Vortragsreihe „ Software Engineering for Everyday Business“

Software Engineering & Software Prozesse

Teil 1c: Ausgewählte Softwareentwicklungsprozesse

Dietmar Winkler, Michael Pernkopf

Technische Universität Wien
Institut für Softwaretechnik und Interaktive Systeme

dietmar.winkler@qse.ifs.tuwien.ac.at
http://qse.ifs.tuwien.ac.at

Inhalt

- **Teil 1a: Einführung in die Softwareentwicklung (ca. 60‘)**
  - Typisierung von Projekten
  - Erfolgs/Misserfolgsfaktoren in der Softwareentwicklung
  - Rollen in der Softwareentwicklung
- **Teil 1b: Life-Cycle Phasen (ca. 60‘)**
  - Softwarelebenszyklus
  - Ausgewählte Aspekte in den jeweiligen Phasen
  - Grundlegende Softwareprozesse in Anlehnung an den Life-Cycle Prozess.
- **Teil 1c: Ausgewählte Softwareentwicklungsprozesse (ca. 60‘)**
  - V-Modell XT
  - Rational Unified Process
  - Agile Softwareentwicklung

Motivation und Zielsetzung

- Zielsetzung bei der Softwareentwicklung ist die Erstellung qualitativ hochwertiger Produkte innerhalb eines definierten Zeitraums und innerhalb des Budgetrahmens.
- Softwareprodukte umfassen ALLE Produkte im Rahmen der Softwareentwicklung (z.B. Spezifikation, Code, Testfälle, Protokolle).

- Software Prozesse unterstützen die Softwareentwickler bei der Erstellung qualitativ hochwertiger Produkte.
  - Unterschiedliche Projekte benötigen jedoch angepasste Vorgehensweisen (verschiedene Software Prozesse bzw. Vorgehensmodelle)
  - Software Prozesse sind abhängig von Projekttyp, Projektgröße, Anwendungsdomäne usw.
  - Software Prozesse orientieren sich am Produkt Life-Cycle Prozess jedoch mit unterschiedlichem Schwerpunkt.

V-Modell XT

- Das V-Modell XT ist eine Weiterentwicklung des V-Modell 97.
- Veröffentlichung im Februar 2005.
- Laufende Weiterentwicklung (derzeit Version 1.3)
- Verpflichtendes Vorgehensmodell für IT Projekte im öffentlichen Bereich in Deutschland.

Zielsetzung der Entwicklung des V-Modell XT

- Verbesserung der Unterstützung von Anpassbarkeit, Anwendbarkeit, Skalierbarkeit und Änder- und Erweiterbarkeit des V-Modells.
- Berücksichtigung des neuesten Stand der Technik.
- Kompatibilität zu formalen Richtlinien und Standards (z.B. ISO 9000 Standard, CMMI).
- Erweiterung des Anwendungsbereiches auf die Betrachtung des Systemlebenszyklus.
- Integration eines Prozessmodells zur “Einführung und Pflege eines organisations-spezifischen Vorgehensmodells).

Philosophie

- Produkte stehen im Mittelpunkt (=Projektergebnisse)
- Projektdurchführungsstrategien und Entscheidungspunkte geben die Reihenfolge der Produktfertigstellung und somit den Projektverlauf vor.
- Projektplanung und –steuerung basiert auf der Bearbeitung und Fertigstellung von Produkten.
- Für jedes Produkt ist eine definierte Rolle verantwortlich.
- Die Produktqualität ist überprüfbar
  - Anforderungen an das Produkt
  - Definierte Produktabhängigkeiten.

→ grundlegende Komponenten des V-Modell XT.

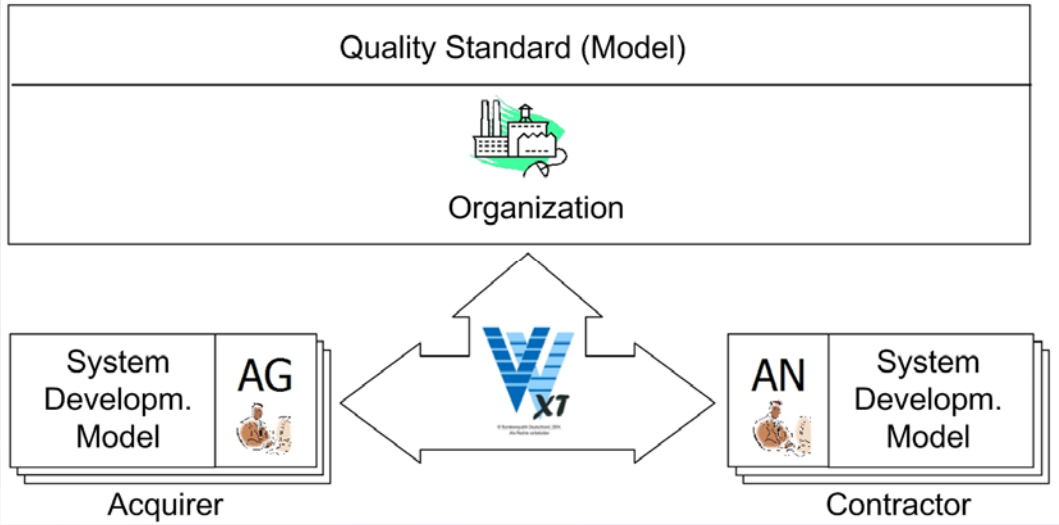
Komponenten des V-Modell XT

- Projekttypen
- Produkte, Aktivitäten, Rollen.
- Integrierte Methoden- und Toolunterstützung zur
  - Erstellung von Produkten durch
  - Aktivitäten und
  - Rollen (verantwortlich für ein Produkt).
- Vorgehensbausteine
  - Verpflichtende Elemente (core elements)
  - Optionale Elemente (um individuelle Projektanforderungen erfüllen zu können)
- Entscheidungspunkte definieren einen Zeitpunkt, an dem eine Fortschrittsentscheidung getroffen wird.
- Projektdurchführungsstrategien definieren die Reihenfolge der im Projekt zu erreichenden Projektfortschrittsstufen (Sequenz von Entscheidungspunkten)
- Durch die Struktur des V-Modell XT ist eine Vergleichbarkeit zu herkömmlichen Prozessmodellen möglich (z.B. Konventionsabbildungen zu Prozessmodellen und Standards)

Projekttypen

Projekttypen werden eingeteilt:

- Nach Projekttyp (z.B. Hardwaresystem, Softwaresystem, Komplexes System)
- Projektrollen (z.B. Auftraggeber / Auftragnehmerprojekte)

