14. Ablauforganisation mit Vorgehensmodellen

Prof. Dr. rer. nat. Uwe Aßmann
Lehrstuhl Softwaretechnologie
Fakultät Informatik
Technische Universität Dresden

http://st.inf.tu-dresden.de/teaching/swm
2016-1.3, 12/05/16

1. Phasenmodelle
2. Vorgehensmodelle
    1. RUP
    2. V-Modell-XT
3. Leichtgewichtige Vorgehensmodelle
    1. XP
    2. SCRUM
    3. EOS

DRESDEN concept
Exzellenz aus Wissenschaft und Kultur

Literatur

- www.v-modell-xt.de Wichtige Adresse!
- http://ftp.tu-clausthal.de/pub/institute/informatik/v-modell-xt/Releases/1.4/V-Modell-XT-Gesamt.pdf
- ftp.tu-clausthal.de/pub/institute/informatik/v-modell-xt/.../2.0/V-Modell-XT-Gesamt.pdf
- [Pichler] Roman Pichler. SCRUM - Agiles Projektmanagement erfolgreich einsetzen. dpunkt-Verlag.
- [Lippert] Lippert, M., Roock, S., Wolf, H.: Software entwickeln mit eXtreme Programming – Erfahrungen aus der Praxis; dpunkt.verlag 2002
- [ProjFachmann] Autorenkollektiv: Projektmanagement Fachmann Band 1 und 2; RKW-Verlag (5.Auflage) 1999
- [Pomberger] Pomberger, G., Pree, W.: Software Engineering - Architektur-Design und Prozessorientierung; Carl Hanser Verlag (3. Aufl.), München 2004
- [Zuser] Zuser, W. u.a.: Software-Engineering mit UML und dem Unified Process; Pearson Studium 2004
- [Dröschel] Dröschel, W., Heuser, W., Midderhoff, R.: Inkrementelle und objektorientierte Vorgehensweisen mit dem V-Modell 97; Oldenbourg-Verlag 1998
- [Hruschka] Hruschka, P., Rupp, Ch.: Agile Softwareentwicklung für Embedded Real-Time Systems mit der UML; Hanser Verlag 2002
- Ove Armbrust, Jan Ebell, Ulrike Hammerschall, Jürgen Münch, Daniela Thoma. Prozesseinführung und -reifung in der Praxis: Erfolgsfaktoren und Erfahrungen. 14. Workshop der Fachgruppe WI-VM der Gesellschaft für Informatik. IESE-Report Nr. 034.07/D Version 1.0 31. Januar 2007
  - vmxt.fraunhofer.de/plaintext/downloads/erfolgsfaktoren034.07.pdf

Ablauforganisation

Unter **Ablauforganisation** versteht man die Aneinanderreihung systeminterner Arbeitsabläufe, die räumlich und zeitlich so angeordnet sind, dass das System funktionsfähig ist und zielgerichtet arbeitet.

- **Arbeitsabläufe (workflows)** organisieren den Ablauf von Arbeitsvorgängen (Aktivitäten)
  - ihre **Abhängigkeiten** (Steuer- und Datenfluss) zu beschreiben
  - Verantwortlichkeiten innerhalb der Vorgänge zu formalisieren (Rollen)
  - zu rationalisieren, optimieren und vereinfachen
- **Prozessmodelle** beschreiben schematisch die Methodik des Vorgehens
  - sie geben also Arbeitsabläufe schematisch vor
  - sequentiell, parallel oder iterativ
- Prozessmodelle müssen zu Arbeitsabläufen ausgeprägt (instanziiert) werden


A380
A380
Lufthansa
Technik

Prozessmodelle als Mittel der Ablauforganisation

Es gibt viele verschiedene Prozessmodelle, die ihre Berechtigung haben, abhängig u.a. vom Produkt (Inhalt) und der Projektgröße

- **Phasenmodelle**: Phasen sind erkennbar, durch Meilensteine getrennt
  - Spiralmodell
  - V-Modell
  - INeCT
- **Vorgehensmodelle**: Ohne Phasen, aber aus Schablonen/Mustern zusammengesetzt
  - V-Modell des Bundes, Rational Unified Process (RUP)
  - Oft zyklisch: Rückkopplung ist wichtig!
    - Modelle zur inkrementellen SWE
    - Evolutionäre Modelle
    - Prototypisches Vorgehen
- **Leichtgewichtige, agile Vorgehensmodelle:** XP, SCRUM, EOS

14.1 Phasenmodelle

Phasenorganisation

Eine **Projektphase** ist ein zeitlicher Abschnitt in einem Projektablauf, der sachlich getrennt von einem anderen abläuft.
Die Projektphase wird durch eine Abnahme an einem **Meilenstein** offiziell abgeschlossen.

- Die *Phasenorganisation* definiert eine Kette logisch aufeinander aufbauender Meilensteine (als Übergänge zwischen den Phasen)
  - Die Phasen enthalten die Summe der Aktivitäten zwischen den Meilensteinen
- Die Phasenorganisation reduziert das technische, wirtschaftliche und terminliche Risikos durch:
  - schrittweises Vorgehen
  - Vorgabe und einfache Überwachung von Zwischenergebnissen (Meilensteinen und Entscheidungspunkte)
  - Transparenz über den Projektstand (Projektkontrolle, -regelung)
- → Phasengliederung immer einsetzen

Vorteile der Projektphasen

- Phasenweises Vorgehen ist ein *einfaches* Planungs- und Controlling-Instrument hilft,
  - den **Überblick** zu behalten und sich nicht im Detail zu verlieren
  - zwingt die Mitarbeiter zur **periodischen Stellungnahme** durch Meilensteine
  - das Risikos einer Fehlentwicklung zu verringern durch **bessere Überwachung**
  - hilft bei der Herstellung einer **fortlaufenden Dokumentation**
- **Entscheidungsfreiheit** des Projektleiters bleibt gewahrt durch Beeinflussung der Entwicklung an den vordefinierten Ergebniszeitpunkten

[Jenny, S. 68]

Meilensteine in der Planung von Phasenorganisation

- **Meilensteine (MS, Barrieren, Transitionen)** sind sind Eckpunkte der Planung und Durchführung einer Phasenorganisation
  - an ihnen wird der Stand des Projekts evaluiert
    - Basis der Bewertung sind: definierte Arbeitsprodukte an einem Termin
    - Formale Grundlage: Petrinetze, Workflow-Sprachen
- Ein Meilenstein, der für eine Gruppe von Aktivitäten gilt, und an dem der Steuerfluss verzweigen kann, heisst **Entscheidungspunkt (EP)**
    - dienen der Entscheidung über weiteren Projektablauf
      - Bestimmung von Korrekturmaßnahmen
      - Freigabe des nächsten Abschnitts
      - Entscheid über den Abbruch des Projekts
- Meilenstein-Spezifikationen müssen von hoher Qualität sein
  - **übergebbar** an andere Personen (z. B. in die Produktbibliothek, dem AG, …)
    - **klar:** Definition mit SMART und CCC
    - **überprüfbar** (als Voraussetzung für nächste Phase)
  - Meilensteinüberprüfung im Controlling durch Meilenstein-Trendanalyse

Bsp: Reihenfolge der Meilensteine im linearen Modell (Wasserfall-Modell)

MS 1: **Projektziele/ Pflichtenheft** ***Was soll erreicht werden?***
- **Anforderungskatalog**

MS 2: **Fachliches Konzept** ***Wie sieht die Lösung aus?***
- **Leistungsbeschreibung**

MS 3: **Technisches Konzept** ***Wie wird die Lösung technisch realisiert?***
- ⇨**Design-Spezifikation (Architektur)**

MS 4: **Realisierung** ***Erstellung der Komponenten***
- **Komponenten**

MS 5: **Integration** ***Arbeiten die Komponenten zusammen?***
- **System**

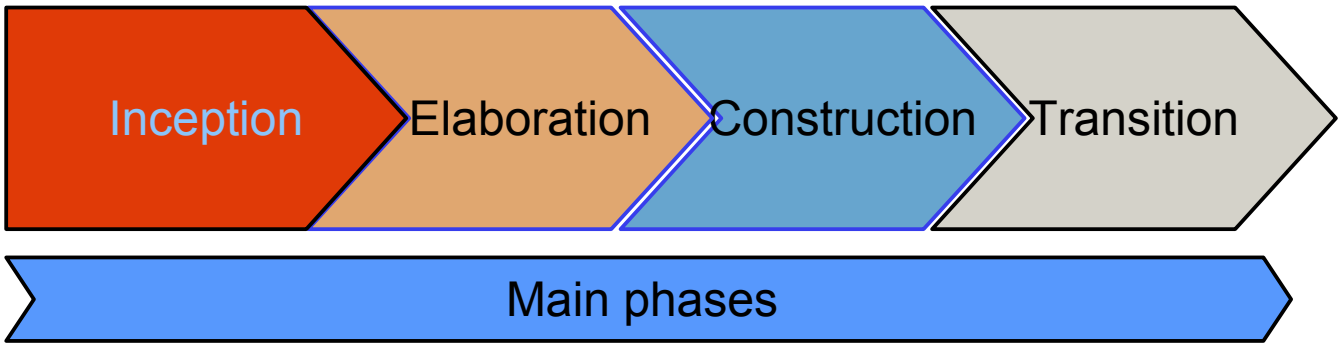
MS 6: **Getestetes System** ***Werden die Funktionen erfüllt?***
- **Produkt**

MS 7: **Eingeführtes System** ***Einführung und Anwendung des Systems***

Phasenmodell des Durchführungsprozesses INECT

Die Phasengliederung INECT des schwergewichtigen Vorgehensmodells RUP ist allgemein für Planung und Controlling verwendbar:

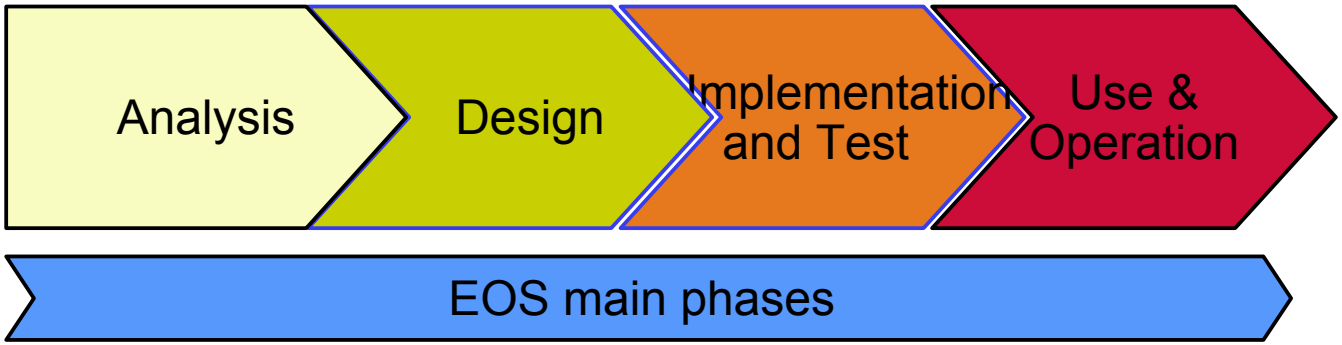
- **Inception:** Festlegung aller Projektbedingungen und Einrichtung einer Umgebung zur Durchführung aller folgenden Arbeitsschritte
- **Elaboration:** Durchführung der Analyse, Festlegung aller Anwendungsfälle und Entwurf der Architektur
- **Construction:** Fortführung des Entwurfs sowie Implementierung der Architektur und Durchführung des Tests
- **Transition**: Übergangsphase in der das Softwareprodukt beim Kunden auf der Zielplattform installiert und integriert wird; Nachstudien; Prozessverbesserung



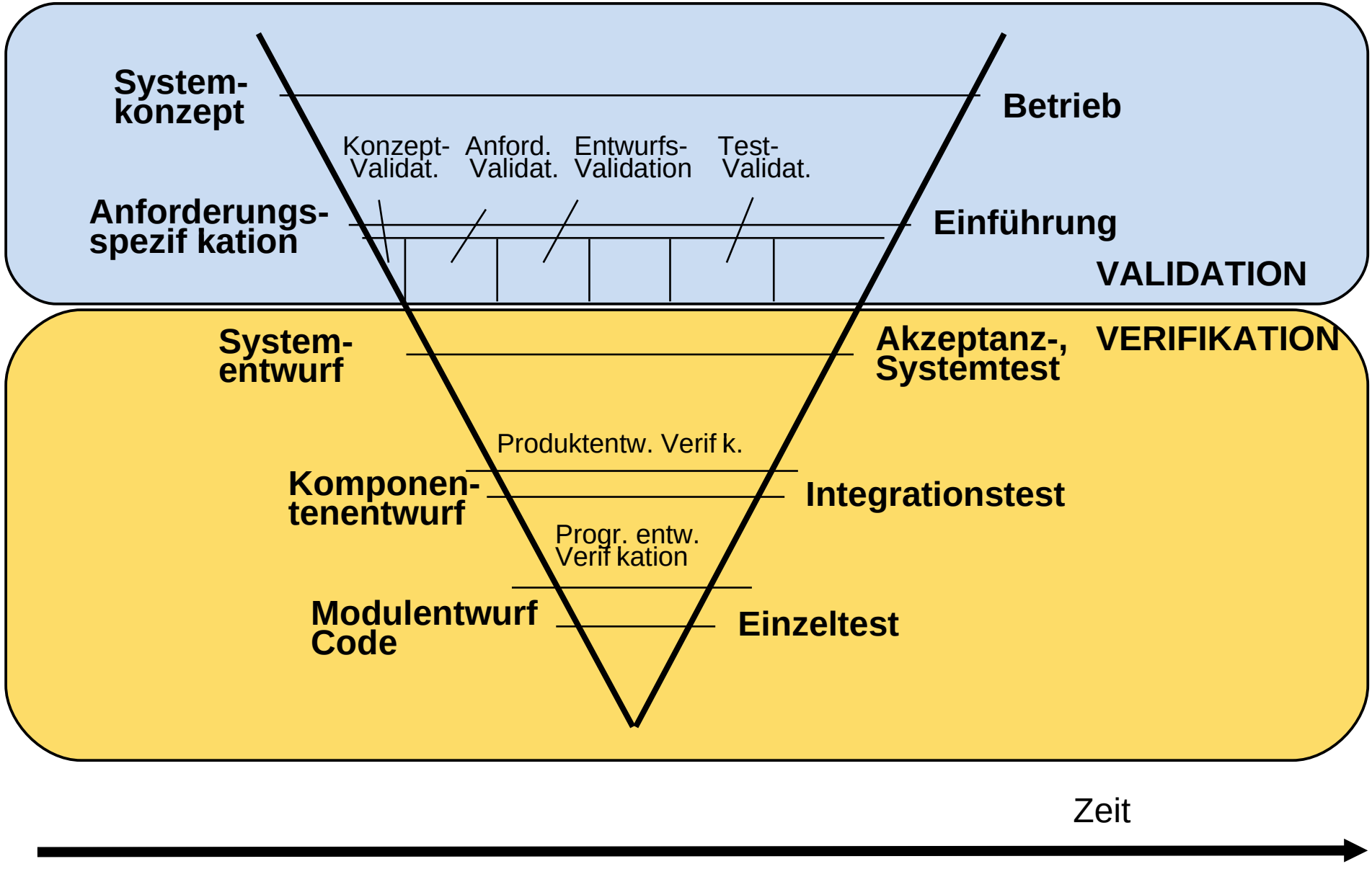
Phasenmodell des Durchführungsprozesses EOS

Die Phasengliederung des schwergewichtigen Vorgehensmodells EOS ist allgemein für Planung und Controlling verwendbar:

- **Analysis:**
- **Design:**
- **Implementation and Test:**
- **Use & Operation**:

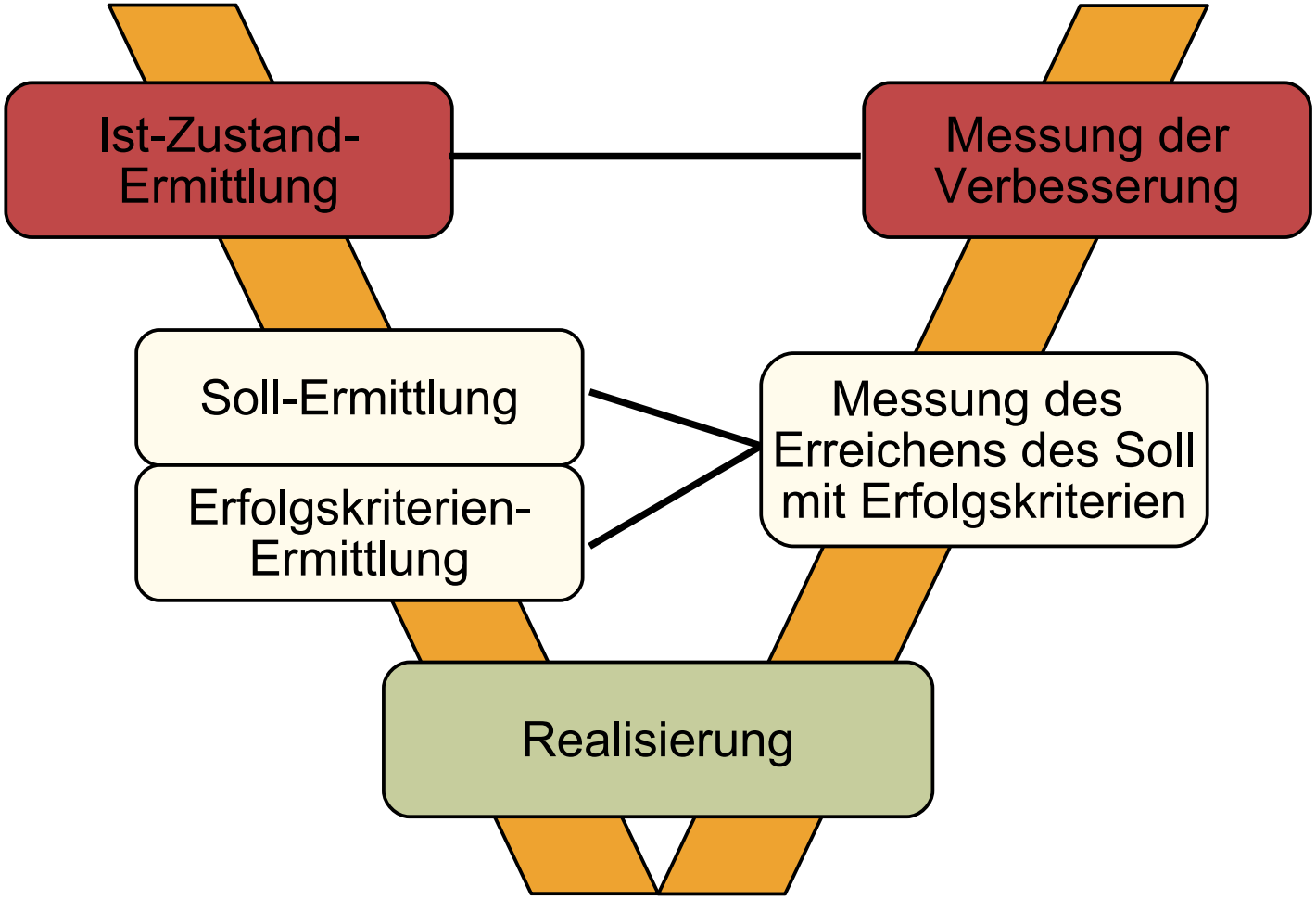


Das V-Modell und seine Meilensteine (nach B. Boehm)



