

Seminar Software Engineering

Inwiefern sind klassische Methoden als agil anzusehen?

Diskussion am Beispiel des Rational Unified Process (RUP)

Seminar Software Engineering

Inwiefern sind klassische Methoden als agil anzusehen?
Diskussion am Beispiel des Rational Unified Process (RUP)

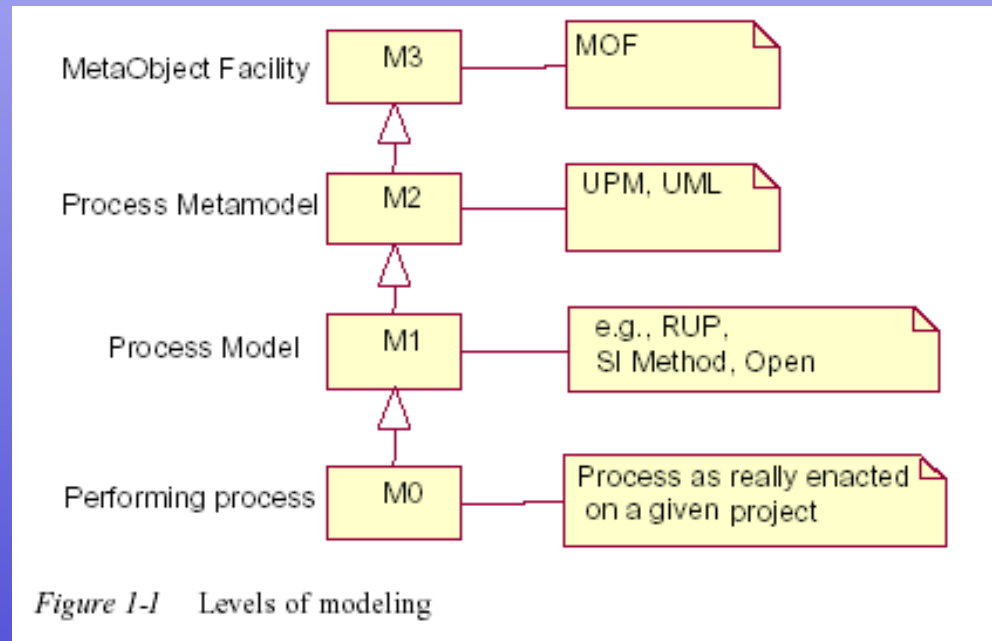
Klassisch? Agil?? RUP??? Vergleichen????

Gliederung

- Einteilung von Software Prozessmodellen
- Klassische Methoden
- Agile Methoden
- Einführung RUP
- RUP-Tools / Kleine Demo
- Gegenüberstellung / Diskussion
- Fazit

Software Prozessmodelle

Gliederung der OMG



Zwei Lager I: Die Klassischen (oder tayloristischen...)

- etabliert
- schwache Bilanz
- Einteilung? (Homogenität)
- Gemeinsamkeiten
 - Grosser Planungshorizont
 - Hoher Detaillierungsgrad der Planung
 - Prozess im Zentrum
- Verbindung zur Geschichte der Informatik

Zwei Lager II: Die Agilen

- jung und dynamisch
- Leistungsnachweis?!
- Klare Front
- Gemeinsamkeiten
 - Agile Manifest
 - Agile Principles
- Einteilung? (Homogenität)

RUP: Geschichte

- Booch + Rumbaugh + Jacobson = The three amigos of Software Engineering
- Personell bedingte Verbindung zu UML
- Seit 1998 bei Rational in der heutigen Form entwickelt
- Rational 2003 von IBM übernommen
- Reicher Erfahrungsschatz
 - Best Practice (im Detail sowie in der Architektur)

RUP: Prinzipien

- Attack major risks early and continuously... or they will attack you.
- Stay focused on executable software.
- Accommodate change early in the project.
- Baseline an executable architecture early on.
- Build your system with components.
- Make quality a way of life, not an afterthought.
- Ensure that you deliver value to your customer.
- Work together as one team.

