von jeweils ein bis vier Wochen notwendig. Auf der untersten Ebene findet eine tägliche Aufgabenplanung durch das Entwicklungsteam statt. Nachfolgend wird das Messen und Steuern auf der Ebene der Produktentwicklungsplanung (Versionsplanung) beleuchtet, das heißt ein Planungshorizont von sechs Monaten bis zu einem Jahr. Messwerte einer Produktentwicklung können natürlich auf der Ebene der Programmplanung aggregiert beziehungsweise auch für die Planung der Iterationen genutzt werden.

### DIE KONTROLLINSTRUMENTE

Auch für agile Projekte gilt: Ein Projekt ohne Messungen und Anzeigen ist wie ein Flugzeugcockpit ohne Kontrollinstrumente. Damit zu fliegen, käme einem Blindflug gleich. Genau wie beim Flugzeugcockpit müssen die Messungen in agilen Projekten aktuell und gut sichtbar sein. Nur so entstehen die notwendige Transparenz und der Ansporn, selbst gesteckte wie auferlegte Ziele auch zu erreichen. Wie viele Kontrollinstrumente eingesetzt werden, hängt letztendlich von der Größe des agilen Projekts ab. Im Folgenden werden unterschiedliche Stufen von einfachen bis zu aufwendigeren Messungen dargestellt. Die einzelnen Stufen bauen aufeinander auf.

### STUFE 1: MANAGEMENT DER ITERATIONEN UND DES GESAMTUMFANGS

In der einfachsten Form der Aufwandsmessung wird am Anfang der Gesamtaufwand für eine Version beziehungsweise das Gesamtprodukt geschätzt – am besten durch das ganze Entwicklungsteam, da es diese Aufwände auch tragen muss. Dieser Aufwand kann in Story Points, normierten Epics, Stunden oder Function Points abgeschätzt werden. Nach jeder Iteration wird erneut bewertet, welcher Restaufwand noch verbleibt, und in einem Diagramm dargestellt. Es ergibt sich eine abfallende Reihe von Einzelmessungen, die mit der letzten Iteration bei einem Restaufwand von null endet. Im hier beschriebenen Beispielprojekt werden Story Points verwendet, um die Höhe des Restaufwands darzustellen. Als Länge einer Iteration wurden im Diagramm eine Dauer von einer Woche sowie absolute Termine anstelle von Iterationen verwendet (siehe Abbildung 2).

Voraussetzung für die Messungen ist eine Abschätzung des Produktumfangs im Vorfeld. Der angestrebte Produktumfang ist normalerweise bekannt und wird zum Beispiel bei Scrum in Form eines Backlogs gepflegt. Natürlich wird es Änderungen am ursprünglich geplanten Umfang geben, der auch gemessen und bewertet werden kann. Vorhaben, bei denen der Gesamtumfang

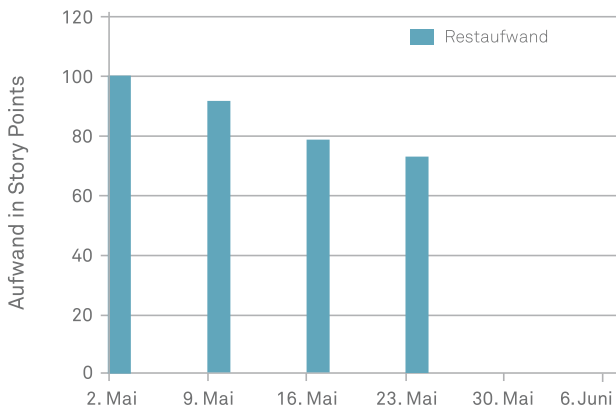


Abbildung 2: Einfache Darstellung des noch verbliebenen Restaufwands über die Zeit

