Requirements-Management und Testen effektiv integrieren

Andreas Birk, Software.Process.Management
Gutenbergstraße 99, 70197 Stuttgart, E-Mail: andreas.birk@swpm.de

Abstract. Requirements-Management und Testen sind in vielen Unternehmen noch zu stark von einander getrennt. In der Folge kommt es zu unklaren Anforderungen, fehlenden Testfällen, Zeit- und Kostenüberschreitungen sowie Qualitätsmängeln. Wenn Requirements-Management und Testen integriert sind, steigen Produktqualität und Produktivität der Software-Entwicklung.

Dieser Artikel zeigt, wie Anforderungen und Tests über den gesamten Software-Lebenszyklus hinweg zusammen hängen. Er beschreibt, wie Projekte diese Zusammenhänge schrittweise nutzen und gestalten können, um Effizienz der Entwicklung und Qualität der Produkte zu steigern.

Prozesse, Produkte und Organisationsstrukturen weiter entwickeln

Damit Requirements-Management und Testen in der Projektpraxis stärker zusammenwachsen, muss man die Prozesse ausbauen, die Dokumentation und Ablage-Infrastrukturen[1] für Requirements und Testfälle ergänzen und aufeinander abstimmen sowie die Rollen und Organisationsstrukturen weiter entwickeln. Die folgenden Kapitel beschreiben, welche Maßnahmen und Lösungen dabei möglich sind.

Häufig wird empfohlen, Entwicklung und Qualitätssicherung voneinander zu trennen, damit die Qualitätssicherung unabhängig bleibt und genügend Ressourcen (Personen, Aufwand) erhält. Diese grundsätzliche Trennung sollte keinesfalls aufgehoben werden. Die hier vorgeschlagene Integration von Requirements und Testen zielt vielmehr darauf ab, die Zusammenarbeit zwischen Entwicklung und Qualitätssicherung trotz der organisatorischen Trennung effizient und wirkungsvoll zu gestalten.

Requirements-Management und Testen im Software-Lebenszyklus

Betrachtet man die Phasen des traditionellen Wasserfallmodells für die Software-Entwicklung, dann liegen Anforderungen und Testen weit auseinander. Die Anforderungen bilden den Startpunkt der Software-Entwicklung und die Grundlage, auf der fast alle weiteren Entwicklungstätigkeiten aufbauen. Das Testen bildet den Abschluss der Entwicklung, in dem sich zeigt, wie gut die vorangegangenen Phasen durchgeführt wurden.

In Wirklichkeit stehen Anforderungen und Testen auf vielfältige Weise eng miteinander in Beziehung. Abbildung 1 zeigt die Phasen der Software-Entwicklung in Anlehnung an das V-Modell. Sie betont die Aktivitäten der Testplanung und -vorbereitung (z.B. *Prepare Acceptance Test*), die möglichst direkt auf die ersten Entwicklungsaktivitäten folgen sollten (z.B. *Develop Requirements Specification*). Das Modell überträgt Elemente aus dem W-Modell [Spi02] in ein reines Aufgabenmodell entsprechend ISO/IEC 12207 [ISO04], das als Grundlage für die Projektplanung dienen kann.

Die Ziffern in Abbildung 1 kennzeichnen Punkte, an denen ein besonders enger Zusammenhang zwischen Anforderungen und Testen besteht: (1) frühzeitige Testplanung und Testvorbereitung, (2) Testbare Anforderungen, (3) anforderungsbasiertes Testen, (4) anforderungsbasierte Testplanung und -priorisierung, (5) integriertes Anforderungs- und Fehlermanagement sowie (6) Organisationsstrukturen und Rollen für Anforderungen und Testen. Die folgenden Kapitel betrachten diese Zusammenhänge näher.

Frühzeitige Testplanung und -vorbereitung

Testen ist keineswegs die „letzte“ Entwicklungsphase, die erst nach Abschluss der Implementierung beginnt. Planung und Vorbereitung des Testens können und sollten bereits parallel zur Anforderungsspezifkation beginnen. Nur dann ist sicher gestellt, dass die Tests gut vorbereitet sind. Außerdem trägt die Definition der Testfälle dazu bei, die Anforderungen früh zu überprüfen und ihre Qualität zu sichern. Zwei Maßnahmen bei der Projektplanung helfen, dass eine frühe Testvorbereitung und -planung gewährleistet ist:

[1] In Kontrast zu den Prozessen auch als *Produkte* und *Zwischenprodukte* bezeichnet.