RUP: Prozesse I

Phasen:

- 4 Phasen (Management Perspektive)



- Konzeption, Entwurf, Realisierung, Einführung
-Evolution
- Die Gewichtung der einzelnen Phasen ist stark abhängig von den Anforderungen des Projektes

RUP: Prozesse II

Iterationen

➤ Definiert:

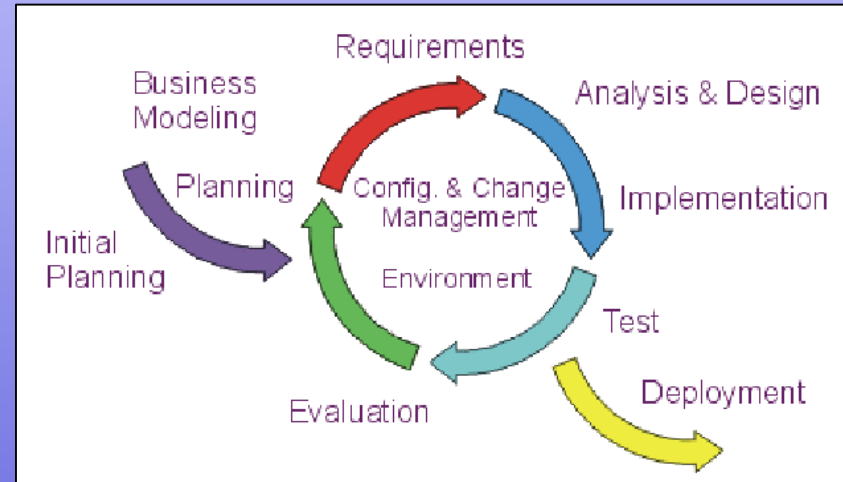
- Aufgaben, Rollen
- Zeitplan, Artifacts

➤ Ziele:

- Lauffähige Software
- Angehen/Lösen der risikoreichsten Tasks
- Phasenziel/Meilenstein untergeordnet

➤ Ablauf

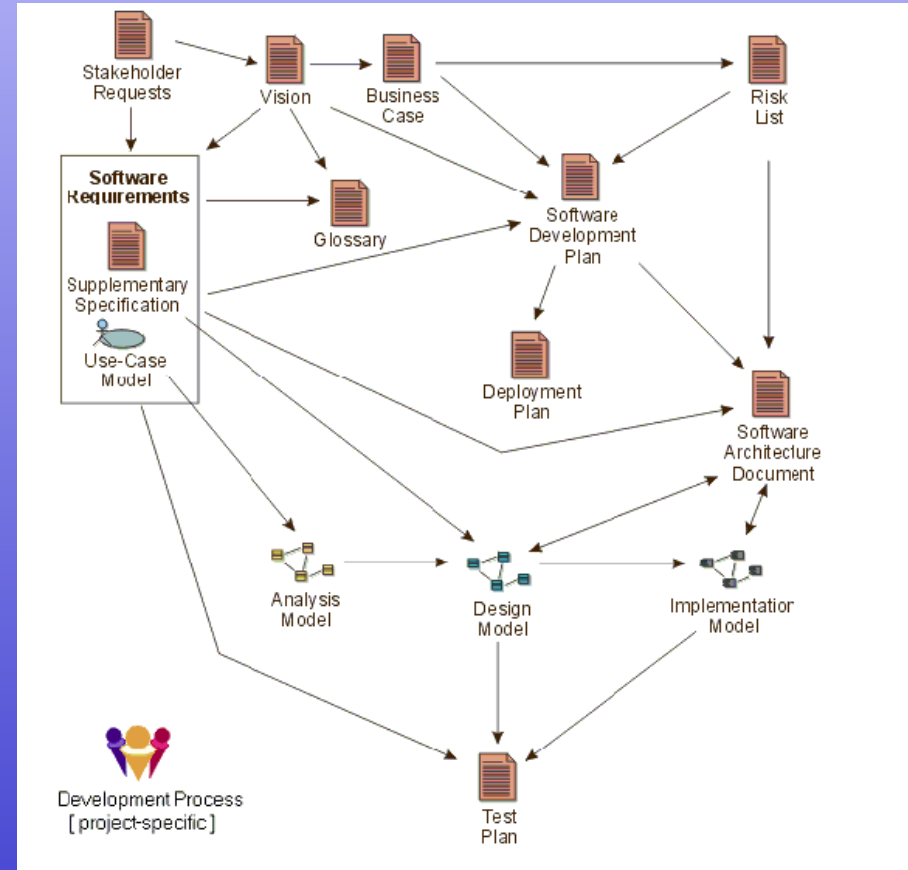
- Sequentiell: Planung, Anforderungsspezifikation, Analyse&Design, Implementation, Test und Evaluation
- Einmalig
 - Projektbeginn: Initialplanung
 - Projektende: Deployment?