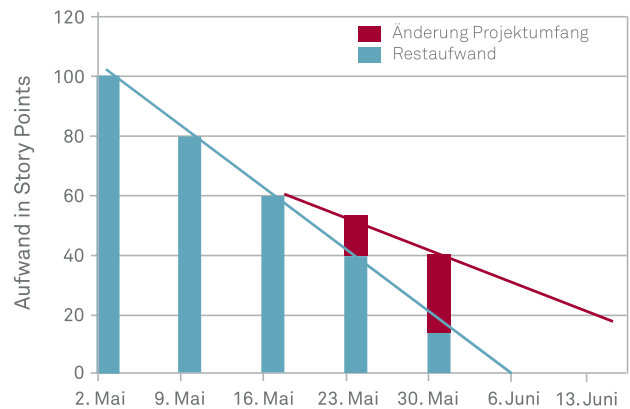


Abbildung 3: Änderungen im Projektumfang

vorweg nicht abgeschätzt werden kann, verbleiben auf der (Vor-) Stufe der reinen Iterations-Planung. Um zeitnah reagieren zu können, ist es unerlässlich, dass Messwerte regelmäßig erhoben werden. Das heißt, es sollte mindestens einmal pro Iteration gemessen werden. Dabei werden der abgearbeitete Produktumfang wie auch die Aufwandsschätzungen für den noch verbleibenden Produktumfang und für zusätzliche Änderungen erhoben und geprüft. Der zusätzliche Aufwand durch die Messungen selbst sollte so gering wie möglich gehalten werden.

### Änderungen im Projektumfang

Es ist hilfreich, zwischen zwei Arten von Änderungen im Projektumfang zu unterscheiden. Die erste Art ergibt sich bei der Detailplanung. So werden üblicherweise Anforderungen (Epics) in User Storys und dann in Aufgaben (Tasks) heruntergebrochen. Das Herunterbrechen findet nicht gleich am Anfang für alle Anforderungen statt, sondern erst zu dem Zeitpunkt, an dem sie auch für die Realisierung benötigt werden. Bei dieser Zerlegung werden immer wieder nicht vorhergesehene Details auftauchen, die wichtig für die Implementierung sind, aber in der ursprünglichen Anforderung nicht erwähnt wurden. Am besten werden diese Änderungen umgesetzt, ohne sie in der Planung extra zu berücksichtigen, denn es ist zu erwarten, dass sich solche Änderungen gleichmäßig über das gesamte Projekt verteilen und sich bei allen Anforderungen ergeben werden. In Summe führen sie entweder zu einer Erhöhung (bei einem systematischen anfänglichen Unterschätzen des Aufwands), zu einer Reduzierung (bei einer systematischen anfänglichen Überschätzung des Aufwands) oder – im Idealfall (bei einem Ausgleich von überschätzten und unterschätzten Anforderungen) – zum Einhalten der ursprünglich geschätzten Aufwände.

Häufig gibt es die Tendenz, User Storys im Projektverlauf auszuschnürceln – besonders, wenn zwischen Anwendern und Entwicklerteams eine zu große Nähe herrscht. Schnell ergeben sich „Extras“, die unter der Hand in das Projekt eingebracht werden und zu einer schleichenden Erhöhung des Projektumfangs führen. Entwicklungsteam und Anwender müssen sich disziplinieren, auf das wirklich Notwendige (minimum viable product) beschränken und den Product Owner als Budgetverantwortlichen immer in den Prozess der Änderungsanforderungen einbeziehen.

Die zweite Art von Änderungen ergibt sich, wenn der Kunde neue Funktionalitäten fordert. Dabei handelt es sich um echte Änderungen am Projektumfang, die in einem Diagramm als gestapelte Balken dargestellt werden können. Damit werden Zeitpunkt und Umfang der Änderungen veranschaulicht und gezeigt, wie sich der Zuwachs der Projektänderungen zur Abarbeitungsgeschwindigkeit

