Entwicklungsmethoden

Prof. Dr. Josef M. Joller
jjoller@hsr.ch

SOFTWARE LIFE-CYCLE MODELLE

Übersicht

- Build-and-fix Modell
- Waterfall Modell
- Rapid Prototyping Modell
- Inkrementelles Modell
- Extreme Programming
- Synchronize-and-stabilize Modell
- Spiral Modell
- Objekt-orientierte Life-Cycle Modelle
- Vergleich der Life-Cycle Modelle

Life-Cycle Modelle (oder Prozess Modelle)

Die folgenden Phasen finden Sie immer:

- Anforderungsphase
- Spezifikationsphase
- Designphase
- Implementationsphase
- Integrationsphase
- Wartungsphase
- Ablösung

Build and Fix Modell

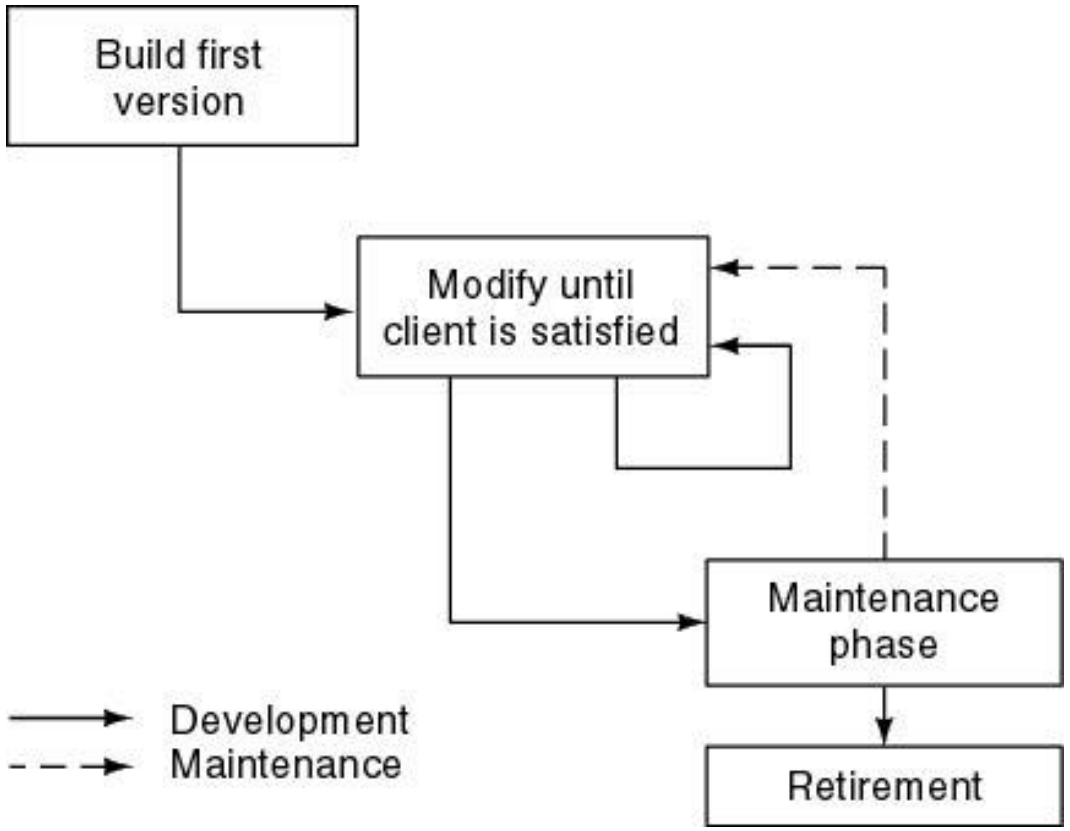
Probleme

- keine Spezifikation
- kein Design

Kaum brauchbar

Man benötigt

- “Game Plan”
- Phasen
- Milestones



Wasserfall Modell

Kennzeichnen

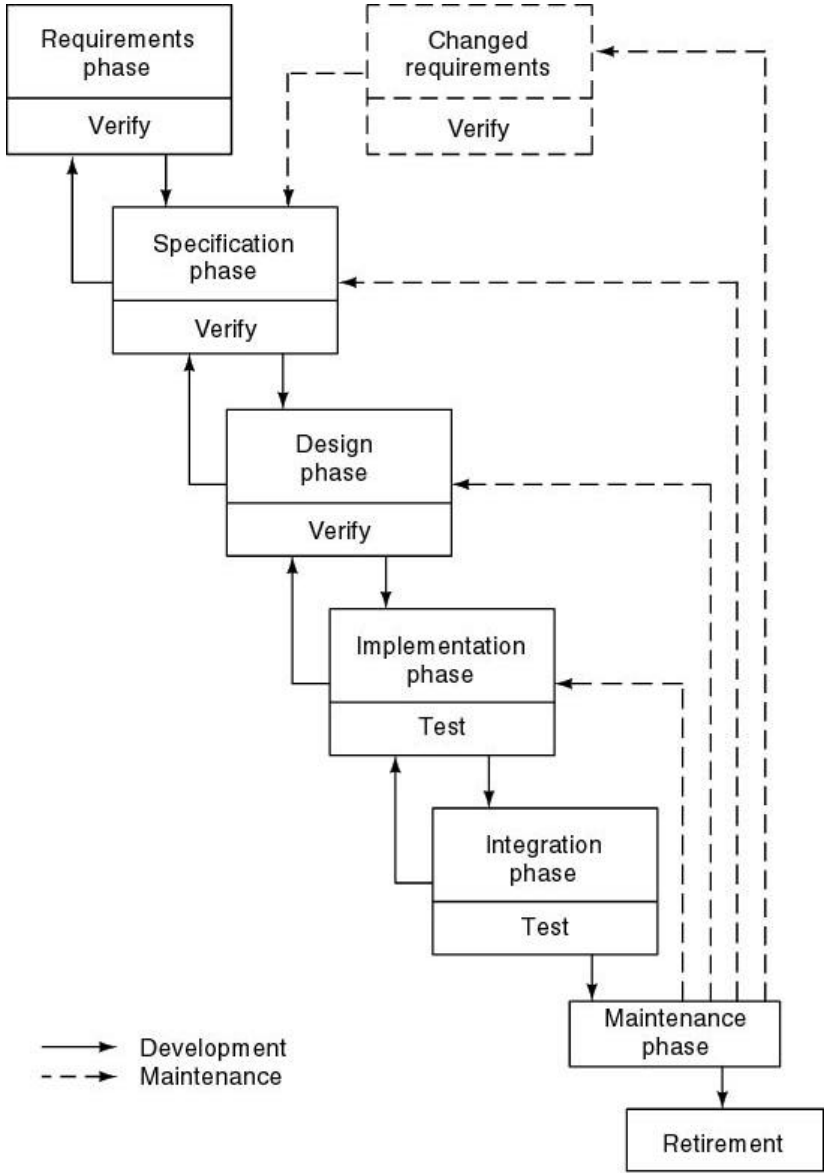
- Feedback Loops
- Dokumentation gesteuert