RUP: Artifacts I

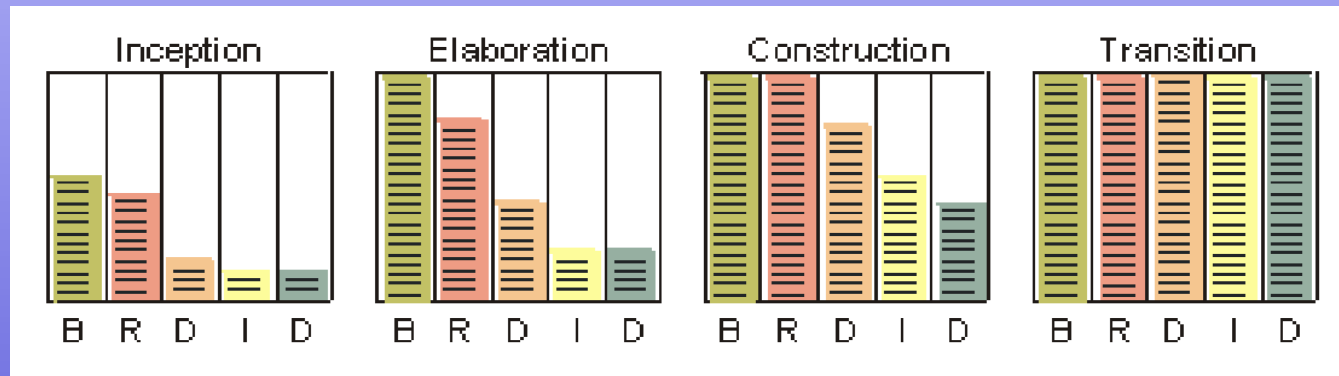
Dokumentation

- Templates
- Rollen und Aufgaben zugeordnet
- Visuelles: UML
- Mehr als 100!
 - RUP Anti-Pattern