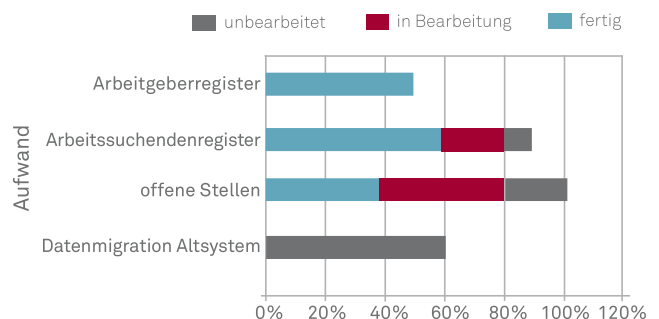


Abbildung 4: Fertigstellungsgrad der Anforderungen in Relation zur Anforderung mit dem größten Aufwand (hier: „offene Stellen“)

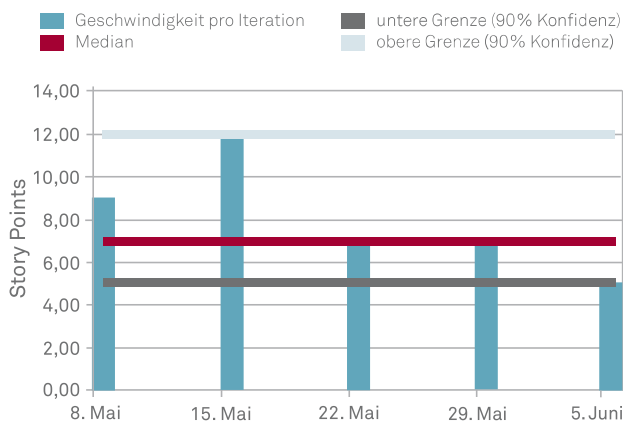


Abbildung 5: Abarbeitungsgeschwindigkeit in Story Points pro Iterationen

verhält. Ist der Zuwachs der Änderungen größer als die Abarbeitungsgeschwindigkeit, verschiebt sich das Projektende potenziell ins Unendliche (siehe Abbildung 3) – ein Indikator dafür, dass Funktionalität in die nächste Version verschoben werden beziehungsweise, dass sie sich auf das Notwendige beschränken muss.

### Bearbeitungsstatus der Anforderungen

Für einen Überblick über den Bearbeitungsstatus der Anforderungen muss der Bearbeitungsfortschritt der einzelnen Anforderungen gemessen werden. Das folgende Diagramm (siehe Abbildung 4) stellt den Fortschritt als Diagramm dar. Es zeigt den Fertigstellungsgrad der Anforderungen in Relation zur Anforderung mit dem größten Aufwand (im Beispiel: „offene Stellen“). So kann man sich schnell einen Eindruck verschaffen, welche Teilsysteme wie weit fertiggestellt sind und für eine Vorführung infrage kommen.

