software quality lab

03.10.2011 ▪ Seite 1

Quality Knowledge Letter

Ausgabe 2011 / 3

Erscheinungsart: ca. 4 x jährlich in elektronischer Form

Change Management

Den geänderten Änderungen auf der Spur

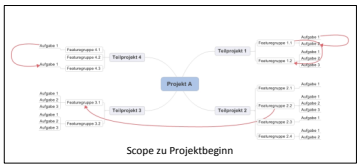
In dieser Ausgabe:

- ⇨ Änderungen in klassischen und agilen Projekten
- ⇨ Probleme der Software Entwicklung mit Change Requests
- ⇨ Der Change Prozess — Wie kann es ablaufen?

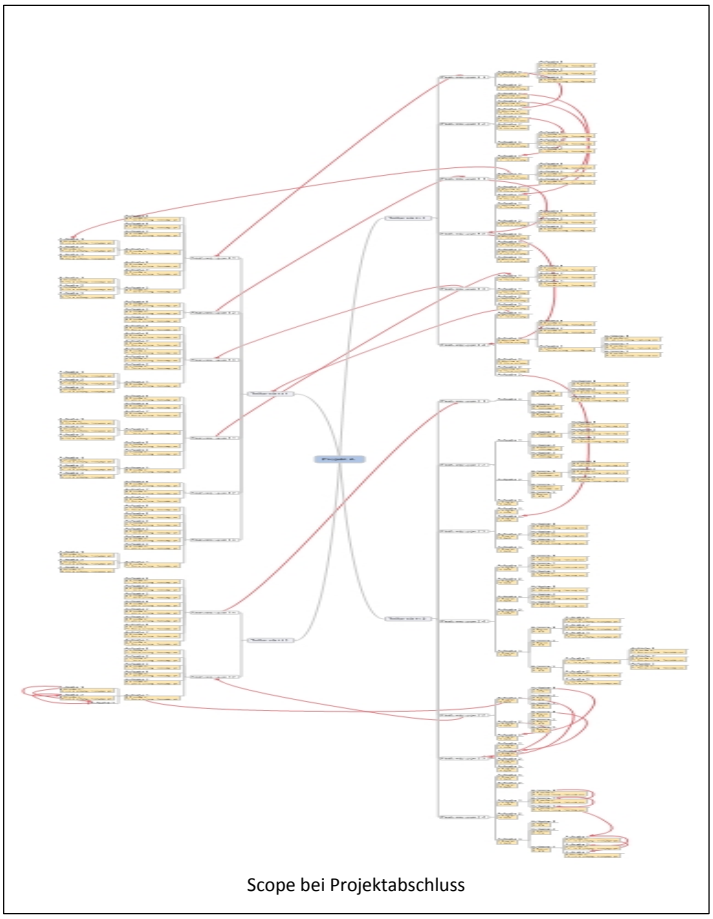
Change Management oder die wundersame Aufwandsvermehrung?

Aus einem kleinen Projektumfang wird in der Praxis vor allem bei Software-Projekten oft ein umfangreiches Projekt (siehe Abbildung).

Die Verantwortlichen sollten dann erklären können, wie es dazu kommen konnte und sich nicht darauf ausreden, dass das ja in jedem Software-Projekt so ist.



Scope Veränderung von Projektbeginn bis zum Projektende



Mögliche Ursachen für dieses Phänomen der zunehmenden Änderungen und einige Methoden, um damit umzugehen, werden nachfolgend in diesem Newsletter dargestellt.

Alles dreht sich!

Es gibt praktisch kein Software-Projekt, in dem bis zum Ende nichts verändert wird. Schon die alten Astronomen hatten irgendwann mal die Erkenntnis gewonnen, dass die Erde nicht der unverrückbare Mittelpunkt des Weltalls ist, sondern sich alles dreht und bewegt.

In der IT wurde dies mittlerweile auch als Faktum erkannt und es haben sich 2 grundlegende methodische Richtungen etabliert, mit diesem „Problem“ umzugehen.