Vorteile

- Dokumentation
- Wartung gesichert

Nachteile

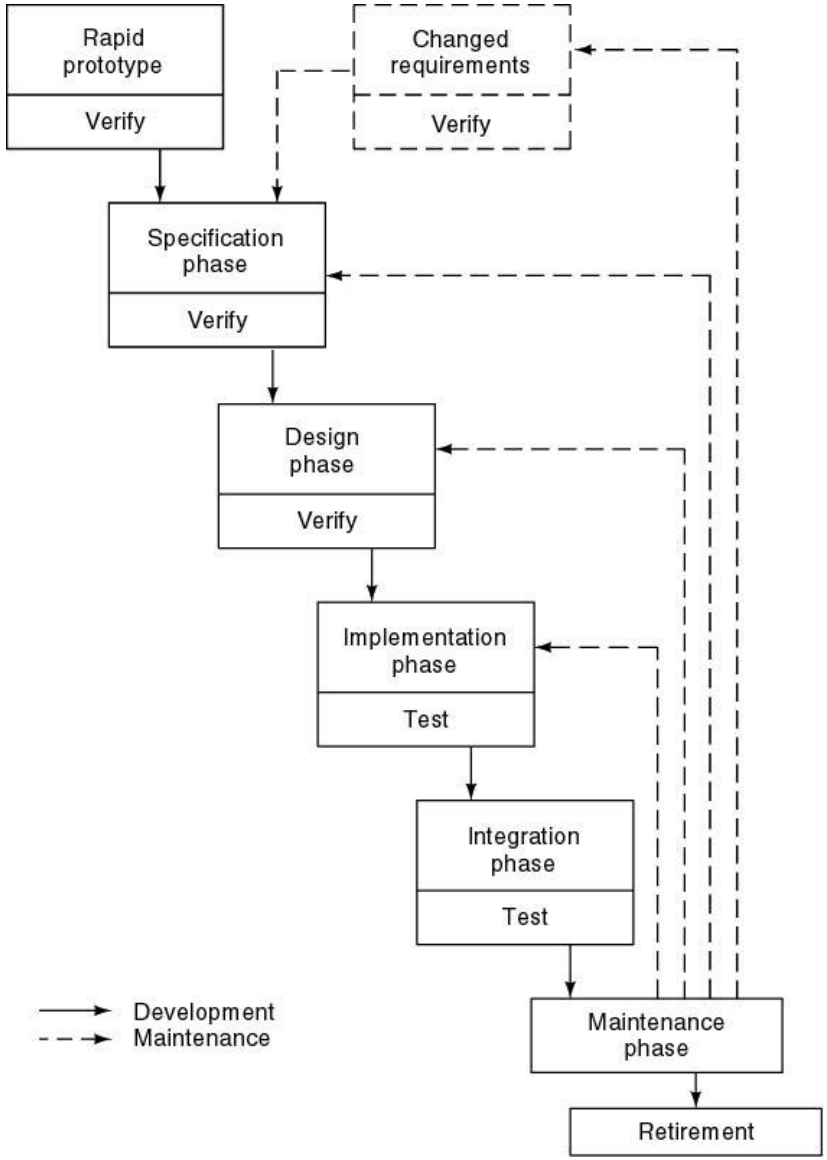
- zu ehrgeiziges Modell



Rapid Prototyping Modell

Lineares Modell

“Rapid”



Drei Schlüsselpunkte des Prototyping

Prototypen sollten nicht zum Produkt werden

Prototypen sind eine sinnvolle Ergänzung der Spezifikation

Prototypen sind nie ein Ersatz für ein Design!

Vergleich:

- Wasserfall Modell—versuche alles gleich beim ersten Mal korrekt zu machen
- Rapid Prototyping—versuchs mal, vielleicht dann noch einmal

Wasserfall und Rapid Prototyping Modell

Wasserfall Modell

- wurde oft erfolgreich eingesetzt
- kommt den Benutzeranforderungen entgegen (klare Planung...)

Rapid prototyping Modell

- keine sauberen statistischen Daten verfügbar
- kann zu Problemen führen

Lösung

- Rapid Prototyping in der Anforderungsphase
- Wasserfall für den Rest

Extreme Programming

Eine kontroverse Methode

- Stories werden festgehalten (Anforderungen der Kunden)
- Kosten und Aufwandschätzung pro Story
- bestimmte Stories bilden zusammen ein “Build”, die Funktionalität, die als nächstes freigegeben wird
- jedes “Build” besteht aus mehreren Tasks
- Zuerst werden die Tests definiert, mit denen die Tasks getestet werden sollen
- Programmiert wird zu zweit
- Integration erfolgt kontinuierlich, auf Task Ebene

Ungewöhnlich

- Rechner stehen alle im selben Raum
- der Client muss anwesend sein
- jeder muss alles können : keine Spezialisierung
- möglichst wenig oder keine Überzeit (bringt nur Stress)
- Refactoring (ist inn: Verbesserungen eines bestehenden Systems)

Synchronize-and Stabilize Model

Microsoft's Life-Cycle Modell

- Anforderungs-Analyse—Interviews mit potentiellen Kunden
- Festhalten der Anforderungen
- Aufteilen des Projekts in 3-4 Builds
- Jedes Build wird durch je ein kleines Team realisiert
- Am Ende des Tages wird synchronisiert
- Am Ende des Build wird zusammengefügt
- Komponenten arbeiten zusammen und das Zusammenspiel wird dauernd getestet
- die Funktionsweise des Gesamtsystems soll möglichst früh sichtbar werden.

Spiral Modell

Einfaches Modell

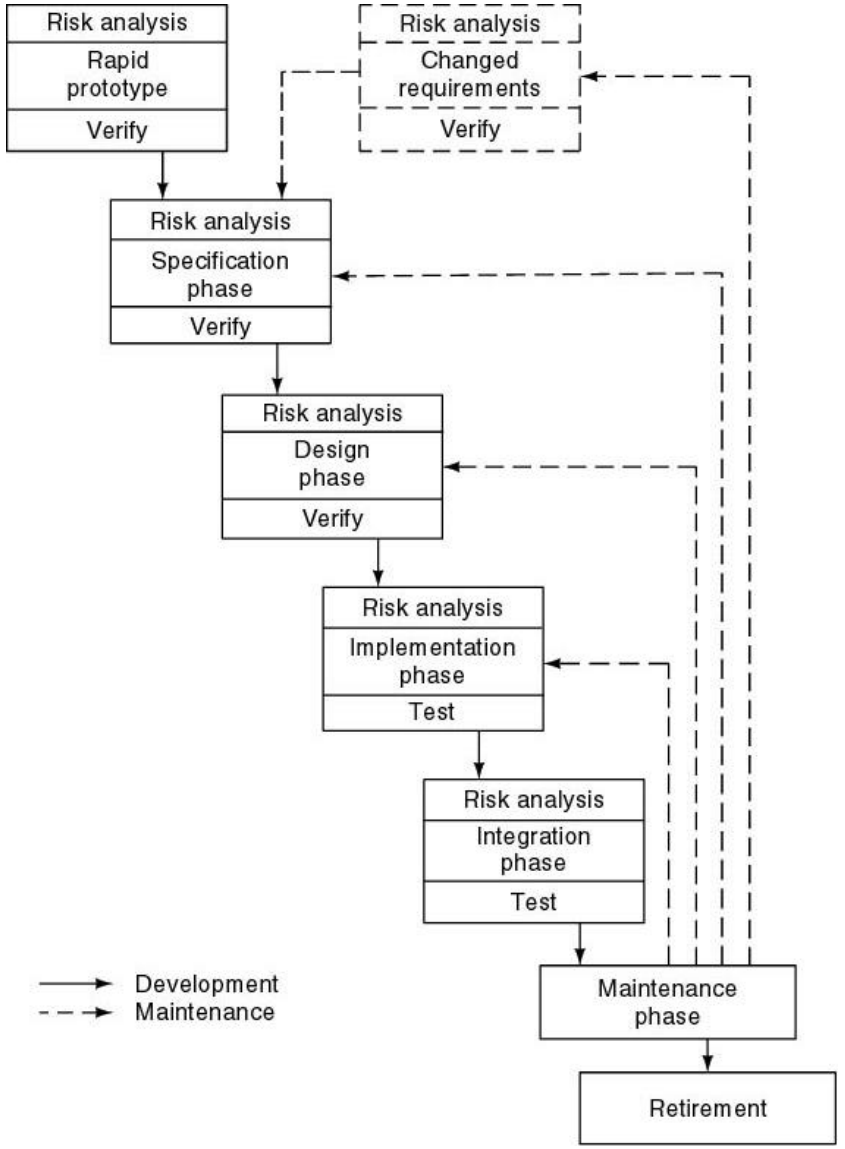
- Wasserfall Modell plus Risk Analyse

Pro Phase Vorarbeit:

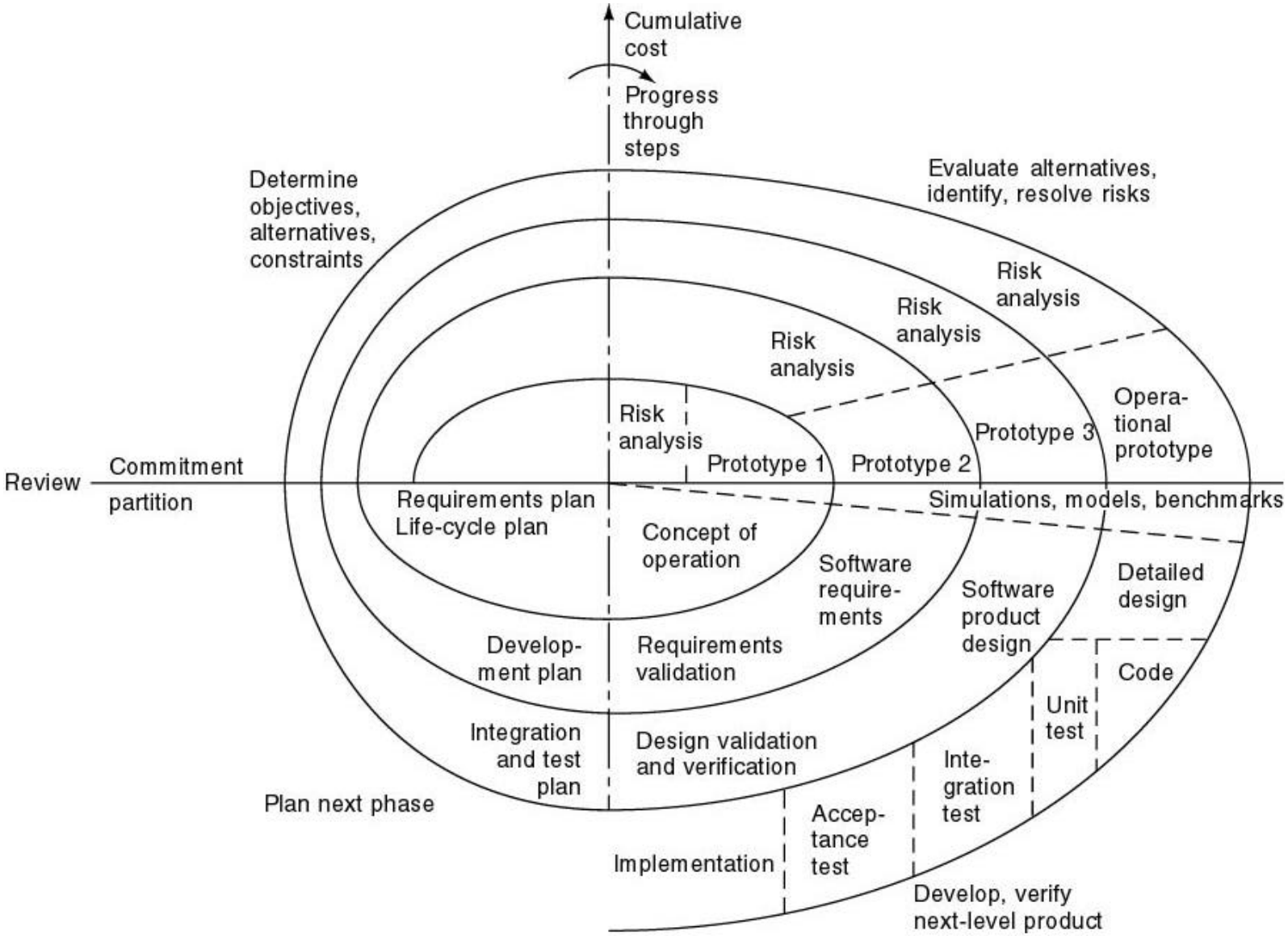
- Alternativen untersuchen
- Risiko Analyse

Pro Phase Nacharbeit

- Evaluation
- Planen der nächsten Phase



Volles Spiralmodell



Analyse des Spiralmodells

Stärken

- sauberes Modell, nachvollziehbar
- Entwicklung und Wartung bilden eine EInheit

Schwächen

- lediglich für grössere Projekte
- Probleme bei vielen Interationen (in diesem Fall eher für Inhouse Projekte)

Objekt-Orientierte Life-Cycle Modelle

Iterationen innerhalb und zwischen den Phasen

- rekursive und parallele Phasen
- round-tripGestaltung
- vereinheitlichter SW / IT Projekt-Prozess