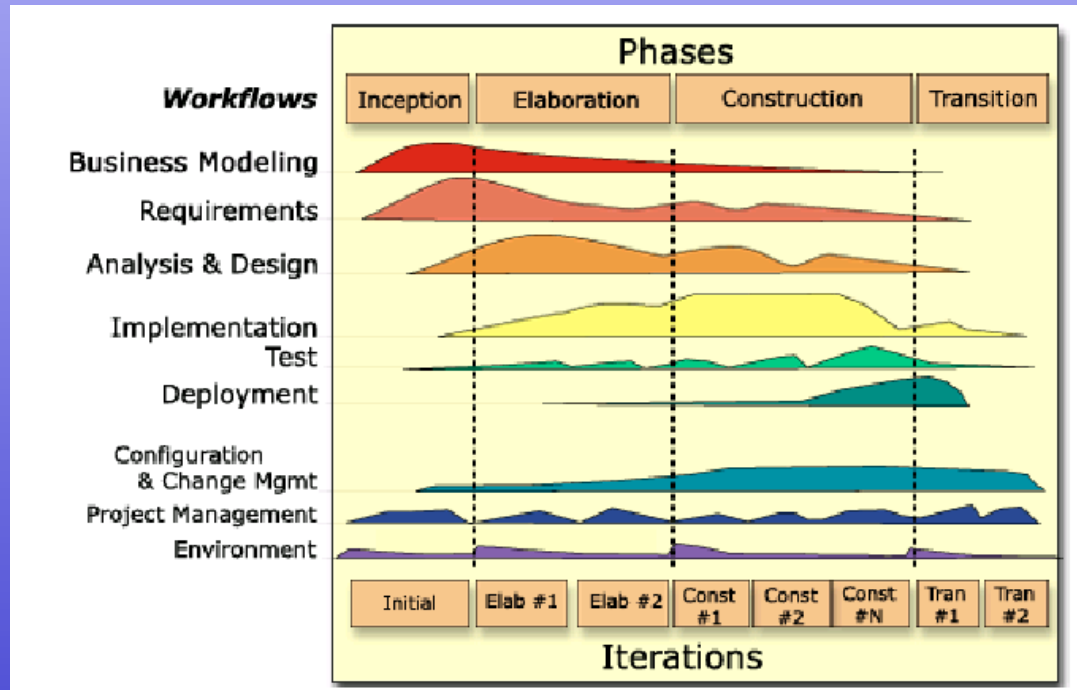
RUP: Artifacts II

Entwicklung



- Zu erstellende Dokumente werden definiert
 - Faustregel: Wenn man unsicher ist ob nötig → nein!
 - Owner und Zeitplan fixieren
- Unterscheidung von Technical- und Management Artifacts

RUP: Rollen & Aufgaben I



RUP: Rollen & Aufgaben II

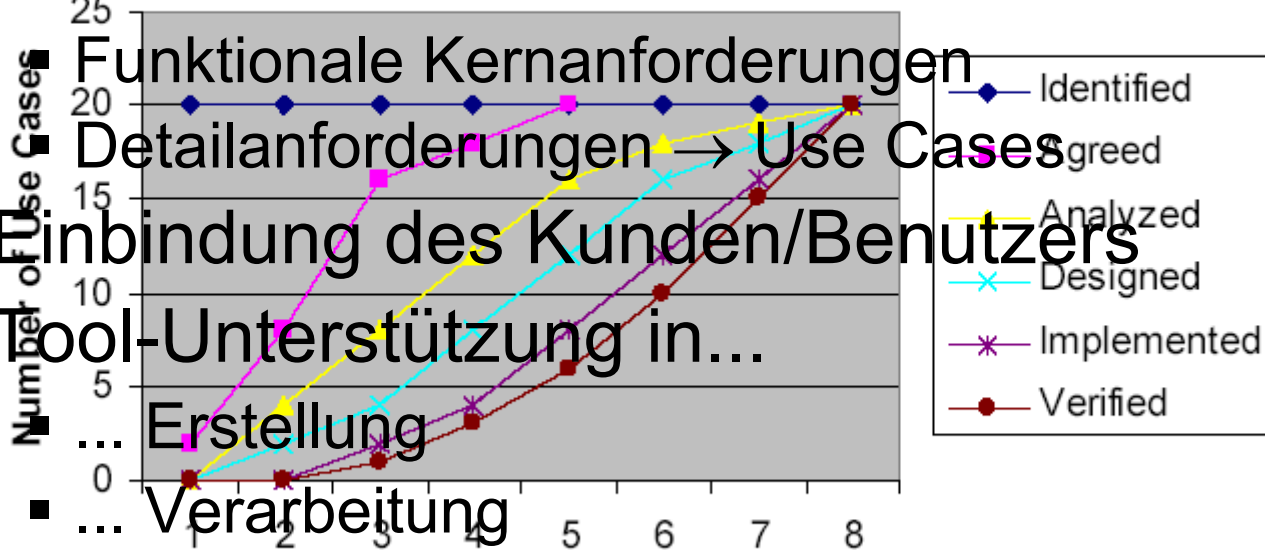
- RUP definiert weit über 40 Rollen
 - Die wichtigsten:
 - Projektleiter
 - Qualitätsmanager
 - Domänenexperte
 - Architekt
 - 1 Person kann mehrere Rollen besetzen (üblich)
 - Die Rollen werden Disziplinen zugeteilt (9)
 - Rollen führen Aufgaben aus, Aufgaben gliedern sich in Schritte...aha!

RUP: Requirements I

➤ 3 Ebenen

Project Progress

- Vision und Business Case



➤ Einbindung des Kunden/Benutzers

➤ Tool-Unterstützung in...