Bei der *agilen* Entwicklung versucht man, das sich verändernde Ziel erst gar nicht genau zu treffen, sondern die Entwicklung an sich im Prinzip als eine geordnete Folge von „Changes“ zu sehen. Ganz unter dem Motto „der Weg das Ziel“ und man geht, soweit man mit dem definierten Budget eben kommt.

In der *sequenziellen* Entwicklung sind Änderungen genauso zulässig, jedoch wird das Ziel am Anfang möglichst genau fixiert und kalkuliert. Jede Abweichung muss daher umso genauer hinterfragt und auch kostenmäßig bewertet werden.

Es muss in beiden Fällen nicht das Ziel um jeden Preis erreicht werden, aber es muss in beiden Fällen klar sein, wie weit es akzeptabel ist, das ursprüngliche Ziel nicht zu erreichen und wer die Kosten für eventuelle Abweichungen oder Umwege zu zahlen hat.

Beide Vorgehensweisen haben je nach Projektsituation ihre Berechtigung.

Wichtig dabei ist, dass der Grundsatz „alles bewegt sich“, in beiden Welten gilt und daher keinesfalls ignoriert werden darf, sondern bestmöglich im Projekt berücksichtigt werden muss.

Dipl.-Ing. Johannes Bergsmann
Staatl. befugter und beeideter Ingenieurkonsulent für Informatik

Der Quality-Knowledgeletter ist ein periodisches Informationsmedium von Software Quality Lab und dessen Partnern mit den Schwerpunkten IT-Qualitätsmanagement, Projekt- und Prozess-Management.

Inhalt: fachliche Beiträge und Schwerpunktthemen, Vorstellung neuer Produkte und Leistungen, neue wissenschaftliche Erkenntnisse und andere Fachbeiträge aus unseren Themenbereichen.

Aktuelle Fach- und Forschungsbeiträge sind willkommen. Einsendungen an info@software-quality-lab.at .

Weitere Infos zu diesem und anderen Themen finden Sie auf http://www.software-quality-lab.at .

Software-Changemanagement

Michael Ulbrich & Johannes Bergsmann — Software Quality Lab

„Die überwältigende Mehrheit der erfolgreichen Innovationen verwertet Veränderung."
Peter F. Drucker

Begriffsdefinition Change Management

Im Best Practice Modell CMMI (Capability Maturity Model Integration) ist Change Management ein Teil des Configuration Managements und befasst sich mit der Festlegung und Kategorisierung von Änderungen am Produkt bzw. Projekt.

Die IT Infrastructure Library (ITIL) definiert das Change Management als „The process responsible for controlling the lifecycle of all changes. The primary objective of Change Management is to enable beneficial changes to be made, with minimum disruption to IT services."
[ITIL v3 Service Transition]

Angewandt auf die Software Entwicklung bedeutet diese Definition, dass das **Change Management** für folgende Punkte verantwortlich ist:

- Erfassen von Änderungen,
- Detaillierte Definition und Kategorisierung,
- Bewerten der Änderungen (Auswirkungen, Aufwand, Budget, Zeitplanung),
- Tracking der Änderungen über den Lebenszyklus,
- Überführung beschlossener Changes in Projektplanung.

In der Software Entwicklung muss bei Änderungswünschen immer der Einfluss auf bereits laufende oder bereits entwickelte Funktionalitäten berücksichtigt werden.

Change Request Schablone - Pflicht und Kür

Wie im Requirements Engineering empfiehlt es sich, auch für die Änderungsanforderungen eine Schablone zu entwerfen und diese für jeden Change Request (CR) zu benutzen.