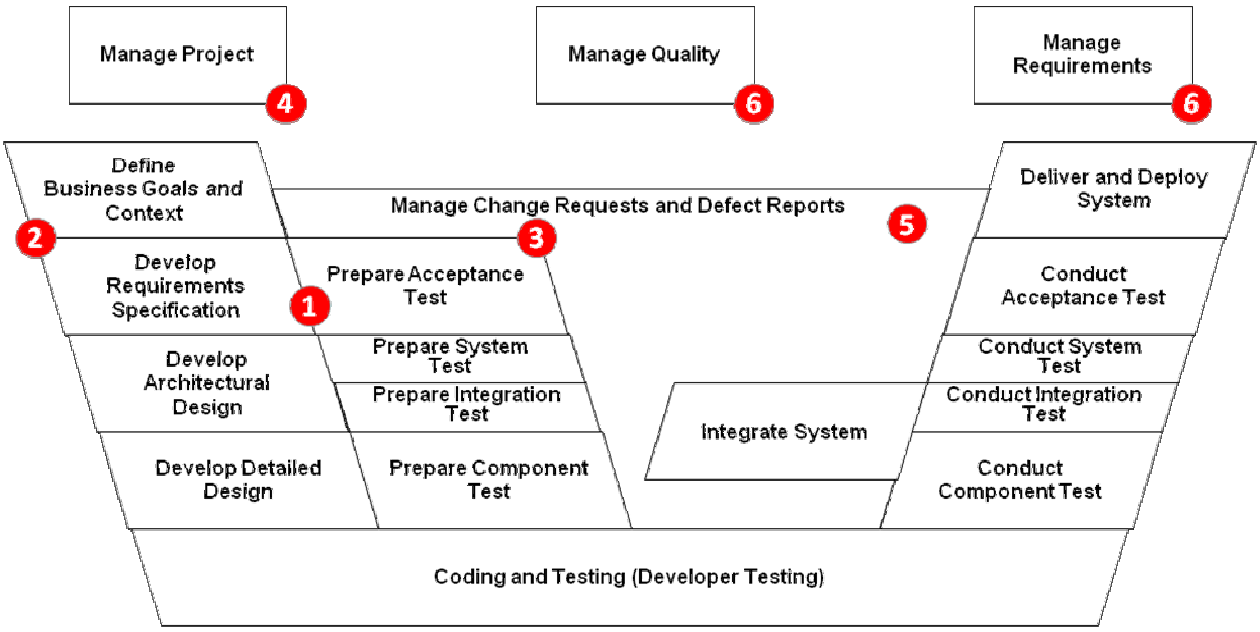

Abbildung 1: Prozessmodell (V-Modell), ergänzt um detailliertere Testaufgaben.

Testen als (Teil-)Projekt: Das Testmanagement von der Testplanung und -vorbereitung bis zur Testdurchführung und -auswertung sollte als ein eigenständiges Teilprojekt organisiert sein. So erhalten die Testaktivitäten den Stellenwert im Projekt, den sie benötigen. Auch das Zusammenspiel zwischen Anforderungen und Testen steht dann auf stabilen Füßen.

Quality Gates zur Verifikation von Testfällen und Anforderungen: Testfälle sollten frühzeitig definiert und ihre Korrektheit gegenüber den Anforderungen in systematischen Reviews verifiziert werden. Diese Prüfungen sind auch eine erste Gelegenheit, um die Korrektheit der Anforderungen abzusichern. Werden derart frühzeitig Unklarheiten oder Unstimmigkeiten in den Anforderungen entdeckt, können sie noch mit vergleichsweise geringem Aufwand korrigiert werden.

Anforderungsstrukturierung: Testbare Anforderungen

Anforderungen sind dann gut testbar, wenn erstens die Form, in der die einzelnen Anforderungen beschrieben sind, bestimmten Qualitätskriterien entspricht und zweitens die Menge aller Anforderungen eines Systems vollständig und gut strukturiert ist.

Damit eine einzelne Anforderung testbar ist, muss sie granular und modular sein sowie präzise und prägnant beschrieben sein. Außerdem sollte sie ein sogenanntes Prüfkriterium (*Fit Criterion*) besitzen. Die erstgenannten Eigenschaften fordern, dass die Anforderungen sozusagen „in kleinen Häppchen“ formell und stilistisch gut definiert sind—also nicht in umfangreichen, kaum strukturierten Prosadokumenten, und ohne Nominalisierungen, Passiv etc. Das Prüfkriterium definiert, wie man später feststellt, ob eine Anforderung erfüllt ist. Es präzisiert die Anforderung und stellt zugleich einen ersten Schritt zu den Testfällen dar.

Damit die Anforderungen in ihrer Gesamtheit testbar sind, müssen sie mithilfe eines geeigneten Kategoriensystems strukturiert sein, beispielsweise entsprechend dem Volere-Template [RR06] oder IEEE Standard 830-1998 [IEE98]. Solche Kategorien machen eine Anforderungsspezifikation übersichtlich und erleichtern es, sie auf Vollständigkeit und Redundanzfreiheit zu überprüfen.