➤ Widersprüchliches

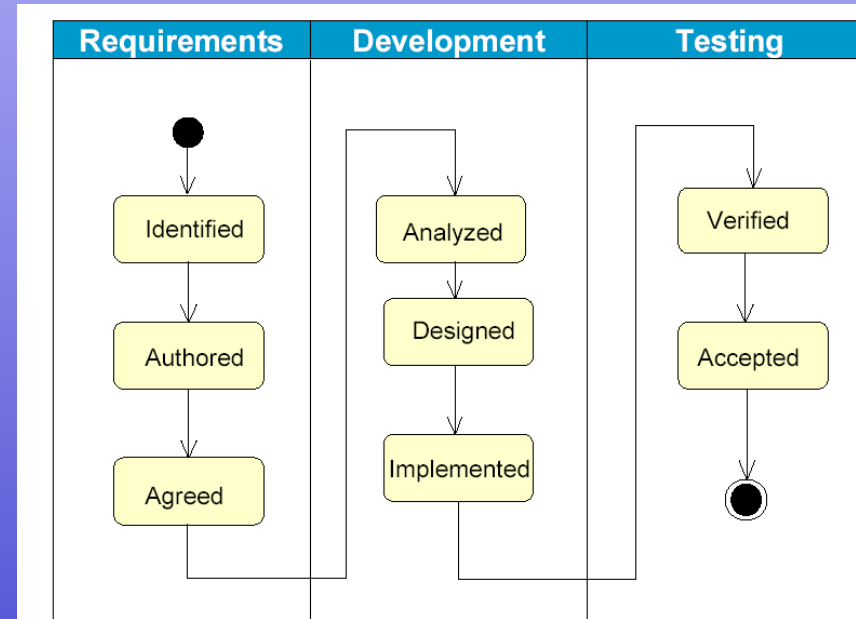
- Use Case Driven <=> Requirements Driven
- Messung des Projektfortschritts

RUP: Requirements II

Lebenslauf

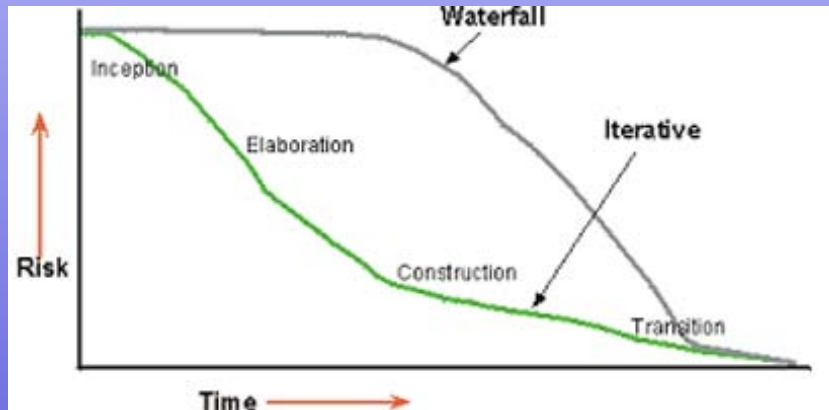
➤ Über alle Phasen

State	Inception	Elaboration	Construction	Transition
Identified	60%	> 80%	100%	100%
Outlined	50%	20-60%	0%	0%
Authored	10%	40-80%	100%	100%
Analysed	< 10%	20-40%	100%	100%
Designed, Impl., Verified	< 5%	< 10%	100%	100%



RUP: Risk Management I

Risikominimierung durch Iteration:



RUP: Risk Management II

Identifikation → Risk List

- Risk 1: We are concerned, based on past experience, that Department X will not understand what requirements we plan to provide, and as a result will request changes after beta software is delivered.
- Risk 2: We do not understand how we can integrate with legacy system Y.
- Risk 3: We have no experience in developing on the Microsoft .NET platform or in using Rational Rose.