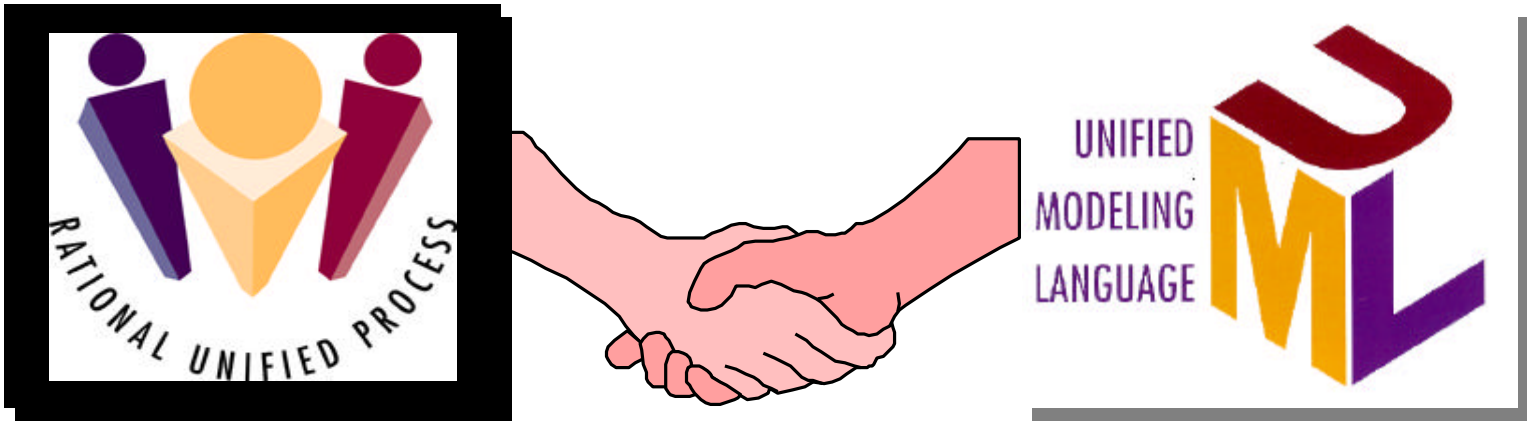
Typische Kennzeichen

- Iteration
- Parallele Bearbeitung mehrerer Phasen
- Inkrementelle Entwicklung

Gefahr

- unendliche Schleifen

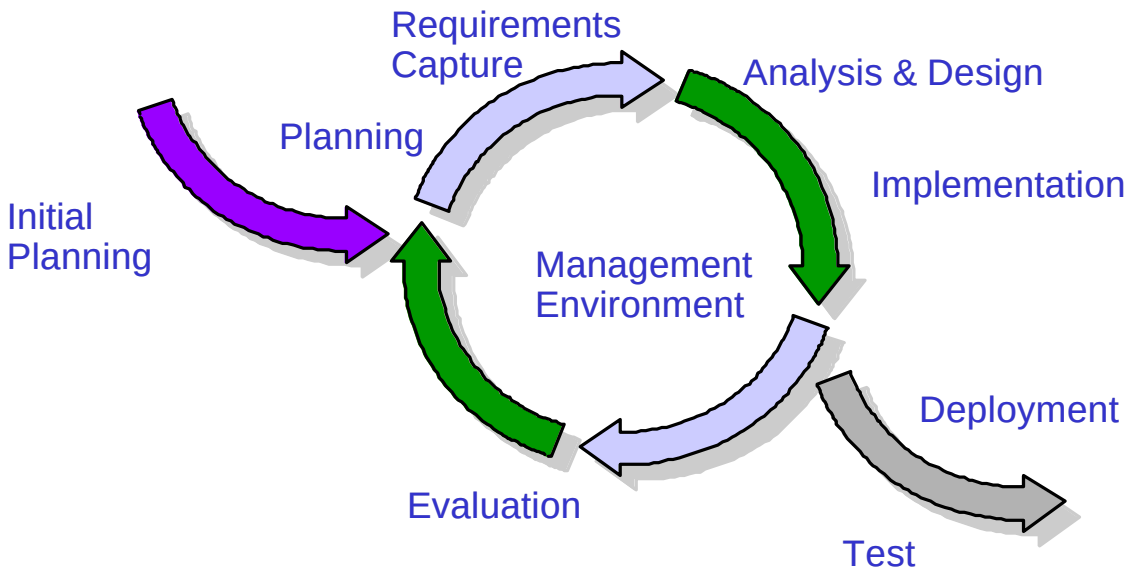
RUP – Rational Unified Process



- Der Rational Unified Process und UML — wurden beide von Rational entwickelt
- Viele Hersteller haben ihren Beitrag geleistet
  - Microsoft
  - HP
  - IBM
  - Oracle
  - Texas Instruments
  - MCI SystemHouse
- Standard durch die OMG (Object Management Group)

1. Wichtige Kennzeichen eines iterativen Prozesses

- Risiken werden abgeschätzt
- Integration geschieht kontinuierlich
- öfters kann ein Teil freigegeben werden (Cash fliesst)
- der Endbenutzer kann öfters seinen Beitrag leisten



2. Managen der Anforderungen

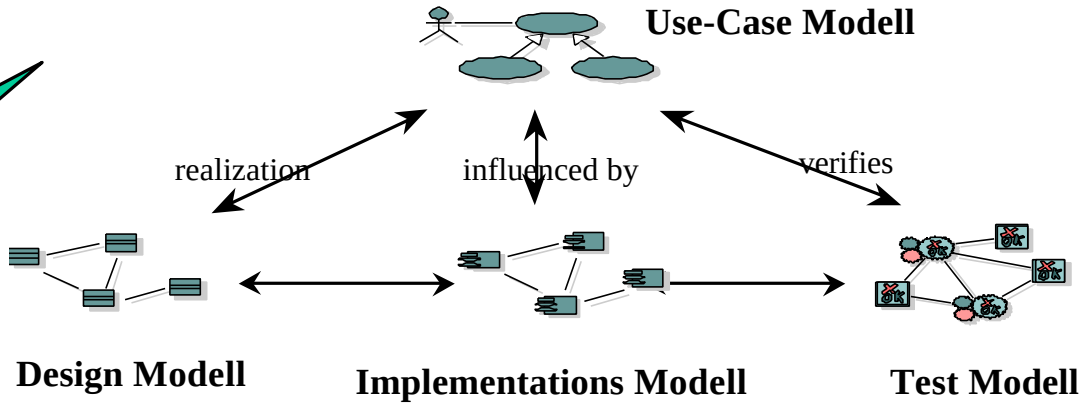
Aufzeigen und dokumentieren der Anforderungen

Entscheide werden dokumentiert

Use Cases beschreiben die Geschäftsfälle / Anforderungen

Use Cases können zur Planung eingesetzt werden

Use Cases werden durchgehend eingesetzt, bilden die Basis aller Modelle

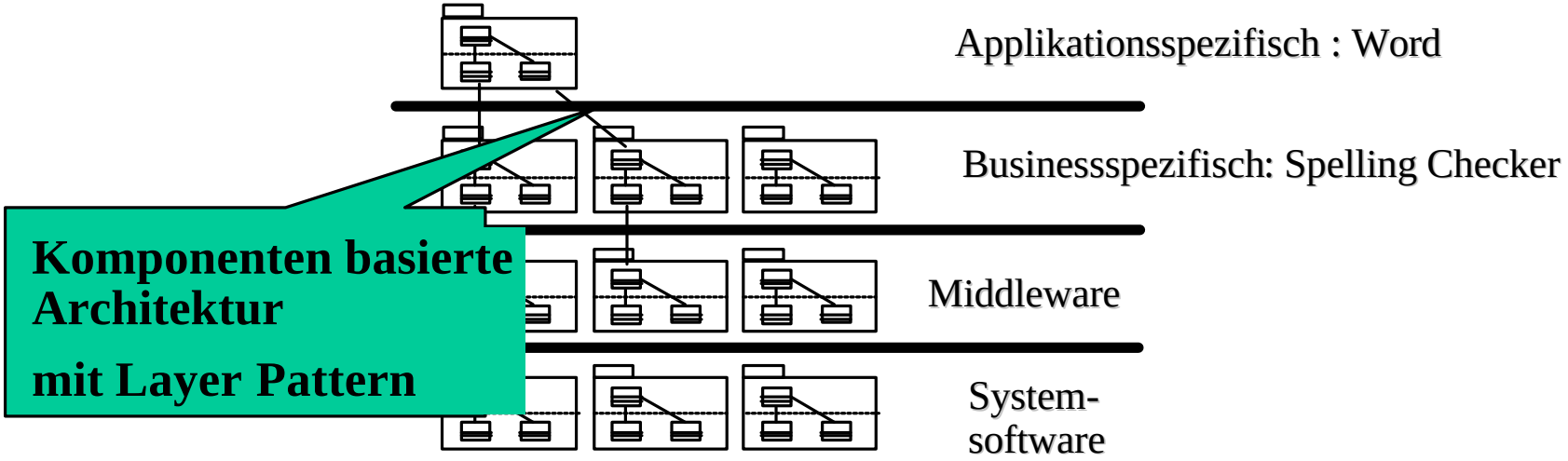


3. Entwickeln einer Komponenten-basierten Architektur

Design und Testen der Architektur sobald wie möglich!