So sollten die Anforderungen mittels Kategorien zunächst in Abstraktionsebenen eingeteilt sein, typischerweise in Geschäftsanforderungen (*Business Requirements*) und Systemanforderungen. Die Geschäftsanforderungen spiegeln die Sicht der Auftraggeber wider. Jeder Geschäftsanforderung entsprechen eine oder mehrere Systemanforderungen, bei denen die Sicht der Software-Entwicklung im Vordergrund steht.

Die Anforderungskategorien erleichtern auch den Übergang von den Anforderungen zur Testplanung und Testvorbereitung. Denn meist werden die Anforderungen einer Kategorie auf eine bestimmte Art getestet.

Die folgenden Anforderungsarten sind für das Testen von besonderer Bedeutung. Sie korrespondieren mit den vier wesentlichen Testarten:

- Funktionale Anforderungen (bzgl. funktionale Tests)
- Performance-Anforderungen (bzgl. Last- und Performance-Tests)
- Benutzbarkeitsanforderungen (bzgl. Usability Tests)
- Sicherheitsanforderungen (bzgl. Application Security Tests)

Weitere Anforderungskategorien zielen speziell auf das Testen ab: *Test-Projektanforderungen* sind Projektanforderungen hinsichtlich der Testdurchführung, wie die zeitliche Verfügbarkeit von Systemteilen oder Festlegungen (z.B. Schnittstellen- oder Datenformatsdefinitionen). Sie gewährleisten die systematische Testvorbereitung und -durchführung. *Testbarkeitsanforderungen* sind funktionale Anforderungen an das System, die speziell dazu dienen, dass die erforderlichen Tests durchgeführt werden können (z.B. besondere Schnittstellenfunktionen, die nur für Tests benötigt werden). Auch die *Betriebsanforderungen* nehmen eine besondere Rolle hinsichtlich des Testens ein. Denn viele Tests werden während des Systembetriebs durchgeführt (z.B. Installationserfolgstests oder Last- und Performance-Tests).

Anforderungsstrukturierung: Anforderungsbasiertes Testen

Anforderungsbasiertes Testen setzt voraus, dass zu den Testfällen die zugrunde liegenden Anforderungen definiert und vermerkt sind (Verfolgbarkeit, Traceability). Das ist der Fall, wenn die Testfälle systematisch aus den Anforderungen abgeleitet werden. Häufig aber existieren Tests, zu denen die Anforderungen nicht ausreichend dokumentiert oder zumindest nicht explizit vermerkt sind. – Wie kann eine Software-Organisation in dieser Situation die Vorteile des anforderungsbasierten Testens schnell nutzen?

Viele Vorteile des anforderungsbasierten Testens erschließen sich bereits, wenn die Testfälle nach Anforderungskategorien gruppiert sind. Dies ermöglicht zum Beispiel risikobasiertes Testen und Monitoring der Testdurchführung nach Geschäftsprozessen. Eine ausführliche Definition der betreffenden Anforderungen kann später schrittweise ergänzt werden.

Dazu werden Anforderungskategorien benötigt, die die existierenden Testfälle strukturieren. Dies sind im Grunde die gleichen Kategorien wie oben unter Testbare Anforderungen beschrieben. In vielen Fällen reichen für die Strukturierung existierender Tests jedoch zunächst die folgenden drei Kategorien:

Geschäftsprozesse: Eine Hierarchie von Geschäftsprozessen, in denen die zu testenden Systeme eingesetzt werden. Diese Hierarchie dient dazu, die Testfälle aus Sicht der Geschäftsanforderungen (*Business Requirements*) zu strukturieren. Beispielsweise können die Tests so nach Relevanz aus Sicht der Auftraggeber gruppiert und priorisiert werden.

Systemstruktur: Eine Hierarchie der Systemaggregation aus den zu testenden Gesamtsystemen, ihren Teilsystemen und Komponenten. Diese Hierarchie dient dazu, Tests nach zu testenden Systemeinheiten auszuwählen, zum Beispiel für Regressionstests nach Änderungen.

Anforderungsarten: Eine Hierarchie aus wesentlichen funktionalen und nichtfunktionalen Anforderungsarten sowie deren Unterkategorien—z.B. Funktionalitätsgruppen und nicht-funktionale Anforderungen wie Performance- und Sicherheitsanforderungen. Diese Hierarchie dient dazu, Testfälle zu bestimmten Aspekten schnell auszuwählen und fokussiert zu testen (z.B. die Benutzerverwaltung oder das Performance-Verhalten).