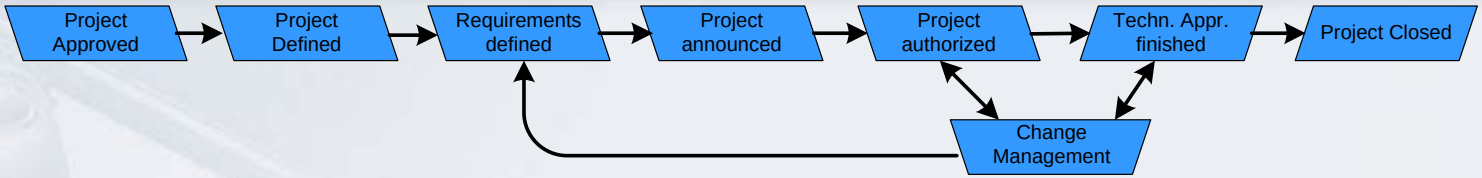


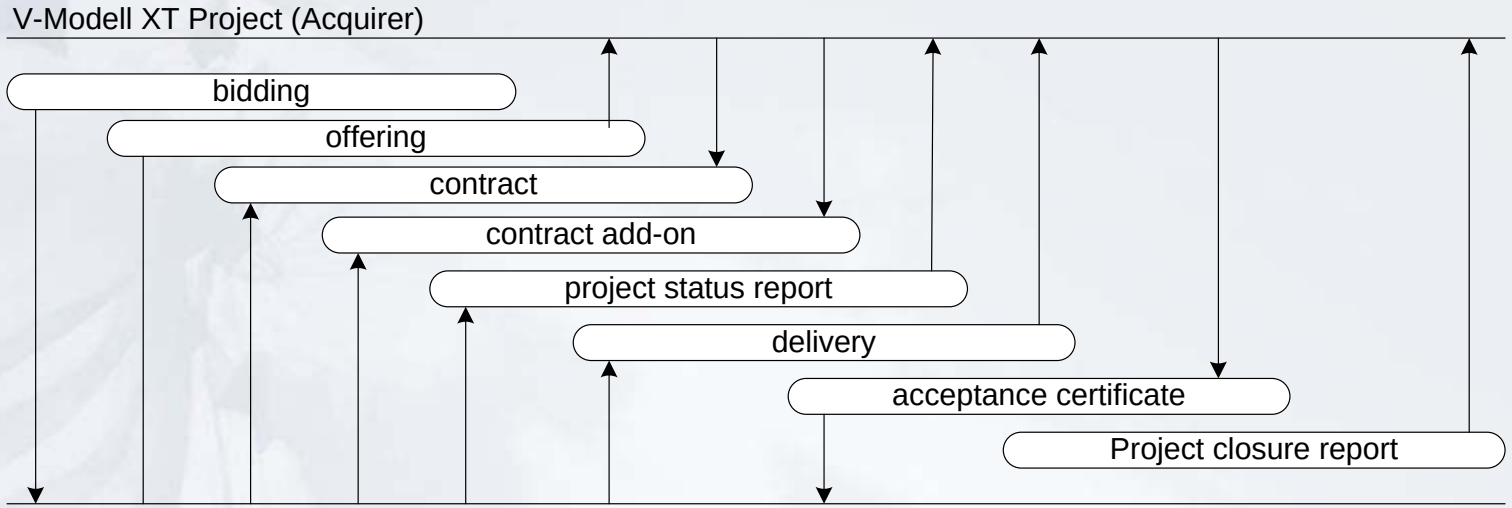
→ daraus resultieren 4 Projekttypen

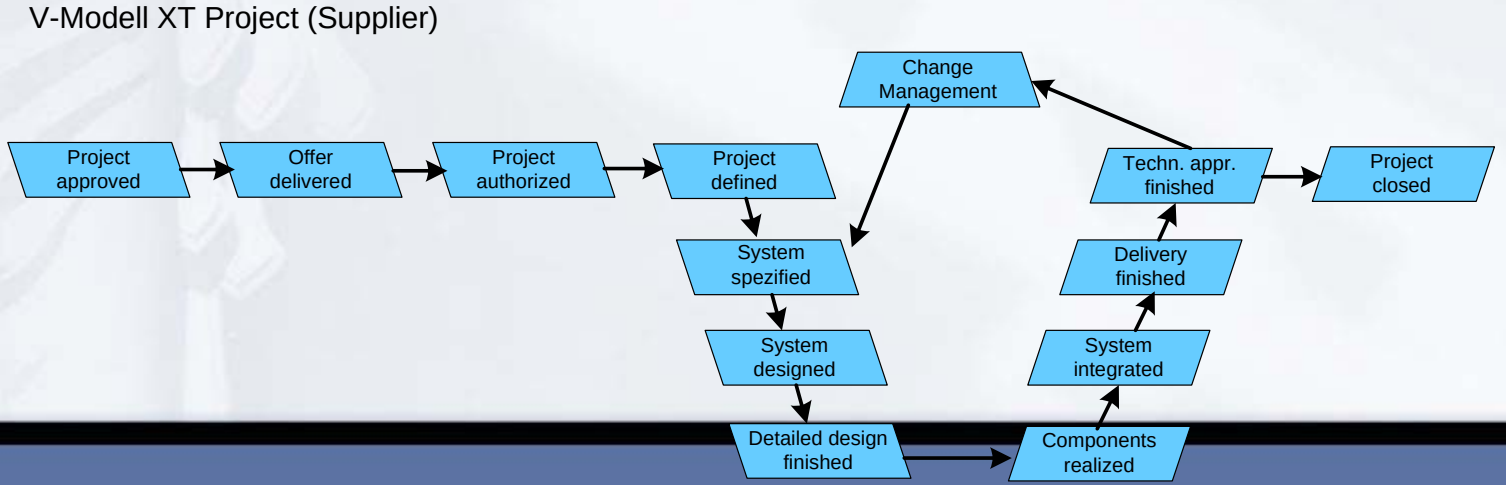
- Systementwicklungsprojekt des Auftraggebers.
- Systementwicklungsprojekt des Auftragnehmers.
- Einführung und Pflege eines organisations-spezifischen Vorgehensmodells

- Neu: Systementwicklungsprojekt (Auftraggeber/Auftragnehmer); z.B. in-house System Entwicklung (seit der Version 1.2 integriert).