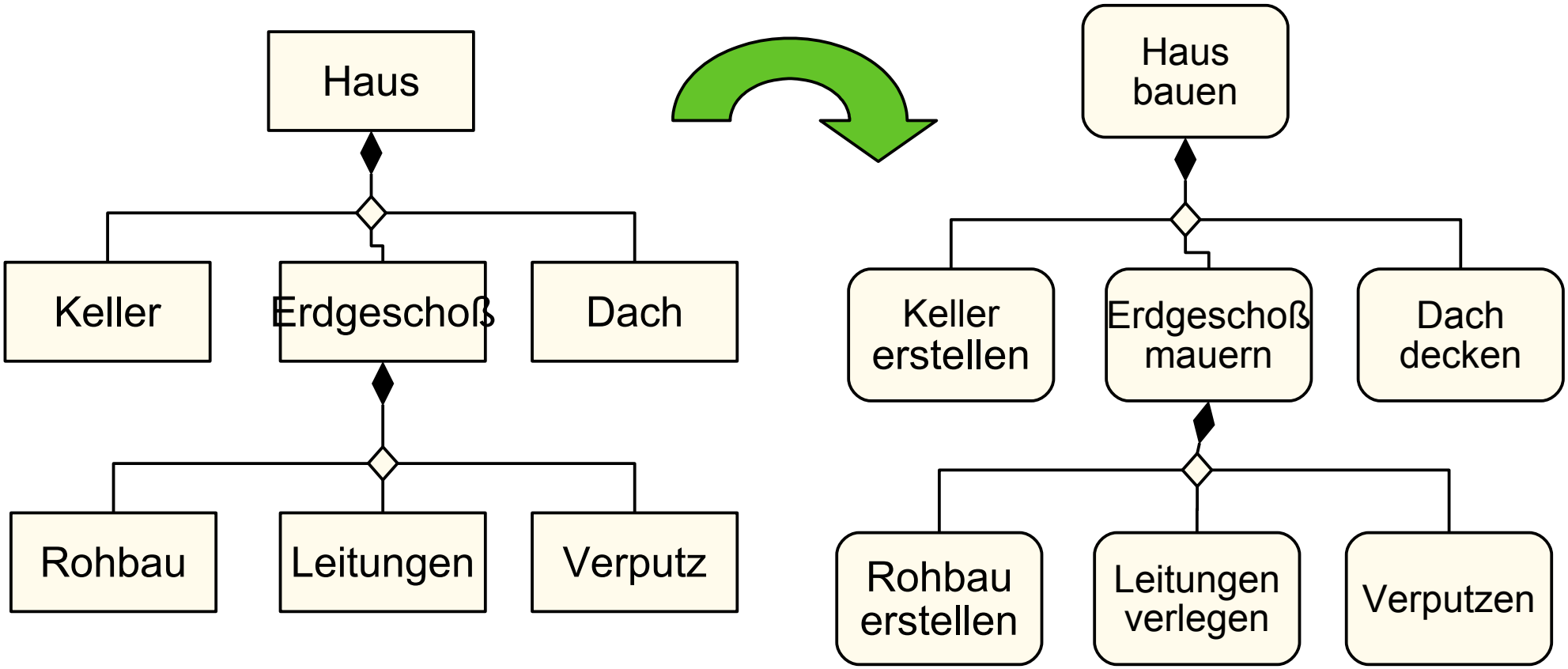
Das Ist-Soll-Modell ist ein generisches Phasenmodell zum Problemlösen

- Ist – Soll - Erfolgskriterien



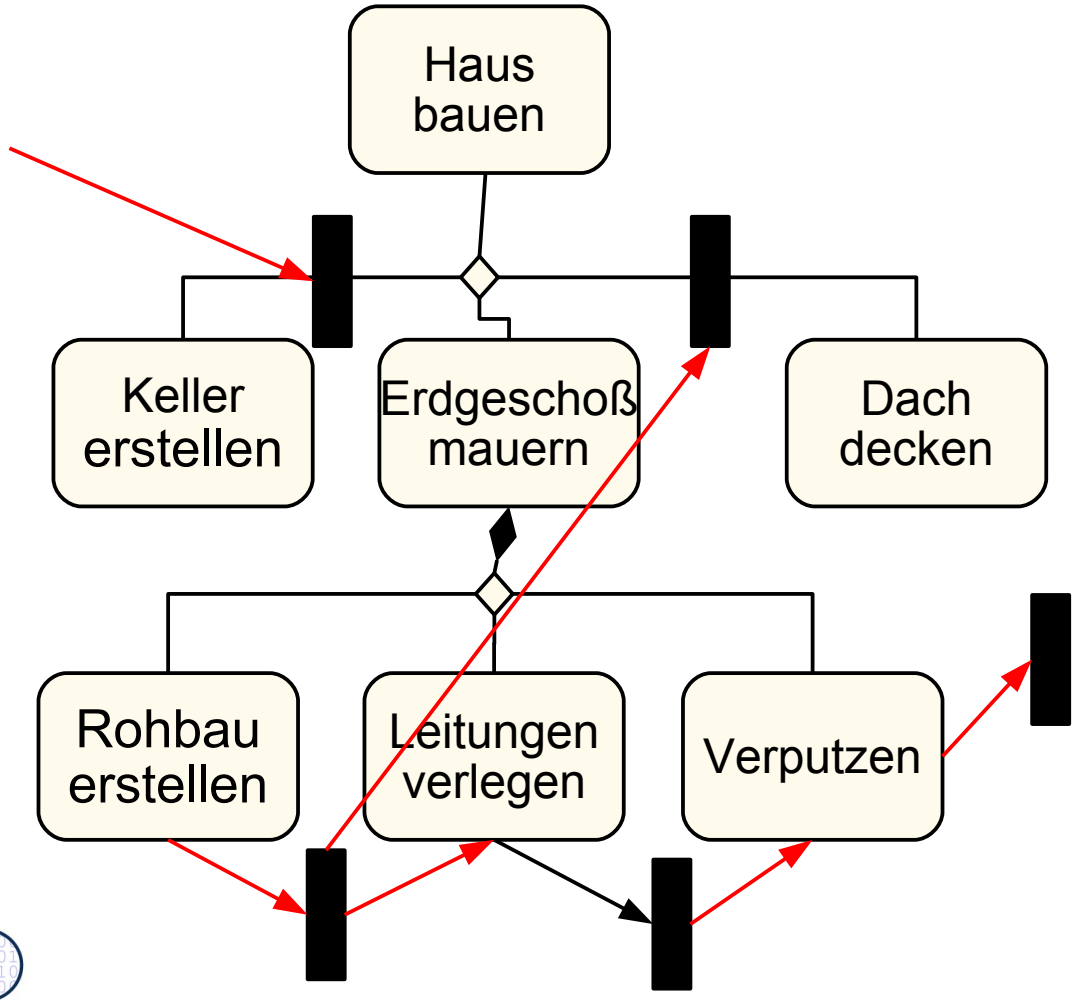
Hierarchische Phasenmodelle

- Ein **hierarchisches Phasenmodell** untergliedert jede Phase in Unterphasen.
- Hat man eine Produktstruktur (Product Breakdown Structure, PBS) aus Komponenten gefunden, lässt sich daraus sehr einfach ein homomorphes hierarchisches Phasenmodell angeben (**hierarchische Ablauforganisation**)



Meilensteine als Transitionen

- Aus einem hierarchischen Phasenmodell können Transitionen (Meilensteine) zwischen den Phasen angebracht werden, um zu einem Petrinetz zu kommen.



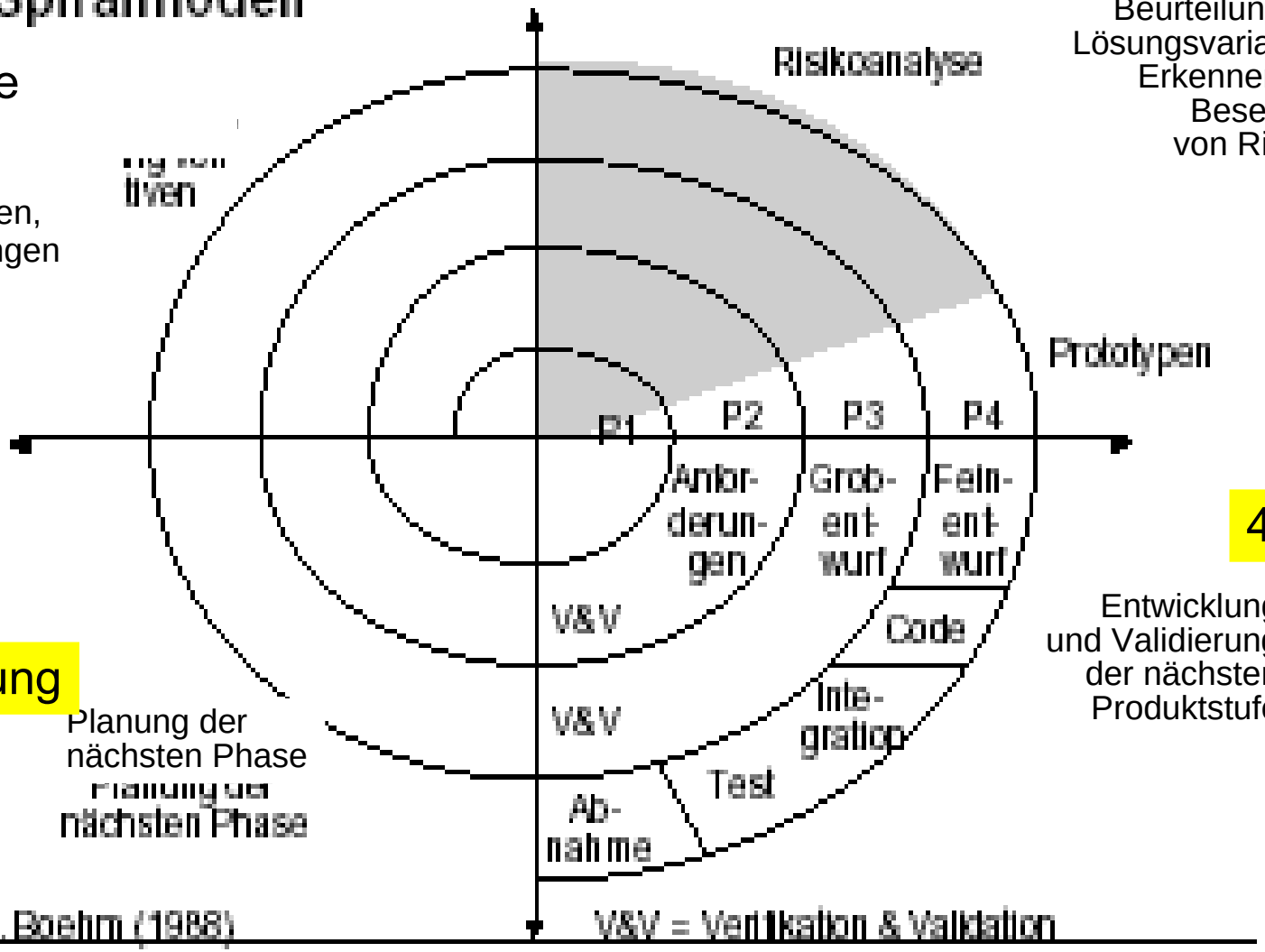
14.1.2 Iterative Phasenmodelle

Ein Phasenmodell mit Rückkopplung: Das Spiralmodell nach Boehm

2) Zielanalyse

3) Risiko- und Alternativenanalyse

Erarbeitung und Beurteilung von Lösungsvarianten, Erkennen und Beseitigen von Risiken

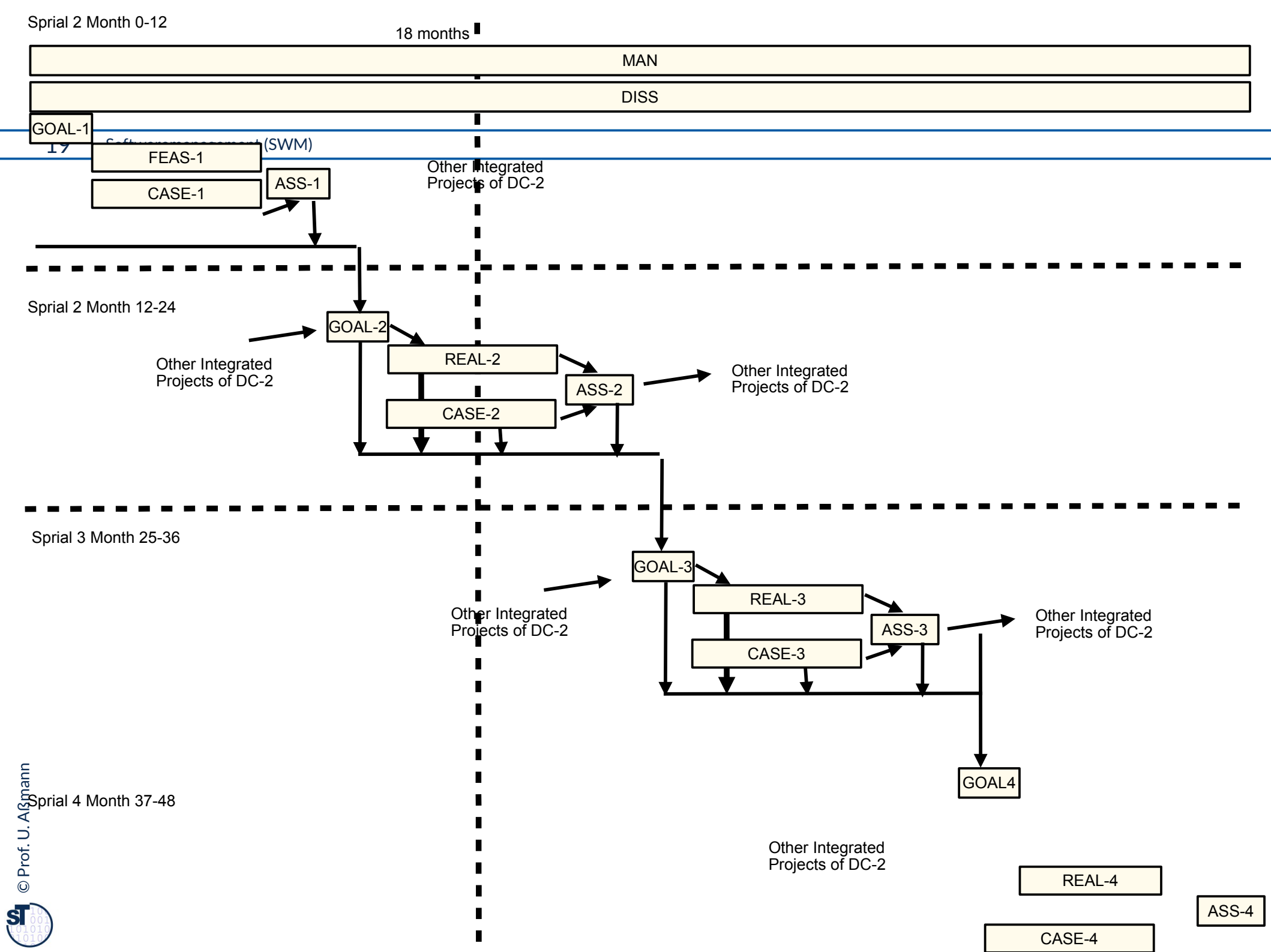


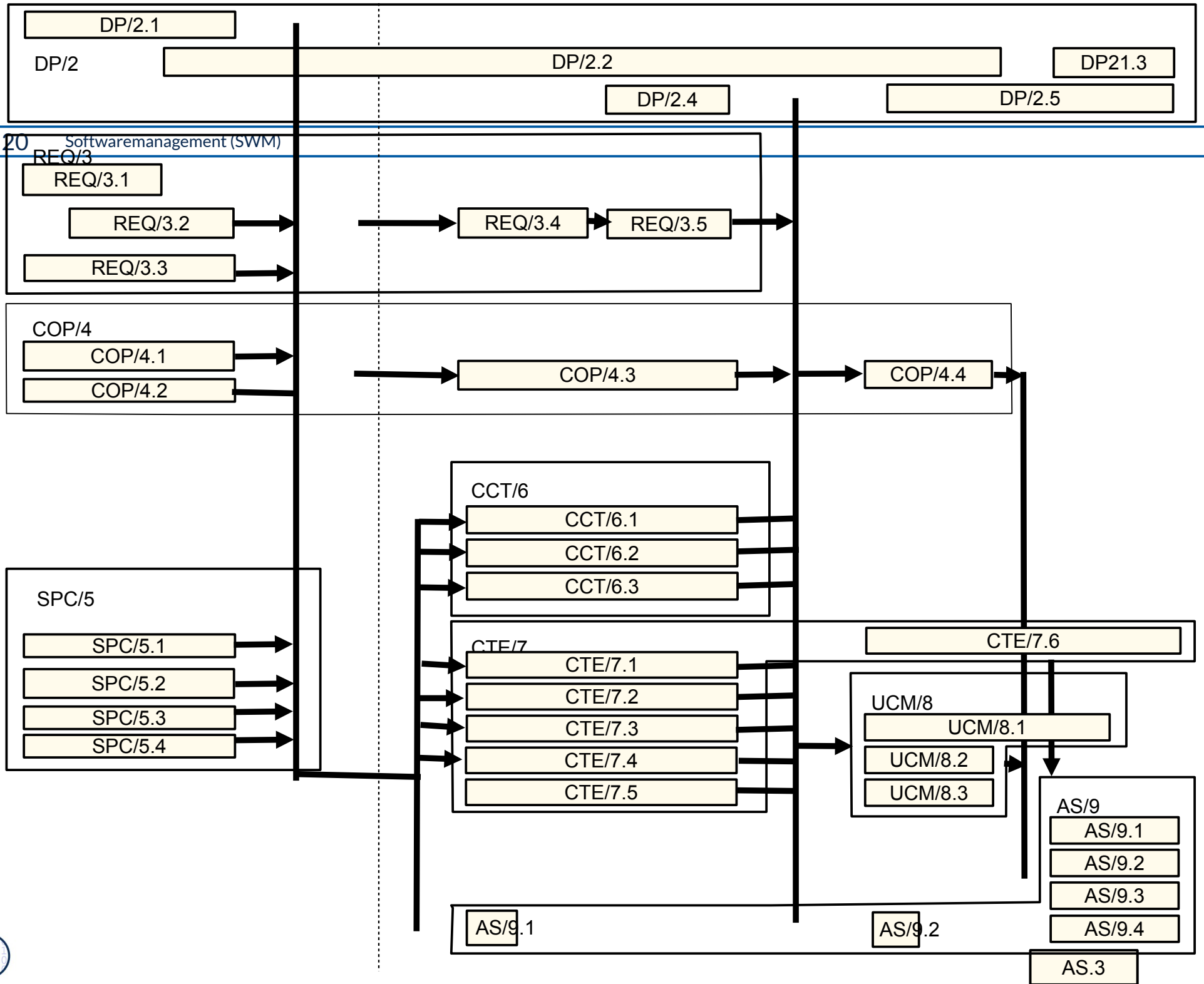
4) Entwicklung

Entwicklung und Validierung der nächsten Produktstufe

1) Planung

[4 Buhl, S.15]





