

2 Vorgehensmodelle in der Softwareentwicklung

2.1 Vorbemerkungen

Aufgrund der Komplexität von Software-Produkten ist es nahezu unmöglich, allein durch Tests die Korrektheit bzw. die Fehlerfreiheit festzustellen. Dies kann nur in Kombination mit einem definierten Vorgehen in der Softwareentwicklung erfolgen, einem sogenannten Vorgehensmodell. Es macht den Entwicklungsprozess übersichtlicher und in der Komplexität beherrschbar.

Vom Grundkonzept bestehen Vorgehensmodelle aus Aktivitäten und den von diesen Aktivitäten erzeugten und verwendeten Daten sowie der Ausführungsreihenfolge der Aktivitäten. Dadurch wird auch im Wesentlichen die Strategie der Softwareentwicklung vorgegeben. Sie geben einen Rahmen vor, in dem ein Projekt geordnet ablaufen kann. Das Vorgehensmodell hilft dabei, den Ablauf eines Projektes zu strukturieren und nachzuvollziehen, da es den Prozess und die Dokumente der Softwareerstellung beschreibt.

2.2 Das Wasserfall-Modell

Das Wasserfallmodell ist ein Vorgehensmodell (auch Prozessmodell) aus dem Bereich der Softwareentwicklung. Es stammt aus den Anfängen der prozessorientierten Planung von Software. Dabei laufen die Einzelschritte wie bei einem Wasserfall in festgelegter Reihenfolge nacheinander ab und sind in der Regel nicht umkehrbar. Eine Erweiterung stellt hier die Rücksprungmöglichkeit dar (das Wasser fließt in dem Fall unnatürlich bergauf). Wird ein Fehler festgestellt, kann man einen Schritt zurückgehen und nach den Ursachen suchen. Entweder wird der Fehler behoben oder es muss noch ein Rücksprung erfolgen, dies kann unter Umständen sehr zeit- und arbeitsaufwendig werden!

Es ist ein stark top-down-orientiertes und dokumentengetriebenes Vorgehensmodell. Es basiert auf der Grundannahme unveränderlicher Anforderungen (die Projektphasen sind sehr starr), was in vielen Projekten nicht zutrifft. Daher ist es nur auf einfache Projekte mit kurzer Laufzeit anwendbar.

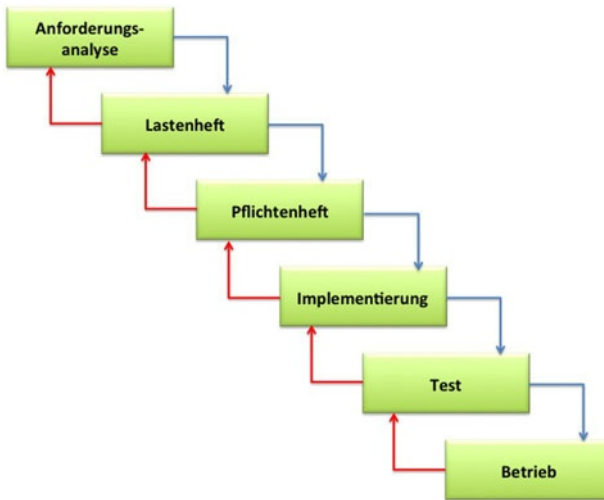


Abbildung 2-1: Wasserfallmodell

Das Wasserfallmodell gliedert sich in sechs Teilschritte:

1. Anforderungsanalyse
Hier geht es um die Analyse der Anforderungen an das zu entwickelnde System. Ziel ist es, ein gemeinsames Verständnis zwischen Auftraggeber und Auftragnehmer zu schaffen.
2. Lastenheft (Systemspezifikation)
Das Lastenheft beschreibt im Detail die Funktionalität des zu entwickelnden Systems. Es beschreibt also was und wofür.
3. Pflichtenheft (Systementwurf)
Das Pflichtenheft beschreibt im Detail wie die Umsetzung erfolgen wird. Es beschreibt also wie und womit.
4. Implementierung
In der Phase Implementierung wird das Pflichtenheft umgesetzt. Es wird sozusagen das bestellte System entwickelt.
5. Test/Abnahme- und Einführungsphase
In dieser Phase wird das entwickelte System getestet, durch den Auftraggeber abgenommen und in die Produktion überführt.
6. Betrieb/Wartungs- und Pflegephase
In dieser Phase wird das neue System betrieben, in Produktion auftretende Fehler behoben und bei Bedarf neue Releases entwickelt.