Definition einer “guten” Achitektur

- stützt sich auf klare Interfaces ab
- verwendet erprobte Komponenen und Konzepte
- basiert auf bewerteten Use Cases
- ist verständlich
- verwendet möglicht erprobte Patterns



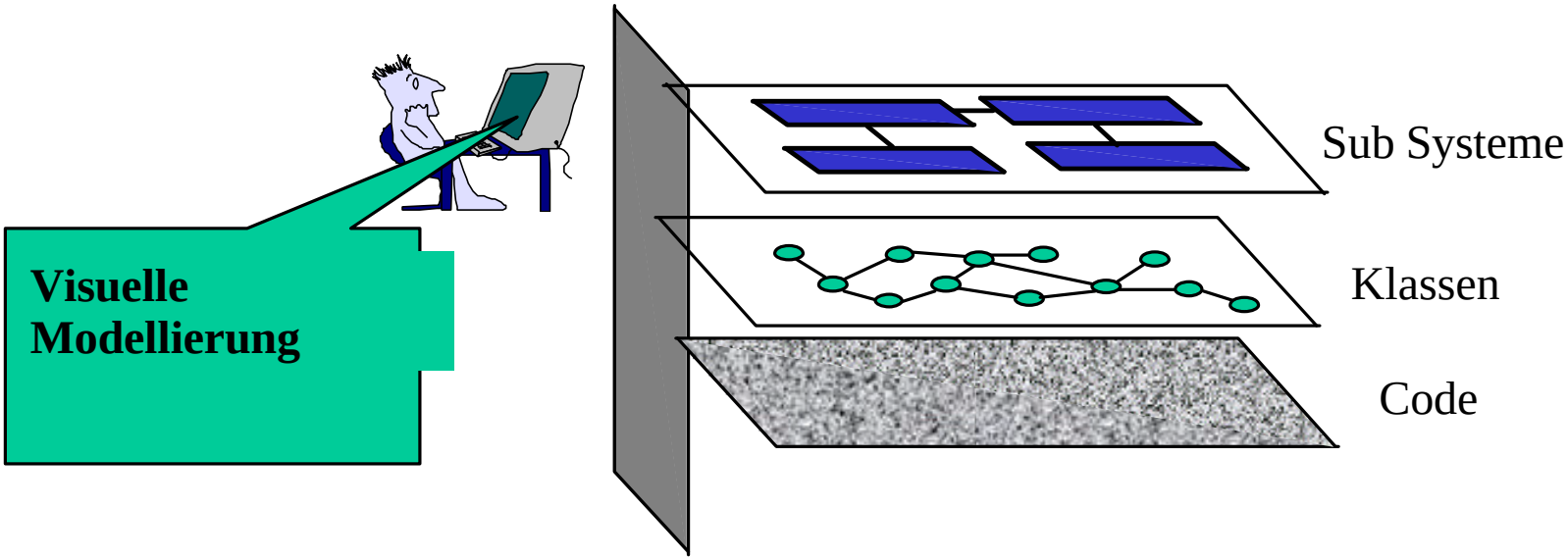
4. Visuelle Modellierung der Software

Festhalten der Struktur und des Verhaltens

Aufzeigen des Zusammenspiels

Konsistente Beschreibung über alle Phasen

Basis für eine klare Kommunikation innerhalb des Teams und mit Kunden



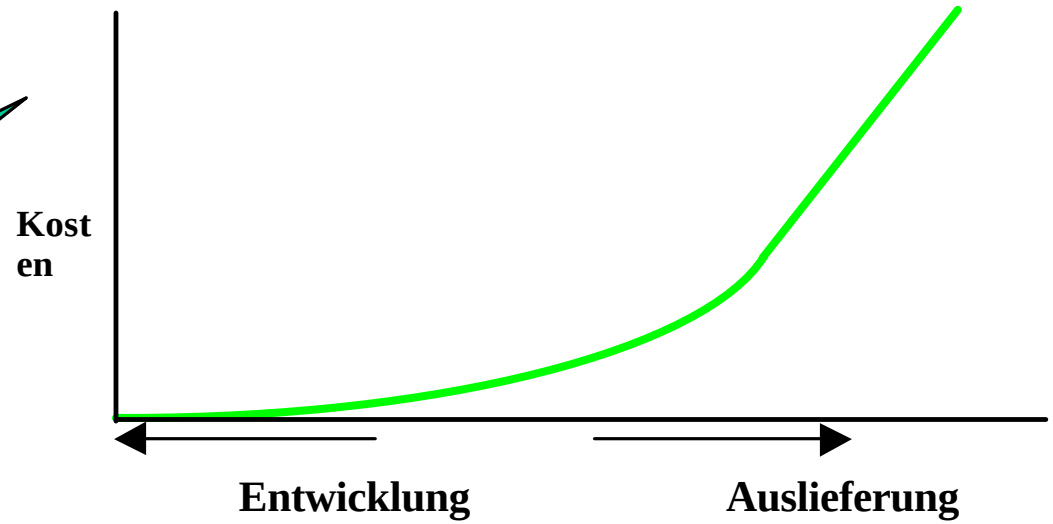
5. Verifikation der Software Qualität

Tests basieren auf Use Cases und können sehr früh festgelegt werden

Die Software kann besser geprüft werden

Tests können automatisiert werden (JUnit)