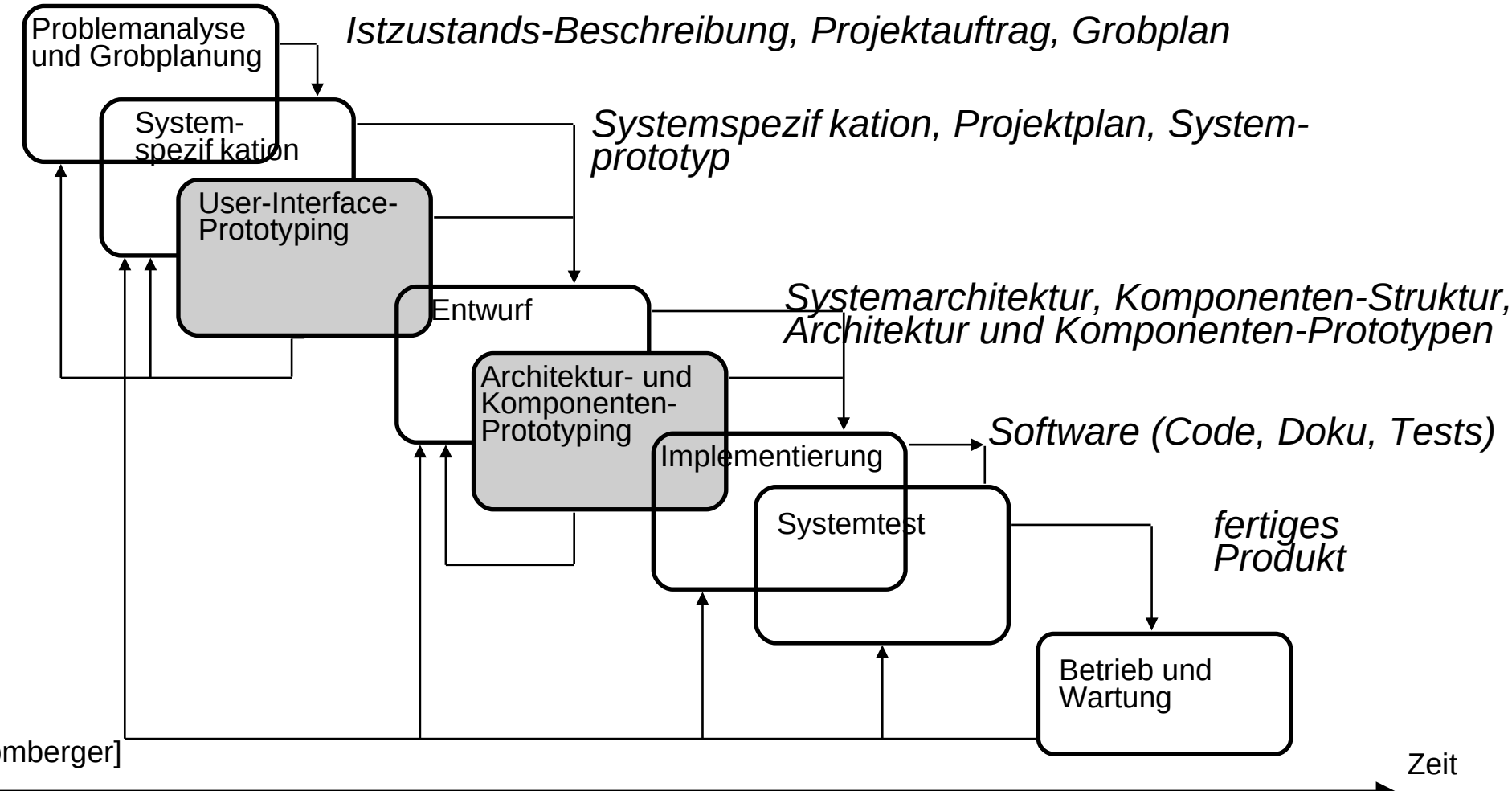
Merkmale des Spiralmodells

- **Merkmale**:
  - Minimierung des Risikos steht im Mittelpunkt
  - Jeder Spiralzyklus durchläuft dieselben Grundschritte
  - Ziele eines Zyklus werden aus Ergebnissen des vorherigen abgeleitet
  - Parallele Spiralzyklen für verschiedene Komponenten einer Anwendung
- **Vorteile**:
  - Hohe Flexibilität des Vorgehens durch rückgekoppelten Prozess (wie PDCA)
  - Integrierte Risikoanalyse
  - Qualitätssicherungsmaßnahmen gut integrierbar
  - Auch für Wartungsprojekte geeignet
- **Nachteile**:
  - Spiralen manchmal zu lang

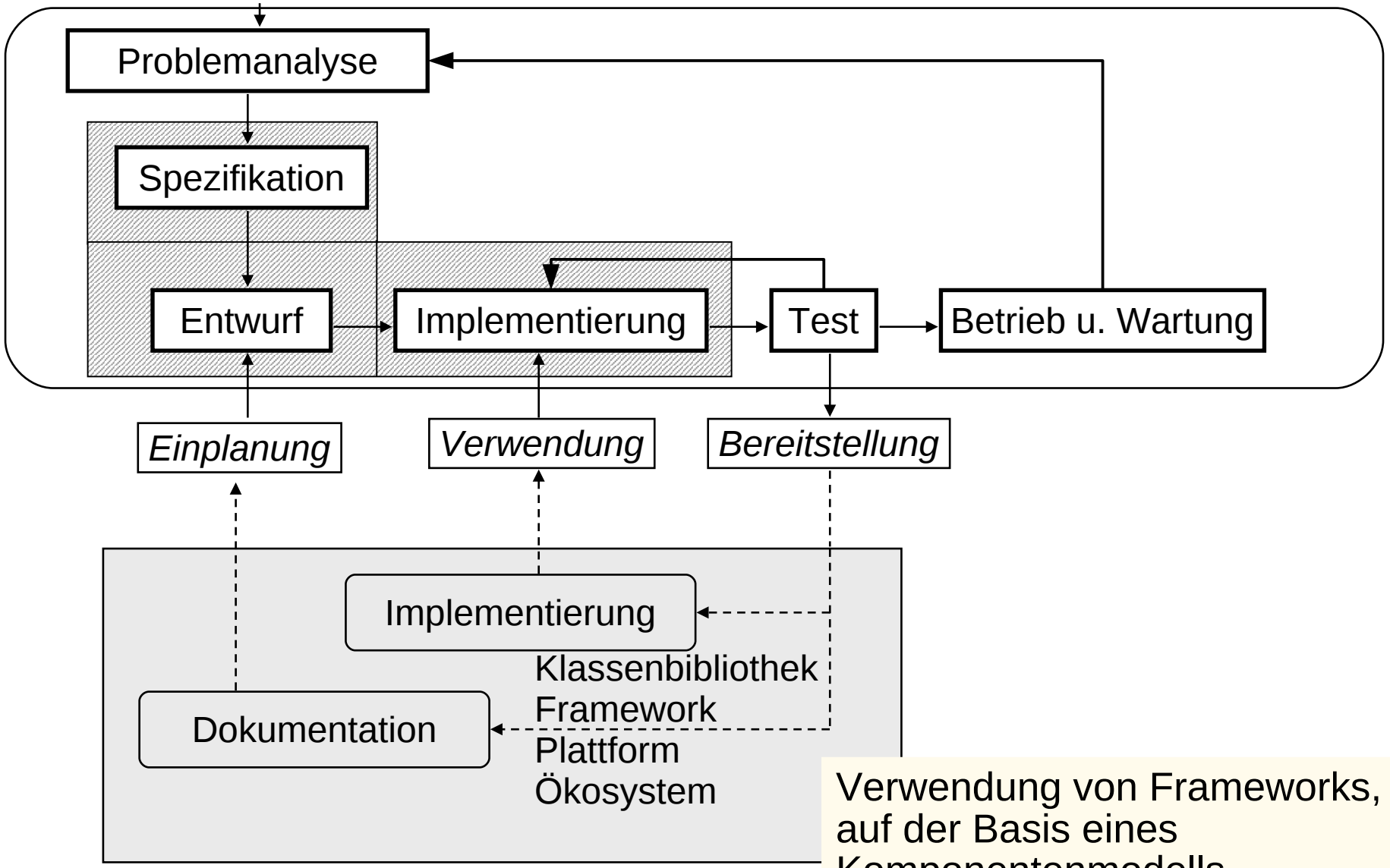
[Prof. Dr. S. Seibert; FH Darmstadt]

Prototyping-orientiertes Phasenmodell mit Rückkopplung

- Teile eines Produktes grob entwickeln, so, dass sie **ausführbar** und **demonstrierbar** werden
- **Minimal viable product (MVP)**: welches ist das wichtigste Feature?



Objektorientiertes Phasenmodell mit Rückkopplung und Wiederverwendung



Verwendung von Frameworks, auf der Basis eines Komponentenmodells und Kompositionssystems [CBSE]

nach [Pomberger]

Aufgabe: Planen Sie Ihr eigenes Haus

- Werten Sie folgende Checkliste zum Hausbau aus
  - http://www.planungschecklisten.de/Checklisten/NB_Vorgehensweise%20Hausbau.pdf
- Welche Phasen können Sie identifizieren? Warum sind Phasen wichtig?
- Erstellen Sie mit Microsoft Project oder einen freien Planungstool wie openproj für jede Phase ein Balkendiagramm
- http://www.bauen.de/ratgeber/hausbau/bauratgeber/planung.html

14.2 Vorgehensmodelle

Vorgehensmodellierung

Vorgehensmodelle definieren **Tätigkeiten** (**Aktivitäten**) und von ihnen erzeugte **Arbeitsergebnisse (Artefakte, Dokumente, Produkte)** sowie ihr komplettes Zusammenwirken einschließlich benötigter **Rollen**, **Produktzustände** und sämtlicher Ressourcen, die für den Projektablauf benötigt werden.

Ziele:

- Erbringung der vom Kunden gewünschten Leistung **planmäßig** in hervorragender Qualität
- **Modellierung des Arbeitsflusses (Workflow)** unter Einschluss der beteiligten Rollen zur Entwicklung begleitende Tätigkeiten, wie Qualitätsmanagement (QM,QS), Konfigurationsmanagement (KM), Projektmanagement (PM)
- **Modellierung der Elemente des Workflows** aus Aktivitäten, Artefakten (Produkten), Zuständen und Rollen und Ressourcen
- **Modellierung der inkrementellen Softwareentwicklung** durch stufenweises Vorgehen auf der Basis insgesamt gefestigter Anwenderforderungen
- organisationsspezifische **Anpassung** an spezielle Anwendungsfälle (**Tailorisierung**)

Einsatz von schwergewichtigen Vorgehensmodellen

Einsatz bei:

- **Festpreisprojekten**
  - Schätzung und Planung spielt dann eine große Rolle
  - Öffentl. Ausschreibungen
- Wenn Kunde nicht vor Ort involviert sein will oder kann
  - Kundeninvolvierung erlaubt agiles Vorgehen
- Wenn große Sicherheitsanforderungen herrschen
  - und der Code gegen Anforderungsspezifikationen verifiziert und zertifiziert werden muss
- Bei langlebiger Software

Nicht einsetzen bei:

- Wenn man schnell auf dem Markt sein muss
  - I.d.R. nicht bei Startups
- Wenn der Kunde nicht weiß, was er will (unklare oder sich wandelnde Anforderungen)

14.2.1 (Rational) Unified Process

- Ursprünglich von der Firma Rational, dann IBM

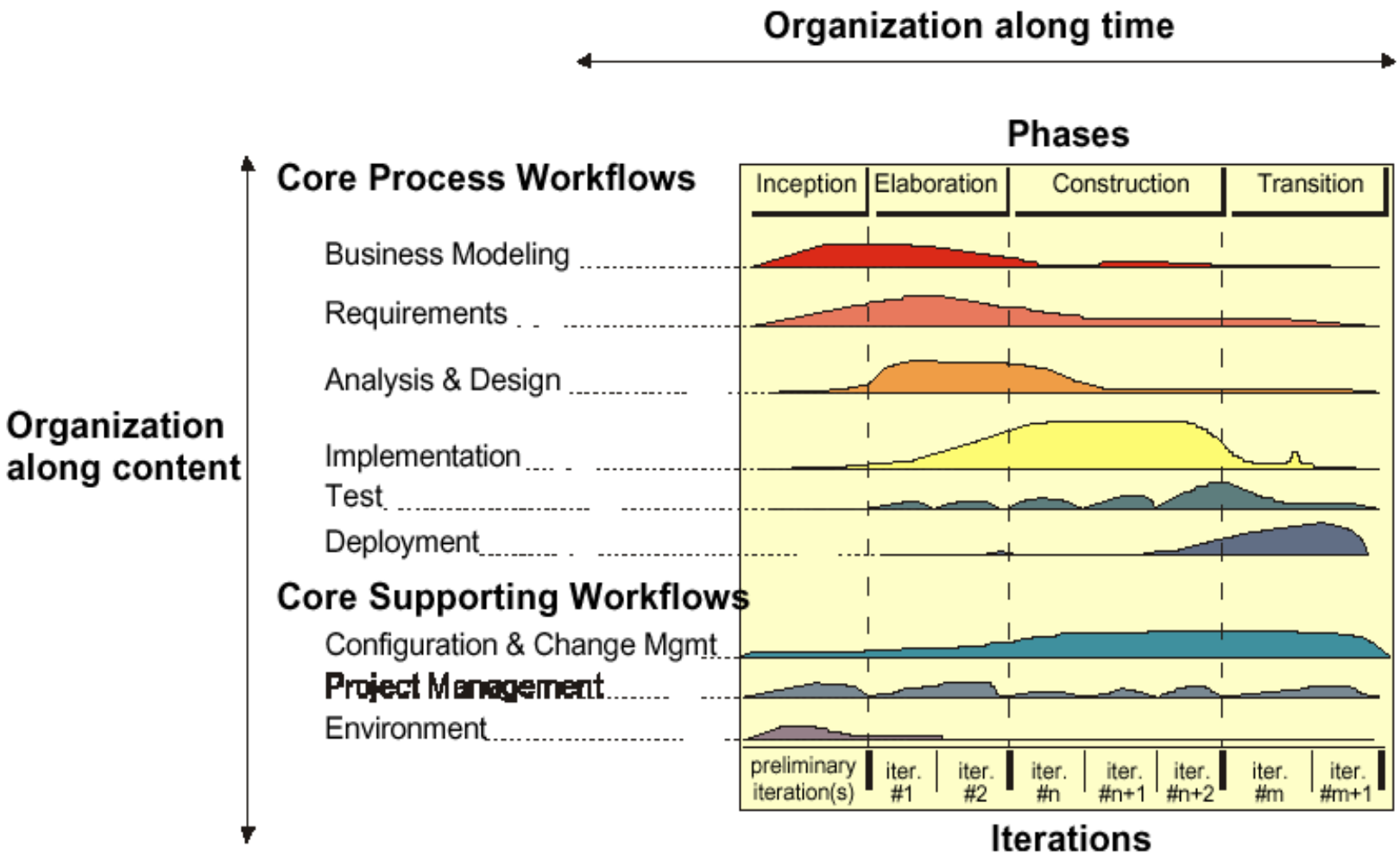
Eigenschaften des Rational Unified Process (RUP)

- Durch Anwendungsfälle gesteuert (use-case driven)
  - Aus ihnen werden alle Modelle bis hin zu den Testdaten zur Überprüfung der Anwendungsfälle entwickelt
- Iterativer und inkrementeller Prozess
  - Die Komplexität der Projekte erfordert nicht nur eine einmalige, sondern eine wiederholte Abfolge der Arbeitsschritte. Im Zuge der Iterationen werden so viel wie möglich Produkte parallel weiterentwickelt.
  - Das gesamte Projekt wächst somit inkrementell.
- Architekturzentriert
  - Die Architektur bildet die Grundlage für das gesamte System. Sie berücksichtigt alle Bedingungen für das Projekt einschließlich der nichtfunktionalen Anforderungen.
  - Mit der Architektur werden die grundlegende Form und alle Anwendungsfälle umgesetzt.
  - Vorhandene Architekturschablonen (Client-Server, Schichten...) können genutzt werden.

[Zuser, W.]

Rational Unified Process (1)

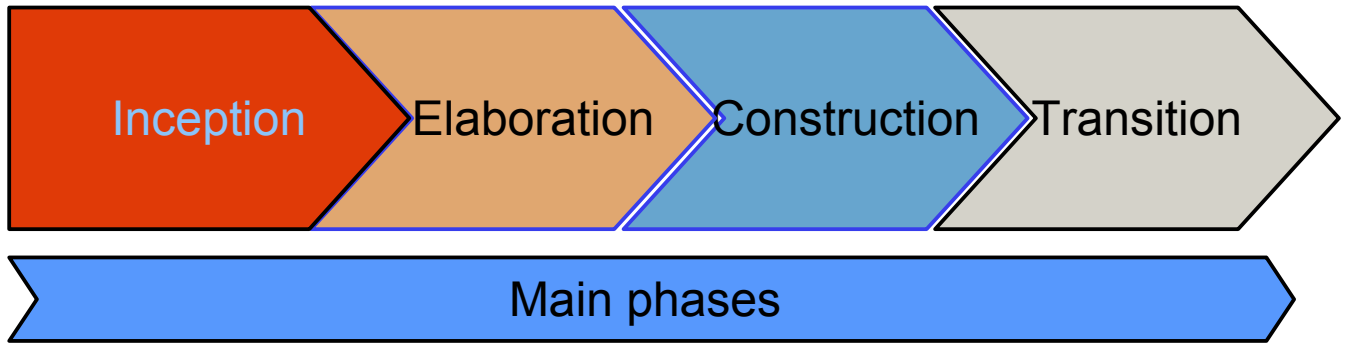
- Jeder Workflow, ob Kern- oder Supporting, wird in die Phasen des INECT eingeordnet.



Phasenmodell des Rational Unified Process (2)

Die Workflows sind über ein Phasenmodell mit 4 Haupt-Phasen gestreut:

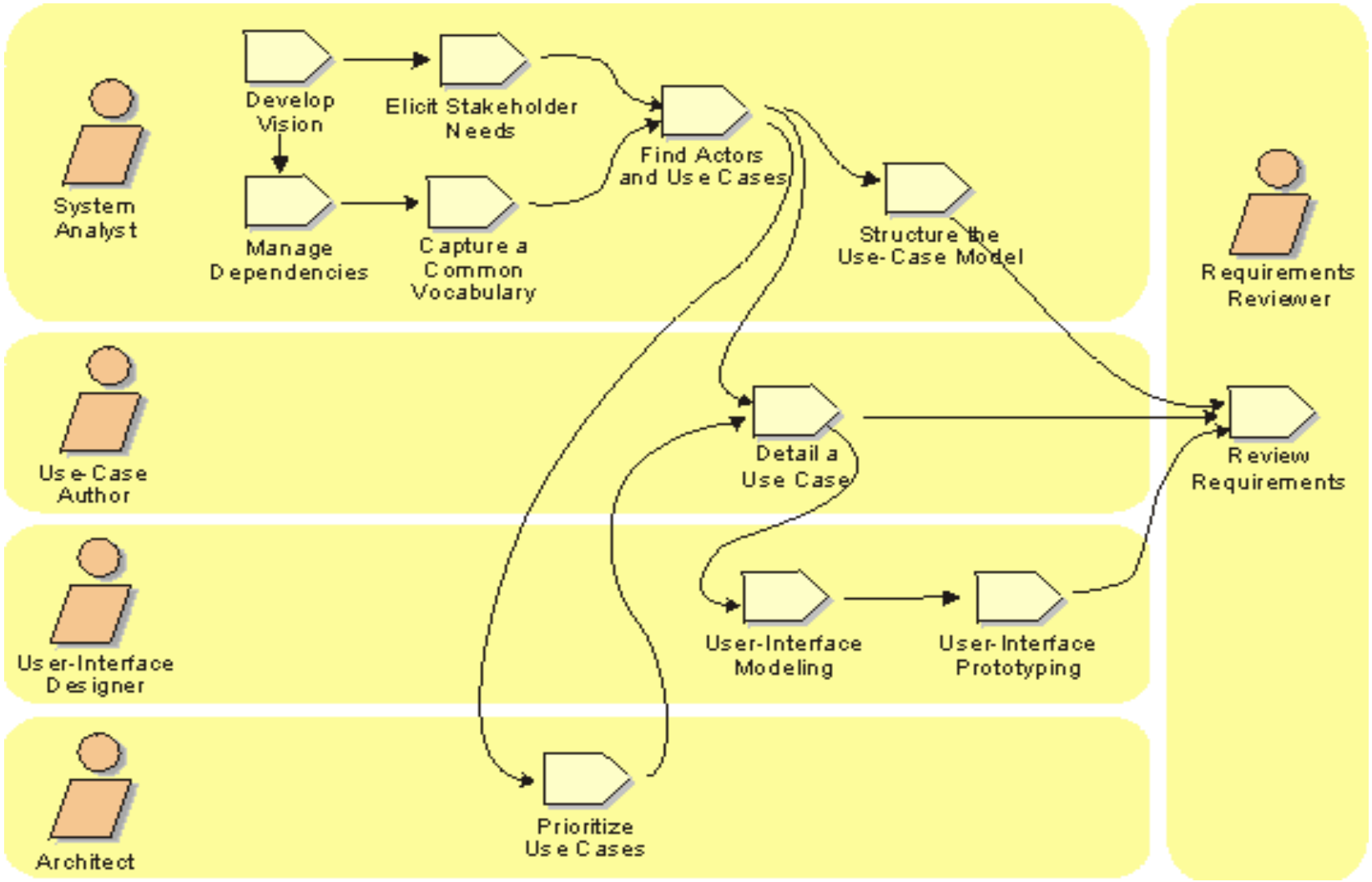
- **Inception:** Festlegung aller Projektbedingungen und Einrichtung einer Umgebung zur Durchführung aller folgenden Arbeitsschritte
- **Elaboration:** Durchführung der Analyse, Festlegung aller Anwendungsfälle und Entwurf der Architektur
- **Construction:** Fortführung des Entwurfs sowie Implementierung der Architektur und Durchführung des Tests
- **Transition**: Übergangsphase in der das Softwareprodukt beim Kunden auf der Zielplattform installiert und integriert wird.
- Die Phasen werden iteriert