Kommunikation Auftraggeber / Auftragnehmer

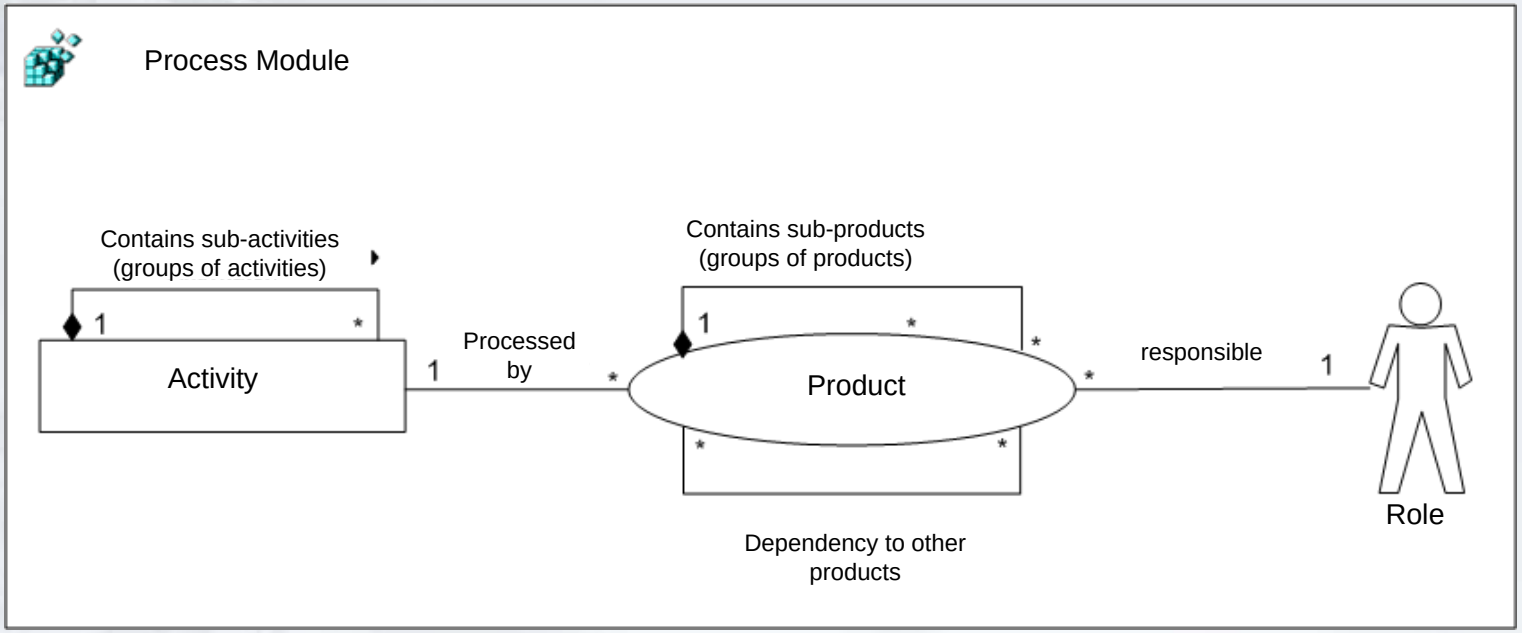






Aufbau von Vorgehensbausteinen

- **Vorgehensbausteine sind die modularen Elemente des Prozessmodells.**



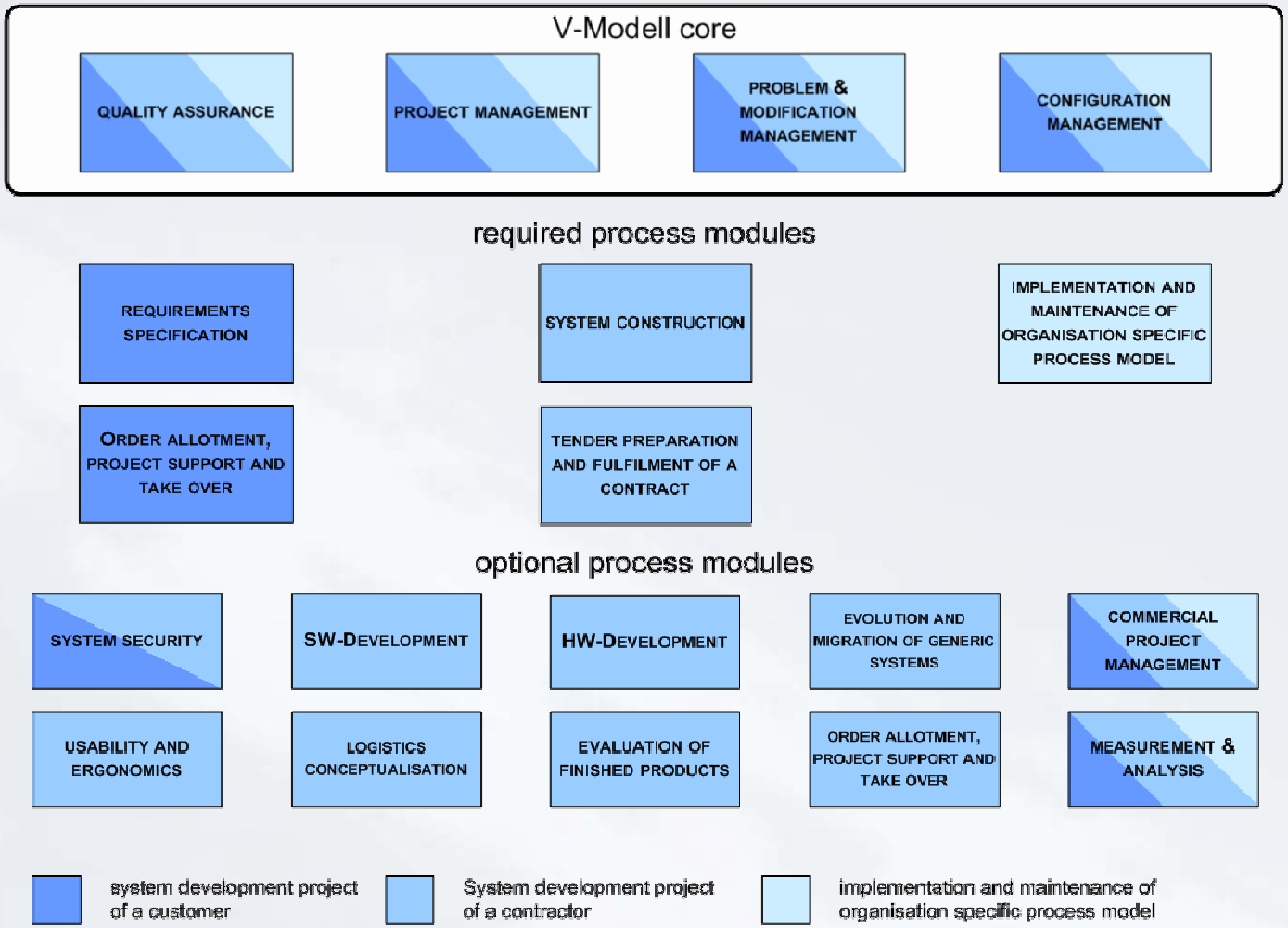
- **Ein Vorgehensbaustein**
  - kapselt Rollen, Produkte und Aktivitäten.
  - Kann als unabhängige Einheit eingesetzt werden.
  - Ist eine Einheit, die unabhängig veränder- und aktualisierbar ist.

Vorgehensbausteine im V-Modell XT

Vorgehensbausteine

- Verpflichtende Elemente (für alle Projekttypen)
- Erforderliche Elemente (abhängig vom Projekttyp)
- Optionale Elemente

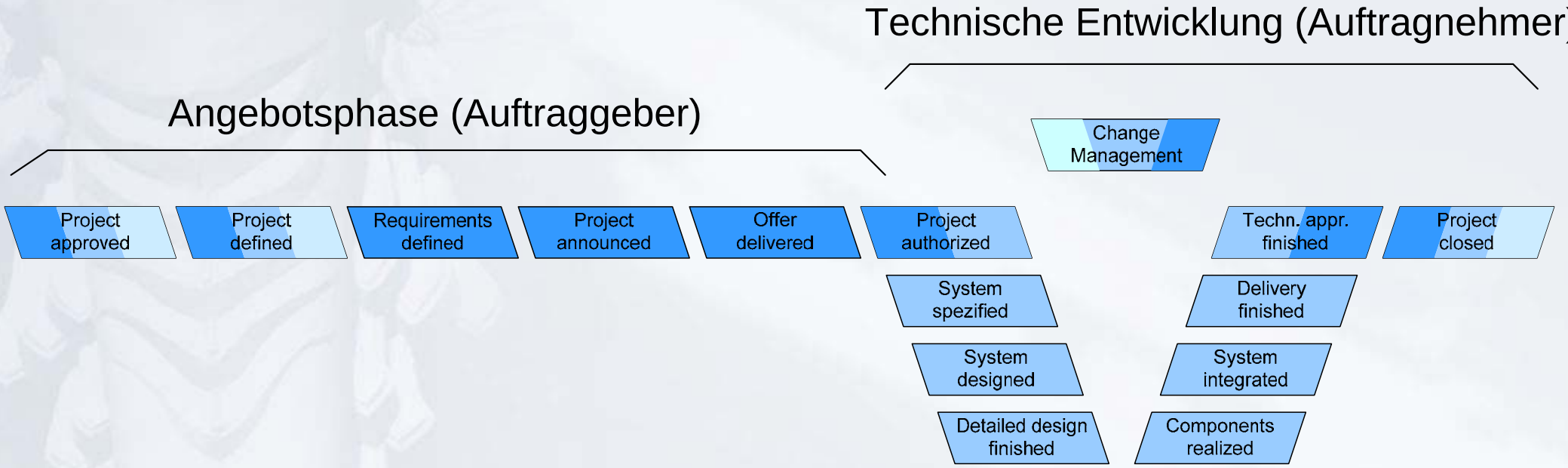
- Tool-Unterstützung durch den V-Modell Assistenten.



