

Philippe Kruchten

Der Rational Unified Process

Eine Einführung

Deutsche Übersetzung von Cornelia Versteegen

 ADDISON-WESLEY

An imprint of Pearson Education

München • Reading, Massachusetts • Menlo Park, California
New York • Harlow, England • Don Mills, Ontario
Sydney • Mexico City • Madrid • Amsterdam

3 Statische Struktur: Prozeßbeschreibung

Dieses Kapitel beschreibt, wie der Rational Unified Process dargestellt wird. Dazu werden die Schlüsselkonzepte für *Worker*, *Aktivitäten*, *Artefakte* und *Workflows* eingeführt. Ebenso wird auf die weiteren Elemente der Prozeßbeschreibung eingegangen.

3.1 Ein Modell des Rational Unified Process

Ein Prozeß beschreibt, *wer was wann* und *wie* tut. Der Rational Unified Process wird durch die folgenden elementaren Modellierungselemente dargestellt:

- Worker: *wer*
- Aktivitäten: *wie*
- Artefakte: *was*
- Workflows: *wann*

Diese Modellierungselemente sind in Abbildung 3-1 dargestellt.

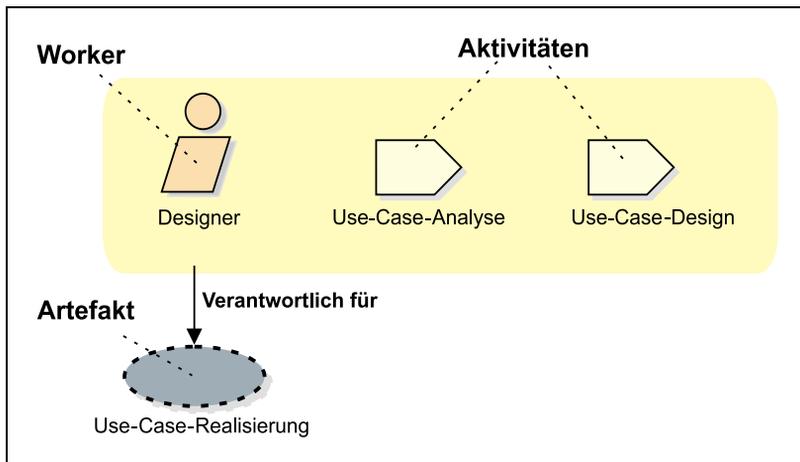


Abbildung 3-1 Worker, Aktivitäten und Artefakte

Der Begriff *Worker* definiert das Verhalten und die Verantwortlichkeit einer einzelnen Person oder einer Gruppe von Personen, die als Team zusammenarbeiten. Das Verhalten wird durch eine Anzahl von *Aktivitäten* beschrieben, die vom Worker durchgeführt werden. Jedem Worker wird dabei eine Anzahl von zusammen-

hängenden Aktivitäten zugeordnet. Unter »zusammenhängend« werden in diesem Kontext diejenigen Aktivitäten verstanden, die am besten von einer Person durchgeführt werden können, wo eine Aufteilung auf weitere Personen keinen Sinn ergibt. Die Verantwortlichkeiten jedes Workers werden in Beziehung zu den *Artefakten* gesetzt, die er erzeugt, ändert oder kontrolliert.

3.2 Worker

Der Begriff Worker umschreibt den »Hut«, den eine Person während der Abwicklungszeit eines Projekts trägt. Dabei kann eine Person durchaus mehrere unterschiedliche Hüte im Projektverlauf tragen. Diese Festlegung ist deshalb von Bedeutung, da normalerweise unter dem Begriff Worker eine Person oder ein Team verstanden wird. Im Rational Unified Process hingegen bezieht sich Worker auf eine Rolle, die festlegt, wie eine Person eine bestimmte Tätigkeit ausführen soll. Ein Worker nimmt eine oder auch mehrere Rollen ein und ist Eigentümer einer bestimmten Anzahl von Artefakten.

Im folgenden werden einige Beispiele für Worker aufgelistet:

- *Systemanalytiker*
Unter einem Systemanalytiker wird eine Person verstanden, die die Anforderungsanalyse verantwortlich leitet, koordiniert und durch eine Use-Case-Modellierung die Funktionalität des Systems darstellt und zu seiner Außenwelt abgrenzt.
- *Designer*
Ein Designer legt die Operationen, Attribute und Beziehungen einer oder auch mehrerer Klassen fest. Ferner definiert er die spätere Integration der Klassen in die Software-Umgebung.
- *Test-Designer*
Der Test-Designer ist für die Planung, das Design, die Implementierung und die Evaluation von Testfällen verantwortlich. Dazu gehört auch die Erstellung des Testplans und des Testmodells, die Implementierung und Durchführung der Testprozeduren sowie die Auswertung der Testberichte, -ergebnisse und -effektivität.