Anforderungsbasierte Testplanung und -priorisierung

Tests müssen immer priorisiert und nach Maßgabe der verfügbaren Ressourcen durchgeführt werden. Ein sehr wichtiges Kriterium für die Priorisierung sind die geschäftlichen Risiken (d.h. Kritikalität und Eintrittswahrscheinlichkeit), die mit den unterschiedlichen Anforderungen verbunden sind. Diejenigen Anforderungen, die besonders geschäftskritisch sind, müssen vordringlich getestet werden.

Grundlagen für eine solche risikobasierte Testplanung sind die oben beschriebenen Anforderungskategorien, insbesondere die Geschäftsanforderungen. Wenn sie mit den Testfällen verbunden sind, können die Testfälle gemäß der Geschäftskritikalität ausgewählt werden. Mithilfe der Geschäftsanforderungen lässt sich die

Risikobewertung gemeinsam mit den Auftraggebern sehr viel leichter vornehmen, als es allein auf Grundlage der Testfälle möglich wäre.

Integriertes Anforderungs- und Fehlermanagement

Das Fehlermanagement vermerkt die im Test identifizierten Fehler und plant und dokumentiert ihre Analyse und Behebung. Die Aktivitäten des Fehlermanagements ähneln sehr denen des kontinuierlichen Anforderungsmanagements [Sch02]. Außerdem kann identifiziertes Fehlverhalten dazu führen, dass neue Anforderungen definiert oder existierende Anforderungen geändert werden.

Aus diesen Gründen sollte in einer Software-Organisation ein zentrales, integriertes Anforderungs- und Fehlermanagement installiert sein. Es sollte sowohl die Software-Entwicklung umfassen wie auch IT-Betrieb, Benutzer der Fachbereiche sowie Support und Service-Desk.

Organisationsstrukturen und Rollen für Anforderungen und Testen

In vielen Software-Organisationen ist das Requirements-Management keiner einzelnen Rolle oder organisatorischen Einheit zugeordnet. Vielmehr sind jeweils Teilaufgaben des Requirements-Managements auf mehrere Rollen verteilt, die eigentlich andere Aufgabenschwerpunkte besitzen: Produkt- und Portfoliomanagement, Software-Architektur und Projektleitung.

Testen und Testmanagement hingegen bilden meist klare Aufgabenschwerpunkte, die einem festen Personenkreis zugeordnet sind, in der Regel als Teil des Qualitätsmanagements. Diese Organisationsstruktur des Testens kann als Ausgangspunkt dienen, um auch das Requirements-Management zu zentralisieren und so die Qualität der Software und die Produktivität der Entwicklung weiter zu steigern.

Das folgende Fallbeispiel belegt den Nutzen und die Durchführbarkeit einer solchen Strategie: Die Software-Abteilung eines Finanzdienstleisters hat die Testaktivitäten schrittweise zentralisiert und damit auch die Grundlage für die Zentralisierung des Requirements-Managements geschaffen. Am Anfang wurde die Verantwortung für Last- und Performance-Tests aller Anwendungen in ein zentrales Testteam überführt. Danach übernahm dieses Team auch die Verwaltung der funktionalen Testfälle, die zuvor von mehreren Fachbereichen getrennt gepflegt wurden. Dadurch ist eine Infrastruktur etabliert worden, die nun auch für das zentrale Anforderungsmanagement und somit für anforderungsbasiertes Testen genutzt werden kann.

Oft leiden Qualität und Effizienz des Testens darunter, dass die Tester zu wenig über die fachlichen Anforderungen und die technische Realisierung des Systems Bescheid wissen. Mögliche Maßnahmen dagegen sind Job-Rotation und enge Kommunikation und Zusammenarbeit zwischen den Testern und den anderen Rollen. Im Rahmen der Job-Rotation sollten Anforderungs- und Architektur-Experten zeitweise Test-Rollen besetzen. Zusammenarbeit und Kommunikation können durch Quality-Gates für Anforderungs- und Test-Dokumente gefördert werden, bei denen immer auch Requirements-Manager, Software-Architekten und Testmanager beteiligt sein müssen.

Literatur

[IEE98] IEEE Recommended Practice for Software Requirements Specification. IEEE STD 830-1998, 1998.

[ISO04] International Standards Organization (ISO) / International Electrotechnical Commission (IEC). Information technology — Software life cycle processes AMENDMENT 2 (ISO/IEC 12207:1995/Amd.1:2004(E)), Mai 2004.

[RR06] Suzanne Robertson and James Robertson. Mastering the requirements process (2nd Ed.). Addison-Wesley Publishing Co., New York, NY, 2006.

[Sch02] Bruno Schienmann. Kontinuierliches Anforderungsmanagement: Prozesse – Techniken - Werkzeuge. Addison-Wesley, München, 2002.

[Spi02] Andreas Spillner. Management des Testprozesses von Anfang an: Das W-Modell. GI-Konferenz Software Management 2002, Hamburg, November 2002.