Entscheidungspunkte

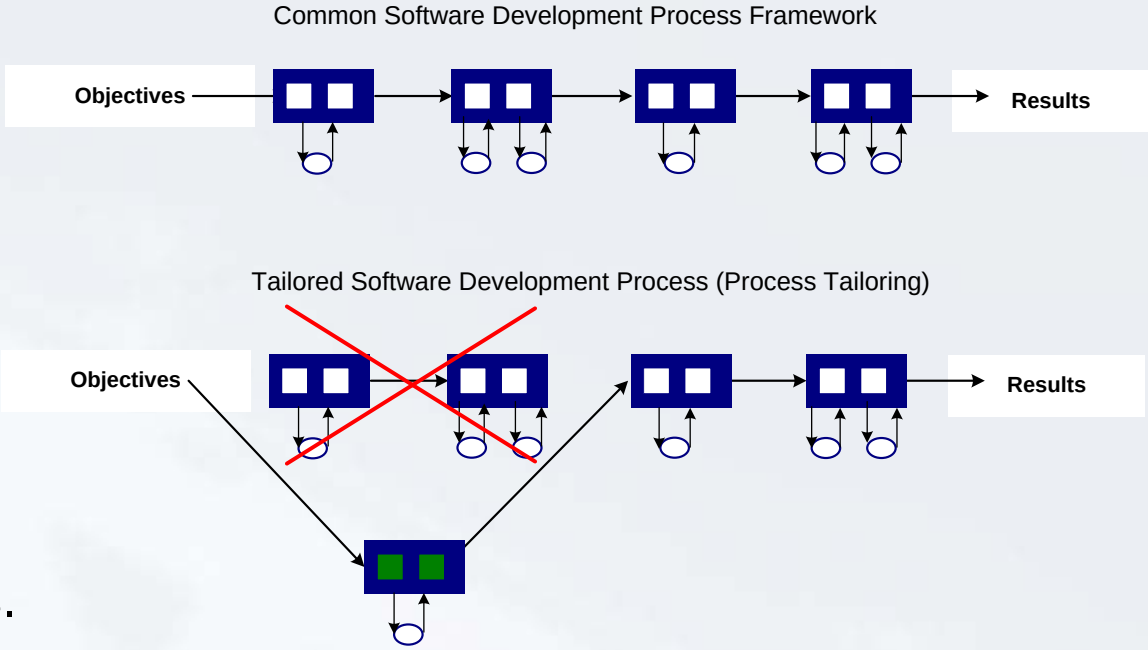
Auswahl einer Projektdurchführungsstrategie

- Definition von Entscheidungspunkten (vergleichbar mit “Meilensteinen”)
- Am Entscheidungspunkt müssen definierte Produkte vorliegen.
- Beispiel:



Flexibilität durch Prozesstailoring

- Anpassbarkeit des generischen Entwicklungsprozesses an spezifische Projektgegebenheiten durch Prozesstailoring.
- Ersetzen einzelner Prozess-Schritte (oder Vorgehensbausteine) durch passende alternative Lösungen.
- Wiederverwendung von Best-Practices (Methoden / Tools).
- Individuelle Anpassung des Projektplans.



- **Achtung:** Tailoring erfordert erfahrene Projektleiter, die Tailoring im Rahmen des Softwareprozesses durchführen (Berücksichtigung von definierten Tailoring-Kriterien).

Tailoring im V-Modell XT

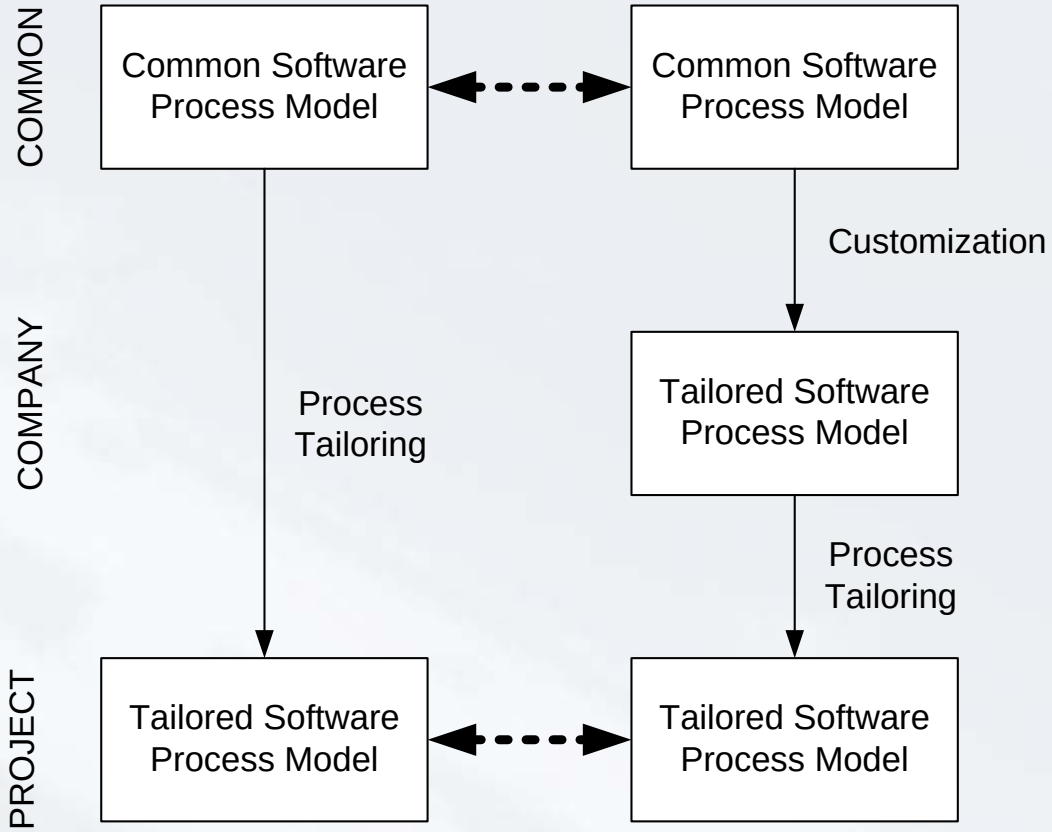
- Tailoring im V-Modell XT umfasst
  a) die Auswahl des Projekttyps,
  b) zugeordnete / wählbare Vorgehensbausteine (core / optional modules)
  c) Auswahl der Projektdurchführungsstrategie (d.h., die Anordnung von Entscheidungspunkten).

- Das V-Modell XT stellt einen Rahmen zur Verfügung, in den existierende Softwareprozesse integriert werden können.

- Tool Unterstützung:
  - Tailoring durch einen Projektassistenten.
  - Anpassung des V-Modell XT an unternehmensspezifische Gegebenheiten durch einen “V-Modell editor”.
  - Open Source Tools via www.v-model-xt.de verfügbar;
    Steigende Anzahl von V-Modell XT Unterstützung in verschiedenen Entwicklungsumgebungen, z.B. microTOOL, 4soft, IBM, Borland.

Prozess Customizing

- Anpassung des Vorgehensmodells an unternehmensspezifische Gegebenheiten.
- Zur Effizienzsteigerung von Tailoring ähnlicher Projekte kann der grundlegende Prozess an das Unternehmen oder Projektfamilien angepasst werden → Customization.
- Diese “angepassten processes” können als Basis für projektspezifisches Tailoring eingesetzt werden.
- Die Verträglichkeit zum zugrunde liegenden Prozess muss sichergestellt werden!