Core und Supporting Workflows

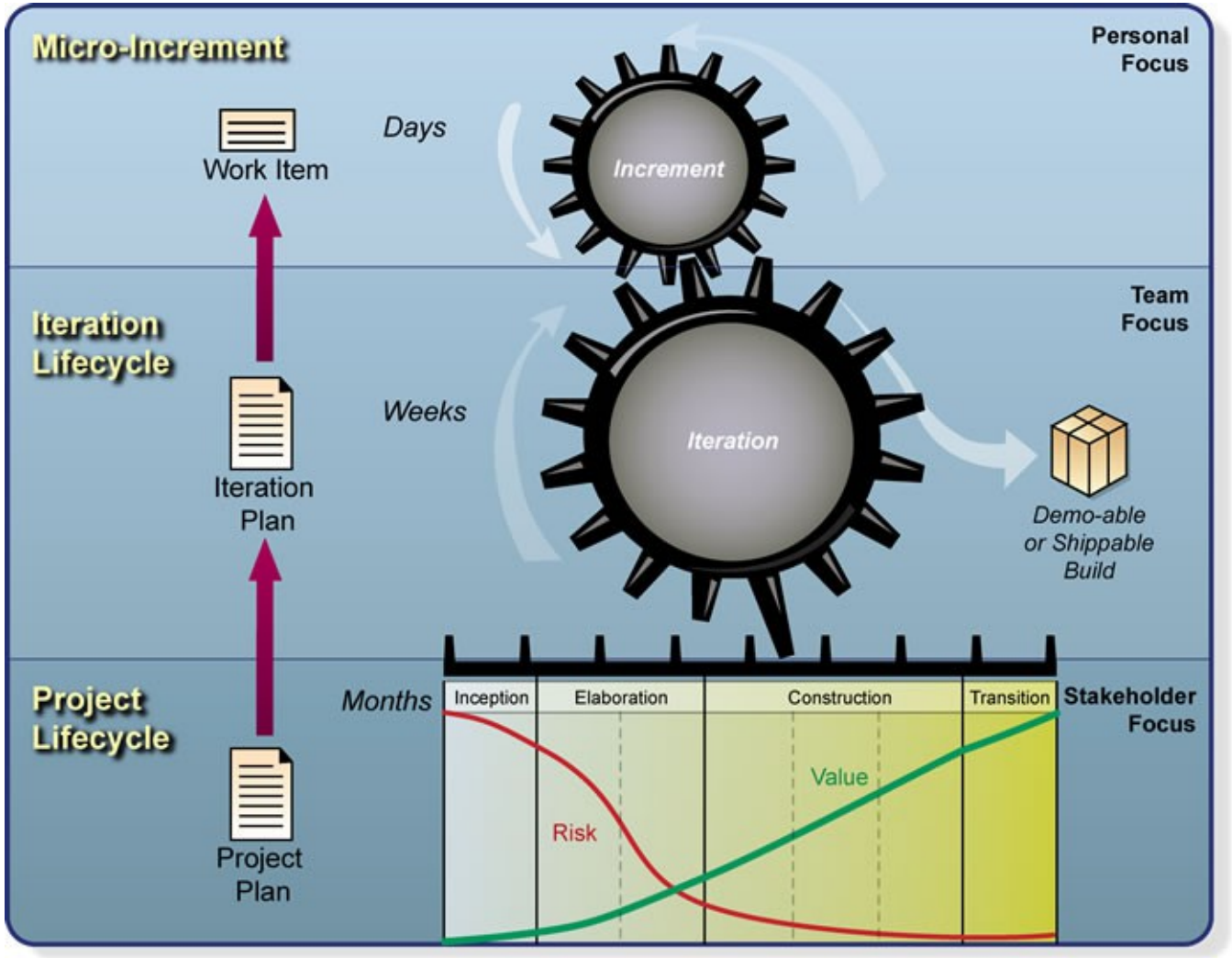
- Der RUP enthält eine Bibliothek von Vorgehensbausteinen (workflows).
- Ein Vorgehensbaustein enthält:
  - Verknüpfung der Aktivitäten über die Abhängigkeitsbeziehungen
  - Aktivität als eine Einheit von Arbeitsschritten
  - Aktivitäten können mehrmals durchgeführt werden
  - Typen der weiterzugebenden Artefakte (Arbeitsprodukte):
    - Textdokumente
    - Modellelemente
    - Modelle
    - (Quell-)Code
    - Testspezifikationen
  - Zuordnung der Rollen zu den durchzuführenden Aktivitäten
    - Wahrnehmung der Verantwortlichkeiten für die Artefakte durch die Rollen
    - Eine konkrete Person kann mehrere Rollen übernehmen u. a. m.

Bsp: RUP Workflow "Requirements Analysis" als Aktivitätendiagramm

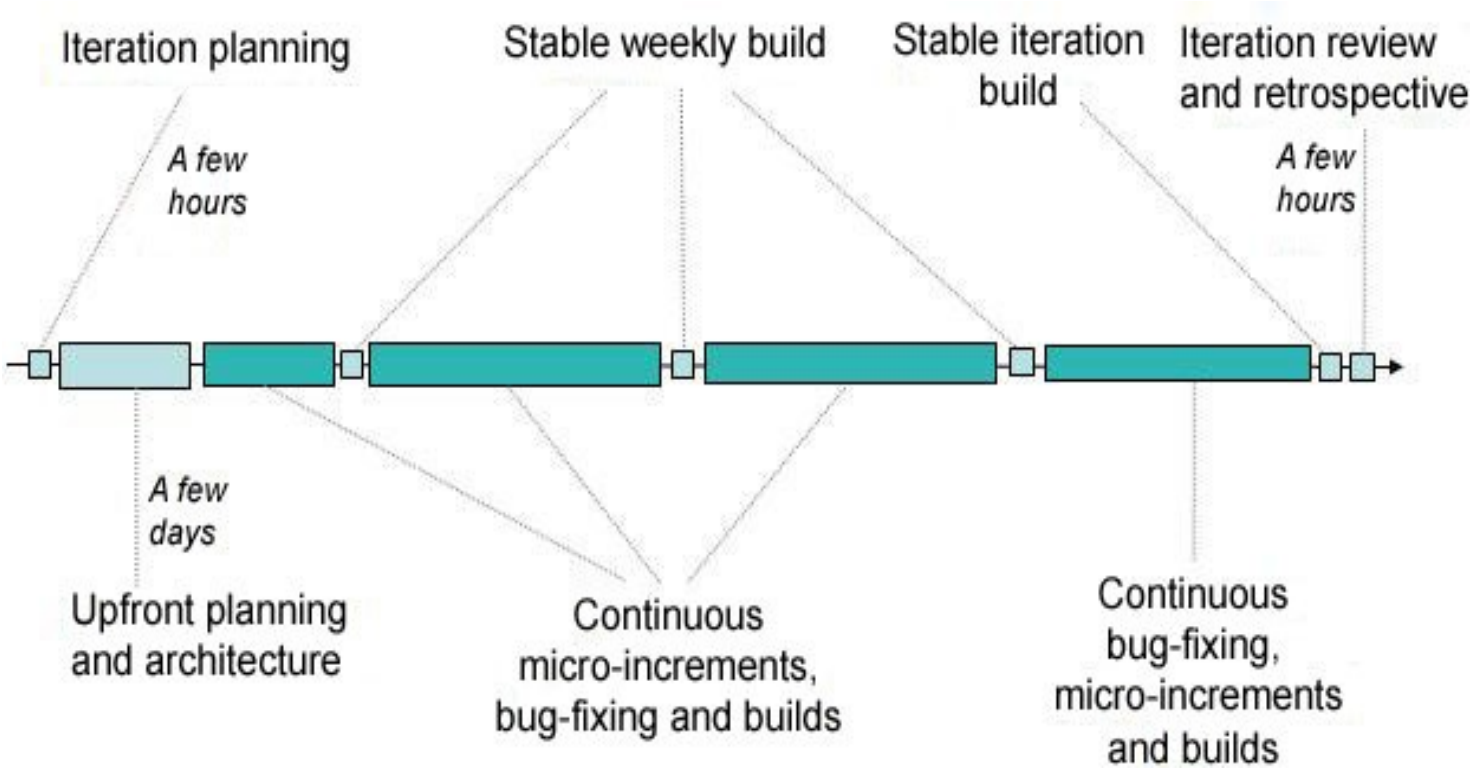


Open Unified Process

- http://epf.eclipse.org/wikis/openup/
- Employs InECT



OpenUP Iteration



14.2.2 V-Modell-XT des Bundes (VMXT)

- ftp.tu-clausthal.de/pub/institute/informatik/v-modell-xt/.../2.0/V-Modell-XT-Gesamt.pdf

Historie

- **1986** Das Bundesministerium der Verteidigung gibt ein Vorgehensmodell für die Softwareentwicklung auf Basis des V-Modells in Auftrag
  - Erstes Modell V-Modell 92 vomBundesamt für Wehrtechnik und Beschaffung (BWB) zusammen mit einem CASE-Tool-Hersteller entwickelt
- **1992** Erste Version wurde für die Bundesbehörden des Verteidigungsministeriums verbindlich
- **1994** Gründung Nutzergemeinschaft ANSSTAND e.V. (Anwender des Softwareentwicklungsstandards der öffentlichen Verwaltung), die regelmäßig Erfahrungsaustausche organisierte
- **1997** Veröffentlichung des verbesserten V-Modell‘97 (mit Objektorientierung) als IT-Standard der Bundes
- **2005** Veröffentlichung des V-Modell XT (VMXT)
- **2009** VMXT 1.3
- **2011** VMXT 1.4+
- **2016** VMXT 2.0

Ziele des VMXT

- **Prozessverbesserung**: Lernen aus abgeschlossenen Projekten durch Erfahrungsmanagement
  - Minimierung der Projektrisiken durch einen einfachen und klaren Prozess
- **Flexibilität** bei der Projekt- und Projektrollenanpassung durch projektspezifisches V-Modell XT (Tailoring)
  - Verbesserung und Gewährleistung der Qualität
- **Erweiterung** von Anwendungsbereichen (durch Projekttypen) und Abstraktionsebenen
- **Inkrementalität**
- **Kostenkontrolle**:
  - Senkung der Kosten mittels klarer Schnittstellen und definierter Rollen in jedem Einzelzyklus des Projektes
  - Transparente Kontrolle der Kosten über den gesamten Systemlebenszyklus

Quelle: Broy, M., Rausch, A.:Das neue V-Modell XT - Ein anpassbares Modell für Software und System Engineering; Informatik-Spektrum 28(2005) H.3, S.220-229

Aufbau und Auslieferungsstruktur des V-Modell XT

Vorgehensbausteine (Moduln) beinhalten schematische Workf bws, die Aktivitäten, Produkte und Rollen in sich organisieren.

Projekttypen legen Grobstrukturen von Projekten und **Projektdurchführungsstrategien** (PDS) fest.

Teil 1:
Grundlagen des V-Modells

Teil 2:
Eine Tour durch das V-Modell

Teil 3:
V-Modell-Referenz-Tailoring

Teil 4:
V-Modell-Referenz-Rollen

Teil 5:
V-Modell-Referenz-Produkte

Teil 6:
V-Modell-Referenz-Aktivitäten

Teil 7:
V-Modell-Referenz-Konventionsabbildungen

Teil 8:
Anhang

Teil 9:
Vorlagen

[www.v-modell.iabg.de]

Projekttypen

Ein **Projekttyp** legt die Infrastruktur und Rahmenbedingungen eines Projekts fest.

1) **Systementwicklungsprojekt**
    1) eines **Auftraggebers (AG-Projekt)**
       Der Projekttyp vereinigt die Projektdurchführungsstrategie des Auftraggebers
    2) eines **Auftragnehmers (AN-Projekt)**
       Der Projekttyp vereinigt die Projektdurchführungsstrategie des Auftragnehmers
    3) eines **Auftragnehmers mit dem Auftraggeber (AG-AN-Projekt)**
       Projekt ist in der gleichen Organisation oder in mehreren Organisationen, die bewusst miteinander arbeiten
2) Einführung eines **organisationsspezifischen Vorgehensmodells**
   Der Projekttyp vereinigt Projektdurchführungsstrategien, die für Auftraggeber und Auftragnehmer alle organisatorischen Maßnahmen bzgl. der Einführung eines organisationsspezifischen V-Modells beinhalten.

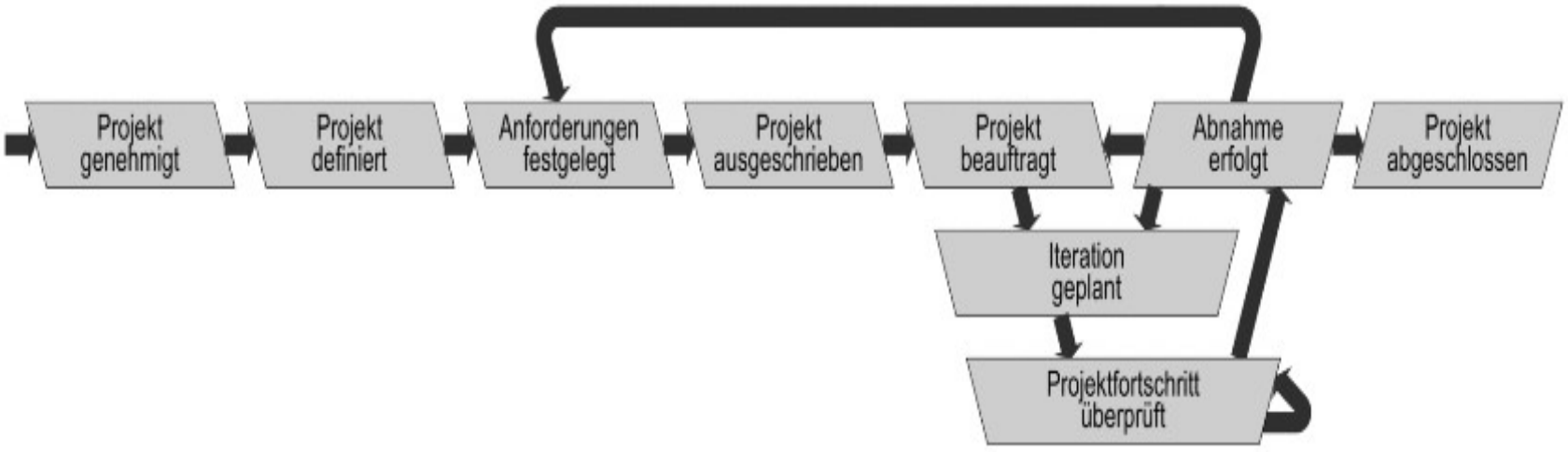
Projekttypvarianten

- Zu jedem Projekttyp gibt es **Projekttypvarianten**, die die Projektdurchführungsstrategie festlegen
  - Beispiel: Projekttypvarianten des Projekttyps Systementwicklungprojekt (AG)
  - AG-Projekt mit einem Auftragnehmer
  - AG-Projekt mit mehreren Auftragnehmern

- Eine Projekttypvariante legt eine **Projekt-Durchführungsstrategie (PDS)** fest, die einen schematisch geordneten Projektablauf festlegt
  - Festlegung der Reihenfolge, in der die für das Projekt relevanten Entscheidungspunkte durchlaufen werden müssen
  - Unterteilung des Projektes in einzelne Abschnitte

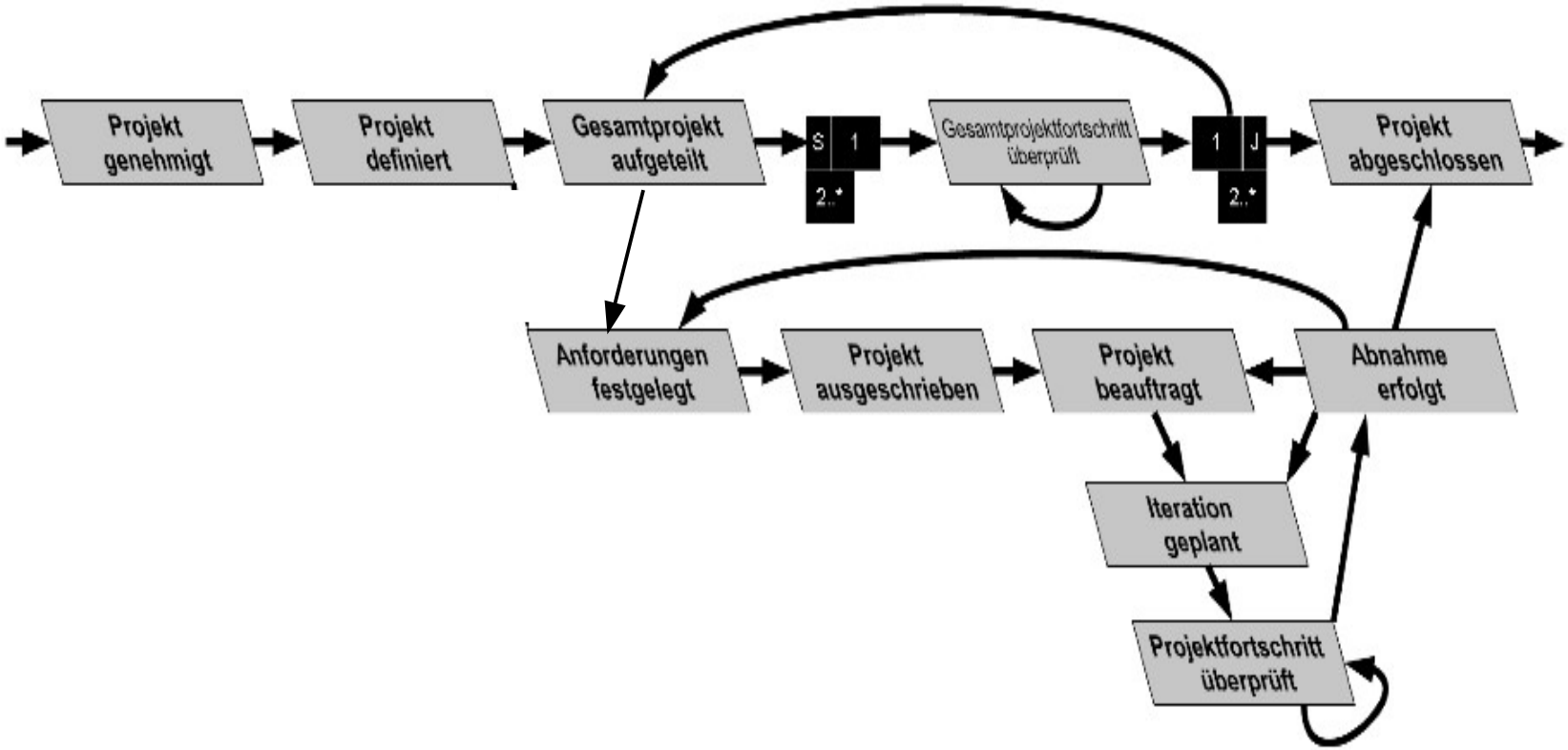
Beispiele für PDS (1): AG-Projekt mit einem Auftragnehmer

- Das VMXT verwendet für die festlegung von PDS **Meilenstein-Graphen (Entscheidungspunkt-Steuerfluß-Graphen)**
  - **Ereignisknotengraph**: Entscheidungspunkte werden durch Steuerfluss verbunden
  - Von den Aktionen wird abstrahiert
- Beispiel: Vergabe und Durchführung von Systementwicklungsprojekten (Schritte des Auftraggebers während des Projektes – System wird nicht selbst entwickelt) mit Abfolge der zugehörigen Entscheidungspunkte