Festzuhalten bleibt, daß Worker keine Personen sind, sondern beschreiben, wie Personen sich im Prozeß verhalten sollen und welche Verantwortlichkeiten sie innehaben. Die einzelnen Personen, die in den gesamten Software-Erstellungsprozeß involviert sind, tragen unterschiedliche Hüte bzw. haben unterschiedliche Rollen.¹ Bei der Projektplanung legt der Projektleiter fest, welche Personen zu welchen Workern werden. Dadurch kann eine Person verschiedene Workerausprägungen zugeordnet bekommen. Ein Worker kann wiederum durch verschiedene Personen repräsentiert werden.

In dem in Abbildung 3-2 dargestellten Beispiel kann die Person Sylvia morgens als Use-Case-Designer arbeiten und bereits am Nachmittag als Design-Reviewer. Paul und Mary sind beide Designer, obwohl sie für verschiedene Klassen oder verschiedene Design-Pakete verantwortlich sind.

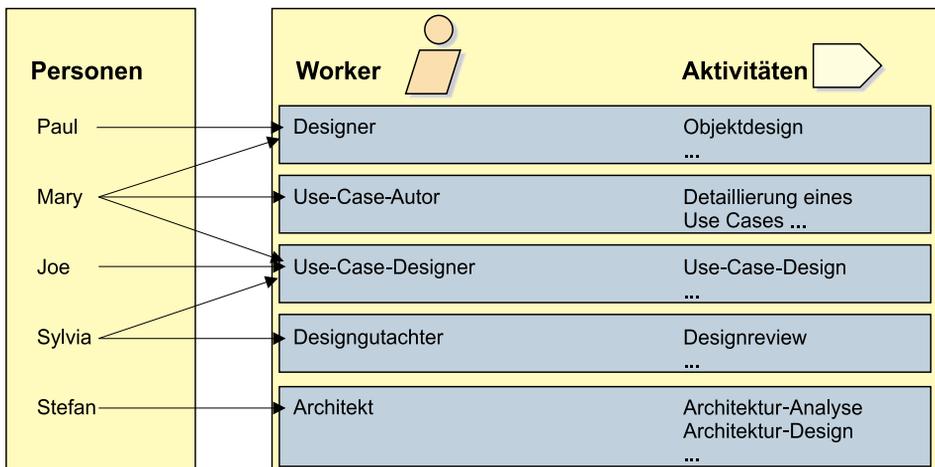


Abbildung 3-2 Personen und Worker

Jede Person, die als Prozeßverantwortlicher eingesetzt wird, muß natürlich auch die notwendigen Voraussetzungen dazu erfüllen. So muß Sylvia einerseits wissen, wie ein Use-Case-Design erstellt wird, und andererseits wie ein Design-Review durchgeführt wird.

1. Es darf also nicht heißen: »Der Designer der Klasse X macht das« sondern: »die Person, die als Designer für Klasse X verantwortlich ist, macht das«.

Worker werden üblicherweise in der folgenden Form dargestellt: *Worker: Use-Case-Designer*.¹ Anhang A gibt eine Übersicht über alle Worker des Rational Unified Process.

3.3 Aktivitäten

Eine Aktivität, die ein Worker durchführen soll, ist ein bestimmter Teil einer Tätigkeit, die eine Person in dieser Rolle ausüben soll. Eine Aktivität erfüllt einen eindeutigen Zweck, üblicherweise die Erstellung oder Überarbeitung von Artefakten wie zum Beispiel einem Modell, einer Klasse oder eines Plans. Jede Aktivität wird einem bestimmten Worker zugeordnet.

Die Bearbeitungsdauer einer Aktivität liegt üblicherweise zwischen einigen Stunden und wenigen Tagen. In der Regel ist damit ein Worker beauftragt und ein einzelnes oder eine kleine Anzahl von Artefakten betroffen. Eine Aktivität sollte ein sinnvolles Element innerhalb der Planung und des Prozeßfortschritts sein. Ist sie zu detailliert gewählt, wird sie vernachlässigt; ist die Aktivität hingegen überdimensioniert, orientiert sich der Prozeßfortschritt an Teilen der Aktivität.

Aktivitäten können verschiedene Male an ein und demselben Artefakt wiederholt werden, insbesondere von einer Iteration zur nächsten, wenn das System verfeinert und ausgebaut wird. Sich wiederholende Aktivitäten werden in der Regel vom selben Worker durchgeführt, jedoch nicht gezwungenermaßen von derselben Person.

