

Management großer Projekte – Ein modellbasierter Ansatz

Dehla Sokenou
GEBIT Solutions, Koenigsallee 75b, D-14193 Berlin
dehla.sokenou@gebit.de

1 Motivation

Betrachtet man ein typisches Softwareprojekt, so werden oft Aussagen wie “Die Komponente X ist zu 80% fertig” gefällt. Solche Aussagen haben den großen Nachteil, dass sie keinerlei Rückschlüsse auf den aktuellen Stand des Projekts erlauben, z.B. ist nicht gesagt, wie viel Zeit die fehlenden 20 % noch kosten. Wenn solche Aussagen zu einem Gesamtstatus kumuliert werden, kann dies zu einer deutlichen Überschätzung des Projektfortschritts führen. Verzögerungen, Fehlplanungen und ungenügende Steuerung sind oft die Folge.

2 Lösungsansatz und Vorgehen

Ziel ist die Planbarkeit und Verfolgbarkeit auch großer Projekte, ohne dass sich dadurch der Aufwand für die Projektleitung in einer Weise vergrößert, dass eine sichere Aussage über den aktuellen Projektstatus nicht mehr möglich ist. Um dieses Ziel zu erreichen, schlagen wir ein Verfahren des Projektmanagements vor, das auf wenigen Regeln basiert und einfach während des Projekts gepflegt werden kann. Unser Verfahren beruht auf Anforderungen, die als Use Cases formuliert werden. Neben funktionalen Anforderungen werden auch nichtfunktionale Anforderungen als Use Cases formuliert, um das Verfahren einheitlich auf alle Anforderungen anwenden zu können. Die Regeln sind im Einzelnen:

Regel 1: *“Die Größe eines Use Case muss überschaubar sein.”*

Aus unserer Erfahrung hat sich gezeigt, dass der Umfang eines Use Cases möglichst klein gehalten werden sollte. Typischerweise sollte man Use Cases mit einem Umfang von etwa 5 Personentagen planen. Eine so feingranulare Planung ist meist nicht bei Projektbeginn möglich. Erleichtert wird die Zerlegung durch die Aufteilung des Projekts in Phasen, denen einzelne grobgranulare Use Cases zugeordnet werden. Diese grobgranularen Use Cases werden zu Beginn einer Phase in kleinere Use Cases verfeinert, bis der geforderte Umfang erreicht ist. Feingranulare Use Cases ändern schneller ihren Status und sind deshalb besser planbar. Hat ein Use Case im Umfang von 5 Tagen eine Verzögerung von z.B. 20%, so fällt dies viel früher auf, als bei einem Use Case mit geplanten 100 Tagen. So lässt sich zu jedem Zeitpunkt eine sehr gute Abschätzung des Gesamtprojektstatus erreichen. Massnahmen zur Gegensteuerung können sehr viel schneller ergriffen werden.

Regel 2: *“Es gibt eine begrenzte Menge von Status. Erst wenn ein Use Case einen neuen Status vollständig erreicht hat, wird sein Status geändert.”*

Oft wird der Projektstatus aus Aussagen kumuliert, die eine teilweise Fertigstellung von Use Cases enthalten. Das betrifft meist nicht nur die Implementierung, sondern auch “teilweise spezifiziert” oder “teilweise getestet” kommen vor. Solche Aussage sind jedoch wenig hilfreich, denn ein teilweise implementierter Use Case kann in der Regel noch nicht in das Gesamtsystem integriert oder getestet werden. Statt Teilaussagen zu machen, wird der Status eines Use Cases hier aus einer begrenzten Menge definiert. Für das erfolgreiche Erreichen eines Status müssen genaue Kriterien definiert werden. Farbliche Kodierung der Use Cases je nach Status erlaubt auch bei großen Projekten einen einfachen Überblick, wieviele Use Cases bereits implementiert oder abgenommen sind. Gleichzeitig kann eine genauere Auswertung und somit eine tagesaktuelle Statusdokumentation durch die Generierung entsprechender Dokumente “on-demand” erzeugt werden.

Regel 3: *“Erstellte Use Cases sollten möglichst schnell fachlich abgenommen werden.”*

Diese Regel garantiert eine schnelle Kontrolle der implementierten Anforderungen durch den Kunden und verhindert ein Implementieren an den Kundenwünschen vorbei. Da jeder Use Case eine in sich geschlossene Einheit implementiert, kann diese auch getrennt fachlich getestet und abgenommen werden. Im Gegensatz dazu stellt ein sehr späte Abnahme durch den Kunden ein hohes Projektrisiko dar.

Regel 4: *“Es gibt keinen Weg zurück.”*

Eine Änderung des Status eines Use Case erfolgt immer nur in eine Richtung, d.h. ein einmal erreichter Status kann nicht wieder zurück auf den vorherigen Status gesetzt werden. Werden z.B. nach der Abnahme Änderungswünsche von Kundenseite geäußert, so werden diese als Change Requests aufgenommen. Die Regel garantiert, dass ein einmal erreichter Projektstatus beibehalten wird. Für alle Projektbeteiligten gibt es keine Überraschungen, so ist z.B. klar, wie mit neuen Kundenanforderungen umzugehen ist.

3 Erfahrungen und Ergebnisse

Das beschriebene Verfahren wurde bereits erfolgreich in diversen, auch größeren Softwareprojekten eingesetzt. Auch wenn es zunächst problematisch erscheint, dass ein Projekt eine große Menge feingranularer Use Case definiert, ist unsere Erfahrung, dass diese zu einer deutlich besseren Übersicht und Planbarkeit des Projekts führen. Durch feingranulare Use Cases weicht der kumulierte Projektstatus nur wenig vom tatsächlichen Status ab. Der Projektstatus ist immer eine “Worst Case” Betrachtung, was eine deutlich höhere Projektsicherheit garantiert als eine zu optimistische Schätzung. Natürlich geht dies nicht ohne die entsprechende Werkzeugunterstützung. GEBIT hat dazu eine modellbasierte Werkzeugkette entwickelt, die von den Anforderungen bis zu Test und Projektmanagement eine durchgängige Softwareentwicklung unterstützt. Als Beispiel sei hier das in Zusammenarbeit von GEBIT und DfG für die OBI-Baufachmärkte entwickelte Marktwarenwirtschaftssystem genannt. In diesem durchgängig modellbasierten Projekt wurden innerhalb von 1,5 Jahren über 1.500 Use Cases realisiert. Trotz des großen Projektumfangs erfolgte die Umsetzung in Time & Budget und in hoher Qualität.