Zusammenfassung

- Das V-Modell XT ist ein verpflichtendes Vorgehensmodell für IT Projekte im öffentlichen Bereich in Deutschland.
- Das V-Modell XT stellt Projekttypen für den Auftraggeber (Kunden) und Auftragnehmer bereit.

- Unterstützung von iterativen Systementwicklungsprojekten.
- Produkte stehen im Vordergrund. Sie werden durch Aktivitäten erzeugt. Für Produkte sind definierte Rollen verantwortlich.
- Die Flexibilität des V-Modell XT ermöglicht die Anwendbarkeit ohne Einschränkung der Anwendungsdomäne.
- Vorgehensbausteine berücksichtigen zusammengehörige Themen, wie z.B. Hard/Softwareprodukte, Logistik, Security usw.
- Das Prozessmodell ermöglicht die flexible Anordnung von verpflichtenden (V-Modell Kern) und optionalen Vorgehensbausteinen.
- Unterstützung durch Open Source Tools.

Rational Unified Process (RUP)

- Das phasenorientierte Prozessmodell wurde von IBM entwickelt.
- Weite Verbreitung im industriellen Umfeld.
- Umfassender Tool Support (Rational ROSE / XDE, Rational ClearCase).

- Fokussiert auf den technischen Entwicklungsprozess
- Aufbau in 4 grundlegende Phasen und 9 Workflows / Disziplinen.
- RUP unterstützt Best-Practices in der modernen Softwareentwicklung, z.B.
  - Iterative Softwareentwicklung (Prototyping)
  - Anforderungsmanagement mit Change Request Handling.
  - Wiederverwendung von Software Komponenten.
  - Visualisierung von Softwaremodellen (UML Diagrammfamilie)
  - Kontinuierliche Überprüfung der Softwarequalität (Reviews, Inspektionen, Tests)
  - Versioning (Rational Clearcase)
- Durch einen umfassenden Prozessansatz unterstützt RUP auch große Projekte.

RUP Konzepte

- **Rollen**
  z.B. Software Architekt, Entwickler, usw.

- **Aktivitäten**
  - Personen (Rollen) führen Aktivitäten aus (d.h. Umsetzung von Arbeitspaketen)
  - Aktivitäten können in Teil-Aktivitäten aufgeteilt werden.
  - Betreffen typischerweise einzelne oder nur wenige Artefakte.
  - Sie stellen die Basis für eine detaillierte Iterationsplanung dar.
  - Beispiele: Definition der Architektur, Implementierung von Komponenten, usw.

- **Artefakte**
  sind die zu liefernden Produkte, z.B. Architekturdokument, Komponenten usw.

- **Workflows / Disciplines**
  definieren konkrete Abläufe für Aktivitäten zur Erstellung von Produkten (Artefakten)

Iterationen

- Die sequentielle Entwicklung basiert auf der Annahme, dass die grundlegenden Anforderungen während des gesamten Projektes konstant sind (z.B. Life-Cylce Modell).
- Änderungen der Anforderungen können nachträglich nur mit erheblichem Aufwand (Zeit, Kosten, Personal) umgesetzt werden.

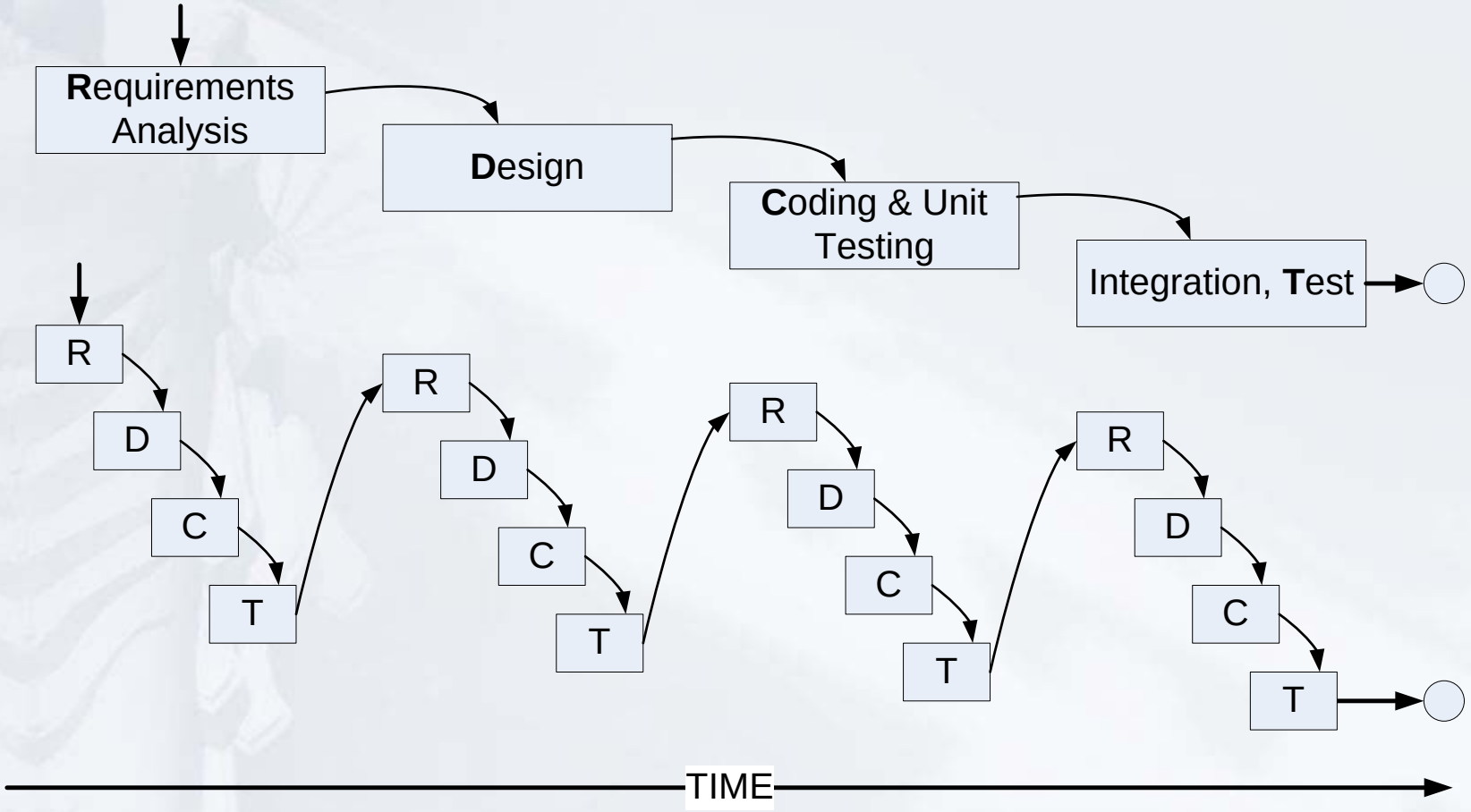
ABER: Anforderungen, Technologien und Marktanforderungen können sich ändern!

→ Iterative Entwicklung durch den Rational Unified Process.