RUP: Risk Management III

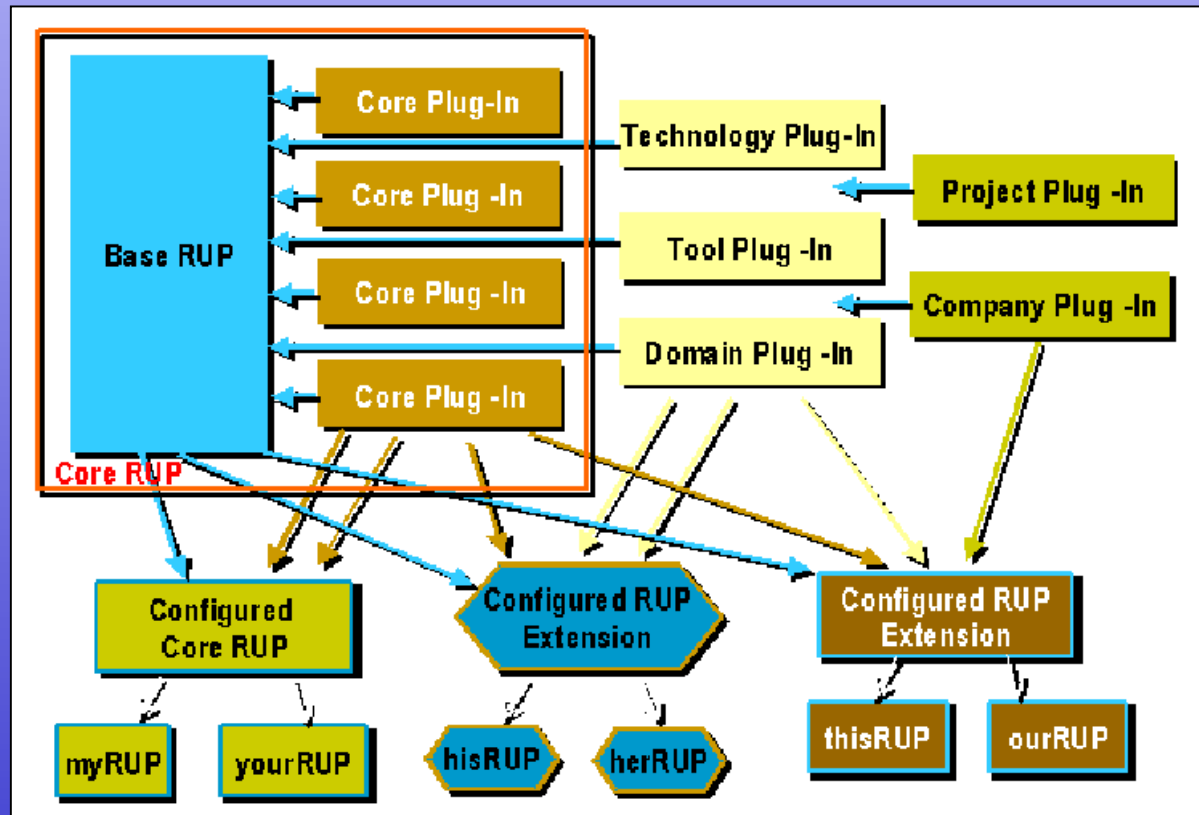
Risk Mitigation / Avoidance

- Risk 1:
 - Ergänzung mit UI Prototyp in Use Case Definitionsphase
 - Meeting zur Durchsicht und formaler Signoff
 - Department X während des Projekts auf dem laufenden halten (kurze Feedback-Zyklen)
- Risk 2 (1):
 - "tiger team" erstellt Wegwerf-Protoypen zur Demonstration der technischen Machbarkeit
 - Kontinuierliches Testen der Integration von SystemY während des Projektverlaufes
- Risk 2 (2):
 - Integration von System Y wird aus Projekt Scope geschmissen
- Risk 3:
 - Schulung von ausgesuchten Leuten auf Microsoft .NET und Rational Rose
 - Rational Rose Mentor für einzelne Tage in früher Projektphase einbinden
 - Rekrutierung von Mitarbeitern mit entsprechender Kompetenz

RUP-Tools I

<p>Die Rational Suite</p> <ul style="list-style-type: none"> • Configuration Management • Process Delivery Tools • Process Authoring Tools • Community / Marketplace 	<p>RUP Kernfunktionalitäten zur Planung des Projektes:</p> <ul style="list-style-type: none"> • Definieren von Parametern • Definieren von Prozessen • Publizieren von Prozessen (web-basiert) • Interaktionstools
Rational ClearCase	Configuration Management Environment
Rational ClearQuest	Change Management
Rational ProjectConsole	Demo..
Rational PurifyPlus	Debugging und Diagnose
Rational Requisite Pro	Verarbeitung von Anforderungen
Rational Robot	Automatisiertes Testen
Rational Rose	Modellierung mit UML
Rational SoDa	Dokumentationsgenerierung in verschiedenen Formaten
Rational TestManager / Test Realtime / TestFactory	Support für Testing: Testgenerierung, Durchführung, Auswertung
Rational Process Workbench	Unterstützungstool für die Projektkonfiguration
RUP Builder	Setup von Projekten, Management von Plugins
<p>Rational sponsored Plug-Ins</p> <p>RUP Plug-In for Sun iAS RUP Plug-In for SunOne RUP Plug-In for Extreme Programming (XP) RUP Plug-In for System Engineering RUP Plug-In for User-Experience Modeling RUP Plug-In for J2EE RUP Plug-In for RealTime RUP Plug-In for Business Modeling RUP Plug-In for Asset-Based Development RUP Plug-In for Creative Web Design</p>	<p>Partner Plug-Ins</p> <p>RUP Plug-In for Existing Software Reuse by Klocwork RUP Plug-In for Business Rules RUP Plug-In for Achieving Straight-Through Processing ICONIX QuickStart Plug-In for RUP RUP Plug-In for CAST Application Mining Suite RUP Plug-In for Operations and Application Surveillance Management</p>

