



XSTUDIE

Änderungsmanagement in der Softwareentwicklung

Werkzeuge und ihre Schnittstellen im Änderungsmanagement
eines Softwareentwicklungsprozesses

Jonathan Koch



Studie

Werkzeuge und ihre Schnittstellen im
Änderungsmanagement eines
Softwareentwicklungsprozesses

Ein Vergleich

von

Jonathan Koch

Dortmund, 11. Oktober 2011

Markenrechtlicher Hinweis

Alle genannten Firmen-, Markennamen und Warenzeichen können auch ohne besondere Kennzeichnung geschützte Marken sein und gesetzlichen Bestimmungen unterliegen.

Inhaltsverzeichnis

1 Einleitung.....	1
1.1 Motivation.....	1
1.2 Zielsetzung.....	2
1.3 Aufbau der Studie.....	2
1.4 Verwandte Arbeiten.....	3
2 Grundlagen.....	9
2.1 Besondere Bedeutung des Änderungsmanagements.....	9
2.2 Verbundene Bereiche im Softwareentwicklungsprozess.....	14
2.2.1 Anforderungsmanagement.....	14
2.2.2 Konfigurationsmanagement.....	17
2.2.3 Testmanagement.....	19
2.3 Kernfunktionalitäten entsprechender Werkzeuge.....	21
2.3.1 Kernfunktionalitäten im Anforderungsmanagement.....	21
2.3.2 Kernfunktionalitäten im Änderungsmanagement.....	22
2.3.3 Kernfunktionalitäten im Konfigurationsmanagement.....	23
2.3.4 Kernfunktionalitäten im Testmanagement.....	25
2.4 Schnittstellen.....	26
2.4.1 Notwendigkeit von Schnittstellen.....	26
2.4.2 Schnittstellen des Änderungsmanagements.....	27
3 Kriterienkatalog.....	29
3.1 Herkunft.....	29
3.2 Rahmenbedingungen.....	30
3.3 Kriterienauswahl.....	30
3.4 Priorisierung.....	31
3.5 Kriterien.....	32
3.5.1 Lizenzierung.....	32
3.5.2 Datenhaltung.....	32
3.5.3 Systemplattformen.....	33
3.5.4 Systemschnittstellen.....	33
3.5.5 Anpassbarkeit.....	35

3.5.6 Mandantenfähigkeit.....	36
3.5.7 Such- und Auswertungsmöglichkeiten.....	37
3.5.8 Benutzerschnittstellen.....	38
3.5.9 Unterstützung und Support.....	39
4 Marktanalyse.....	41
4.1 Werkzeugtypen.....	41
4.2 Suchkriterien.....	43
4.3 Werkzeuge.....	43
4.3.1 ATLIASSIAN – JIRA.....	43
4.3.2 Axosoft – OnTime.....	45
4.3.3 Borland – StarTeam.....	46
4.3.4 IBM – Rational Change.....	48
4.3.5 IBM – Rational ClearQuest.....	50
4.3.6 IBM – Rational Team Concert.....	51
4.3.7 Seapine Software – TestTrack Pro.....	53
5 Werkzeugvergleich.....	55
5.1 Vergleichsverfahren.....	56
5.2 Vergleich.....	56
5.2.1 Lizenzierung.....	57
5.2.2 Datenhaltung.....	57
5.2.3 Systemplattformen.....	58
5.2.4 Systemschnittstellen.....	60
5.2.5 Benutzerschnittstellen.....	62
5.2.6 Unterstützung und Support.....	63
5.2.7 Anpassbarkeit.....	65
5.2.8 Mandantenfähigkeit.....	68
5.2.9 Such- und Auswertungsmöglichkeiten.....	70
6 Ergebnis des Vergleichs.....	73
6.1 Bewertungsverfahren.....	73
6.2 Auswertung.....	75
6.3 Ergebnis.....	78