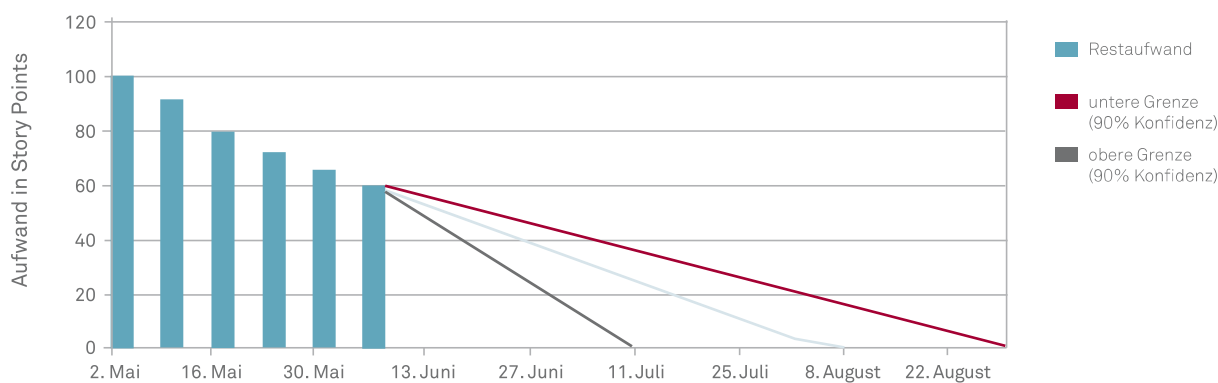


Abbildung 6: Projektion des voraussichtlichen Endtermins

## STUFE 2: INTEGRIERTES UMFANGS- UND ZEITMANAGEMENT

Auf der nächsten Stufe wird dem Restaufwandsdiagramm eine Trendlinie für den geplanten Restaufwand hinzugefügt. Spätestens nach fünf Iterationen sollte so viel Erfahrung vorhanden sein, um eine Abschätzung über den voraussichtlichen Fertigstellungstermin und dessen Eintrittswahrscheinlichkeit treffen zu können. Auftretende Schätzungenauigkeiten werden im Verlauf der Iterationen kleiner.

### Abarbeitungsgeschwindigkeit

Die Abarbeitungsgeschwindigkeit wird über die unterschiedlichen Iterationen nicht gleich sein, auch wenn die Geschwindigkeit über die Teamstärke und die wirkliche Arbeitszeit (ohne Krankheiten, Urlaube, ungeplante Einsätze in anderen Projekten ...) normiert wird. In Abbildung 5 hat die Geschwindigkeit in den letzten Iterationen abgenommen. Hier ist ein genauere Blick auf die möglichen Ursachen und gegebenenfalls ein direktes Eingreifen angezeigt. Hat das Team in den ersten beiden Iterationen die Architektur vernachlässigt (also eine technische Schuld angehäuft)? Gibt es Hindernisse für das Team? Ist ein weiteres Absinken zu erwarten?

Aus den verschiedenen historischen Abarbeitungsgeschwindigkeiten kann ein unterer und ein oberer Schätzwert für die zu erwartende Geschwindigkeit abgeleitet werden. Ziel ist, eine Vorhersage treffen zu können, wie viel Arbeit das Team in einer geplanten Anzahl von Iterationen erledigen kann. Hier ist es besser, statt einen bestimmten Wert zu nennen (zum Beispiel 8 Story Points pro Iteration), die Geschwindigkeit als Bandbreite von Werten zu betrachten, also zum Beispiel: „Aufgrund unserer historischen Daten sind wir zu 90 Prozent sicher, dass die Geschwindigkeit für die noch verbliebenen Iterationen in diesem

Projekt zwischen 5 und 12 liegen werden.“ Um die Bandbreite der Geschwindigkeit zu errechnen, werden die Daten aus mindestens fünf Iterationen benötigt. Man geht von einer Normalverteilung aus und legt in diese Verteilung ein Konfidenzintervall (auch Vertrauensbereich, Vertrauensintervall oder Erwartungsbereich genannt) hinein. Ein Vorteil gegenüber einer Punktschätzung ist, dass man neben dem Konfidenzintervall direkt die Signifikanz, das heißt die Wahrscheinlichkeit ablesen kann, innerhalb einer gewissen Schwelle zu bleiben.

### Zeitmanagement

Mit den Ergebnissen der Geschwindigkeitsmessung kann das Instrument der Restaufwandsschätzung um eine Aussage des Intervalls der Fertigstellung angereichert werden (siehe Abbildung 6). Im Diagramm mit den Restaufwänden kann eine Abschätzung über den noch verbleibenden Zeitbedarf aufgezeichnet werden. Vom Zeitpunkt der letzten Restaufwandsschätzung trägt man die untere und obere Grenze sowie den Median auf und erhält in diesem Beispiel als Enddatum im besten Fall den 11. Juli, für den Median den 8. August und im ungünstigsten Fall (mit 90 Prozent Konfidenz) den 29. August.

Im Diagramm wird ein linearer Verlauf des geplanten Restaufwands angezeigt. Im Gegensatz zu klassischen Projektmethoden mit ihrem typischen S-Verlauf im Aufwand (wenige Mitarbeiter in Analyse und Design, viele Mitarbeiter in Programmierung und Test und wenig Personal beim Einsatz) kann bei agilen Methoden von einem konstanten Team ausgegangen werden. Sollte das nicht der Fall sein, so muss die Trendlinie entsprechend angepasst werden.