Die Kosten für die Fehlersuche variieren sehr stark



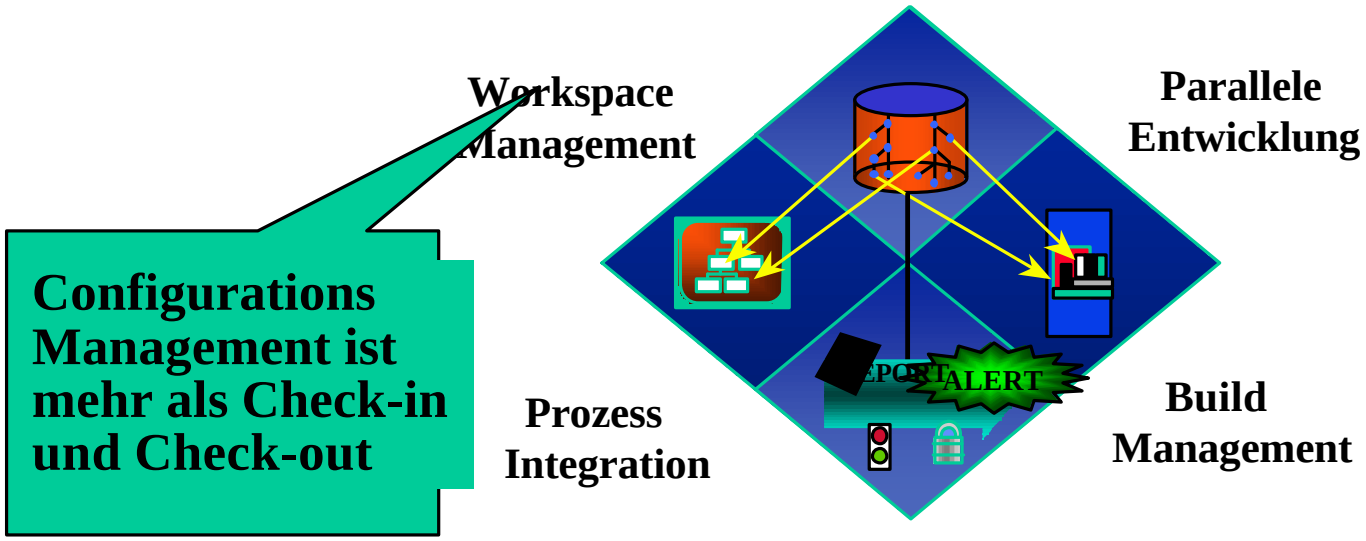
6. Änderungsmanagement

Änderungen und Ergänzungen werden erfolgt

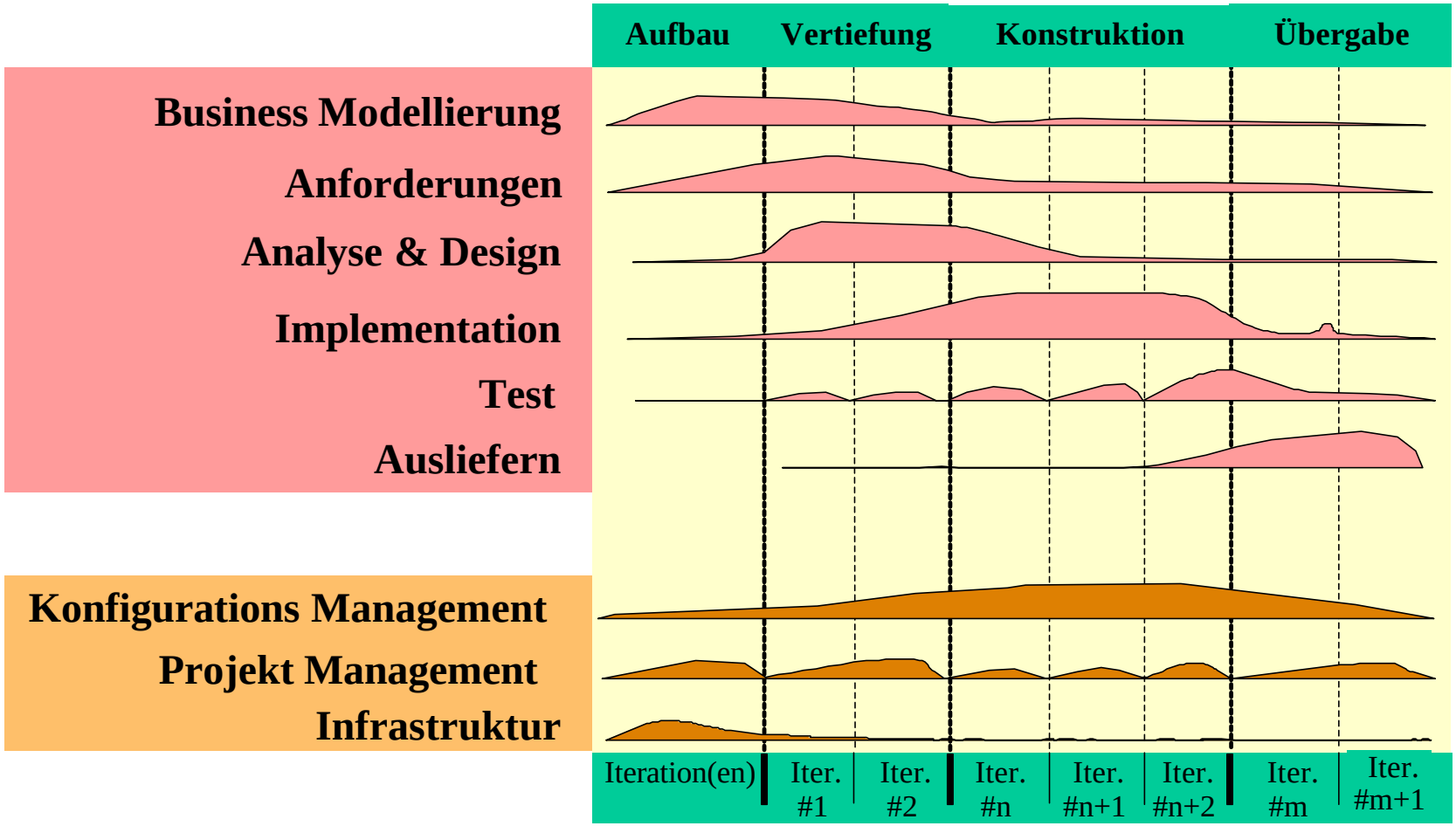
Jeder Entwickler arbeitet in einem gesicherten Bereich

- Fehler und Änderungen verbreiten sich nicht
- Die SW Artifakten (Dokumente, Code) werden verfolgbar

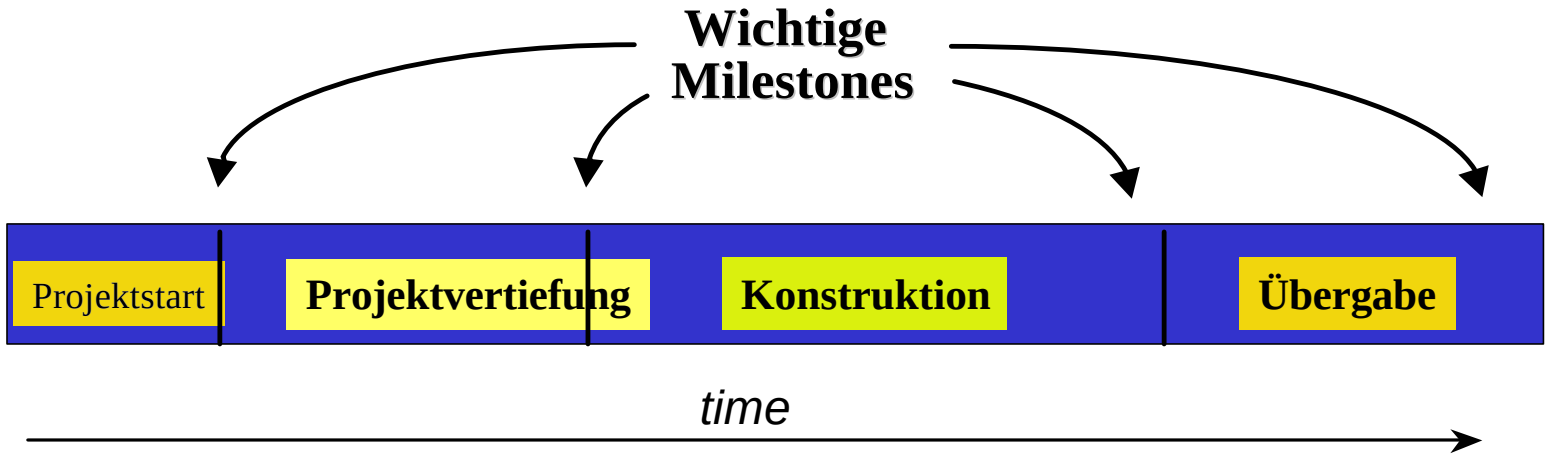
Integration und Release des Produkts können vereinfacht werden