Das Modell besticht durch seine Einfachheit, sofern die Vorgaben präzise sind. Es ist leicht zu erlernen und auf jeden Fall besser, als eine Software ohne Konzept zu planen. Es eignet sich gerade wegen seiner Einfachheit eher für sehr kleine und

übersichtliche Projekte. Bei größeren Projekten sollte man auf andere Modelle zurückgreifen. Der Aufwand bei sich ändernden Grundbedingungen kann sehr groß werden und nimmt viel Zeit in Anspruch.

2.3 Das V-Modell

Das V-Modell ist in Deutschland als Entwicklungsstandard für IT-Systeme des Bundes für die Planung und Durchführung von IT Projekten verbindlich vorgeschrieben. Es ist ein Vorgehensmodell in der Softwareentwicklung, bei dem der Softwareentwicklungsprozess in Phasen organisiert wird. Neben diesen Entwicklungsphasen definiert das V-Modell auch das Vorgehen zur Qualitätssicherung (Testen) phasenweise.

Das V-Modell basiert auf dem Wasserfallmodell: Die Phasenergebnisse sind bindende Vorgaben für die nächsttiefere Projektphase. Der linke, nach unten führende Ast für die Spezifizierungsphasen schließt mit der Realisierungsphase ab. Eine Erweiterung gegenüber dem Wasserfallmodell sind die zeitlich nachfolgenden Testphasen, die im rechten, nach oben führenden Ast dargestellt werden. Den spezifizierenden Phasen stehen jeweils testende Phasen gegenüber, was in der Darstellung ein charakteristisches "V" ergibt, das dem Modell auch den Namen gab. Diese Gegenüberstellung soll zu einer möglichst hohen Testabdeckung führen, weil die Spezifikationen der jeweiligen Entwicklungsstufen die Grundlage für die Tests (Testfälle) in den entsprechenden Teststufen sind.

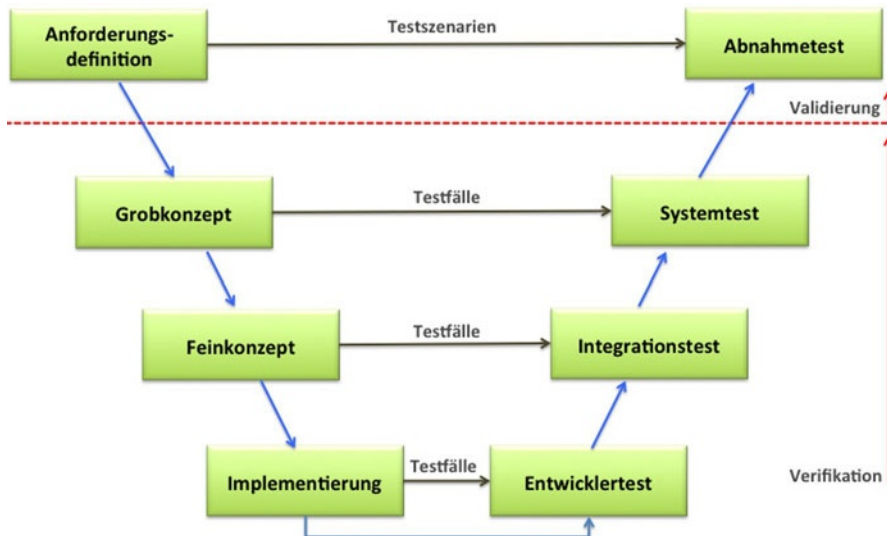


Abbildung 2-2: V-Modell

Zum V-Modell im Allgemeinen werden in der Literatur die Anzahl der Phasen und auch deren Bezeichnungen unterschiedlich dargestellt, jedoch immer mit 1:1-Gegenüberstellung von Entwurfs- und Teststufen.