Im objektorientierten Jargon könnte man sagen, daß der Worker ein aktives Objekt ist und die durchzuführende Aktivität eine Methode des Objekts. Im folgenden werden einige Beispiele für Aktivitäten aufgelistet:

- Planung einer Iteration – durchgeführt von Worker: Projektleiter
- Auffinden von Use Cases und Akteuren – durchgeführt von Worker: Systemanalytiker
- Design Review durchführen – durchgeführt von Worker: Design-Reviewer
- Durchführen eines Performance Tests – durchgeführt von Worker: Performance-Tester

Aktivitäten werden üblicherweise durch das Voranstellen des Wortes *Aktivität* gekennzeichnet. So zum Beispiel *Aktivität: Auffinden von Use Cases und Akteuren*.²

1. Anmerkung der Übersetzerin: im Original: *worker: Use-Case Designer*

2. Anmerkung der Übersetzerin: im Original: *Activity: Find use cases and actors*

Die Kapitel 7 bis 15 geben eine Übersicht über alle Aktivitäten des Rational Unified Process.

3.3.1 Aktivitäten und Schritte

Aktivitäten werden in Schritte unterteilt. Dabei unterscheidet man drei Kategorien:

- *Denkschritte*
Der Worker sammelt und untersucht die Eingangsartefakte und formuliert die Ausgangsartefakte. Hier steht der Sinn der Aufgabe im Vordergrund.
- *Durchführungsschritte*
Der Worker erzeugt oder ändert einige Artefakte.
- *Review-Schritte*
Der Worker wertet die Ergebnisse anhand von vordefinierten Kriterien aus.

Diese Schritte sind als Alternativen zu verstehen, nicht in jeder Aktivität müssen alle drei Kategorien vertreten sein. Zum Beispiel kann obige Aktivität *Aktivität: Auffinden von Use Cases und Akteuren* in die folgenden Schritte zerlegt werden:

1. Finden der Akteure
2. Finden der Use Cases
3. Beschreiben des Zusammenspiels von Use Cases und Akteuren
4. Bilden von Paketen
5. Transformation des Use-Case-Modells in ein Use-Case-Diagramm
6. Überblick über das Use-Case-Modell anfertigen
7. Ergebnisse auswerten

Die ersten drei Schritte sind als Denkschritte einzugruppieren, während die Schritte 4 bis 6 den Durchführungsschritten zuzurechnen sind, da hier die Ergebnisse der vorherigen Schritte im Use-Case-Modell abgebildet werden. Der Review-Anteil (Schritt 7) erfordert die Überprüfung des Ergebnisses hinsichtlich Vollständigkeit, Robustheit, Deutlichkeit und anderer Qualitätsmerkmale.

3.4 Artefakte

Ein Artefakt ist ein Teil einer Information, die von einem Prozeß erstellt, geändert oder genutzt wird. Bei Artefakten handelt es sich um konkrete Produkte eines Projekts, oder anders ausgedrückt: Artefakte werden während der Projektlaufzeit erstellt oder benutzt, bis letztendlich zum Projektende ein Endprodukt feststeht. Artefakte werden von den Workern als Input genutzt, um eine Aktivität auszuführen. Ebenso sind Artefakte als Output von Aktivitäten zu betrachten. Im objektorientierten Jargon sind Artefakte die Parameter der Aktivitäten, die wiederum Methoden eines aktiven Objektes sind. Artefakte können unterschiedliche Ausprägungen haben:

- Ein Modell (zum Beispiel ein Use-Case-Modell oder ein Designmodell)
- Ein Element eines Modells (zum Beispiel eine Klasse, ein Use Case oder ein Subsystem)
- Ein Dokument, das zum Beispiel die Software-Architektur beschreibt
- Quellcode
- Ausführbare Dateien

Artefakt ist ein Begriff, der speziell im Rational Unified Process genutzt wird. In anderen Prozessen tauchen Begriffe wie Produkt, *work unit* und ähnliches auf, die damit das gleiche beschreiben. Ausgeliefert an den Endkunden wird immer die Gesamtmenge aller Artefakte.

Artefakte können sich auch aus anderen Artefakten zusammensetzen. So besteht zum Beispiel das Designmodell aus einer Vielzahl von Klassen, und der Software-Entwicklungsplan beinhaltet eine Menge anderer Pläne, wie zum Beispiel einen Ressourcenplan, einen Metrikenplan, einen Iterationsplan usw.

Artefakte sind stets Gegenstand des Versions- und Konfigurationsmanagements. Dabei kann es durchaus möglich sein, daß wenn einzelne Artefakte sich nicht versionieren lassen, eine Art Container-Artefakt, das diese einzelnen Artefakte subsummiert, versioniert wird. So könnte man zum Beispiel die einzelnen Designmodelle oder Pakete versionieren, anstatt jede Klasse einzeln dem Versionsmanagement zu unterwerfen.