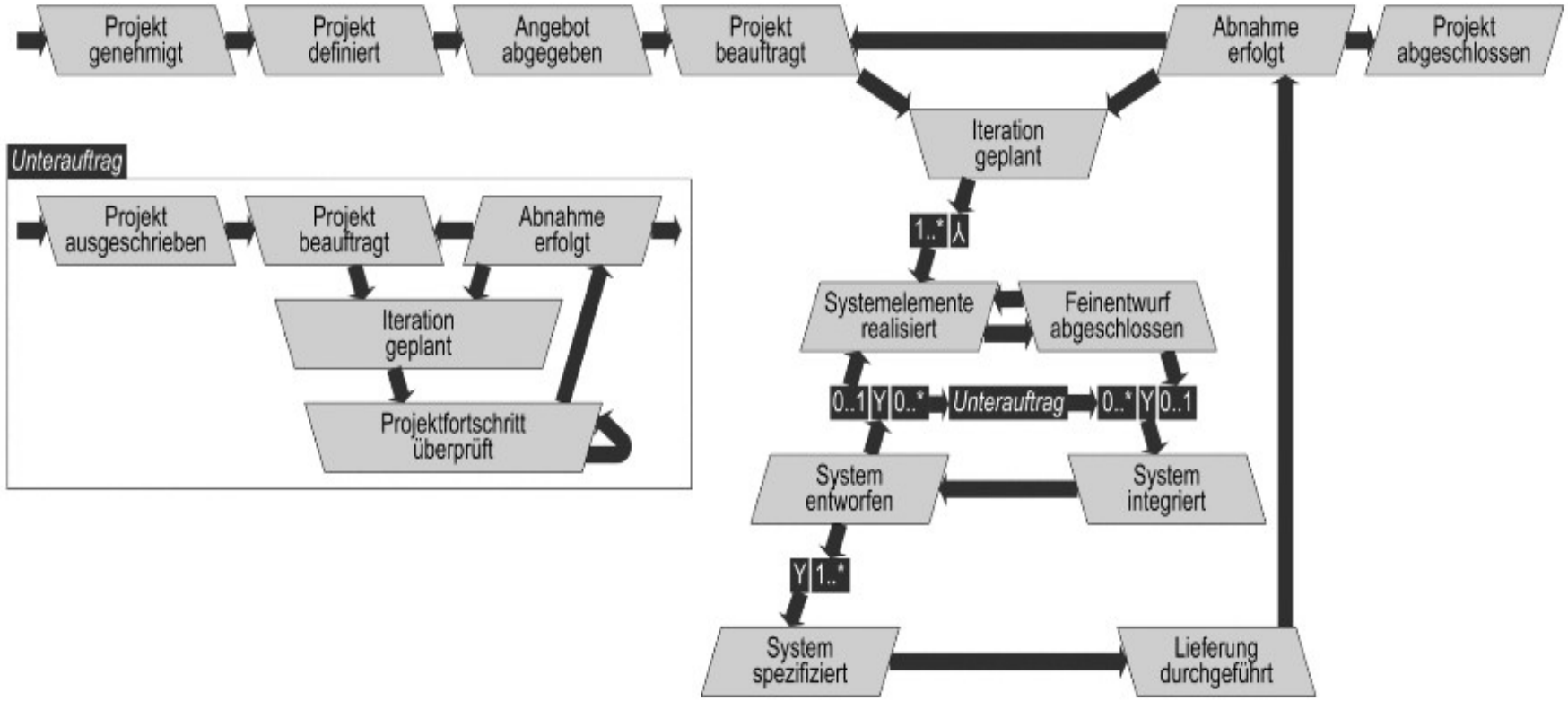
http://v-modell.iabg.de/v-modell-xt-html/index.html

Beispiele für PDS (2): Auftraggeber-Projekt mit mehreren Auftragnehmern



Beispiele für PDS (3): Agile Systementwicklung

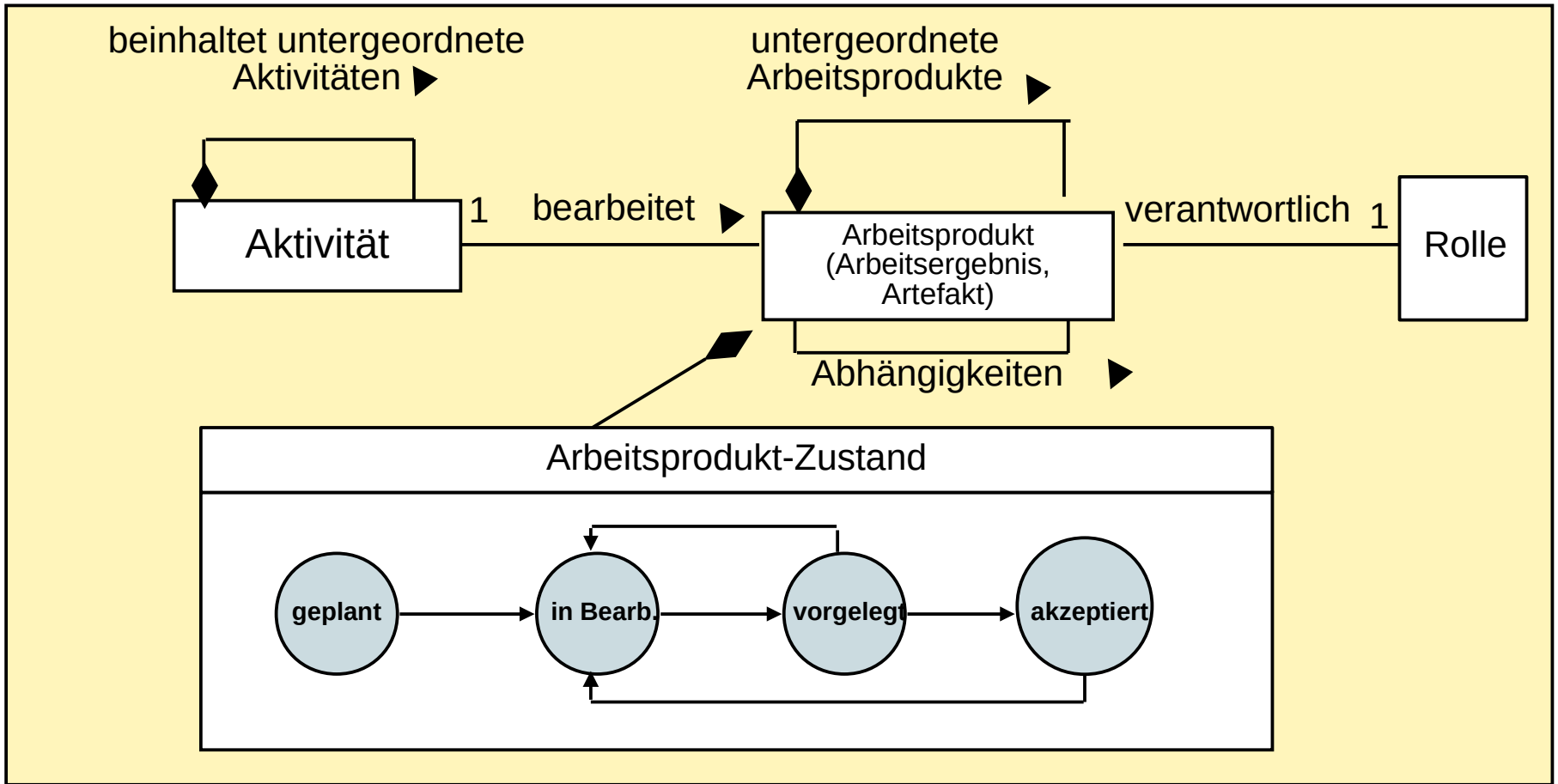
- Diese Strategie basiert auf der Erwartung eines hohen Realisierungsrisikos, d.h. sie wird eingesetzt, wenn Anforderungen von vornherein nicht eindeutig definierbar sind.



http://v-modell.iabg.de/v-modell-xt-html/index.html

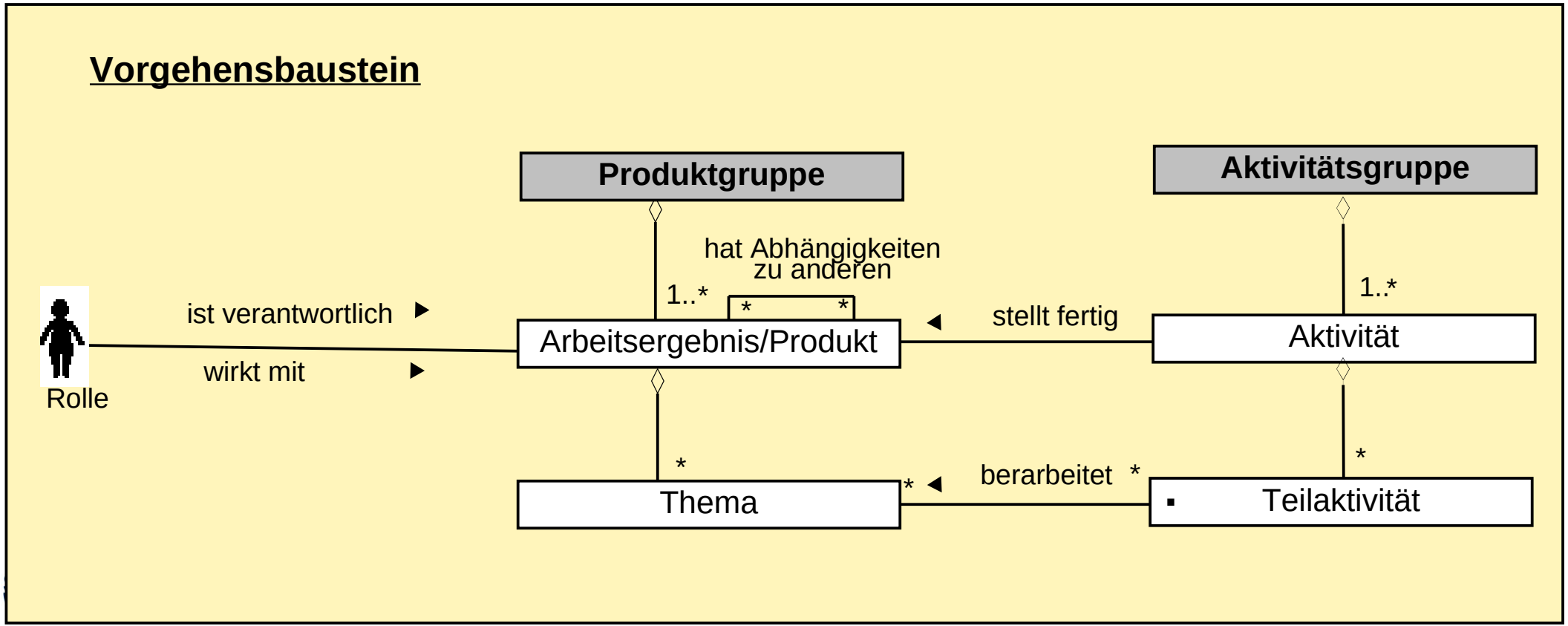
Metamodell eines Vorgehensbausteins im VMXT

- **Arbeitsergebnisse (Arbeitsprodukte)** sind bestimmte Dokumentations- und Teilergebnisse der Entwicklung. (Produkt == Artefakt == Objekt == Ergebnis)
- Der **Produkttyp** beschreibt auf abstrakter Ebene Produktexemplare, die während eines Entwicklungsprozesses entstehen.
- Ein **Produktexemplar** ist die Ausprägung eines Produkttyps.



Vorgehensbausteine und ihre Bestandteile

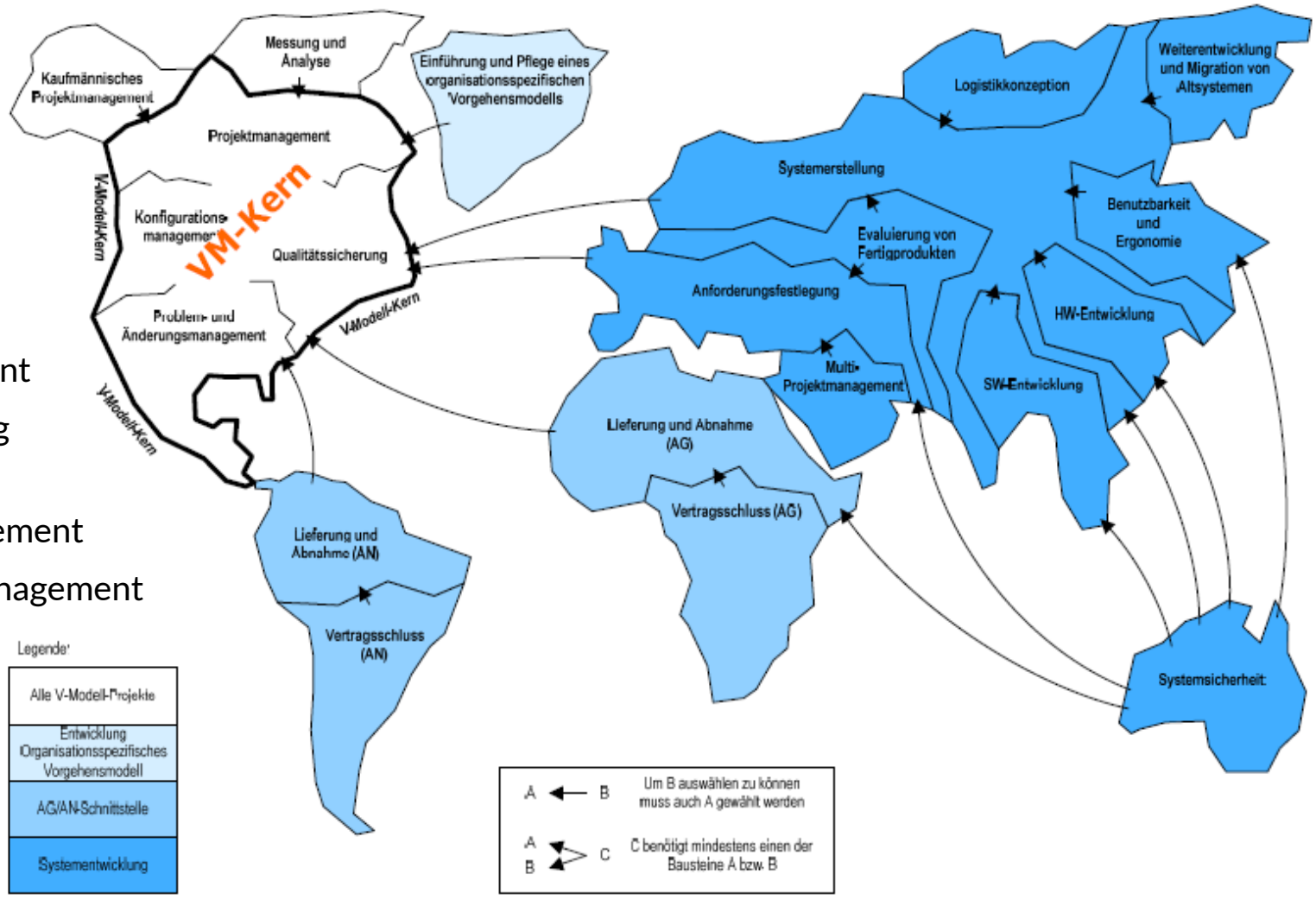
- Eine **Produktgruppe** ist eine Sammlung von einzelnen Produkten, welche einem Vorgehensbaustein zugeordnet sind.
- **Externe Produkte** werden nicht im Rahmen des V-Modell Projekts erstellt.
- Ein **initiales Produkt** ist ein Produkt, das in jedem Fall genau einmal erstellt werden muss.
- Eine **Aktivitätengruppe** ist eine zusammenhängende Gruppe von Aktivitäten



“Landkarte” der Vorgehensbausteine: Modulbibliothek des VMXT

Kernmodule sind

- Projektmanagement
- Qualitätssicherung
- Problem- und Änderungsmanagement
- Konfigurationsmanagement



http://v-modell.iabg.de/v-modell-xt-html/index.htm

(Arbeits-)Produktgruppen

- (Arbeits-)Produktgruppen (Artefaktgruppen) können den Bereichen Projekt, Entwicklung und Organisation zugeordnet werden.

Beispiele:

- **Bereich Projekt** – 6 Produktgruppen
  - Planung und Steuerung beinhaltet organisatorische und essentielle Produkte des Projektmanagements
  - Berichte beinhalten alle management-begleitenden Dokumente oder Produkte
- **Bereich Entwicklung** – 6 Produktgruppen
  - Im Systementwurf werden alle Produkte gesammelt, welche zur technischen Realisierung benötigt werden
- **Bereich Organisation** – 1 Produktgruppe
  - Prozessverbesserung wird für die Einführung und Pflege eines spezifischen Vorgehensmodell verwendet

Produktgruppe Kernmodul Projektmanagement

Planung und Steuerung	
	E Projektfortschrittsentscheidung
I	Projekthandbuch
I	QS-Handbuch
	Projektmanagement-Infrastruktur
	Schätzung
	Risikoliste
I	Projektplan
	Arbeitsauftrag
	Kaufmännische Projektkalkulation

Berichtswesen	
	Besprechungsdokument
E	Projektstatusbericht (von AN)
E	Projektabschlussbericht (von AN)
	Projekttagebuch
	Messdaten
	Metrikauswertung
	Kaufmännischer Projektstatusbericht
	Projektstatusbericht
	QS-Bericht
	Projektabschlussbericht

Konfigurations- und Änderungsmanagement	
E	Problemmeldung/Änderungsantrag
	Problem-/Änderungsbewertung
	Änderungsentscheidung
	Änderungsstatusliste
I	Produktbibliothek
	Produktkonfiguration

Prüfung
Prüfspezifikation Dokument
Prüfprotokoll Dokument
Prüfspezifikation Prozess
Prüfprotokoll Prozess
Prüfspezifikation Benutzbarkeit
Prüfprotokoll Benutzbarkeit
Prüfspezifikation Systemelement
Prüfprozedur Systemelement
Prüfprotokoll Systemelement
Prüfspezifikation Lieferung
Prüfprotokoll Lieferung

Ausschreibungs- und Vertragswesen	
	Ausschreibungskonzept
	Ausschreibung
	Kriterienkatalog für die Angebotsbewertung
E	Angebot (von AN)
	Angebotsbewertung
	Vertrag
	Vertragszusatz
E	Lieferung (von AN)
	Abnahmeerklärung

Angebots- und Vertragswesen	
I E	Ausschreibung (von AG)
I E	Bewertung der Ausschreibung
	Angebot
I E	Vertrag (von AG)
E	Vertragszusatz (von AG)
	Lieferung
E	Abnahmeerklärung (von AG)

6 Produktgruppen

V-Modell XT Teil 5

Rollen im VMXT

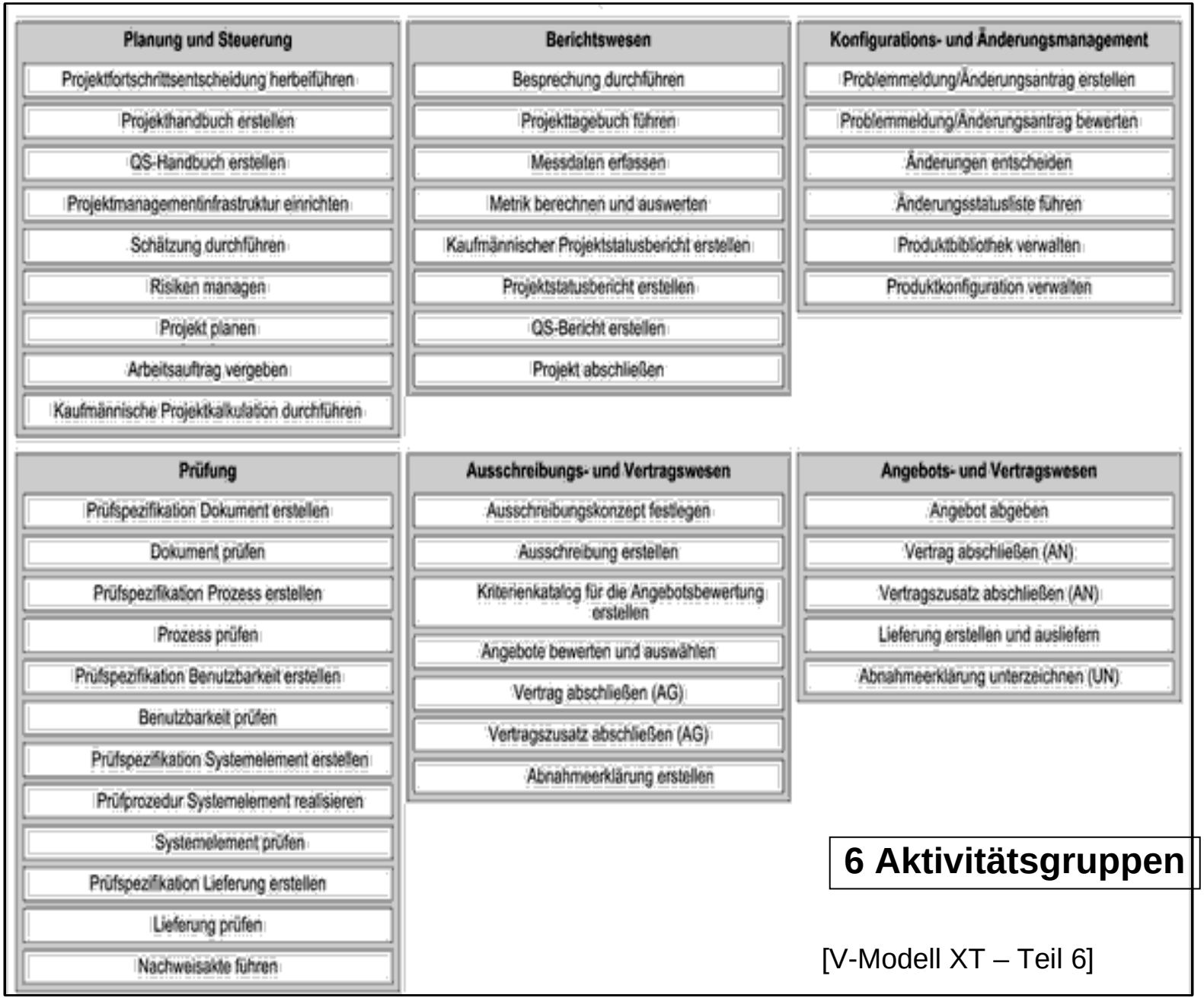
- Jedem Arbeitsergebnis/Produkt wird eine eindeutige Rolle zugewiesen, der eine Person im Projekt entspricht.
  - Eine Person kann auch mehrere Rollen einnehmen.
- Beispiele für Rollen sind:

Anforderungsanalytiker(AN oder AG)	Projektleiter
Ausschreibungsverantwortlicher	Projektmanager
System-Designer	Projektassistent
DV-Analytiker	Projektmanager beim AG
DV-Designer	
SW-Architekt	QS-Manager
SW-Entwickler	QS-Verantwortlicher (AN oder AG)
Programmierer	Qualitätsprüfer
	QS-Assistent
Support-Berater	
Applikationsberater	KM-Verantwortlicher
HW-Berater	Konf.-Administrator
Technischer Autor	Datenschutz- und Sicherheitsberater

Aktivitäten

- In einer **Aktivität** wird festgelegt, welches Arbeitsergebnis (Produkt, Artefakt) von wem, wie erstellt wird.
- Ein **Aktivitätstyp** bildet ein Schema für Aktivitäten.
- Aktivitäten können in **Aktivitätsgruppen** zusammengefasst werden.
  - Die Aktivitätsgruppen können analog den Produktgruppen den Bereichen zugeordnet werden:
    - Projekt
    - Entwicklung
    - Organisation
- Die **Aktivitätsstruktur** bildet die Menge aller Aktivitätsexemplare eines Projekts und deren Beziehungen zueinander.

Aktivitätsgruppen im Kernmodul Projektmanagement

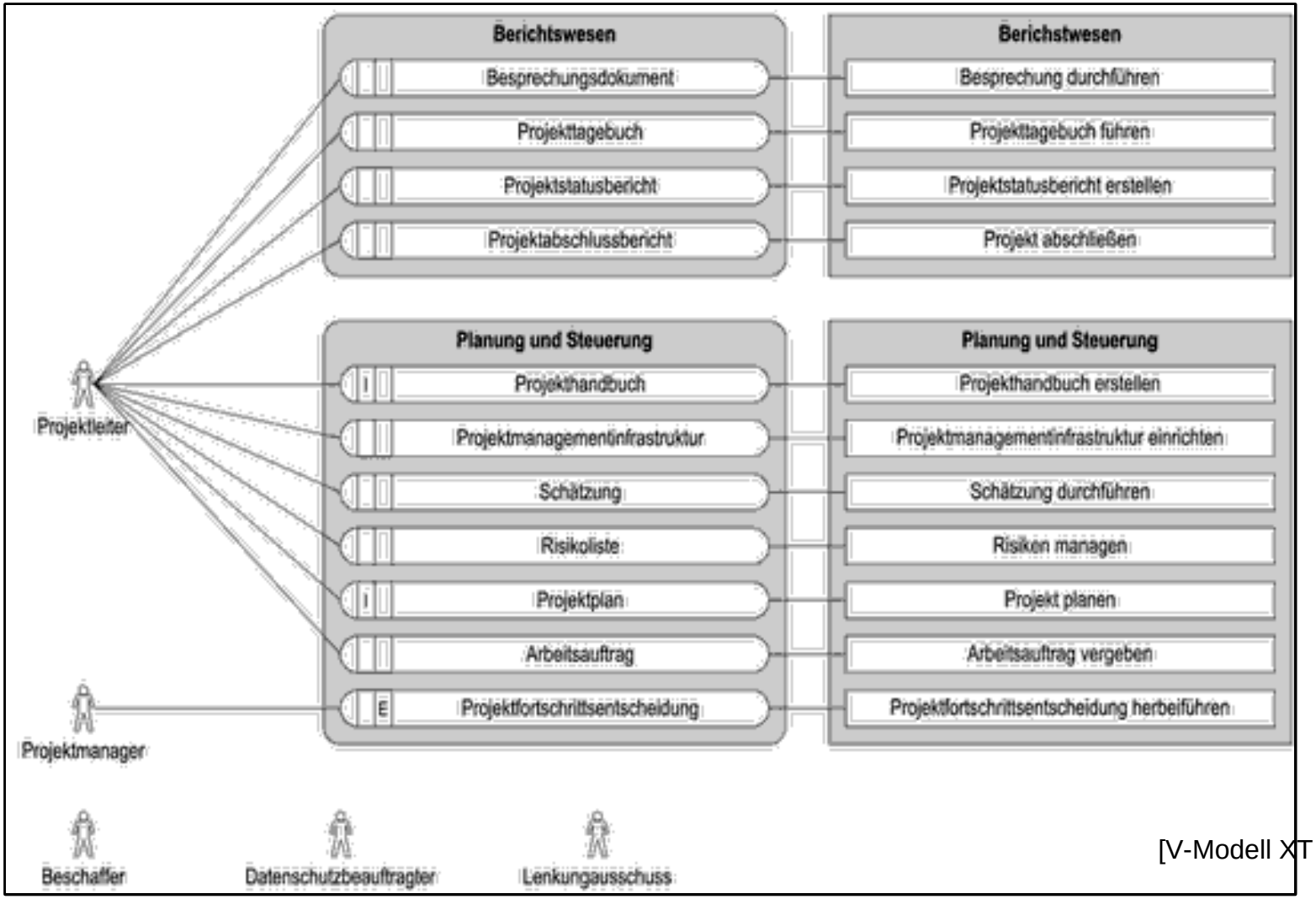


6 Aktivitätsgruppen

[V-Modell XT – Teil 6]

Kernmodul Projektmanagement: Gruppierung von Aktivitäten und Arbeitsprodukten

- durch Kombination

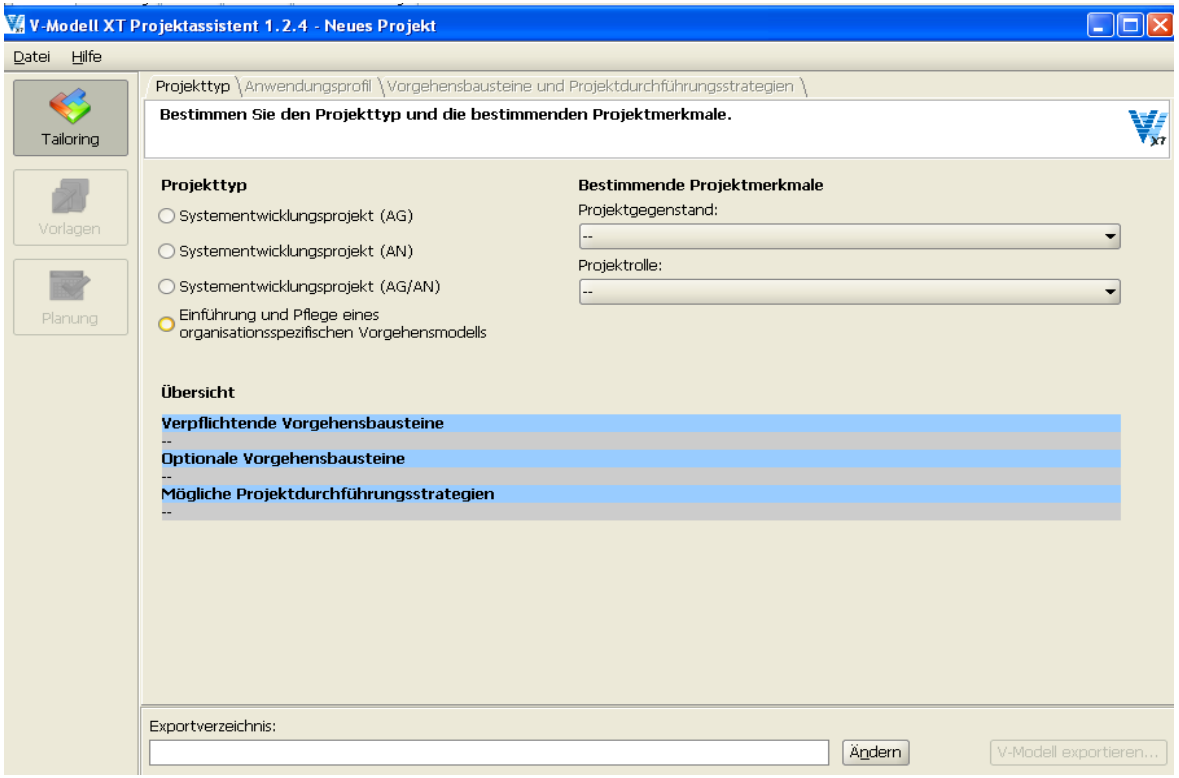


[V-Modell XT – Teil 3]

Projektspezifische Anpassung - Tailoring

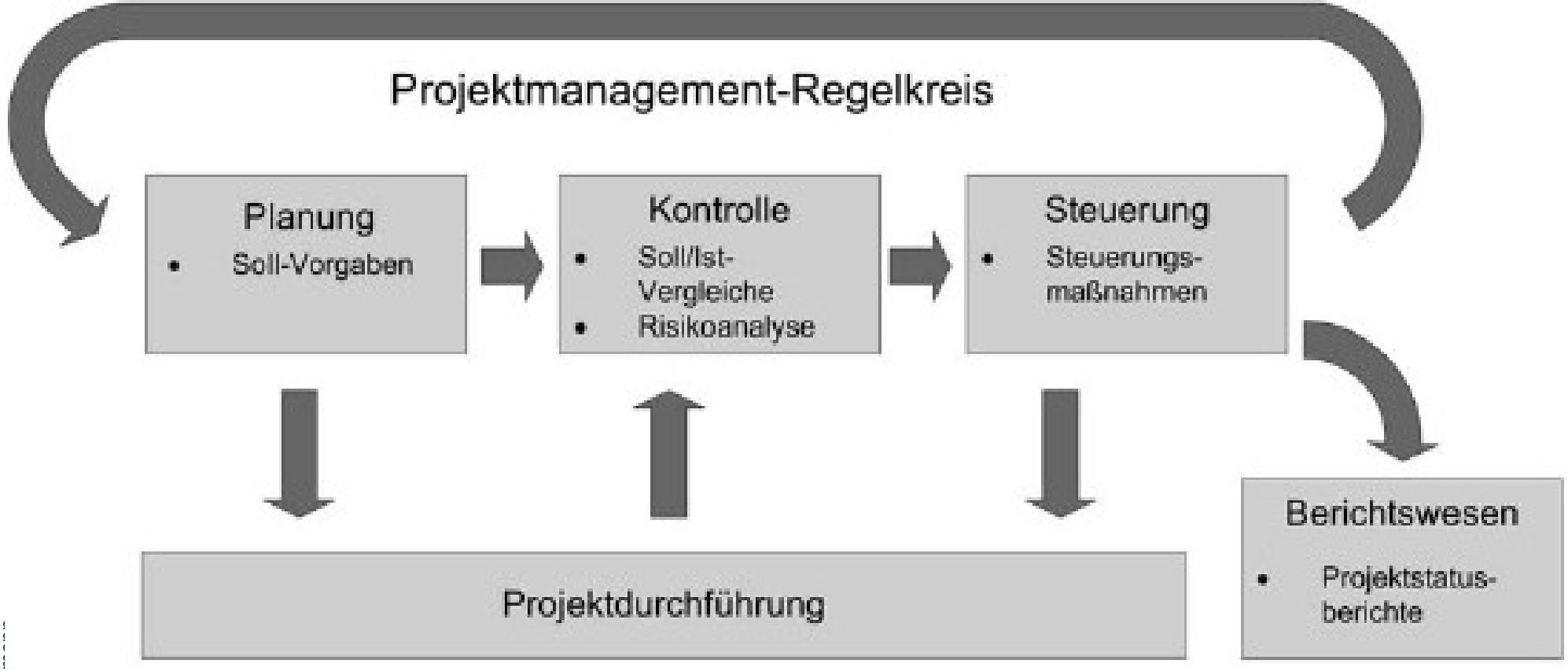
Die Anpassung des V-Modells an konkrete Projektbedingungen wird **projekt-spezifisches Tailoring** genannt.

- Auswahl einer der vier unterstützten **Projekttypen** u. **Projekttypvarianten**
- Definition von **Streichbedingungen** für Produkte, Rollen und Aktivitäten, um ihre Anzahl auf das notwendige Maß zu reduzieren
- Festlegung des **Anwendungsprofils**
- Auswahl der zu verwendenden Vorgehensbausteine und Projektdurchführungsstrategie



VMXT Regelkreis als Ausprägung des PDCA

- Mit dem VMXT kann man Projekte im Regelkreis führen



Bewertung des V–Modells des Bundes

Vorteile: „nichts vergessen“

- **Vollständige Behandlung** von Entwicklung, Qualitätssicherung, Konfigurationsmanagement, Projektmanagement sowie anderen Prozessen (nichts vergessen!)
- **Generisches Vorgehensmodell** mit definierten Möglichkeiten zum Maßschneidern auf projektspezifische Anforderungen
- Ermöglicht eine standardisierte Abwicklung von Systemerstellungs-Projekten
- Eine sehr gute Basis für die Prozesszertifizierung nach ISO 9000
- Gut geeignet für große Projekte, auch für eingebettete Systeme

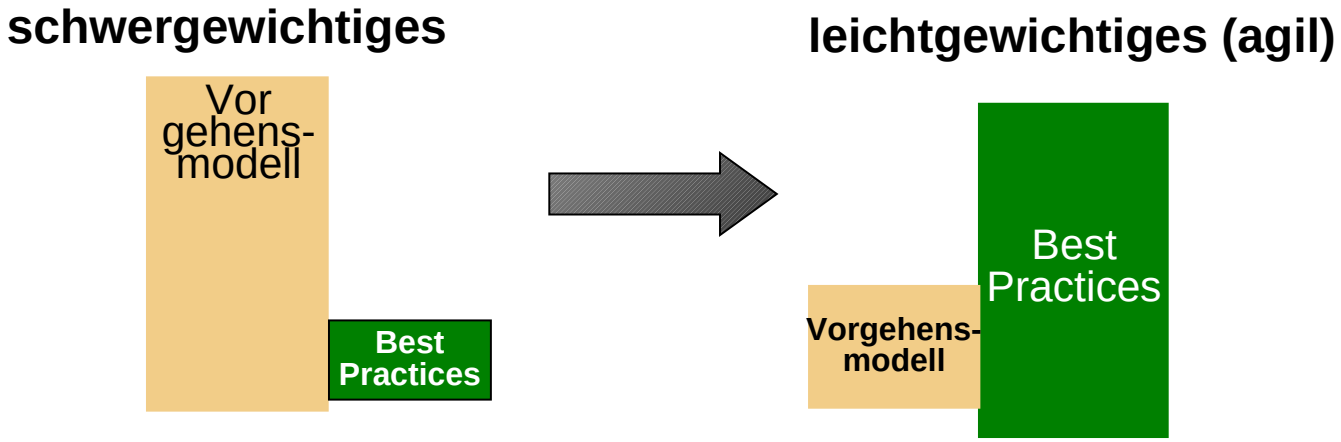
Nachteile:

- Führt zu einer großen Produktvielfalt und Softwarebürokratie bei kleinen und mittleren Softwareentwicklungen bzw. -unternehmen
- Ohne Werkzeugunterstützung insbesondere zur Unterstützung der Produkterstellung (Dokumentation) ist V-Modell schwer handhabbar
- 23 definierte Rollen erscheinen überdimensioniert
- Gefahr des unkritischen Übertragens der Vorgehenskonzepte auf andere Produkttypen bzw. Anwendungsprofile

14.3 Leichtgewichtige Vorgehensmodelle

Evolutionäres Vorgehen auf der Basis des PDCA, in kleinen, kurzen Zyklen mit rascher Rückkopplung

Agile Softwareentwicklung



„Best Practices“

- heißt kontinuierlich die Projektsituation beurteilen und entscheiden (kurze Feedback-Zyklen)
- so wenig wie möglich, aber soviel wie nötig (“lean management”)
- der Ablauf wird ständig an neue Rahmen-bedingungen angepasst und verbessert
- am wichtigsten ist, zuerst der Mensch, dann die Methode und zuletzt das Werkzeug
- wenn ein Tool hilft, wird der Bearbeiter es kaum missen wollen
- Veränderungspotential aus der Gruppe heraus entwickeln. Mitarbeiter akzeptieren Veränderungen erst nach und nach – Change Management
- eher Angemessenheit als Extremismus

embrace change!

[16]

Maxime des agilen Handelns

Eher offen für Änderungen	⇒	als starres Festhalten an Plänen
Eher Menschen und Motivation	⇒	als Prozesse und Tools
Eher Vertrauen	⇒	als Kontrolle
Eher ergebnis-orientiert	⇒	als prozess-orientiert
Eher „darüber miteinander reden!“	⇒	als „gegeneinander schreiben“
Eher „Best Practices“ aus Erfahrung	⇒	als verordnete Vorgaben

Cziharz, T.: Agilität – Spagat zwischen Ultraleichtbau und Giganten; Vortrag GIRegDD am 15.07.04

14.3.1 Extreme Programming

Was heißt ‚Extreme Programming'?