7	Werkzeugtest.....	80
7.1	TestszENARIO.....	80
7.2	Testsysteme.....	86
7.2.1	TestTrack-Server.....	86
7.2.2	TestTrack-Linux-Client.....	87
7.2.3	TestTrack-Windows-Client.....	88
7.2.4	JIRA-Server.....	89
7.2.5	JIRA-Eclipse-Interface.....	90
7.3	Testdurchführung.....	90
7.3.1	TestTrack Pro.....	91
7.3.2	JIRA.....	96
7.4	Fazit.....	101
8	Schnittstellenanalyse.....	104
8.1	Kriterienkatalog.....	104
8.1.1	Herkunft.....	104
8.1.2	Kriterien.....	105
8.2	TestszENARIO.....	107
8.3	Testumgebungen.....	109
8.3.1	TestTrack Pro – TestTrack RM.....	109
8.3.2	TestTrack Pro – Surround SCM.....	110
8.3.3	TestTrack Pro – TestTrack TCM.....	110
8.3.4	JIRA – Jama Contour.....	110
8.3.5	JIRA – Perforce SCM.....	111
8.3.6	JIRA – Enterprise Tester.....	112
8.4	Testdurchführung.....	112
8.4.1	TestTrack Pro.....	113
8.4.2	JIRA.....	115
8.5	Fazit.....	116
9	Anhang.....	119
9.1	Werkzeugvergleich.....	119
9.1.1	Legende.....	119
9.1.2	Grober Teilvergleich.....	120

Inhaltsverzeichnis

9.1.3 Ausführlicher Komplettvergleich.....	122
9.1.4 Gesamtübersicht.....	143
9.2 Vergleichsergebnis.....	149
9.3 Werkzeugtest.....	154
9.3.1 Testsysteme.....	154
9.3.2 TestTrack-Server.....	154
9.3.3 TestTrack-Linux-Client.....	158
9.3.4 TestTrack-Windows-Client.....	161
9.3.5 JIRA-Server.....	164
9.4 Schnittstellenanalyse.....	167
9.4.1 TestTrack Pro – TestTrack RM / TCM.....	167
9.4.2 TestTrack Pro – Surround SCM.....	168
9.4.3 JIRA – Jama Contour.....	173
9.4.4 JIRA – Perforce SCM.....	177
9.4.5 JIRA – Enterprise Tester.....	182
10 Abbildungsverzeichnis.....	185
11 Tabellenverzeichnis.....	186
12 Literaturverzeichnis.....	187

Kurzfassung

Diese Studie beschäftigt sich mit Werkzeugen für das Änderungsmanagement in der Softwareentwicklung und ihren Schnittstellen zum Anforderungs-, Konfigurations- und Testmanagement. Dabei wird zuerst erläutert, warum das Änderungsmanagement an sich und vor allem die Verbindung zu umliegenden Entwicklungsdisziplinen wichtig ist und anschließend ein Katalog mit 54 priorisierten Kriterien für Änderungsmanagement-Werkzeuge aufgestellt. Daraufhin werden nach einer Marktanalyse die Werkzeuge

- ATlassian – JIRA
- IBM – Rational Change
- Axosoft – OnTime
- IBM – Rational ClearQuest
- Borland – StarTeam
- IBM – Rational Team Concert
- Seapine Software – TestTrack Pro

vorgestellt und auf Basis des Kriterienkatalogs ausführlich miteinander verglichen. Nach der Vergleichsauswertung werden die Produkte JIRA und TestTrack Pro – die im Vergleich am besten abgeschnitten haben – installiert, eingerichtet und in einem Testszenario, welches die Hälfte der aufgestellten Kriterien enthält, näher untersucht. Zum Schluss wird eine Schnittstellenanalyse durchgeführt. Hier werden die Schnittstellen der beiden Werkzeuge zum jeweiligen Anforderungs-, Konfigurations- und Testmanagement-Werkzeug konfiguriert und wiederum in einem Szenario auf zuvor durch Kriterien festgelegte Eigenschaften überprüft.

Bei JIRA konnte nur die Schnittstelle zum Anforderungsmanagement-Werkzeug konfiguriert werden. Die Einrichtung der restlichen Schnittstellen schlug leider fehl.

1 Einleitung

1.1 Motivation

Seit vielen Jahren werden in den unterschiedlichsten Branchen – beispielsweise der Luft- und Raumfahrttechnik, der Automobil- und Verteidigungsindustrie, im Gesundheits- und Transportwesen, in der Informations- und Kommunikationstechnik sowie der Logistik – verschiedene Software-Lösungen für das Anforderungs-, Konfigurations- und Änderungsmanagement eingesetzt.