Normalerweise gehören Dokumente nicht zu den Artefakten. Hingegen wird bei vielen anderen Prozessen ein expliziter Schwerpunkt auf Dokumente gelegt¹ – besonders auf Dokumente in Papierform. Der Rational Unified Process dagegen verringert diese systematische Produktion von Papier-Dokumenten deutlich. Der effizienteste und pragmatischste Ansatz, um Projektartefakte zu managen, besteht darin, diese in dem Werkzeug zu warten, mit dem sie erstellt wurden. Falls notwendig, kann dann bei Bedarf immer noch ein Dokument (als Schnappschuß) aus dem Werkzeug generiert werden.

Für die betroffenen Parteien ist es ohnehin interessanter, daß die Artefakte zusammen mit dem Werkzeug ausgeliefert werden, statt in Papierform. Dieser Ansatz stellt einerseits sicher, daß die Artefakte immer aktuell und auf dem neuesten Projektstand sind, und andererseits fällt kein zusätzlicher Aufwand an. Hier einige Beispiele für solche Artefakte:

- Ein Designmodell, gespeichert in Rational Rose
- Ein Projektplan, gespeichert in Microsoft Project
- Ein Fehler, gespeichert in ClearQuest
- Eine Anforderungsdatenbank, gespeichert in Requisite Pro

Trotzdem existieren Artefakte, die ausschließlich in Papierform vorliegen. Dabei kann es sich zum Beispiel um externe Vorgaben für das Projekt handeln oder auch um Beschreibungen, die auf Papier am einfachsten darstellbar sind.

3.4.1 Reports

Modelle und Modellelemente können zugeordnete Reports besitzen. Ein Report extrahiert Modellinformationen aus einem Werkzeug. So kann zum Beispiel ein Report ein Artefakt oder eine Sammlung von Artefakten für ein Review sein. Im Gegensatz zu den bisher besprochenen Artefakten unterliegen Reports nicht der Versionskontrolle. Sie können jederzeit reproduziert werden, indem zu dem AusgangsArtefakt, das den Report generiert hatte, zurückgegangen wird.

Artefakte werden üblicherweise durch Voranstellen des Wortes *Artefakt* gekennzeichnet, so zum Beispiel *Artefakt: Use-Case-Drehbuch*.

1. Anmerkung der Übersetzerin: Dies ist zum Beispiel besonders beim hierzulande verbreiteten V-Modell der Fall.

3.4.2 Kategorien von Artefakten

Die Artefakte des Rational Unified Process lassen sich in fünf Kategorien¹ aufteilen:

- Managementkategorie
- Anforderungskategorie
- Designkategorie
- Implementierungskategorie
- Verteilungskategorie

Die Managementkategorie umfaßt alle Artefakte, die in Zusammenhang mit dem Software-Geschäft und dem Projektmanagement stehen:

- Planungsartefakte, wie zum Beispiel ein Software-Entwicklungsplan, der Geschäftsvorgang, die aktuelle Prozeßinstanz, die gerade vom Projekt genutzt wird², usw.
- Operationale Artefakte, wie zum Beispiel eine Release-Beschreibung, Statusbeurteilung, Verteilungsdokumente oder Fehlerdaten

Die *Anforderungskategorie* faßt alle Artefakte zusammen, die zur Definition des zu entwickelnden Software-Systems genutzt werden:

- Das Ausgangsdokument
- Anforderungen in Form von Use Cases und ergänzenden Spezifikationen
- Das Geschäftsmodell, sofern es zum Verständnis der Geschäftsprozesse, für die das System entwickelt wird, notwendig ist.

Die *Designkategorie* enthält Beschreibungen des zu entwickelnden Systems in folgenden Formen:

- Designmodell
- Architekturbeschreibung
- Testmodell

1. Anmerkung der Übersetzerin: P. Kruchten spricht auch von *information sets*.

2. Anmerkung der Übersetzerin: auch als *development case* bezeichnet.

Die *Implementierungskategorie* enthält:

- Den Quellcode und die ausführbaren Dateien
- Die betroffenen Datendateien oder diejenigen Dateien, die notwendig sind, um sie zu produzieren

Die *Verteilungskategorie* enthält alle Informationen im Hinblick auf die Auslieferung einschließlich:

- Installationskripten
- Benutzerdokumentation
- Schulungsunterlagen

In einem iterativen Entwicklungsprozeß werden die verschiedenen Artefakte nicht in einer einzelnen Phase erstellt, vervollständigt oder auch eingefroren, bevor die nächste Phase begonnen hat. Im Gegenteil, die fünf Artefaktkategorien entwickeln sich über den gesamten Entwicklungsprozeß hinweg, wie in Abbildung 3-3 dargestellt wird. Anhang B listet alle Artefakte auf, die im Rational Unified Process definiert werden.