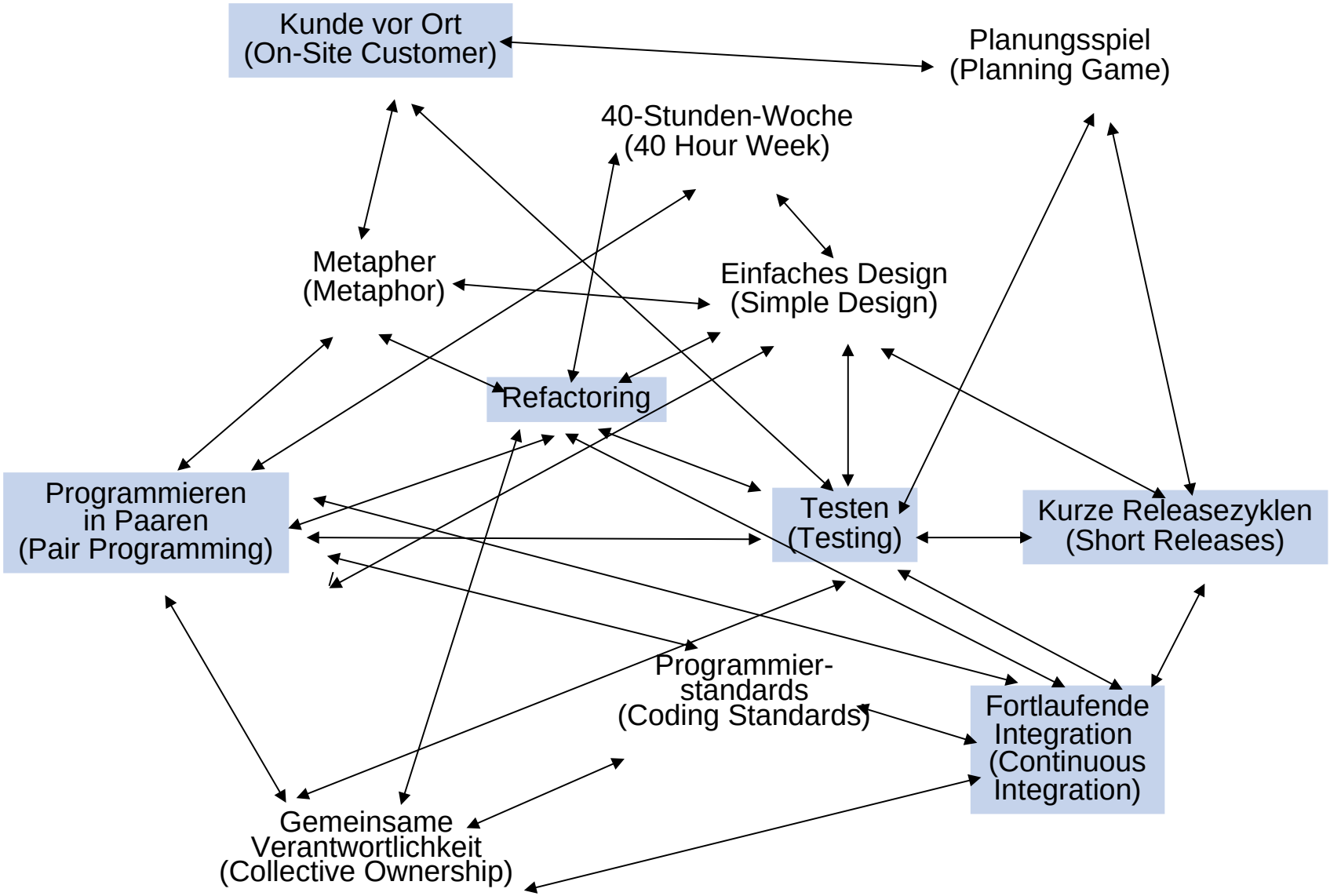
- **XP** ist ein evolutionärer (agiler) Softwareentwicklungsprozess für kleine Teams in der Größe von zwei bis etwa zwölf Programmierern
- Werte des XP:
  - Kommunikation, Einfachheit, Feedback, Disziplin, Lernen, Qualität und Respekt
- Prinzipien des XP:
  - kurze **Iterationen** in Perioden von ein bis drei Wochen, am Ende einer Periode steht ein getestetes System mit neuer Funktionalität
    - offene Arbeitsumgebung in größerem Raum mit Flipcharts mit täglichem Standup-Meeting
    - verständliche Sprache und gemeinsames Vokabular im Team, um über das zu erstellende System effektiv diskutieren zu können
  - jede Iteration endet damit, in einem Rückblick die eigene Arbeitsweise kritisch zu reflektieren --> **Retrospektive**
  - Anforderungsanalyse in Form einfacher Geschichten als **User Stories**, die mittels vom Nutzer geschriebener Story-Karte festgehalten werden
- Techniken

Westphal, F.: Extreme Programming; Copyright 2000-2004; URL: http://www.frankwestphal.de/ExtremeProgramming.html

5 zentrale Prinzipien von XP

- Unmittelbares Feedback
  - Je kürzer der zeitliche Abstand zwischen Aktion und Rückmeldung ist, desto größer ist der Lernerfolg
- Einfachheit anstreben
  - Einfache Lösungen haben eine Reihe wichtiger Vorteile gegenüber komplizierten Lösungen; daher verständlicher, leichter änderbar, schnellere Reaktion
- Inkrementelle Veränderung
  - Durch Änderungen in kleinen Schritten bleiben Effekte beherrschbar; jede Änderung basiert auf der vorherigen, so dass Reihenfolge überschaubar bleibt (Vermeidung von Seiteneffekten)
- Veränderung willkommen heißen (embrace change)
  - Änderungen auf Basis von Anforderungen, Entwicklungsprozessen, Systembestandteilen sind nichts Ungewolltes; sie führen in ihrer Gesamtheit zum neuen Produkt
- Qualitätsarbeit [15 Balzert2]
  - Gute Software befriedigt Softwareentwickler; sie möchten Qualitätsarbeit abliefern, dazu müssen die Qualitätsmaßstäbe vorgegeben werden.

XP-Techniken im Zusammenhang



14.3.2 SCRUM

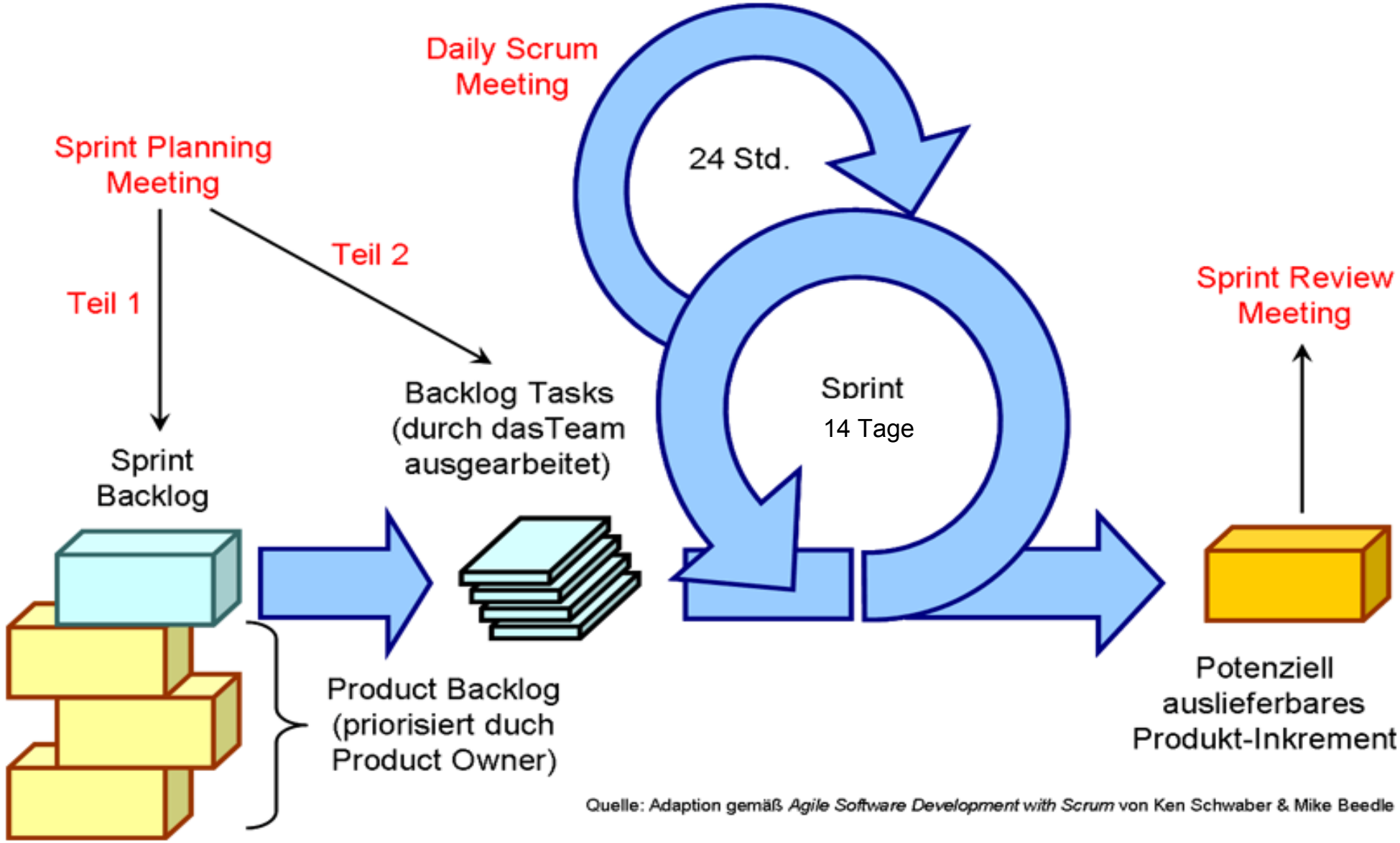
- SCRUM Alliance
  - http://www.scrumalliance.org/
- SCRUM certification (SCRUM master)
  - http://www.scrumalliance.org/certifications
- List of SCRUM tools (Florian Markert)
  - http://www.scrum-tools.de/
- ICESCRUM Freemium SCRUM tool, web based
  - http://www.icescrum.org/

Scrum

- Leichtgewichtiges Vorgehensmodell im Rahmen der agilen SW-Entwicklung mit **Zeit-Schachtelung (time boxing)**
- Hauptmerkmale:
  - Sorgfältige Planung mit “Sprint Planning Meetings” für 14 Tage
  - für komplexe Projekte mit unklar definierten Anforderungen
  - Einbeziehung des Kunden, um nicht fehl zu entwickeln
  - Iteratives Vorgehen, ständige Kontrolle
  - Wenig Rollen
  - Teams organisieren ihren Tagesablauf selbst
  - Ständige Neupriorisierung der Anforderungen
- http://www.scrum.org/Portals/0/Documents/Scrum%20Guides/Scrum_Guide.pdf

http://scrum-master.de

Scrum: Vorgehensweise



Quelle: Adaption gemäß *Agile Software Development with Scrum* von Ken Schwaber & Mike Beedle

Rollen im Scrum

- **Product Owner**
  - vertritt den Auftraggeber aus fachlicher Sicht
  - verantwortlich für das Product-Backlog
  - Priorisierung der einzelnen Product-Backlog-Elemente
  - ist Ansprechpartner für das Team
- **Scrum-Master**
  - ist verantwortlich für den gesamten Prozess
  - moderiert die Meetings
  - überwacht die Entwicklung (Product-Backlog; Sprint-Backlog ..)
- **Team**
  - 5 bis 10 Personen
  - verantwortlich für die Umsetzung der Product-Backlogs, bzw. der einzelnen Sprint-Backlogs
  - Aufwandsschätzung der einzelnen Backlog-Elemente
  - Team organisiert sich selbst

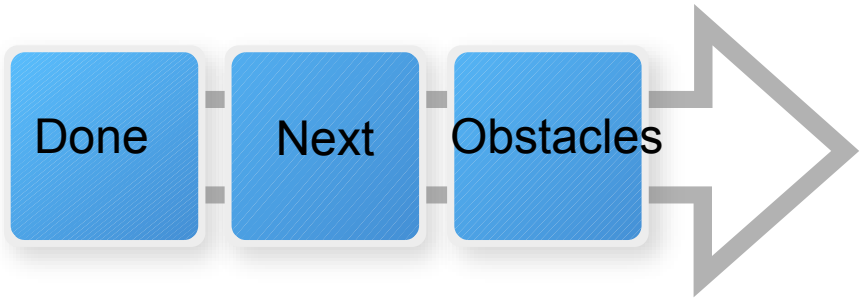
Scrum-Rollen – Pigs & Chickens

- Diese direkt am Prozess beteiligte Rollen werden **Pigs** genannt,
- Außenstehende **Chickens**.
- Der Ursprung kommt aus einem englischen Witz
  - A chicken and a pig were brainstorming …
  - Chicken: Let`s start a restaurant!
  - Pig : What would we call it?
  - Chicken: Ham `n`Eggs!
  - Pig : No thanks. I`d be commited, but you`d only be involved!

scrum.master.de

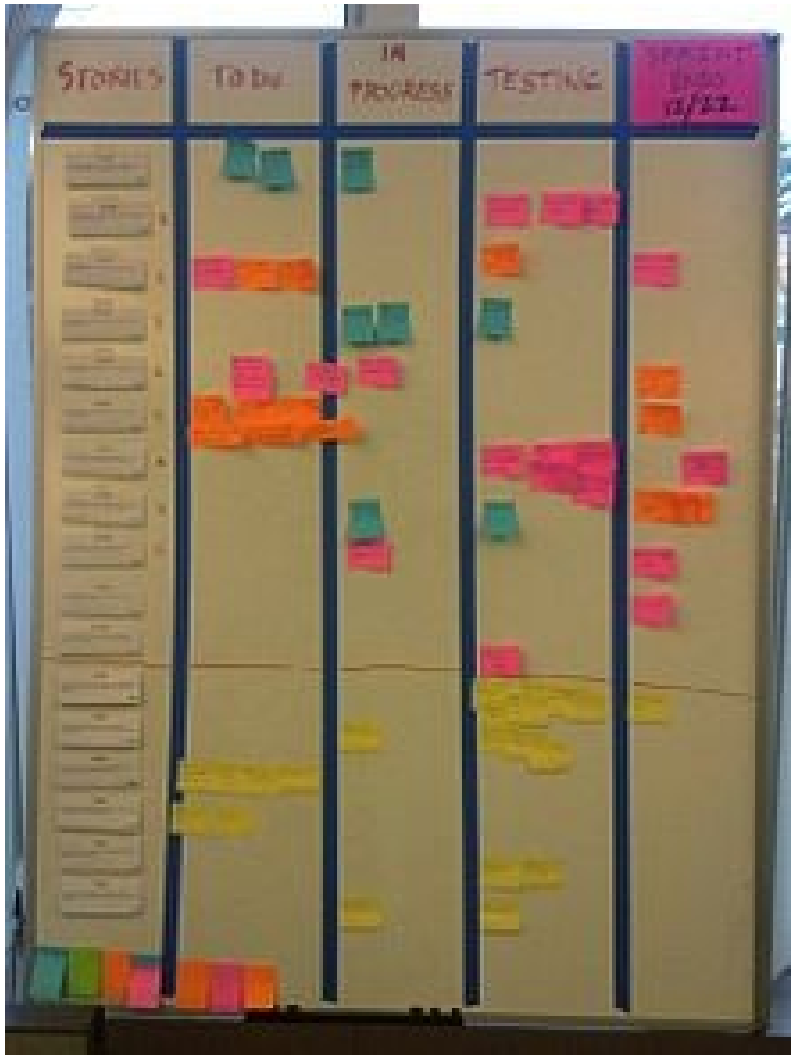
“Daily SCRUM Meetings” sind Problem-Meldungstreffen (einfaches, agiles Berichtswesen)

- Berichten über den Zustand des Sprints und aufgetretene Probleme.
  - “Stand-up meeting”: man steht, damit es kurz bleibt
- KEIN Meeting zum Diskutieren der Architektur oder von Problemen!
- 3 Fragen soll jedes Team-Mitglied beantworten (DNO questions): [ScrumGuide]
  - Was habe ich seit dem letzten Meeting getan?
  - Was werde ich nächstens tun?
  - Welche Hindernisse habe ich erkannt?
- Der SCRUM-Master muss die erkannten Hindernisse asap ausräumen. Er kämpft für das Team nach außen



Sprint Kanban Board

- Requirements eines Sprints werden als “user stories” an eine Wand gepinnt
- Ihr Zustand wird in weiteren Spalten verfolgt: “noch zu tun” → “laufend” → “zu testen” → “getan”
- ETEOBoard von Saxonia für IT-gestütztes SCRUM
  - http://www.eteoboard.de/de/

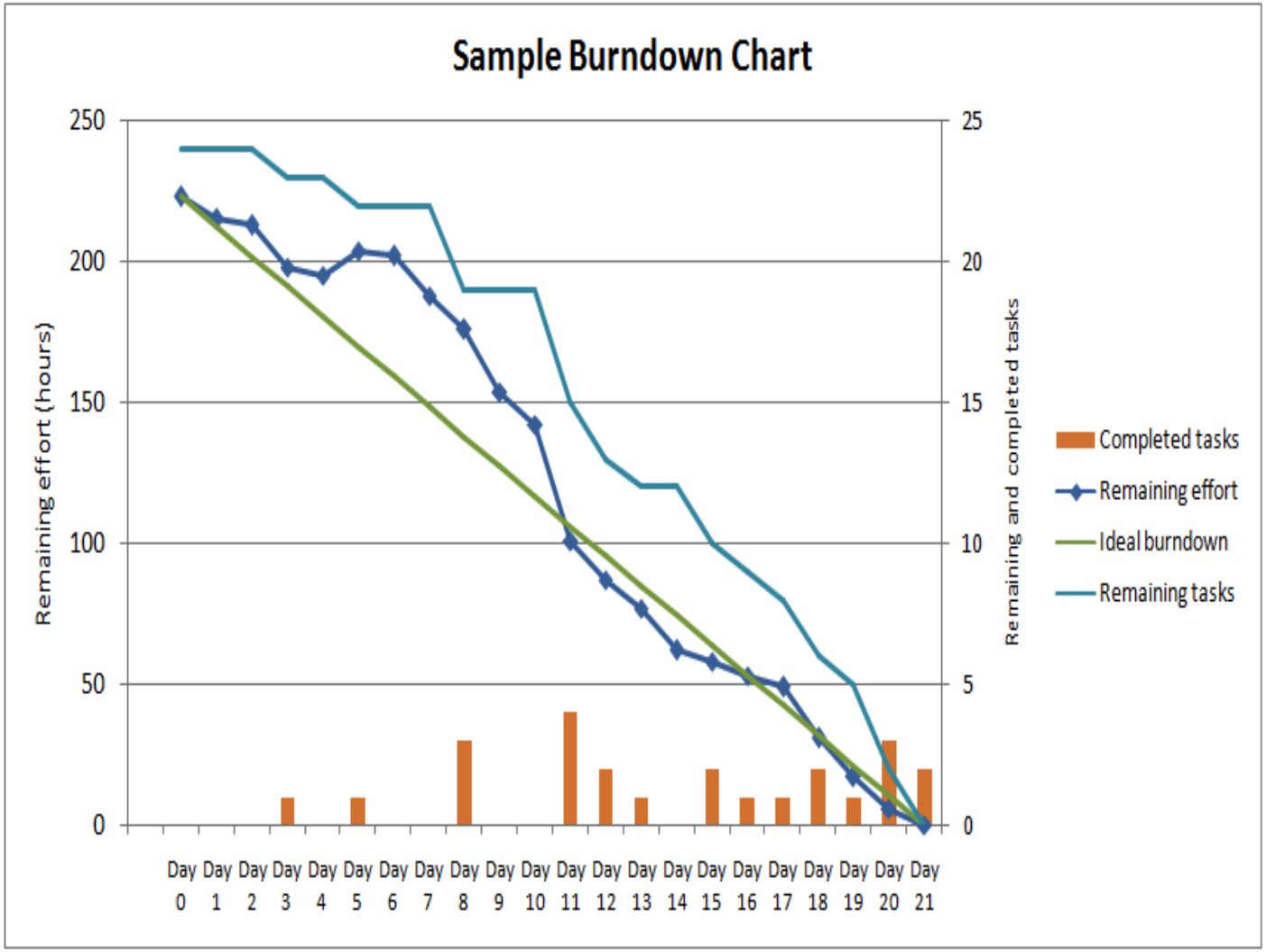


http://en.wikipedia.org/wiki/Kanban_board

http://en.wikipedia.org/wiki/Scrum_(development)

Sprint Burndown Chart

- Ein **burndown chart** hält den Fortschritt des Sprints fest, in Termini von erfüllten Anforderungen



http://upload.wikimedia.org/wikipedia/commons/0/05/SampleBurndownChart.png

Sprint Review Meetings

- **Sprint Review Meeting:**
  - Nach Abschluss des Springs zur Diskussion, was alles erreicht und nicht erreicht wurde
  - zur Repriorisierung des Product Backlogs
- **Sprint Retrospektive Meeting:**
  - Diskussion über Prozessverbesserung: was lief gut? was lief schlecht? was kann man besser machen?
  - Ständige Prozessverbesserung
- **Sprint Planning Meeting:**
  - zur Fixierung der User Stories für den nächsten Sprint

SCRUM

- SCRUM ist sehr beliebt; wird schätzungsweise heute in ca. 60% aller Firmen angewendet
- SCRUM wird oft in parallel arbeitenden Teams durch divide-and-conquer auf größere Probleme angewandt (many scrums)
- SCRUM wird auch mit VMXT und SPICE kombiniert
- SCRUM braucht aber den Kunden
- **SCRUM funktioniert nicht bei Festpreis-Projekten,** z.B. auf Ausschreibungen hin. Hier sollte ein stärker planendes Vorgehensmodell verwendet werden