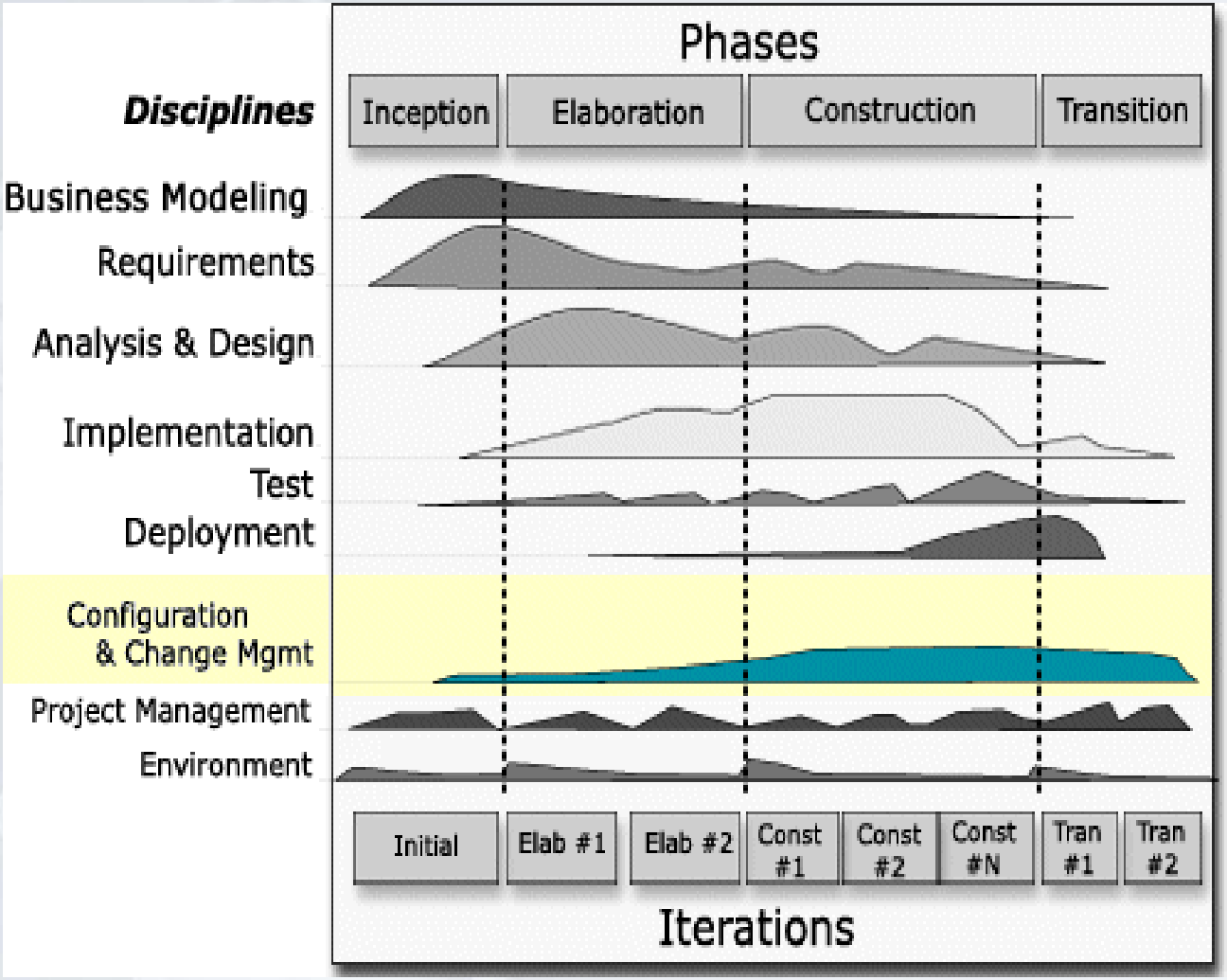
- Softwareentwicklung in „kleinen Schritten“ (statt „Big Bang“ Lösungen).
- Jede Iteration trägt ihren Anteil an der Produkt- und Projektfertigstellung.
- Kleinere Schritte erhöhen die Möglichkeit zur Fortschrittskontrolle.
- Kontinuierliches Lernen während des Projektverlaufs.
- Kleinere Module können einfacher wieder verwendet werden.

Beispiel für Iterative Entwicklung

Von der sequentiellen Entwicklung zum Iterativen Life-Cylce.



RUP Prozessmodell



4 Phasen:

- Inception (Begin)
- Elaboration (Concept & Design)
- Construction
- Transition (Delivery)

- Jede Phase schließt mit einem Meilenstein ab.

Workflows / Disciplines:

- 6 Engineering Workflows
- 3 Supporting Workflows
- Several iterations within one phase.

RUP – 4 Phases

Inception (Begin)

- Definition von Projektkontext und Akzeptanzkriterien.
- Definition von Anwendungsfällen (Use Cases)
- Kostenschätzung und Terminplanung.
- Risikoabschätzung.
- Typischerweise 1 Iteration.

Elaboration (Concept and Design)

- Analyse der Anwendungsdomäne
- Definition der Architektur
- Verfeinern des Projektplans
- Typischerweise 1-2 Iterationen.

Construction

- Erstellung der Software Produkte
- Anwenderhandbuch.
- Zahlreiche Iterationen gemäß Projektplanung.

Transition (Delivery)

- Beta Testphase
- Schulung von Anwendern und Servicepersonal.
- Übergabe an den Kunden.
- Typischerweise 1-2 Iterationen.

RUP – 9 Disciplines

6 Engineering Disciplines

- Business Modelling Discipline (Geschäftsfälle definieren)
- Requirements Discipline (Anforderungen festlegen)
- Analysis and Design Discipline (Analyse und Design)
- Implementation Discipline (Softwareerstellung)
- Test Discipline (Verifikation und Validierung)
- Deployment Discipline (Auslieferung und Inbetriebnahme)

3 Supporting Disciplines

- Project Management Disciplin
- Configuration and Change Management Discipline
- Environment Discipline

Beispiel für eine Disziplin

Requirements Discipline → Engineering Discipline

Ziel und Zweck

- Festlegung der grundlegenden Systemfunktionalität (“was soll das System leisten”)
- Definition der Systemgrenzen.
- Grundlegende Information über Projektkosten/-dauer und Planung.
- Festlegen der Benutzerschnittstellen.
- Definition von nicht-funktionalen Anforderungen (z.B. Verfügbarkeit, Performance)

Beispiel: Grundlegenden Anforderungen für ein Computerspiel.

Purpose	Allow users to select game level and number of players. Show high score and supporting screen.
Inputs	Mouse clicks on various options.
Processing	None
Outputs	The selected action is performed.

Requirements Discipline – Rollen

- **System Analyst**
  - Erstellt ein Dokument zur Beschreibung der Projektziele (Vision).
- **Requirement Specifier & Stakeholders**
  - Definition und Vereinbarung der Systemanforderungen (aus Benutzersicht).
- **Software Architect**
  - Anwendungsfalldiagramm (Use-Case Modell) inkl. beteiligter Akteure.
  - Ergänzende Spezifikationen.
  - Glossar zur einheitlichen Begriffsbildung.
  - Storyboards für Benutzerschnittstellen.

Toolunterstützung (Beispiel)

- IBM Rational Requisite Pro zur Bearbeitung von Anforderungen und Anwendungsfällen.

Zusammenfassung

- Iterative vs. inkrementelle Prozess-Workflows.
- Integriertes Anforderungsmanagement.
- Visuelle Modellierung durch die UML Diagrammfamilie.
- Modellierung von Anforderungen aus Kundensicht (Use-Cases)
  → “real world scenarios”
- Unterstützung komponenten-basierter Architektur.
- Überprüfung der Softwarequalität (aller Produkte) an Meilensteinen, die die jeweiligen Phasen / Iterationen abschließen.
- Fokus auf Change Management.
- Unterstützung durch umfassende Dokumentation.