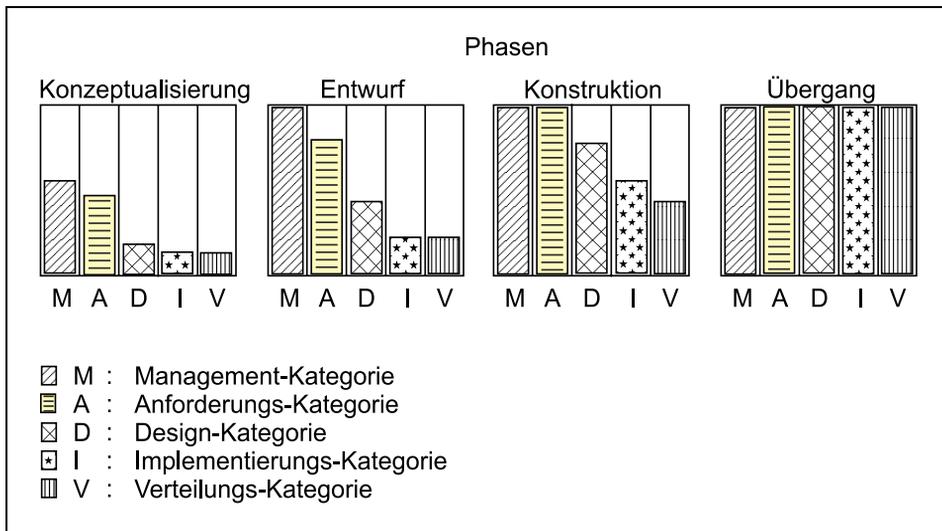


Abbildung 3-3 Entwicklung der Artefaktkategorien

3.5 Workflows

Die bloße Aufzählung aller Worker, Aktivitäten und Artefakte allein sorgt noch nicht für einen vollständigen Prozeß. Zusätzlich wird ein Weg benötigt, der einerseits aussagekräftig die Abfolge der Aktivitäten, die ein sinnvolles Ergebnis produzieren, beschreibt und andererseits die Interaktion zwischen den Workern aufzeigt.

Ein Workflow ist eine Abfolge von Aktivitäten, die ein Ergebnis von nachweisbarem Wert erzeugen. Im UML-Jargon ist ein Workflow mit einem Sequence-Diagramm, einem Collaboration-Diagramm oder einem Aktivitätsdiagramm vergleichbar. In diesem Buch wird die Notation der Aktivitätsdiagramme benutzt. Abbildung 3-4 zeigt ein Beispiel für einen Workflow¹.

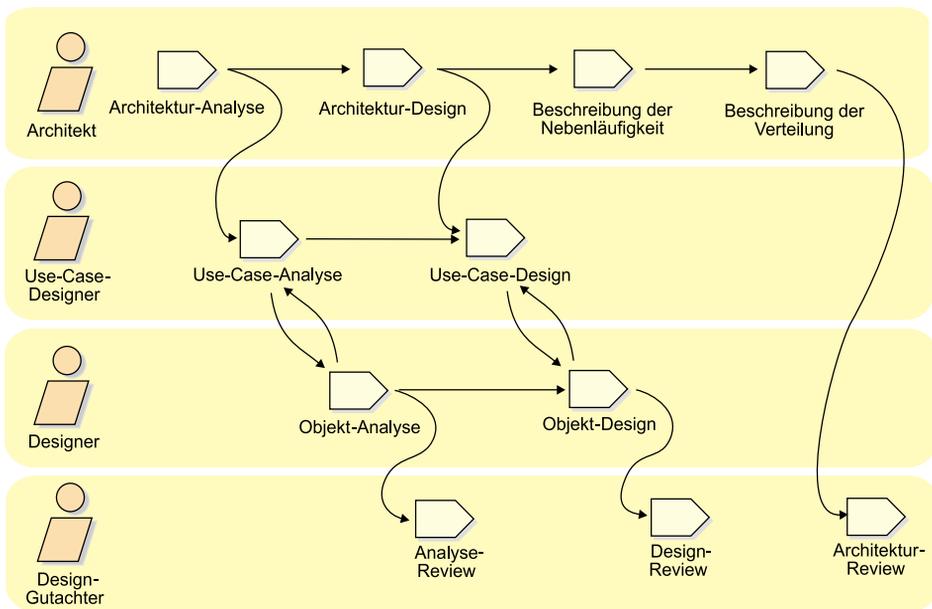


Abbildung 3-4 Beispiel für einen Workflow

Nicht immer ist es sinnvoll (oder auch möglich), alle Abhängigkeiten zwischen Aktivitäten darzustellen. Manchmal sind zwei Aktivitäten viel enger miteinander verknüpft, als darstellbar ist, besonders wenn derselbe Worker oder dieselbe Per-

1. Genauer gesagt handelt es sich nicht um Workflows, sondern um Workflow Klassen, die eine Vielzahl von Workflow Instanzen haben können.

son an der Erstellung beteiligt war. Menschen sind eben keine Maschinen, und ein Workflow kann nicht als ein Programm interpretiert werden, dem Menschen genau und mechanisch zu folgen haben.