Prozess Übersicht



Phases im Prozess

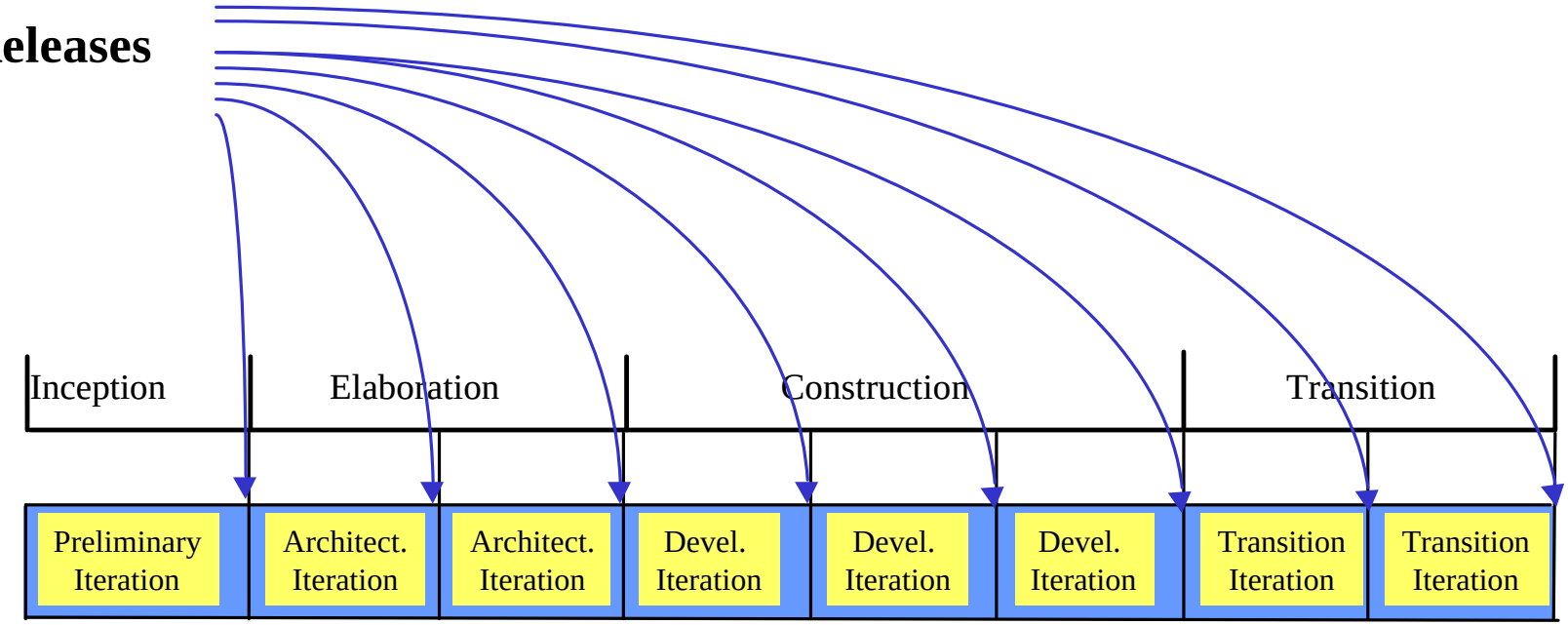


Der Rational Unified Process besteht aus folgenden Grobphasen

- Inception / Start / Aufbau des Projekts - Definieren des Projekts
- Vertiefung - Projektplan, Festlegen der Grobarchitektur
- Konstruktion - Bau des Produkts
- Übergabe - Ausliefern des Produkts an den Kunden

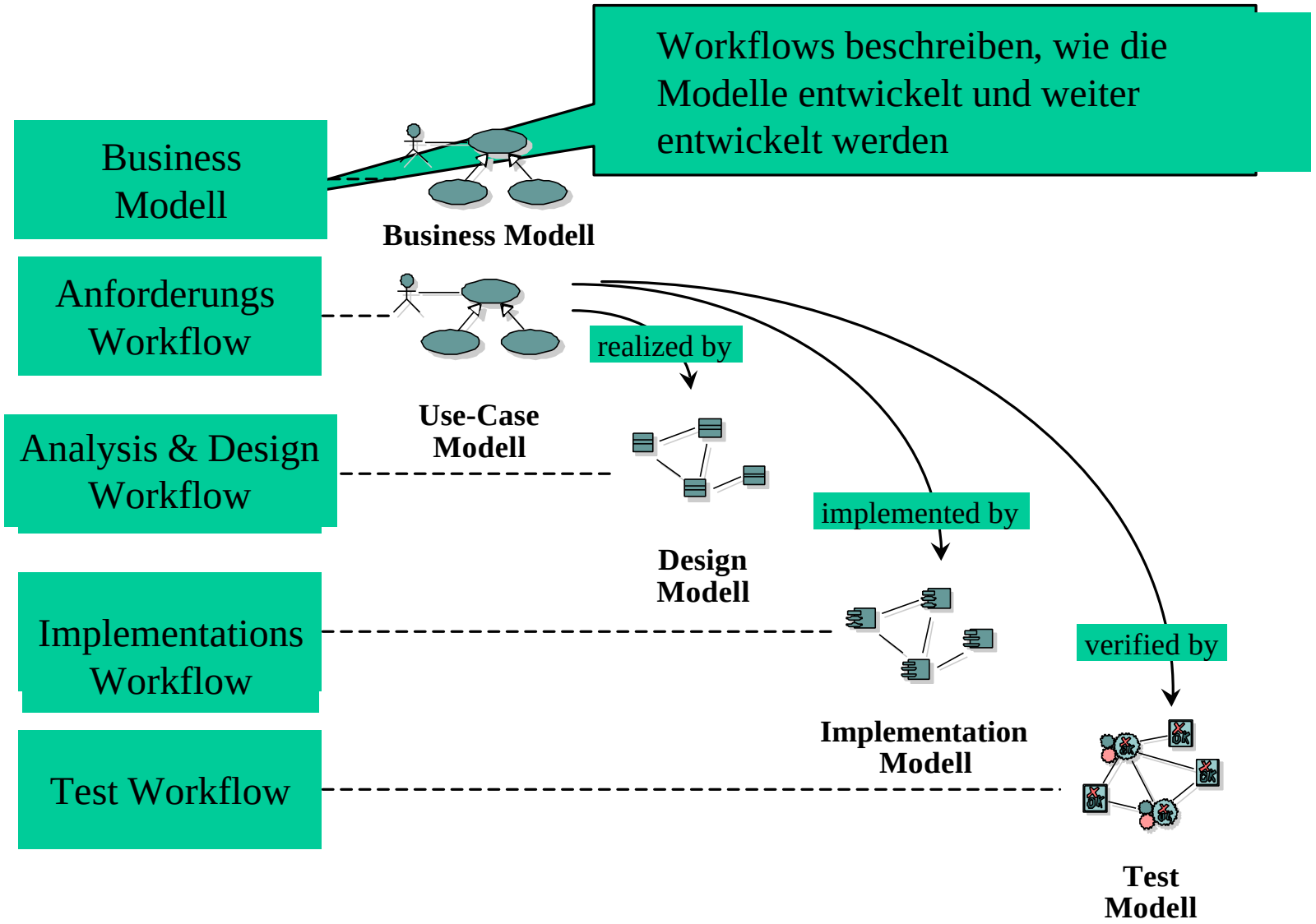
Iterationen und Phases

Releases



Eine *Iteration* besteht aus einer Sequenz von Aktivitäten, mit einem Plan und Bemessungskriterien und dem Ziel ein auslieferbares Teilsystem herzustellen (intern oder extern).

Modelle und Workflows



HAAS & PARTNER
THE LEARNING SOLUTION COMPANY

Alles auf einen Blick...