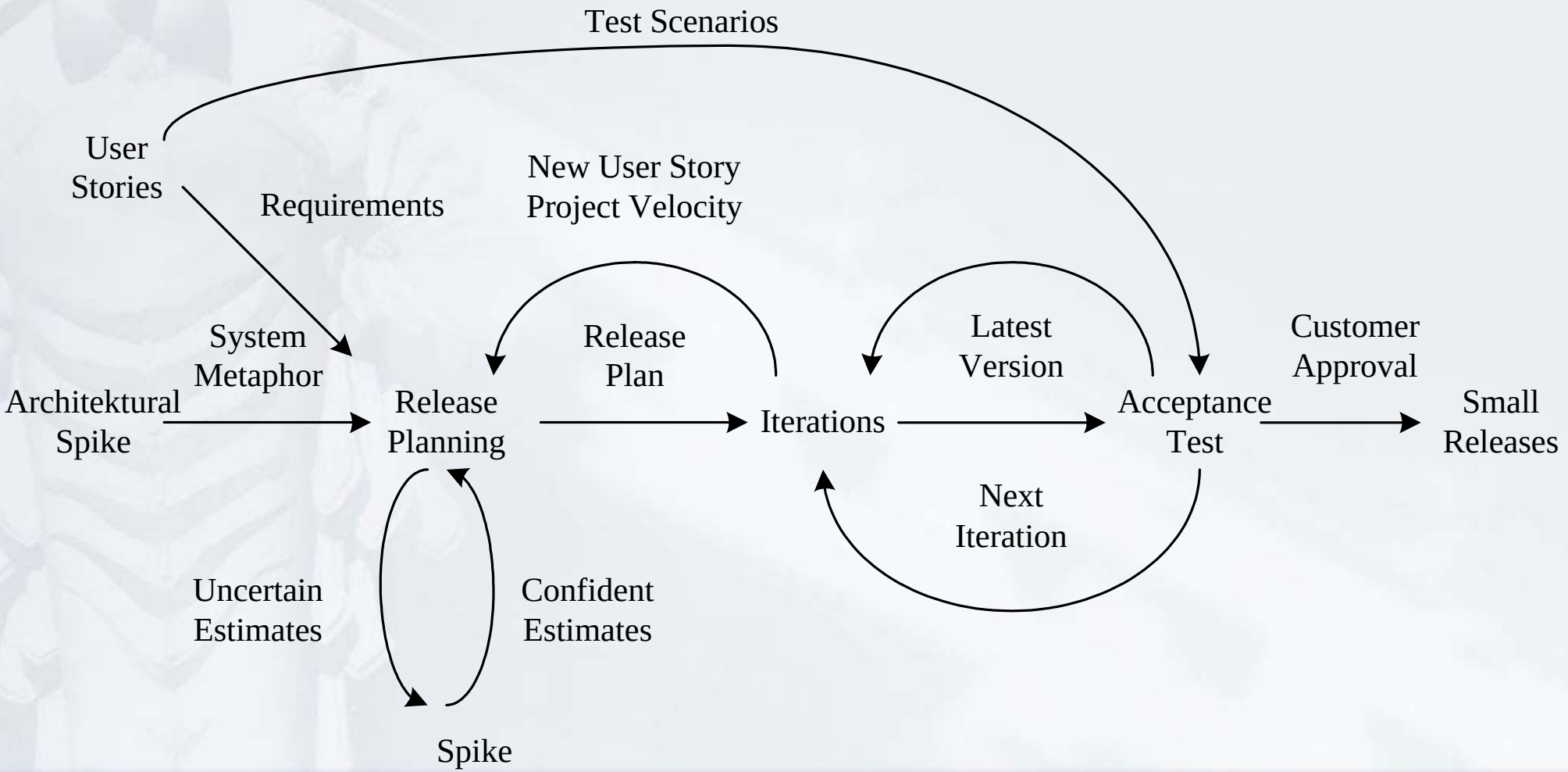
- Gut anwendbar für große Projekte.
- Hoher Dokumentationsaufwand

eXtreme Programming (XP)

- Gehört zur Familie der agilen Softwareentwicklungsprozesse
- Iterativer Softwareprozess mit sehr kurzen Iterationen und engem Kundenkontakt (typischerweise wöchentliche Iterationen)
  - **Coding** mit laufender Systemintegration, Tests und Reviews. Laufendes „refactoring“ (entfernen überflüssiger Code-Fragmente)
  - **Design**: Einfache Aufgaben sollen nur 1x durchgeführt werden.
  - **Documentation**: innerhalb des Source Codes; “keep it simple“.
  - **Test**: Unit Tests (Testfalldefinition findet vor der Codierungsphase statt).
  - **Integration**: findet sehr häufig (mehrmals pro Tag) statt; Versionskontrolle durch “Einchecken” von Source-Code Teilen in eine zentrale “Coding Base” (einschließlich Software Tests).
- Daher sehr flexibel für Änderungen der Anforderungen.

Konzept von eXtreme Programming

- Iterativer, evolutionärer und agiler Software Process.



Vor- und Nachteile von XP

Vorteile

- Flexible Reaktion auf Änderung der Anforderungen.
- Kundenorientiert durch häufige direkte Interaktion mit dem Kunden.
- Minimum an Dokumentation (“so viel wie unbedingt notwendig”)
- Prototypen sind innerhalb sehr kurzer Zeit verfügbar.

Nachteile

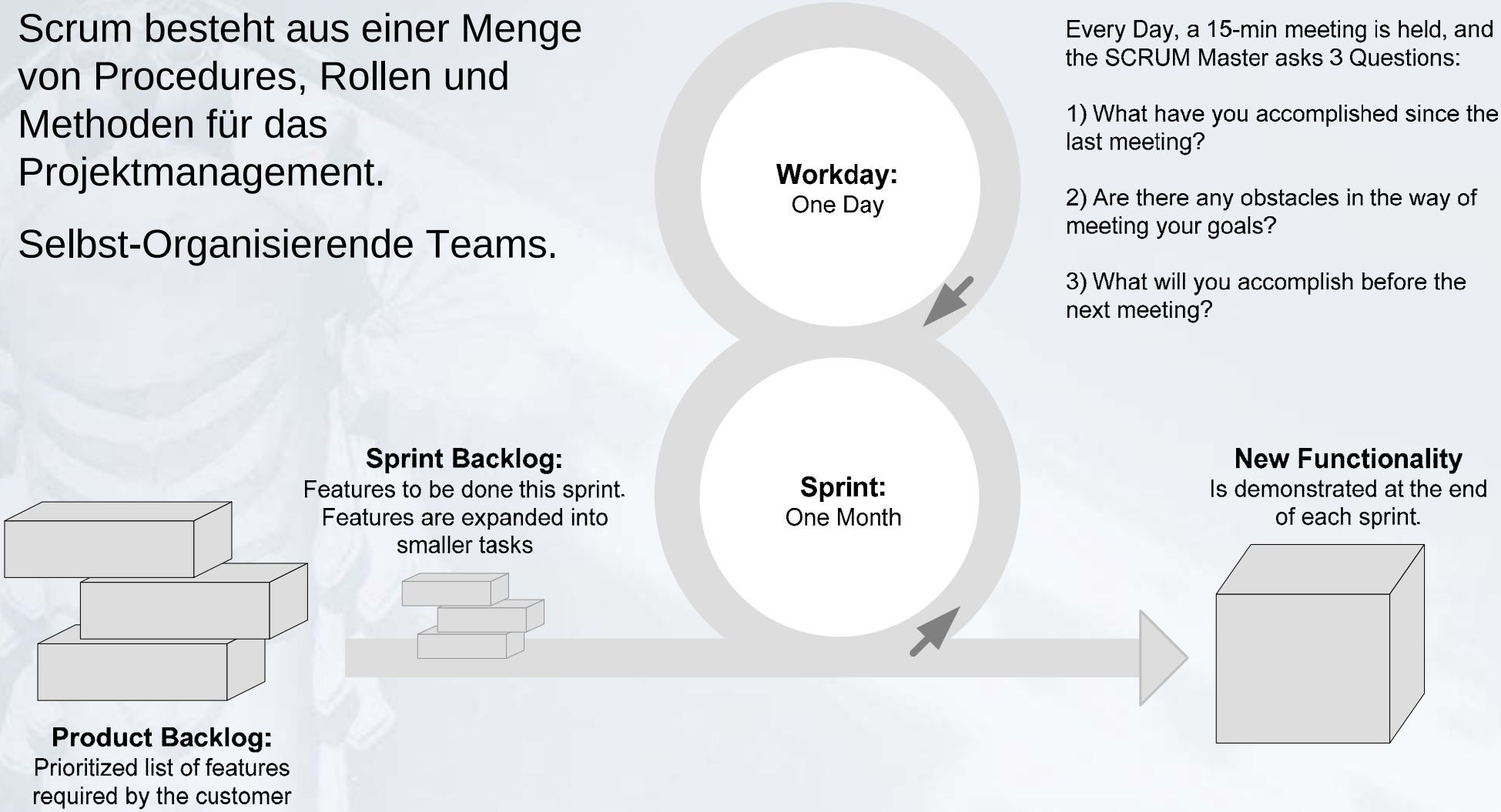
- Zusätzlicher Kommunikationsaufwand und “starkes” Projektmanagement erforderlich
- Anwendbar in Teams mit bis zu 12 Personen.
- Wiederverwendung von Komponenten kaum möglich.
- Wenig bzw. Keine formale Verifikation vorgesehen.