Es gibt verschiedene Möglichkeiten, wie die Vielzahl von Aktivitäten in Form von Workflows organisiert werden kann. Der Rational Unified Process ist durch die folgenden Workflow-Typen organisiert:

- Core-Workflows
- Iterations-Workflows
- Workflow-Details

3.5.1 Core-Workflows

Im Rational Unified Process existieren neun Core-Workflows. Diese repräsentieren eine Aufteilung von allen Workern und Aktivitäten in logische Einheiten (vergleiche Abbildung 3-5). Diese Core-Workflows sind in sechs Engineering Workflows und drei Unterstützungs-Workflows unterteilt. Zu den Engineering-Workflows zählen:

1. Geschäftsprozeßmodellierungs-Workflow
2. Anforderungs-Workflow
3. Analyse- und Design-Workflow
4. Implementierungs-Workflow
5. Test-Workflow
6. Verteilungs-Workflow

Die drei Unterstützungs-Workflows sind:

1. Projektmanagement-Workflow
2. Konfigurations- und Änderungsmanagement-Workflow
3. Umgebungs-Workflow

Obwohl die Namen der sechs Core-Workflows an die sequentiellen Phasen des traditionellen Wasserfallmodells erinnern, wird in Kapitel 4 dargestellt, daß die Phasen eines iterativen Prozesses unterschiedlich sind und daß die sechs Workflows während dem gesamten Lebenszyklus immer wieder durchlaufen werden. Der gesamte Workflow eines Projekts durchläuft diese neun Core-Workflows und wiederholt sie mit unterschiedlicher Intensität in den einzelnen Iterationen. Die Core-Workflows werden in den Kapiteln 7 bis 15 näher besprochen.

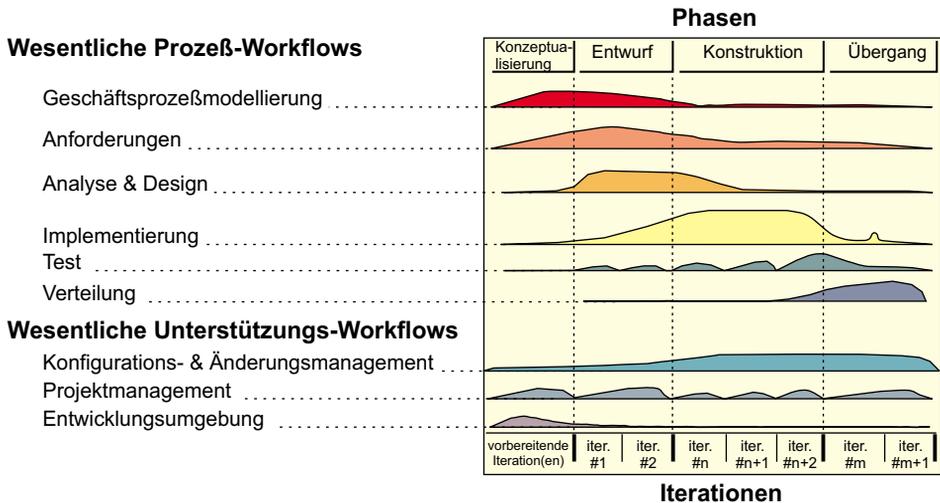


Abbildung 3-5 Die neun Core-Workflows

3.5.2 Iterations-Workflows

Iterations-Workflows sind ein anderer Blickwinkel auf die Prozeßdarstellung. Der Schwerpunkt liegt dabei mehr auf der Perspektive, was in einer typischen Iteration passiert. Iterations-Workflows können als Zusammensetzung der Instanzen des Prozesses für eine Iteration betrachtet werden. Es gibt unzählige Möglichkeiten, wie der Prozeß ausgeprägt sein kann. Der Rational Unified Process enthält Beschreibungen einiger typischer Iterations-Workflows. Sie werden eher aus pädagogischen Gründen aufgeführt, wie aus den in Kapitel 16 aufgeführten Beispielen hervorgeht.