Alle Aktivitäten werden in jeder Iteration ausgeführt

Phasen

Prozess Workflows

Business Modeling

Anforderungen

Analysis & Design

Implementation

Test

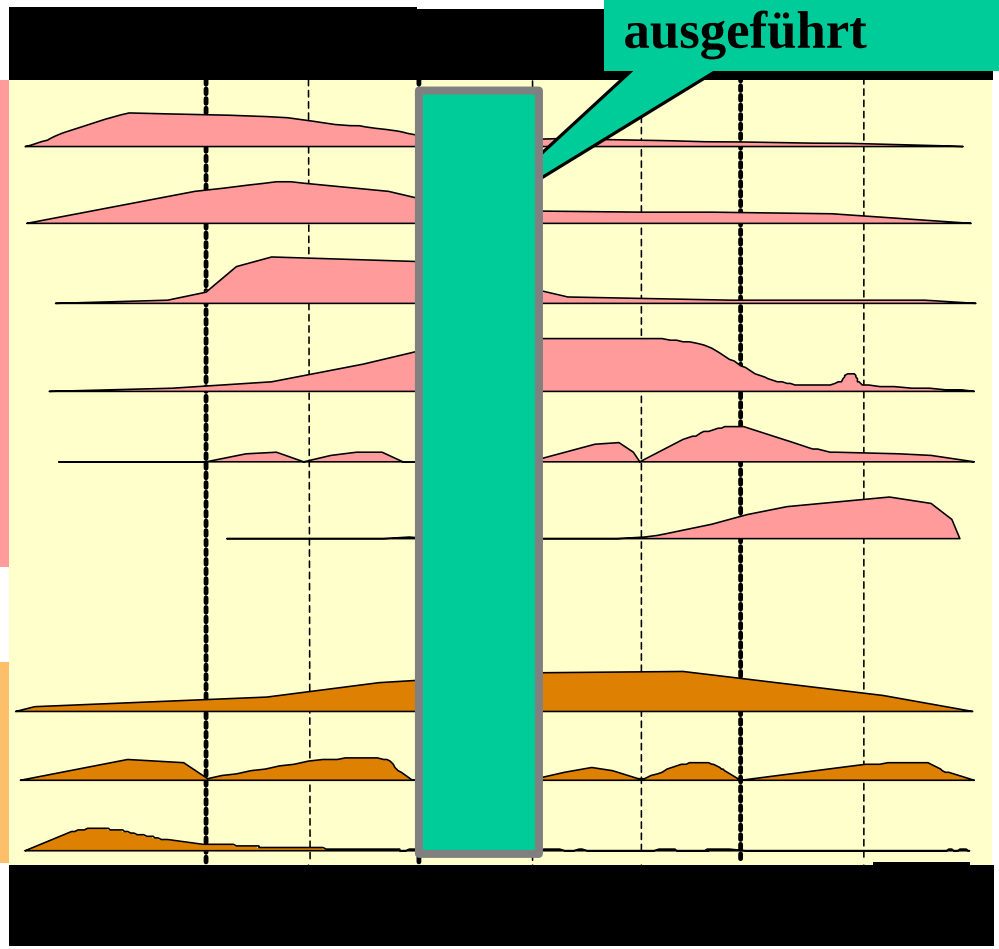
Ausllieferung

Unterstützende Workflows

Configuration Mgmt

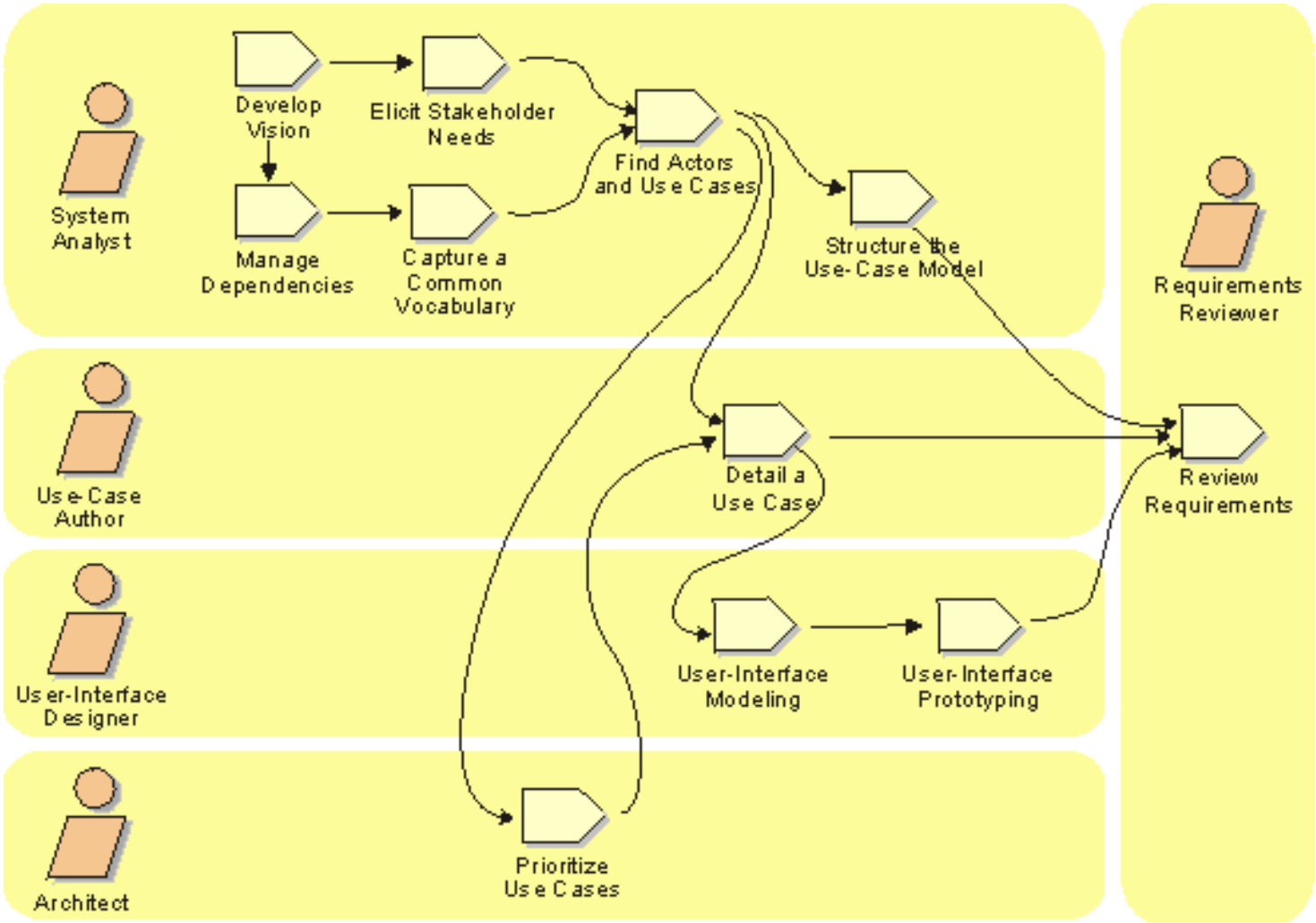
Management

Environment



Iterationen

Beispiel eines Workflows



Rational Unified Process (RUP)

Testversion

http://www.rational.com/products/rup/tryit/eval/gen_eval.jsp

- Username
  - jjoller@hsr.ch
- Password
  - KmiVF1wP
  - bis Ende April 2002!

Schlussfolgerungen

Es gibt unterschiedliche Lebensphasen-Modelle

Jedes Modell hat sein Stärken und Schwächen

Welches Modell soll ich wählen?

- Welche Organisation ist etabliert?
- Wie denkt das Management?
- Welche Fähigkeiten haben die Mitarbeiter?
- Welches Produkt soll entwickelt werden?

In der Regel

- wird eine Mischung das Beste sein!