2.4 RUP – Rational Unified Process

Der Rational Unified Process (RUP) ist ein kommerzielles Produkt der Firma Rational Software³. Es beinhaltet sowohl ein Vorgehensmodell zur Softwareentwicklung als auch die dazugehörigen Softwareentwicklungsprogramme. IBM entwickelt den RUP und die zugehörige Software weiter. Das RUP-Modell ist als Sammlung von Best Practices im IBM Rational Method Composer umgesetzt.

Die Berücksichtigung des Aspekts der Qualität ist jedoch beim Rational Unified Process geringer als zum Beispiel bei der Vorgehensweise nach dem V-Modell.

Das RUP-Modell teilt den Ablauf eines Projektes in vier zeitlich geordnete Phasen auf, die in mehreren Iterationen behandelt werden können:

- Inception (Projektsetup/Konzeption)
Diese erste Konzeptionsphase hat das Ziel einer gemeinsamen Vision, eines klaren Zieles sowie der Erstellung eines rudimentären Anwendungsfallmodells, welches die wesentliche Funktionalität beschreibt sowie einer tentativen/provisorischen Architektur. Darüber hinaus werden die wesentlichsten Risiken identifiziert und die Ausarbeitungsphase geplant.
- Elaboration (Ausarbeitung/Entwurf)
In dieser Phase wird ein Architekturprototyp sowie eine grobe Beschreibung der Anwendungsfälle ausgearbeitet. Planung der Konstruktionsphase, Machbarkeitstests, Systemevaluierung und erste Programmteile von Schlüsselkomponenten sind Teil der Elaborationsphase.
- Construction (Implementierung)
Nachdem die Architektur ausgearbeitet wurde, konzentriert sich diese Phase auf die Entwicklung und das Testen des Produktes. In dieser Phase werden sämtliche Anforderungen unter laufender Abstimmung mit dem Kunden realisiert.
- Transition (Inbetriebnahme)
Übergabephase und Freigabe der Software an den Kunden.

3 Seit 2003 Teil des IBM-Konzerns

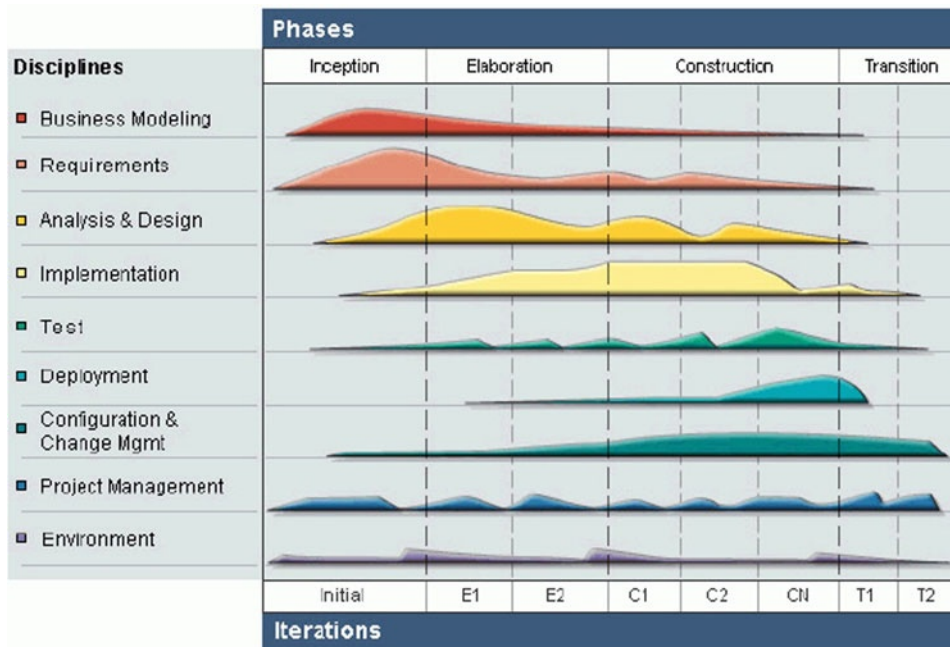


Abbildung 2-3: Rational Unified Process (RUP)⁴

Die Struktur der Phasen und Iterationen im RUP-Modell teilt sich in Kern-Arbeitsprozesse (Core Workflow):