Anwendungsbereich:

- Projekte mit unklaren und sich schnell ändernden Anforderungen.
- Kleine Entwicklungsteams.
- Zeitkritische Projekte.

SCRUM

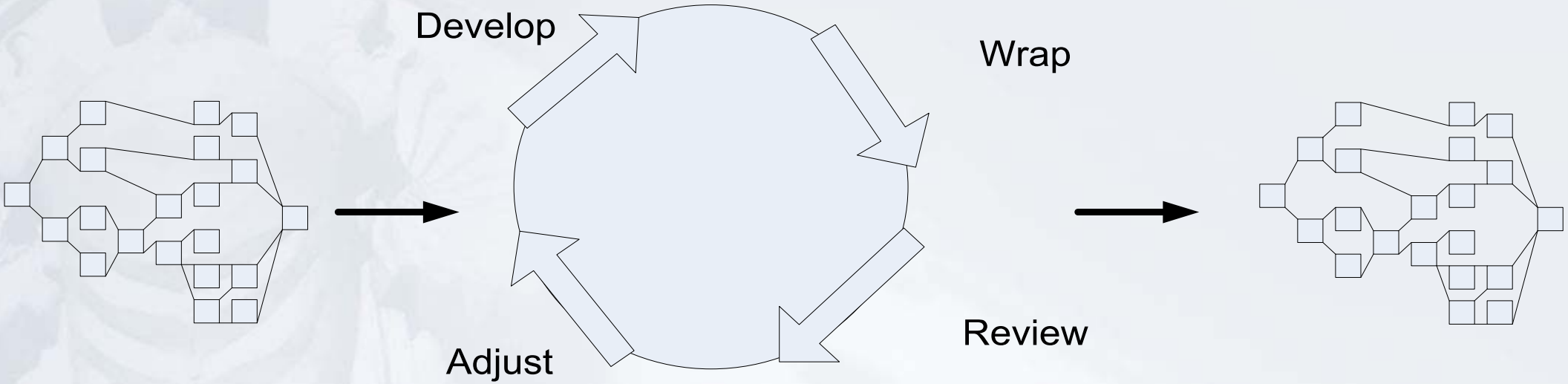
- Scrum besteht aus einer Menge von Procedures, Rollen und Methoden für das Projektmanagement.
- Selbst-Organisierende Teams.



SCRUM – Begriffsbildung

- **Backlog**:
  beinhaltet alle Arbeitspakete, die in “nächster” Zeit umgesetzt werden müssen (sowohl klar definierte als auch vage Anforderungen).
- In einem **Sprint** werden im Zeitraum von max. 30 Tagen definierte Produkte (Sprint Backlog) erstellt. In diesem Zeitraum sind nur geringe / keine Änderungen der Anforderungen möglich.
- **Sprint Backlog**:
  beinhaltet Arbeitspakete für einen Sprint (inkrementelle Produkterzeugung). Nach dem Start des Sprints sind keine wesentlichen Änderungen möglich.
- Ein **Scrum** ist eine tägliche Projektbesprechung um etwaige Fragen / Missverständnisse zu klären; Definition des täglichen Arbeitsauftrags.
- **Scrum Meeting Regeln**: Vorgehensweise für eine effiziente Projektbesprechung.
- **Scrum Team**: Funktionsübergreifendes Team, zuständig für den Sprint Backlog.

SCRUM – Phasen



Planning & System Architecture

Sprints

Closure

Zusammenfassung

- Software Prozesse sind zur Erstellung qualitativ hochwertiger Produkte notwendig.
- Je nach Projekttyp, -größe, -risiko muss der passende Softwareprozess ausgewählt und gegebenenfalls angepasst werden.
- Customizing und Tailoring ermöglichen die Anpassung eines Software Prozesses auf individuelle Gegebenheiten eines Unternehmens bzw. eines Projektes.

- Das V-Modell XT ist ein verpflichtendes flexibles Prozessmodell für öffentliche IT-Projekte in Deutschland. Durch den modularen Aufbau ist es auch für die Anwendung in KMUs geeignet.
- Der Rational Unified Process (RUP) ist ein phasenorientiertes Prozessmodell, das in im Industriellen Einsatz für große Projekte häufig eingesetzt wird.
- Agile Entwicklungsprozesse wie XP und SCRUM sind gut geeignet, um innerhalb kleiner Entwicklungsteams schnell auf geänderte Anforderungen reagieren zu können. Beide Modelle zeichnen sich durch intensiven Kundenkontakt aus.

References

- SWEBOK: Guide to the Software Engineering Body of Knowledge, http://www.swebok.org, 2005.
- Boehm B.: Software Risk Management: Principles and Practices, IEEE Software 8(1), pp. 32-41, 1991.
- Duncan William R.: A Guide to the Project Management Body of Knowledge, 1996.
- Kruchten P.: The Rational Unified Process, Addison-Wesley, 1999.
- Schach, S.: Object-Oriented and Classical Software Engineering, WCB/McGraw-Hill, 2002. http://www.mhhe.com/engcs/compsci/schach5/
- Schwaber Ken: SCRUM Development Process, 1995 http://www.controlchaos.com/old-site/scrumwp.htm
- SCRUM, www.controlchaos.com, February 2006.
- Wells, D.: When should Extreme Programming be Used, http://www.extremeprogramming.org/when.html
- [VMXT] http://www.v-model-xt.de.

Vielen Dank für die Aufmerksamkeit !

Kontakt: Dipl.-Ing. Dietmar Winkler, Michael Pernkopf

Technische Universität Wien
Institut für Softwaretechnik und Interaktive Systeme
Favoritenstr. 11/188, A-1040 Vienna, Austria

dietmar.winkler@qse.ifs.tuwien.ac.at
http://qse.ifs.tuwien.ac.at