3.5.3 Workflow-Details

Jeder der Core-Workflows umfaßt eine Vielzahl von Aktivitäten. Um diese zu identifizieren, benutzt der Rational Unified Process die Workflowdetails. Hier werden spezifische Gruppen von Aktivitäten gebildet, die

- entweder in einem engen Zusammenhang stehen,
- oder gemeinsam oder in einem Zyklus durchgeführt werden,
- oder die von einer Gruppe von Personen zum Beispiel innerhalb eines Workshops behandelt werden,
- oder die ein relevantes Zwischenergebnis produzieren.

Workflow Details zeigen ebenfalls Informationsflüsse auf, also die Artefakte, die Input für die Aktivität und Output von der Aktivität sind. Dadurch wird verdeutlicht, wie Aktivitäten innerhalb der unterschiedlichen Artefakte zusammenwirken.

3.6 Ergänzende Prozesselemente

Worker, Aktivitäten (organisiert in Workflows) und Artefakte bilden das Rückgrat der statischen Struktur des Rational Unified Process. Es gibt jedoch noch weitere Elemente, die zu den Aktivitäten und Artefakten hinzugefügt werden, um den Prozeß verständlicher und leichter anwendbar zu machen. Dabei handelt es sich um:

- Richtlinien
- Vorlagen
- Tool-Mentoren
- Konzepte

Diese Elemente erweitern die primären Elemente, wie in Abbildung 3-6 dargestellt.

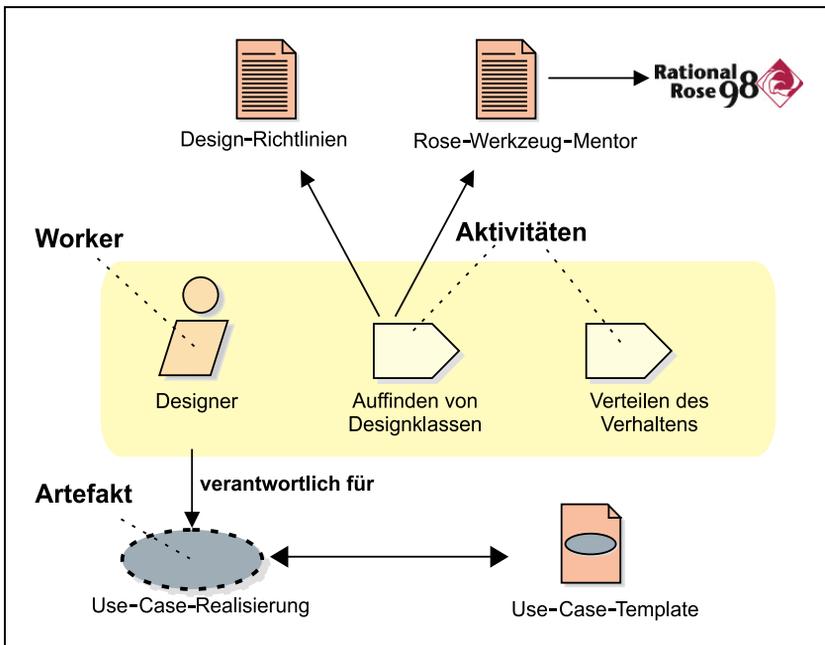


Abbildung 3-6 Templates, Tool-Mentoren und Richtlinien im RP

3.6.1 Richtlinien

Aktivitäten und Schritte werden kurz und konkret gehalten, da sie als Referenz dienen sollen, was wann zu tun ist. Um diesen Anspruch zu gewährleisten, müssen sie sowohl für Neueinsteiger, die eine konkrete Anleitung benötigen, als auch für erfahrene Anwender, die lediglich eine Erinnerungsfunktionalität brauchen, konzipiert sein.

Daher wurden den Aktivitäten und Schritten die Richtlinien beigefügt. Dabei handelt es sich um Regeln, Empfehlungen oder methodische Anleitungen, die die Aktivitäten und Schritte unterstützen. Sie beschreiben Musterartefakte mit dem Schwerpunkt auf der Qualität. Richtlinien beschreiben des weiteren bestimmte Techniken, wie zum Beispiel die Transformation von einem Artefakt zu einem anderen oder die richtige Anwendung des Unified Modeling Process. Ferner können Richtlinien genutzt werden, um die Qualität von Artefakten einzuschätzen – in Form von Checklisten, die Artefakten zugeordnet sind – oder auch um Aktivitäten einer Überprüfung zu unterwerfen. Im folgenden werden einige Beispiele für Richtlinien aufgeführt:

- Modellierungsrichtlinien, die ideale Modellierungselemente wie Use Cases, Klassen und Testfälle beschreiben
- Programmierrichtlinien für Sprachen wie C++ oder ADA, die optimale Programmierungstechniken beschreiben
- Richtlinien für Benutzerschnittstellen
- Richtlinien, wie ein bestimmtes Artefakt erstellt werden muß, zum Beispiel eine Risikenliste oder ein Iterationsplan
- Arbeitsrichtlinien, die praktische Hinweise geben, wie zum Beispiel innerhalb einer Gruppe eine Aktivität durchgeführt wird
- Checklisten, die als Teil eines Reviews genutzt werden können oder von einem Worker zur Feststellung, ob seine Aktivität vollständig abgewickelt ist

Es ist durchaus möglich, daß manche Richtlinien überarbeitet oder an bestimmte Unternehmenskulturen oder Projekte angepaßt werden müssen, um den Projektgegebenheiten wie der Verwendung bestimmter Techniken oder Werkzeuge zu entsprechen. Beispiele hierfür wären:

- Richtlinien für Benutzerschnittstellen, wie die Beschreibung einer Gestaltung von Masken im Hinblick auf Farbpalette, Fonts, Icons usw.
- Programmierrichtlinien, zum Beispiel für die Konvention hinsichtlich der Namensvergabe bei Variablen, die für das Projekt spezifisch ist

3.6.2 Vorlagen

Unter Vorlagen werden Modelle oder Prototypen von Artefakten verstanden. Zusammen mit der Artefaktbeschreibung gibt es eine oder auch mehrere Vorlagen, die zur Erstellung des zugehörigen Artefakts benutzt werden können. Diese Vorlagen haben einen Link zu dem Werkzeug, mit dem das Artefakt erstellt werden soll, so zum Beispiel:

- Microsoft Word-Dokumentvorlagen für Dokumente und Reports
- SoDA-Vorlagen für Microsoft Word oder FrameMaker, die wichtige Informationen aus Werkzeugen wie Rational Rose, Requisite Pro oder TeamTest extrahieren
- Microsoft FrontPage-Vorlagen für verschiedene Elemente des Prozesses
- Microsoft Project-Vorlagen für den Projektplan

Ähnlich wie bei den Richtlinien kann es durchaus auch bei den Vorlagen vorkommen, daß Unternehmen diese vor der Benutzung anpassen möchten, zum Beispiel durch Integration des Firmenlogos, Projektidentifikationsinformationen oder Informationen bezüglich des Projekttyps.

3.6.3 Tool-Mentoren

Aktivitäten, Schritte und die zugeordneten Richtlinien bieten in erster Linie eine Unterstützung für erfahrene Anwender. Geht man einen Schritt zurück, sind die Tool-Mentoren für Neuanwender eine wertvolle Hilfe zur Einarbeitung in spezielle Software-Produkte. Im Rational Unified Process sind Tool-Mentoren derart integriert, daß ein Link zwischen Aktivitäten und Produkten wie Rational Rose, Requisite Pro, Clear Case, ClearQuest und TestStudio existiert. Die Tool-Mentoren stellen die Abhängigkeiten der Werkzeuge innerhalb des Prozesses dar, wobei jedoch alle Aktivitäten unabhängig von den Werkzeugeigenschaften betrachtet werden. Jedes Unternehmen kann das Konzept der Tool-Mentoren ausbauen, um andere Werkzeuge zu unterstützen.

3.6.4 Konzepte

Einige der Schlüsselkonzepte, wie zum Beispiel *Iteration*, *Phase*, *Risiko*, *Performance Tests* usw. werden in separaten Abschnitten des Rational Unified Process behandelt. Sie sind im Regelfall dem entsprechenden Core-Workflow zugeordnet. Einige der Konzepte werden daher auch in diesem Buch besprochen.

3.7 Ein Prozeß-Framework

Mit der oben beschriebenen Struktur bildet der Rational Unified Process ein Prozeß-Framework. Worker, Artefakte, Aktivitäten, Richtlinien, Konzepte und Mentoren sind die Elemente, die entweder hinzugefügt oder ausgetauscht werden können, um den Prozeß an ein Unternehmen anzupassen oder ihn dort einzuführen. Wie dies durchgeführt werden kann, wird in Kapitel 17 erläutert und im Umgebungs-Workflow beschrieben.

3.8 Zusammenfassung

- Das Rational-Unified-Process-Modell basiert auf drei wesentlichen Entitäten: Worker, Aktivitäten und Artefakten.
- Workflows koordinieren Aktivitäten und Worker anhand der Ergebnisse der Produkte.
- Richtlinien, Vorlagen und Tool-Mentoren ergänzen die Prozeßbeschreibung, indem sie dem Anwender eine detaillierte Unterstützung anbieten.
- Der Rational Unified Process ist ein Prozeß-Framework, das die unternehmensspezifische Konfiguration seiner statischen Struktur ermöglicht.