RUP-Tools II: Componentized RUP



RUP-Tools III: Demo RUP-Trial

RUP

(Das ist noch nicht der Hammer...)

RUP-Builder

Idee eines „Process Market Place“

Diskussion: Agilität des RUP I

Agile Elemente:

- Iteratives Vorgehen
 - Inkrementelle Entwicklung
 - Lauffähige Software
 - Kampf der Analysis-Paralysis
 - Verwendung von Use Cases
 - Betonung der Wichtigkeit von direkter Kommunikation und Zusammenarbeit
- RUP wird gerne als agil verkauft

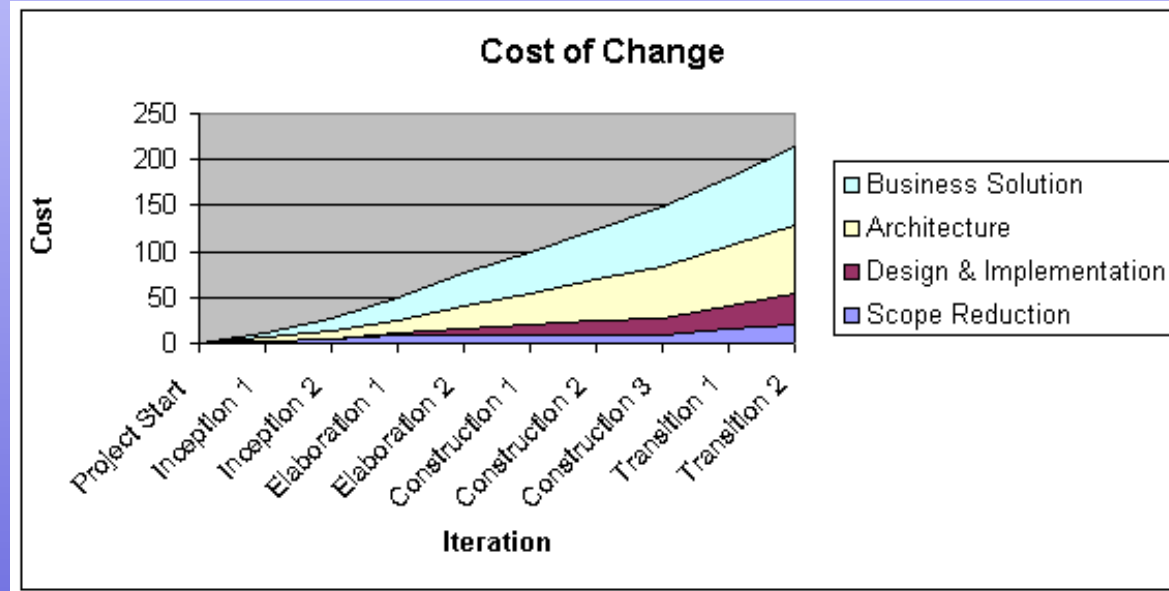
Diskussion: Agilität des RUP II

Kontrapunkte:

- Zu iterativem Vorgehen/inkrementelle Entwicklung: die Iterationen sind geplant → Projekt Grobplanung deckt gesamte Laufzeit ab
- Kampf der Analysis-Paralysis: Schwergewichtige und viele Artifacts
- Betonung der Wichtigkeit von direkter Kommunikation und Zusammenarbeit: Diese ist erwünscht aber nicht gefordert.
- RUP legt grossen Wert auf die Architektur, zudem wird diese sowie Kernanforderungen als relativ stabil angesehen