In dieser Studie sollen nun unterschiedliche Software-Lösungen für das Änderungsmanagement miteinander verglichen und anhand eines vorher definierten Kriterienkatalogs bewertet werden. Der Grund für den Vergleich ist der, dass sich manche Lösungen zwar für große Projekte eignen, für kleinere Projekte jedoch – vor allem aus Kostengründen – eher ungeeignet sind. Oft wird in diesen Projekten auf kostenlose oder selbst entwickelte Lösungen zurückgegriffen, die nicht nur in ihrem Funktionsumfang den kommerziellen Produkten nachstehen, sondern meistens auch in ihrer Stabilität und Performance. Die vielfältigen Anforderungen, die von den Projektbeteiligten durch unterschiedliche Projekteigenschaften und Rahmenbedingungen an ein Änderungsmanagement-Werkzeug gestellt werden, können in der Regel nicht erfüllt werden. So fehlt teilweise eine ausführliche und ausreichende Dokumentation, ein ständig verfügbarer Support oder die Möglichkeit, individuelle Anpassungen vorzunehmen. Bei den selbst entwickelten Lösungen müssen außerdem genügend Ressourcen für die Weiterentwicklung und die Fehlerbehebung vorhanden sein. Zusätzlich sind in den hochpreisigen Werkzeugen Rational DOORS, Rational Synergy und Rational Change Schnittstellen vorhanden, um zwischen Anforderungen, Konfigurationseinheiten und Änderungsanträgen Beziehungen zu erstellen und zu verwalten. Die Vorteile, die sich daraus ergeben, sind unter anderem ein nachvollziehbarer und durchgehend dokumentierter Entwicklungsprozess sowie die Reduktion von Datenredundanzen und Fehlerquellen. In den kostenlosen und selbst entwickelten Lösungen fehlen derartige Schnittstellen meistens oder sind nur ansatzweise vorhanden.

Das Problem besteht also darin, dass ein Werkzeug benötigt wird, welches auf der einen Seite aus finanzieller Sicht auch für kleine Projekte geeignet ist und auf der anderen Seite die umfangreichen und unterschiedlichen Anforderungen in verschiedenartigen Projekten erfüllt.

Zudem müssen mindestens Schnittstellen zum Anforderungs- und Konfigurationsmanagement existieren.

Die Evaluation von alternativen Änderungsmanagement-Werkzeugen, die die jeweiligen Anforderungen erfüllen und Schnittstellen zu anderen Entwicklungsmanagement-Werkzeugen besitzen, ist deshalb die Hauptaufgabe dieser Studie. Dabei kann gesagt werden, dass die im Laufe der Studie aufgestellten Anforderungen allgemeingültige Kriterien an ein Änderungsmanagement-Werkzeug darstellen.

Des Weiteren wird erläutert, warum die erwähnten Schnittstellen von besonderer Bedeutung sind und bei einem Änderungsmanagement-Werkzeug nicht fehlen sollten.

1.2 Zielsetzung

1. Das Hauptziel dieser Studie ist es, durch einen detaillierten Vergleich herauszufinden, welche Werkzeuge für das Änderungsmanagement in der Softwareentwicklung am ehesten geeignet sind. Dabei sollen die in einem Kriterienkatalog aufgestellten und priorisierten Anforderungen berücksichtigt werden und für die Beurteilung ausschlaggebend sein. Nachdem auf Basis des Kriterienkatalogs verschiedene Werkzeuge ausgewählt worden sind, sind diese in einem Szenario zu testen.
2. Bevor der Werkzeugvergleich und die zuvor notwendige Marktanalyse durchgeführt wird, soll außerdem deutlich werden, warum das Änderungsmanagement in der Softwareentwicklung von besonderer Bedeutung ist und nicht als autonome Disziplin angesehen werden darf.
3. Das dritte Ziel dieser Studie ist es, aufzuzeigen, welche Schnittstellen die Werkzeuge anbieten, um mit anderen in der Softwareentwicklung existierenden Werkzeugen – speziell für das Anforderungs-, Konfigurations- und Testmanagement – verbunden zu werden. Anschließend sollen diese Schnittstellen bei den ausgewählten Produkten durch eine Analyse näher untersucht werden.

1.3 Aufbau der Studie

Nach der Einleitung folgt ein Grundlagenkapitel (Kapitel 2). Dieses Kapitel erläutert die besondere Bedeutung des Änderungsmanagements und stellt wichtige verbundene Bereiche im Softwareentwicklungsprozess vor. Anschließend werden die Kernfunktionalitäten

entsprechender Entwicklungswerkzeuge aufgeführt und es wird auf die Notwendigkeit von Schnittstellen sowie auf die konkreten Vorteile der Schnittstellen eingegangen.