1. Geschäftsprozessmodellierung (Business Modeling/Business Analysis): Im Rahmen der Geschäftsprozessmodellierung gilt es, insbesondere Geschäftsprozesse oder Ausschnitte daraus zu dokumentieren. Die Definition von Geschäftsanforderungen und die sich eventuell ergebende Geschäftsprozessoptimierung sind Grundlage zur Vorbereitung einer Automatisierung bzw. IT-Unterstützung der Geschäftsprozesse. In diesen Bereich fallen vor allem die analysierten Anforderungen zur Systemintegration sowie die Planung der Zuständigkeiten im Zusammenhang mit der Websitepflege.
2. Anforderungsanalyse und -management (Requirements): Die Anforderungsanalyse zieht sich bis über die Construction-Phase hinaus, ist jedoch besonders bei Projektbeginn sehr ausgeprägt. Hier werden Anforderungen analysiert und definiert sowie laufend angepasst. Das gewonnene Feedback hat über den gesamten Prozess hinweg starken Einfluss auf die Anforderungen.

⁴ siehe auch <http://www.ibm.com/developerworks/library/ws-soa-term2/index.html>

3. Analyse und Design (Analysis & Design): In diesen Bereich fallen vor allem die konzeptionelle Aufbereitung sowie die Software-Architektur für Integration, Design und Struktur.
4. Implementierung (Implementation): Im Rahmen der Implementierung werden die Entwicklungsarbeiten durchgeführt, wie das Customizing von Systemen, Softwareentwicklung und Oberflächengestaltung. Die Realisierung aller funktionalen und nicht funktionalen Anforderungen ist Teil dieser Disziplin.
5. Test: Die Erstellung von Testplänen in erster Linie in der Initialisierungs- und Elaborationsphase fallen in diesen Bereich. Aber auch die Erstellung von Unit-Tests und manuellen Tests während der Entwicklungsphase sowie Integrations- und Abnahmetests in der Transitionsphase sind Teil dieser Disziplin.
6. Auslieferung (Deployment): Das Deployment umfasst insbesondere die Installation im kundeneigenen Rechenzentrum oder als Cloud-Service sowie die Inbetriebnahme.
.... und unterstützende Arbeitsprozesse (Core Supporting Workflow):
7. Konfigurations- und Änderungsmanagement (Configuration und Change Management): Das Änderungsmanagement nimmt eine wichtige Rolle ein. Neue Anforderungen werden als Change Requests erfasst und fließen nach Abstimmung unmittelbar in das Projekt ein. Absehbare variable Konfigurationseinstellungen werden bereits in der Software berücksichtigt, auch Veränderungen, falls erforderlich.
8. Projektmanagement (Project Management): Das Projektmanagement zieht sich durch alle Phasen des Projekts und umfasst die Planung, Steuerung und Kontrolle des Projektverlaufes.
9. Infrastruktur (Environment): Unter dieser Disziplin wird der Einbezug der Systemlandschaft in sämtliche andere Bereiche (Anforderungen, Analyse, Design, Implementierung, etc.) verstanden. Systemeinrichtung, Backup, Einbindung in die IT- Umgebung, Netzwerk- und Zugriffssicherheit sind hier zu sehen.

Testmanagement bei SAP-Projekten
Erfolgreich Planen • Steuern • Reporten bei der
Einführung von SAP-Banking

Vivenzio, A.; Vivenzio, D.

2013, XII, 174 S. 79 Abb., Softcover

ISBN: 978-3-8348-1623-8