Als Pflichtfelder sollten immer mindestens folgende Eigenschaften erfasst und gepflegt werden:

Attribut	*Beschreibung*
ID	eindeutige Nummer zur Identifikation
Titel	kurze Zusammenfassung des wesentlichen Inhalts
Beschreibung	möglichst klare Beschreibung d. Anforderungsänderung
Begründung	wichtige Gründe für die Änderung, gibt es keinen triftigen Grund, sollte die Umsetzung des Change Requests nochmals hinterfragt werden
Datum	Datum der Erstellung des Änderungsantrags
Initiator	der Anforderer der Änderung; gibt es niemand, der diese Änderung tatsächlich anfordert, sollte die Umsetzung des Änderungsauftrags nochmals hinterfragt werden
Priorität	subjektive Priorisierung aus Sicht des Initiators
Klassifikation	Einstufung der Art der Änderung durch Change-Management-Verantwortlichen in Abstimmung mit dem Initiator

Es empfiehlt sich, auch die folgenden Attribute eines Change-Requests noch zusätzlich zu erfassen:

Attribut	*Beschreibung*
Budget	wird im Zuge der Aufwandsschätzung eingepflegt und ist bei der Entscheidung für oder gegen die Umsetzung ein wichtiges Entscheidungskriterium
Gewünschter Umsetzungstermin	dies ist der vom Initiator angegebene Terminwunsch für die Umsetzung und hilft bei der Planung
Einfluss auf andere Features	die beeinflussten anderen Features sowie der „Impact" der Änderung (Hoch, Mittel, Gering) sollten hier angeführt werden
Status	Zustand, in dem sich der Change-Request befindet wie z.B. Neu, gesichtet, geschätzt, beauftragt, abgenommen, geschlossen, abgerechnet, zurückgestellt.
Name des Freigebenden	wird bei der Freigabe gefüllt.
Datum der Freigabe	wird bei der Freigabe gefüllt und dient dazu um die Durchlaufzeiten auszuwerten.

Klassifikation von Change Requests

Änderungen kann man grundlegend in drei grobe Gruppen einteilen:

- **Korrektive Anforderungsänderungen**: Man spricht von einer korrektiven Änderung, wenn diese ein Fehlverhalten behebt, das während der Entwicklung oder im Betrieb der Software zum Vorschein gekommen ist.

- **Adaptive Anforderungsänderungen**: Wenn eine beantragte Änderung eine Anpassung oder Erweiterung des Systems erfordert, ist dies eine adaptive Änderung. Grund für solche Anpassungen sind sehr oft z.B. Zusatzwünsche des Auftraggebers oder Änderungen im Kontext oder der Systemgrenzen.

- **Ausnahmeänderung**: Bei diesen so genannten **„Hotfixes"** handelt es sich um Änderungen, die außerhalb der normal geplanten Umsetzungszyklen meist unmittelbar durchgeführt werden müssen. Sie können sowohl korrektiv als auch adaptiv sein.

Die Klassifizierung bestimmt meist auch die Dringlichkeit der Umsetzung. Hotfixes werden so rasch als möglich

Vollständiger Knowledge Letter Zugang

Wir freuen uns, dass Sie an diesem Thema Interesse haben und den Knowledge Letter von Software Quality Lab bis hierher gelesen haben.

Dieser Knowledge Letter ist eine Vorschau (gekürzte Version des gesamten Artikels).

Wenn Sie den ungekürzten Knowledge Letter lesen möchten, registrieren Sie sich bitte unter
http://www.software-quality-lab.com/download/knowledge-letter/anfrage-knowledge-letter/

Sie erhalten nach der Registrierung vollen Zugang zu allen bisherigen Knowledge Lettern von Software Quality Lab und erhalten automatisch künftige Knowledge Letter per E-Mail.

Software Quality Days — Die größte Konferenz zum Thema „Software Qualität" in Europa!

Besuchen Sie die Top-Konferenz mit allen Infos rund um Software Qualität.

Beste Qualität der Vorträge und Tutorials sowie eine Mischung aus praktischen und wissenschaftlichen Beiträgen machen die Software Quality Days zum Top-Event.

In den 3 praktischen Tracks werden anwendungsorientierte Vorträge präsentiert. Der wissenschaftliche Track zeigt Beiträge mit hohem Innovationsgrad und praktischer Anwendbarkeit, basierend auf Forschungs-ergebnissen. Im Solution Provider Forum präsentieren Aussteller ihre neuesten Tools mit Praxis-Beispielen.

Nähere Infos unter
www.software-quality-days.com