Daraufhin wird ein Kriterienkatalog (Kapitel 3) aufgestellt, der neben bestimmten Rahmenbedingungen priorisierte Anforderungen an ein Änderungsmanagement-Werkzeug enthält.

Nach der Definition der Kriterien wird eine Marktanalyse (Kapitel 4) durchgeführt und es werden die gefundenen Werkzeuge sowie ihre Hersteller vorgestellt.

Anschließend werden die Werkzeuge nach einem festgelegten Vergleichsverfahren miteinander verglichen (Kapitel 5) und das Ergebnis wird ausgewertet (Kapitel 6).

Bei den Werkzeugen, die den Vergleich am besten abgeschlossen haben, wird daraufhin ein Werkzeugtest (Kapitel 7) und zum Schluss eine Schnittstellenanalyse (Kapitel 8) durchgeführt. Der Werkzeugtest verifiziert die aufgestellten Kriterien und untersucht in der Schnittstellenanalyse die Schnittstellen der Werkzeuge genauer.

1.4 Verwandte Arbeiten

Die in diesem Kapitel aufgelisteten Arbeiten (Bücher, Studien und Umfragen) sind mit dem Thema dieser Studie verwandt oder behandeln vergleichbare Aufgabenstellungen.

Zuerst werden drei Bücher genannt, die das Änderungsmanagement in der Softwareentwicklung darstellen. Im Anschluss ist ein Buch aufgeführt, welches die Bedeutung von Schnittstellen im Softwareentwicklungsprozess erläutert und die Schnittstellen im Anforderungsmanagement benennt. Im dritten Teil sind fünf Studien und eine Umfrage angegeben, die verschiedene Werkzeuge im Anforderungs-, Konfigurations-, Versions-, Änderungs-, Fehler- oder Testmanagement miteinander vergleichen.

Eine Arbeit, die einen detaillierten Vergleich speziell von Änderungsmanagement-Werkzeugen durchführt, konnte leider nicht gefunden werden. Die im dritten Teil aufgeführte Studie „Bessere Software – den Entwicklungsprozess im Griff“ ([iX-Studie 05]) vergleicht zwar unterschiedliche Konfigurations- und Änderungsmanagement-Werkzeuge, konzentriert sich allerdings auf die Funktionalitäten für das Konfigurationsmanagement (Kapitel 2.3.3).

1. Änderungsmanagement

Im Folgenden sind drei Bücher vorgestellt, die sich unter anderem mit dem Änderungsmanagement in der Softwareentwicklung beschäftigen und dessen Bedeutung in einem Softwareentwicklungsprozess erläutern.

- | |
|--|
| ○ Titel: Change Management bei Software-Projekten
[Versteegen 01] |
|--|

- *„Eine der wesentlichen Ursachen für das Scheitern von Softwareentwicklungsprojekten ist, dass sich die vorab formulierten Anforderungen ändern. Nur wenn diese Änderungen professionell gemanagt werden, kann ein Software-Projekt erfolgreich in Zeit und Budget fertiggestellt werden. Schwerpunkte dieses Buches bilden Grundlagen und Konzepte eines professionellen Anforderungs- und Änderungsmanagements. Mögliche Werkzeuge, die das Anforderungs- und Änderungsmanagement erleichtern und zum Teil auch automatisieren, werden ebenfalls vorgestellt. [...]“*

[Versteegen 01] Buchbeschreibung

- In diesem Buch werden zuerst die möglichen Ursachen für Anforderungsänderungen erörtert (S. 67 ff.). Daraufhin wird das Anforderungsmanagement im Rational-Unified-Prozess ([RUP]) und mit den von Rational Software¹ entwickelten Produkten Rational Suite AnalystStudio² und Rational RequisitePro betrachtet (S. 93 ff., S. 123 ff.). Zum Schluss wird die besondere Bedeutung des Anforderungs- und Änderungsmanagements in Entwicklungsprojekten beschrieben (S. 209 ff.).