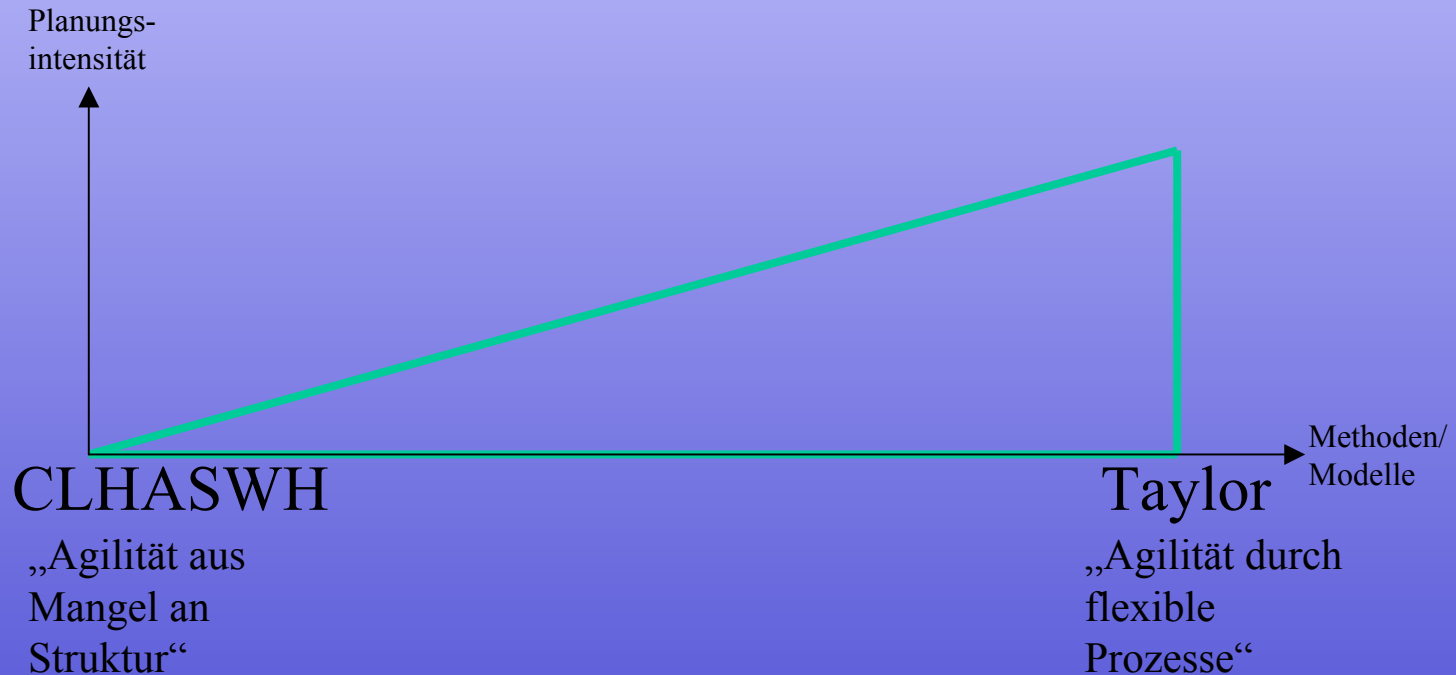
Diskussion: Agilität des RUP III

➤ Change



- Mächtigkeit des Frameworks / Tools etc.
 - Des einen Freude, des anderen...
 - RUP selbst sieht sich als reich gedecktes Buffet, wo jeder eingeladen ist, das raus zu picken was er am meisten mag. Es empfiehlt sich jedoch nicht, das ganze Buffet zu verschlingen...
- Agilität durch flexibel steuerbare Prozesse
 - Process Manager

Fazit I: 2 Pole des SE?



- Kontinuum
- Wo auf der x-Achse für welches Projekt?
 - Management Job

Fazit II

„...the big system people see RUP as the answer to their problems; the small system community sees XP as the solution to their problems. Our experience indicates that most software projects are somewhere in between - trying to achieve the right level of process for their situation. Neither end of the spectrum is sufficient for them...“ (Gary Pollice, Rational; Expanding Upon eXtreme Programming)