### STUFE 3: INTEGRIERTES UMFANGS-, ZEIT- UND KOSTENMANAGEMENT

Zusätzlich zur Aufwandsmessung kann das noch dem Projekt zur Verfügung stehende Budget gemessen werden. Dieses Budget kann in einem Diagramm neben dem Restaufwand aufgetragen werden (Abbildung 7, graue Trendlinie „Restbudget“). Bei Software-Entwicklungsprojekten wird der Löwenanteil der Kosten durch Personalkosten verursacht, die entweder direkt aus den Buchhaltungssystemen oder aus Stundenzetteln (und den Stundensätzen) der Entwickler ermittelt werden können. Die Messung des Budgets ist insbesondere dann hilfreich, wenn die Entwickler immer wieder zu „Noteinsätzen“ in anderen Projekten abgerufen werden. Abbildung 7 zeigt, dass die Restbudget-Trendlinie unter die Trendlinie des geplanten Restaufwands gefallen ist, das heißt, dass bisher mehr Budget verbraucht wurde als geplant. Kommt es zu keiner Änderung im Trend, wird das Budget vor der Fertigstellung aufgebraucht sein.

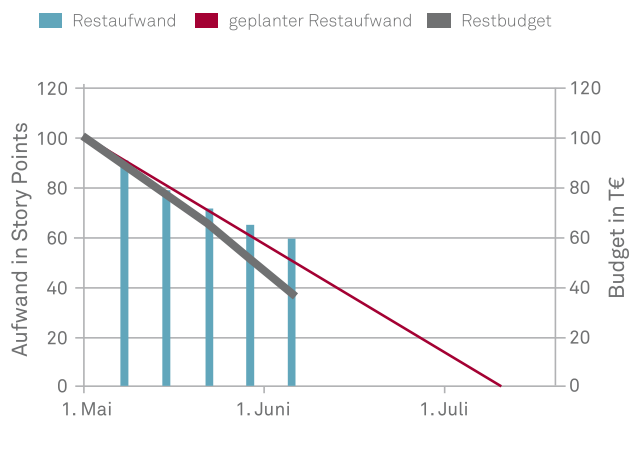


Abbildung 7: Restaufwand mit Zeit- und Kostendarstellung

### FAZIT

Viele Organisationen versuchen, sofort ein aufwendiges Reporting für agile Projekte einzuführen. Das führt zu hohen Kosten am Anfang und bringt wenig Nutzen. Andere Organisationen erledigen die Projektbudgetierung und -buchhaltung ad hoc: Sie führen zwar auch eine Zeitplanung durch, können die verschiedenen Controlling-Informationen aber nicht zusammenzuführen. Wieder andere Organisationen beherrschen den Projektumfang nicht. Einfach, weil sie ihn nicht messen. So haben sie kein solides Fundament für das Kosten- und Zeitmanagement.

Die im Artikel beschriebenen drei Stufen sind ein guter Weg, in agilen Projekten ein „angemessenes“ Projektcontrolling einzuführen. Jede Stufe dieses skalierbaren Ansatzes liefert objektive Messungen, die der Steuerung, Berichterstattung und Zusammenarbeit helfen. Jede Stufe ermöglicht einen zunehmend besseren Überblick – auch wenn sie einen größeren Controlling-Aufwand bedeutet. Die Beherrschung jeder Stufe in der beschriebenen Reihenfolge heißt auch, die Fähigkeit zur Leistungsmessung durch praktische Erfahrung schrittweise aufzubauen. So entsteht nicht nur das Produkt, sondern auch das agile Cockpit in Iterationen. ●

**ANSPRECHPARTNER – UWE KOBLITZ**  
 Lead Project Manager  
 Public Sector Solutions Consulting