- | |
|---|
| ○ Titel: Lehrbuch der Softwaretechnik: Softwaremanagement
[Balzert 08] |
|---|

- *„Während der Entwicklung – aber insbesondere in der Nutzungsphase – entstehen aus verschiedenen Gründen Änderungswünsche an das Produkt. [...] Um solche Probleme zu vermeiden, muss sowohl für bereits freigegebene*

1 Das Unternehmen *Rational Software* wurde inzwischen von IBM aufgekauft (siehe auch Kapitel 4.3.4).

2 Das Produkt *Rational Suite AnalystStudio* ist im Produktkatalog vom IBM nicht mehr vorhanden.

Softwareelemente während der Entwicklung als auch für Änderungen in der Nutzungsphase ein definierter Managementprozess existieren.“

[Balzert 08] S. 429

- In diesem Buch wird unter anderem das Konfigurations- und Änderungsmanagement als notwendige Teilaufgabe des Softwaremanagements beschrieben (S. 427 ff.).

<ul style="list-style-type: none">○ Titel: Managing the Change: Software Configuration and Change Management [Haug 01]

- *„The current volume offers a variety of perspectives on software configuration/change management and requirements engineering. These areas have a significant impact on the success of a software product and present a high potential for improvement as many organizations, especially small and medium-sized enterprises, have no purposeful process addressing these issues.“*

[Haug 01] Buchbeschreibung

- In diesem Buch werden die Erfahrungen zusammengefasst, die im Laufe mehrerer Software-Projekte im Konfigurations- und Änderungsmanagement gesammelt wurden. Als Ergebnis werden Best Practices formuliert, um in der Softwareentwicklung erfolgreiches Konfigurations- und Änderungsmanagement zu betreiben.

2. Schnittstellen

Im Folgenden ist ein Buch genannt, welches die Notwendigkeit von Schnittstellen zwischen dem Anforderungs-, Konfigurations-, Änderungs- und Testmanagement beschreibt.

<ul style="list-style-type: none">○ Titel: Requirements Management: The Interface Between Requirements Development and All Other Systems Engineering Processes [Hood 08]

- *„This book focuses on the interfaces of Requirements Management to the other disciplines of Systems Engineering, for example Project Management, Change Management and Configuration and Version Management.“*

[Hood 08] Buchbeschreibung

- Nach einer Einführung in die Anforderungsverwaltung und -entwicklung werden in diesem Buch die notwendigen Schnittstellen im Anforderungsmanagement genannt. In Kapitel 11 („Change Management Interface“, S. 175 ff.) wird die Schnittstelle zum Änderungsmanagement beschrieben.

3. Werkzeugvergleiche

Im Folgenden sind Studien und Umfragen aufgeführt, die verschiedene Werkzeuge im Anforderungs-, Konfigurations-, Versions-, Änderungs-, Fehler- oder Testmanagement miteinander vergleichen.

○ Titel: Anforderungsmanagement [iX-Studie 07]

- *„Die Studie hat dabei einerseits die Zielrichtung, als einführende Literatur in die Thematik des Anforderungsmanagements zu dienen, andererseits liegt der Schwerpunkt der Studie in der Betrachtung der derzeit auf dem deutschsprachigen Markt verfügbaren Werkzeuge und deren Hersteller.“*

[iX-Studie 07] S. 11

- In dieser Studie werden Grundlagen, Methoden und Techniken der Anforderungsentwicklung und -verwaltung erläutert (S. 15 ff.) sowie 14 Anforderungsmanagement-Werkzeuge untersucht (S. 122 ff.).

○ Titel: INCOSE Requirements Management Tools Survey [INCOSE RM Tools]

- Bei dieser Veröffentlichung handelt es sich um eine – auf einer Umfrage basierenden – Zusammenstellung von zurzeit 34 verschiedenen Anforderungs-

rungsmanagement-Werkzeugen. Alle Informationen stammen direkt von den Herstellern der Werkzeuge und können regelmäßig aktualisiert werden.

○ **Titel: Bessere Software – den Entwicklungsprozess im Griff**
 [iX-Studie 05]

- *„Danach stellen wir [...] die Konfigurations- und Change-Management-Produkte vor, die derzeit auf dem deutschen Markt verbreitet sind.“*

[iX-Studie 05] S. 10

- In dieser Studie werden zu Anfang 12 „Prinzipien für bessere Software“ genannt und daraufhin acht Konfigurationsmanagement-Werkzeuge – die teilweise auch das Änderungsmanagement unterstützen – sowie zehn UML-Werkzeuge untersucht. Neben den möglichen Plattformen werden bei den Konfigurations- und Änderungsmanagement-Werkzeugen vor allem die Versions-, Change-, Release- und Build-Management-Funktionalitäten näher betrachtet.

○ **Titel: Study of Available Tools**
 [FLOSS 08]

- Diese Studie beschäftigt sich unter anderem mit der Frage, ob es eine Möglichkeit gibt, aus verschiedenen Versions- und Fehlermanagement-Werkzeugen, Mail-Diensten und Wiki-Systemen Daten zu exportieren. Daher wird in dieser Studie unter anderem die Datenhaltung bei verschiedenen Versions- und Fehlermanagement-Werkzeugen betrachtet.

○ **Titel: Testmanagement- & Testautomatisierungswerkzeuge**
 [Quality Lab 08]

- In dieser Studie werden sieben Testmanagement- und Testautomatisierungswerkzeuge miteinander verglichen. Die untersuchten Kriterien stammen unter anderem aus den Bereichen „Administration“, „Benutzerverwaltung“, „Design“ und „Testaktivitäten“ (Testplanung, Steuerung, Analyse usw.).

○ Titel: Software-Testmanagement [iX-Studie 06]
--

- Nachdem am Anfang der Studie verschiedene Aspekte des Testmanagements – beispielsweise die Testplanung, -durchführung und -steuerung, das Testdesign und die Testautomatisierung – beleuchtet wurden, wird ein Fragenkatalog aufgestellt und ein Vergleich von 13 verschiedenen Testmanagement-Werkzeugen durchgeführt.

2 Grundlagen

2.1 Besondere Bedeutung des Änderungsmanagements

„Beachte immer, dass nichts bleibt, wie es ist, und denke daran, dass die Natur immer wieder ihre Formen wechselt.“ Marc Aurel (121–180), römischer Kaiser

Veränderungen sind das Natürlichste, was es gibt, und begleiten schon seit jeher die Menschheit in ihrem Dasein. Das hat damals auch Marc Aurel erkannt. Egal, ob man sich die Natur ansieht, die Menschen oder ihr vielfältiges Handeln: Veränderungen sind stets allgegenwärtig. Das Beständigste, was es gibt, sind Veränderungen. Trotzdem rechnet man selten mit ihnen und ist oft verwundert, wenn sie eintreten. Manche Veränderungen sind gewollt, man kann sie einleiten und steuern. Andere wiederum sind nicht gewollt, treten von ganz alleine ein und lassen sich nicht lenken. Aber unabhängig davon, wie eine Veränderung aussieht und zustande kommt, sie hinterlässt meistens Spuren.

Eine bekannte Redensart sagt aus, dass der Mensch ein Gewohnheitstier ist. Er gewöhnt sich an seine Umgebung und seinen Alltag. Sein täglicher Tagesablauf ist Routine. Tätigkeiten, an die er sich gewöhnt hat, und Aufgaben, die immer auf demselben Weg zu erledigen sind, kann er am schnellsten und zuverlässigsten bewältigen. Steht er vor einer neuen Aufgabe, plant er mehr oder weniger einen entsprechenden Lösungsweg mit Handlungsschritten. Treten auf einem seiner Wege ungeplante Veränderungen ein und veranlassen diese ihn dazu, an seinem Handeln Änderungen vorzunehmen, wird er aufgehalten oder schlimmstenfalls zum Scheitern verurteilt. Je komplexer sein Lösungsweg ist, desto komplexer können die Änderungen sein, die er vornehmen muss, um sich an eintretende Veränderungen anzupassen und das Gesamtergebnis zu erreichen. Dazu kommt, dass einzelne Handlungsschritte voneinander abhängig sein können. Ist die zu ändernde Handlung – der zu ändernde Schritt – weitgehend autonom, sind Anpassungen schnell und sicher durchgeführt. Gibt es dagegen Schnittstellen zwischen einzelnen Handlungsschritten und weisen diese eine hohe Komplexität auf, muss er bei Änderungen immer darauf achten, dass verbundene Handlungsschritte in ihrem eigentlichen Sinn und Zweck nicht negativ beeinflusst werden. Im Endeffekt muss stets das Gesamtergebnis kontrolliert werden, wenn für ein Teilergebnis eine Änderung vorgenommen wurde.

Änderungsmanagement im Allgemeinen beschäftigt sich mit diesem Dilemma. Auf der einen Seite ist es keine Neuigkeit, dass in allen Bereichen Veränderungen gewollt oder ungewollt eintreten. Auf der anderen Seite besteht die Tatsache, dass diese uns oft beeinflussen, Änderungen an unserem Handeln notwendig machen und für Verzögerungen, erhöhte Fehleranfälligkeit und Qualitätsverlust sorgen. Daher ist es unausweichlich, von vornherein mit Änderungen zu rechnen, sie einzuplanen und ihr Management festzulegen, gerade dann, wenn es sich um ein Netz von Teilaufgaben handelt und Änderungen an deren Lösung stets das Gesamtergebnis beeinflussen können. In vielen Bereichen der Industrie wurde dies erkannt und wird Änderungsmanagement dazu verwendet, jegliche Form von hier stattfindender komplexerer Entwicklung zu unterstützen. Sei es die Organisations-, die Prozess-, die Projekt- oder die Produktentwicklung, jede dieser Disziplinen hat als ausschlaggebende Eigenschaft, dass sie sich aus vielen miteinander in Verbindung stehenden Aufgaben zusammensetzt, deren Lösungsqualität zur Qualität des Gesamtergebnisses beiträgt. Werden Änderungen zur Lösung einer Teilaufgabe vorgenommen und in einer anderen fälschlicherweise nicht berücksichtigt, ist das Gesamtergebnis gefährdet und der Entwicklungserfolg aussichtslos. Wirken sich Änderungen aus der einen Aufgabenlösung auf eine zweite negativ aus, ist der Änderungsprozess gescheitert. Viele Qualitäts-, Prozess- und Projektmodelle fordern daher explizit den Einsatz eines formalen Änderungsmanagements. So zum Beispiel die ISO 9000 ([ISO 9000]), das CMMI (Capability Maturity Model Integration, [CMMI]), PRINCE2 (Projects in Controlled Environments, [PRINCE2]) oder das vielfach in Behörden eingesetzte V-Modell XT ([V-Modell XT]).

„Das Änderungsmanagement oder Veränderungsmanagement beschreibt Funktionen und Prozesse, die in einer Organisation etabliert werden, um Änderungen an Produkten der Organisation kontrolliert und dokumentiert vorzunehmen.“

[Kannengiesser 07]

„Im Problem- und Änderungsmanagement werden Änderungswünsche, Fehler und Probleme, die während der Systementwicklung oder -nutzung auftreten, behandelt und gelöst.“ [V-Modell XT]

Wie die genannten Definitionen zeigen, geht es im Änderungsmanagement vor allem um eines: Prozesse bereitzustellen, die es ermöglichen, bereichsübergreifende und inhaltlich weit reichende Änderungen während der Systementwicklung oder -nutzung kontrolliert und dokumentiert zu behandeln und umzusetzen.

Auch die IT Infrastructure Library in ihrer aktuellen Version 3 ([ITIL V3]), die Best Practices für das IT-Service-Management aufzählt und inzwischen international als De-facto-Standard gilt, beschreibt im Buch „Service Transition“ das Change Management als eine eigene und notwendige Disziplin.

Betrachtet man die reine Softwareentwicklung, findet man immer öfter agile Entwicklungsmethoden wie Scrum ([Scrum]) oder Extreme Programming (XP, [Beck 03]). Diese gehen von vornherein von der Hypothese aus, dass Änderungen bei Softwareprojekten allgegenwärtig sind und zu Projektbeginn nicht alle Anforderungen bis ins kleinste Detail endgültig feststehen können. Stattdessen wird inkrementell entwickelt und phasenweise werden immer genauere Anforderungen definiert und umgesetzt, wobei am Anfang die Anforderungen mit der höchsten Priorität und dem höchsten Umsetzungsrisiko stehen ([iX 02/06] S. 39 f.).

Trotz der möglichen Komplikationen, die Änderungen mit sich bringen, haben sie durchaus ihre Existenzberechtigung. So sind Änderungen für jede Art von Entwicklung zwingend notwendig, um diese am Leben zu erhalten. Vergisst man beispielsweise, ein Produkt rechtzeitig an aktuelle Kundenwünsche anzupassen oder Organisationsstrukturen an die aktuelle Marktsituation, sind Verluste programmiert. Auch Prozess- und Projektabläufe müssen stetig Änderungen durchlaufen, um einerseits optimiert werden zu können und andererseits individuellen Prozess- und Projekteigenschaften zu genügen. Verpasst man den Augenblick, eine entsprechende Änderung durchzuführen, kann es passieren, dass Produkte nicht mehr verkauft werden, Prozesse und Projekte scheitern und Organisationen Verluste einfahren.

Egal in welcher Entwicklungsphase man sich gerade befindet, am Anfang bei der Anforderungsdefinition, in der eigentlichen Entwicklung oder am Ende in der Testphase, stets müssen notwendige Änderungen festgehalten, kontrolliert gesteuert und bereichsübergreifend fehlerfrei umgesetzt werden. Damit dies funktionieren kann, muss jede dieser Phasen mit dem Änderungsmanagement verbunden werden und bis zum Ende der Entwicklung verbunden bleiben. Nur so ist gewährleistet, dass notwendige Änderungen ausreichend bekannt, dokumentiert und kontrollierbar sind und nicht vergessen werden.

Auch nach der Entwicklung darf das Änderungsmanagement nicht eingestellt werden, da das Auftreten von Veränderungen und die damit bestehende Notwendigkeit von vorzunehmenden Änderungen so lange nicht abgeschlossen ist, wie das Produkt existiert ([Balzert 08] S. 429). Ändern sich zum Beispiel gesetzliche Bestimmungen, technische Komponenten oder kundenspezifische Interessen, gibt es auch nach der ersten Fertigstellung notwendige Änderungen, die umgesetzt werden müssen ([Versteegen 01] S. 67 ff.).

Damit das Verwalten von Änderungen funktioniert, muss jede Änderungsanforderung einen festgelegten Workflow durchlaufen. Dieser verhindert, dass einerseits Änderungen voreilig durchgeführt werden und andererseits notwendige Änderungen in Vergessenheit geraten. Außerdem muss jede vorgenommene Änderung – wie jede andere Entwicklung auch – vor ihrer Auslieferung kontrolliert und getestet werden. Abbildung 1 zeigt, wie solch ein Workflow aussehen kann.

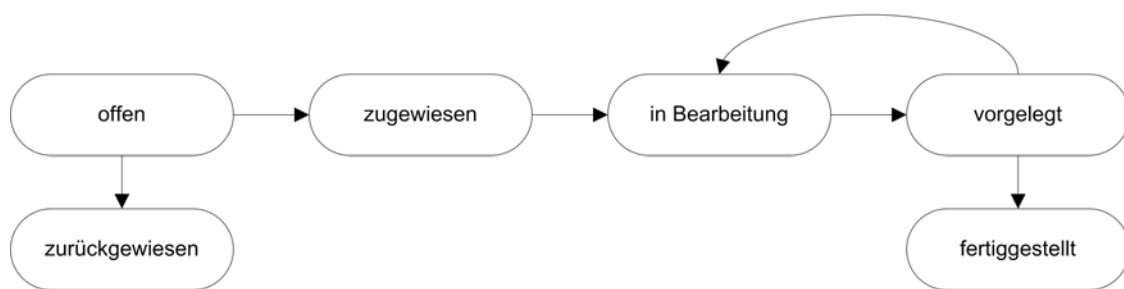


Abbildung 1: Grundlagen – Möglicher Workflow einer Änderungsanforderung

Eine neue Änderungsanforderung wird im Status „offen“ angelegt. Nach einer Prüfung, ob diese umgesetzt werden soll, wird sie einem Bearbeiter „zugewiesen“. Beginnt dieser mit seiner Arbeit, überführt er die Anforderung in den Status „in Bearbeitung“. In diesem Status verweilt die Änderungsanforderung, bis die Umsetzung vollzogen ist, und bekommt daraufhin den Status „vorgelegt“. Sind die folgenden Tests erfolgreich, bekommt die Änderungsanforderung den Status „fertiggestellt“ und ist damit bereit für die Auslieferung. Treten während der Tests Fehler auf, muss sie noch einmal bearbeitet werden und erhält erneut den Status „in Bearbeitung“. Soll nach der am Anfang stattfindenden Prüfung die Änderungsanforderung nicht umgesetzt werden, wird sie „zurückgewiesen“.

Wie bereits erwähnt, sind die Auswirkungen, die eine notwendige Änderung haben kann, von der Komplexität des zur Lösung der Aufgabe notwendigen Lösungswegs abhängig.