14.3.3. Evolutionary Object-Oriented Software Development (EOS)

An agile process based on product-breakdown structure (PBS)

DRESDEN concept
Exzellenz aus Wissenschaft und Kultur

courtesy Prof. Wolfgang Hesse, University of Marburg

Obligatory Literature

- [Sarferaz] S. Sarferaz: "Methods and tool support for evolutionary, object oriented software development", Ph. D. thesis, Univ. of Marburg
- [Hesse 97a] W. Hesse: From WOON to EOS: New development methods require a new software process model; Bericht Nr. 12, Fachbereich Mathematik, Univ. Marburg; and: Proc. WOON ´96, 1st Int. Conf. on OO technology, St. Petersburg 1997
- [Hesse 97b] W. Hesse: Improving the software process guided by the EOS model. In: Proc. SPI '97 European Conference on Software Process Improvement. Barcelona 1997
- [Hesse, Weltz 94] W. Hesse, F. Weltz: Projektmanagement für evolutionäre Software-Entwicklung; Information Management 3/94, pp. 20-33, (1994)
- [Sarferaz, Hesse 00] S. Sarferaz, W. Hesse: CEOS – A Cost Estimation Method for Evolutionary, Object-Oriented Software Development . In.: R. Dumke, A. Abran (Eds.): New Approaches in Software Measurement. Proc. 10th Int. Workshop, IWSM 2000, Springer LNCS 2006, pp. 29-43

References

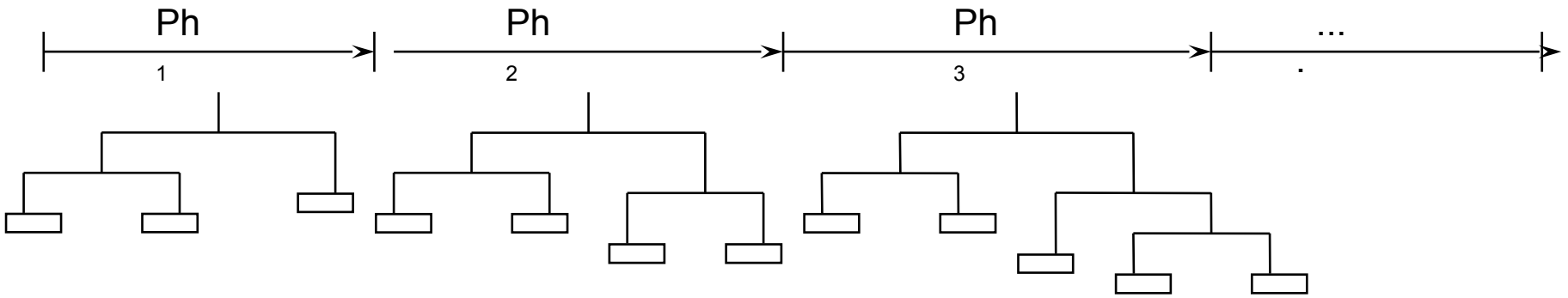
- [Beyer, Hesse 2002] Use of UML for software process modelling. Internal report, Univ. Marburg 2002
- [Bittner, Hesse, Schnath 95] U. Bittner, W. Hesse, J. Schnath: Praxis der Software-Entwicklung, Methoden, Werkzeuge, Projektmanagement - Eine Bestandsaufnahme, Oldenbourg 1995
- [Frese, Hesse 93] M. Frese, W. Hesse: The work situation in software development - Results of an empirical study, ACM SIGSOFT Software Engineering Notes, Vol. 18, No. 3, pp. A-65 - A-72 (1993)
- [Floyd, Reisin, Schmidt 89] Ch. Floyd, F.-M. Reisin, G. schmidt: STEPS to software development with users; in: C. Ghezzi, J. McDermid (eds.): ESEC '89, 2nd European Software Engineering Conference; LNCS 387, pp. 48-64, Springer 1989
- [Hesse, Merbeth, Frölich 92] W. Hesse, G. Merbeth, R. Frölich: Softwaretechnik - Vorgehensmodelle, Projektführung und Produktverwaltung, Handbuch der Informatik Bd. 5.2, Oldenbourg 1992
- [Hesse 96] W. Hesse: Theory and practice of the software process - a field study and its implications for project management; in: C. Montangero (ed.): Software Process Tech-nology, 5th Europ. Workshop EWSPT 96; Springer LNCS 1149, pp. 241-256 (1996)

14.3.3.1 The EOS Process Model

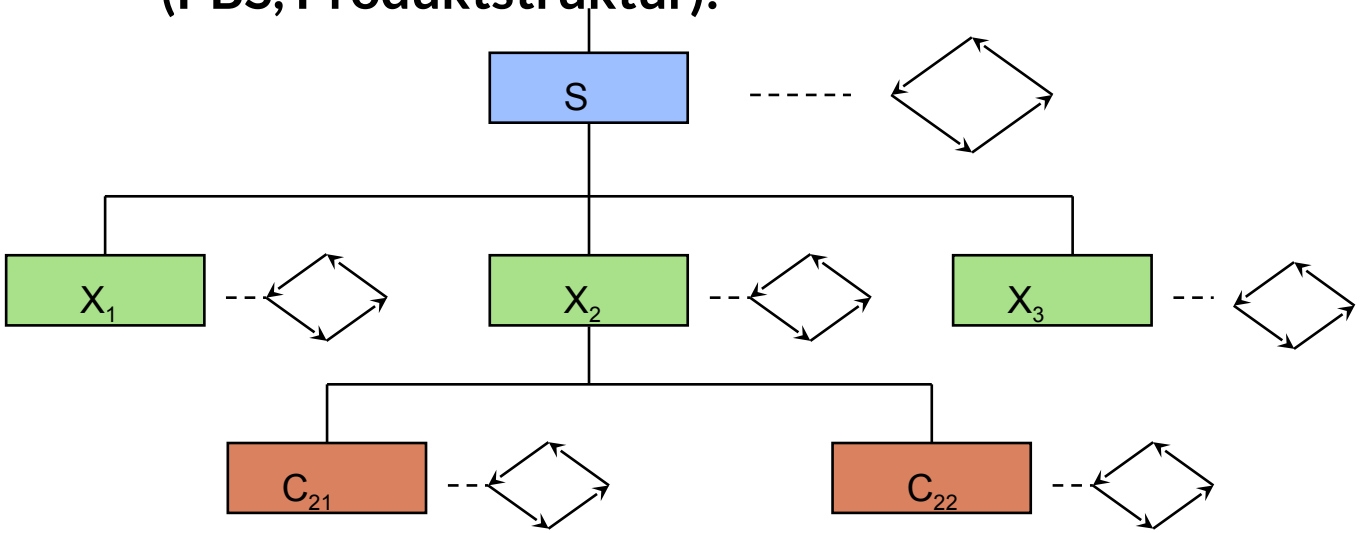
- Problem: Heavy-weight process models are often too bureaucratic and not scalable
  - The aspect of software *evolution* is hardly reflected
  - Planning relies on assumptions and may go wrong
  - *Unforeseen* discoveries, such as change of customer requirements, change the planning
- Component-oriented, distributed and web-based SW development requires flexible and well-adaptable processes
- EOS is a *product-structure-based process*, i.e., works if the architecture of the system is clear (standard architecture, standard architectural style, clear component hierarchy)
  - Well-known domain, low innovation project
  - It treats unforeseen dependencies between the components
  - Different availabilities of resources
  - Parallel work possible

Phase-oriented vs. component-oriented process

- Process in *linear* phases (Phasenmodell):

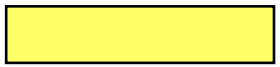


- EOS is a process *recursively* structured along **product breakdown strukture (PBS, Produktstruktur):**



- S System level
- X Subsystem level
- C Class level

Legend:



Building block

Phase or activity

Objects and features of the software process

- The **product breakdown structure (PBS, component hierarchy, Produktstruktur)** is a decomposition of the software product into components
- In EOS, it is assumed that the PBS is organised in a hierarchy with three level system development structure with three forms of components:
  - S System level
  - X Subsystem level
  - C Class level
- What are the features of those objects?
  - Attributes: Size, Responsible_person, Start_date_of_work, Delivery_date, ...
  - Operations: Development activities: Analysis, Design, Implementation, Operational_Use
  - State: active, interrupted, completed

Development Cycle of Components

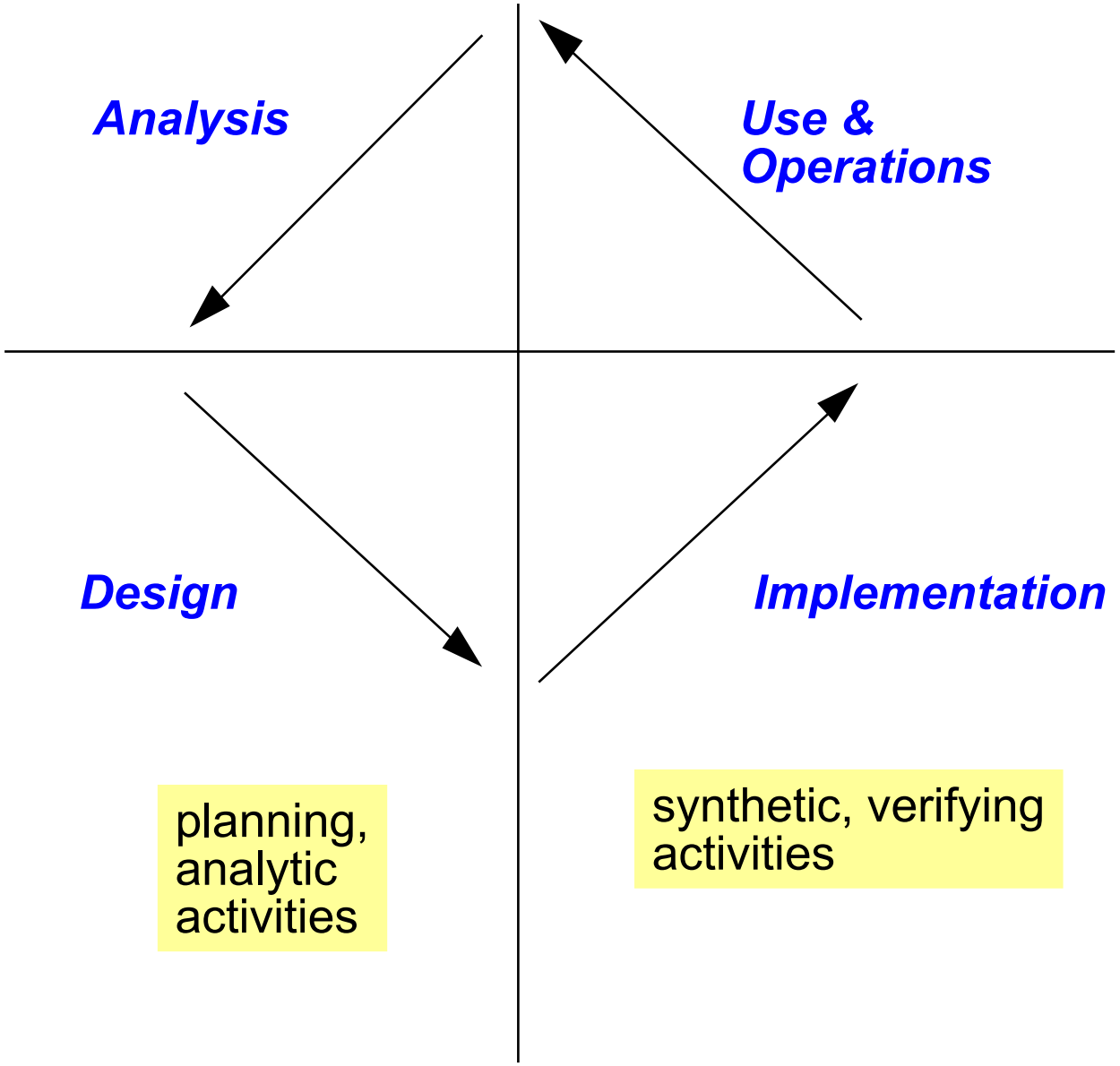
- Each development cycle, for every component on every level, has the *same structure* and consists of
  - *(.A) Analysis:* Define requirements, build model, consult building block (BB) library
  - *(.D) Design:* Specify and construct BB's
  - *(.I) Implementation:* Transform designed BB's to code, test, integrate
  - *(.O) Operational use:* installation, acceptance test, usage, revision

- *Evolutionary development* is supported by:
  - Integration of *operational use* (incl. "maintenance" and revision) into development cycles
  - Further development and re-use of components
  - Dynamic project planning and control based on cycles and activities

Phases of the EOS Micro-Increment Cycle

Use environment

Development environment

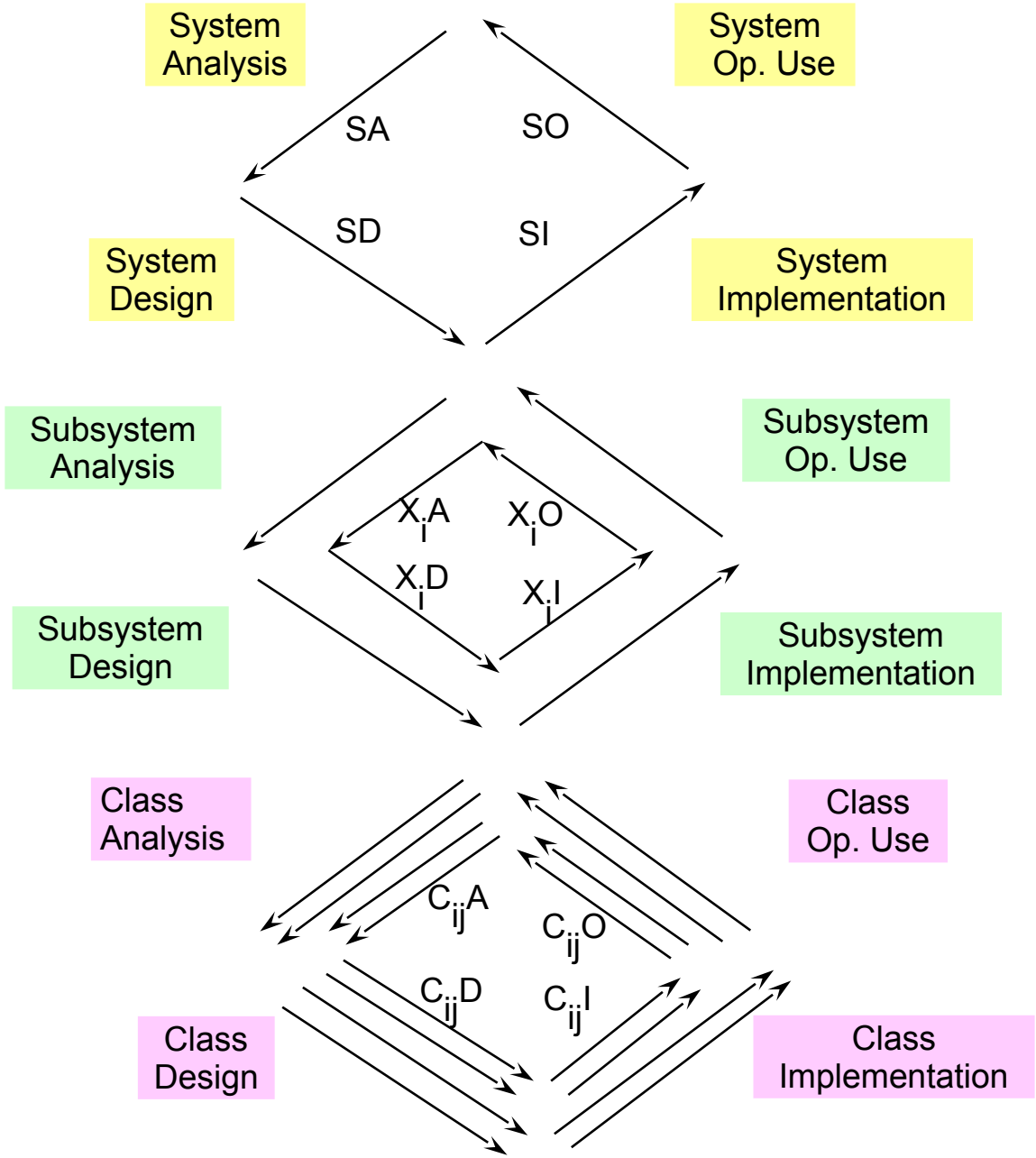


planning, analytic activities

synthetic, verifying activities

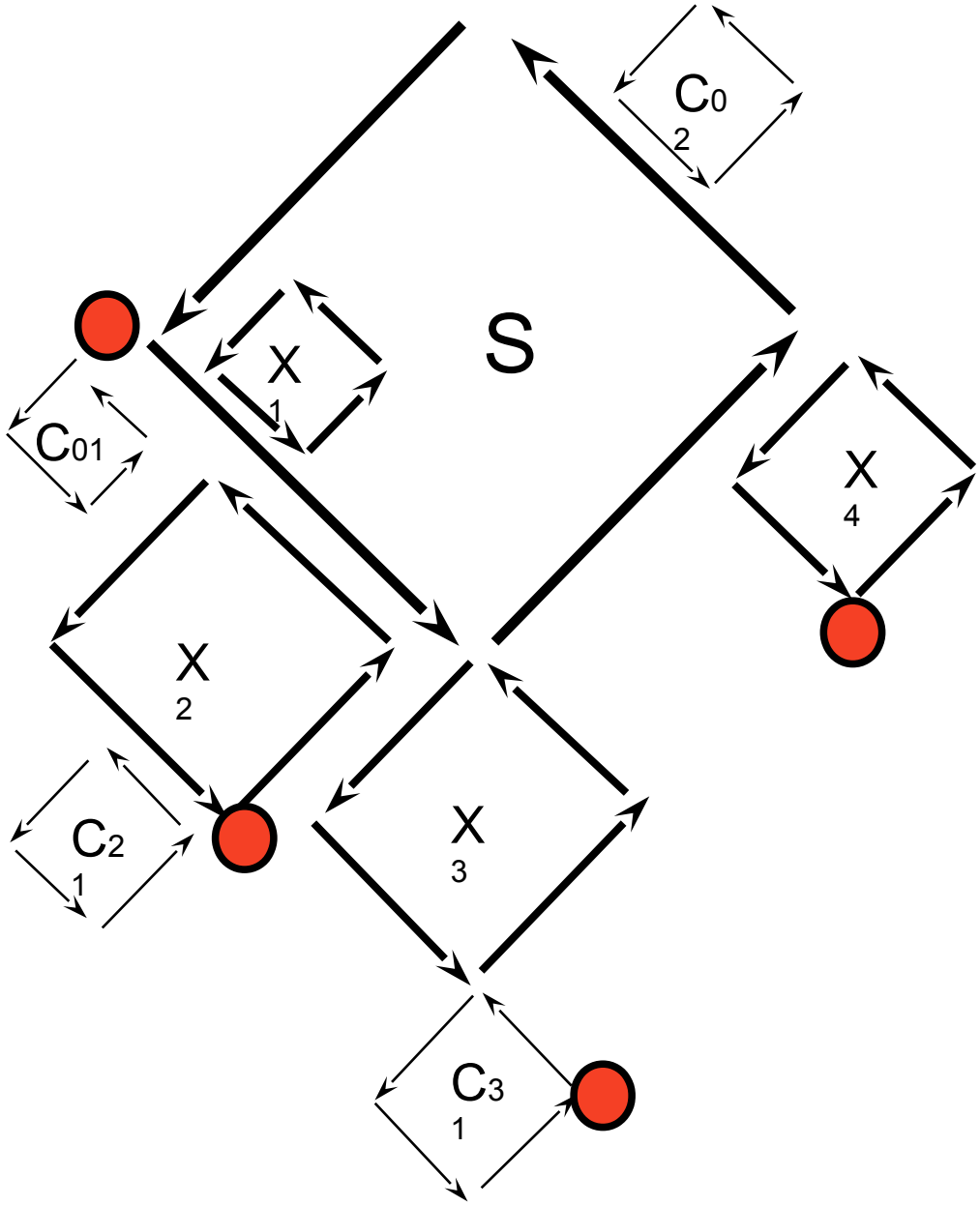
Combining development cycles in a traditional way

- Development phases for the components overlap
- System S has n subsystems X_i
- Subsystem X_i has m classes C_{ij}



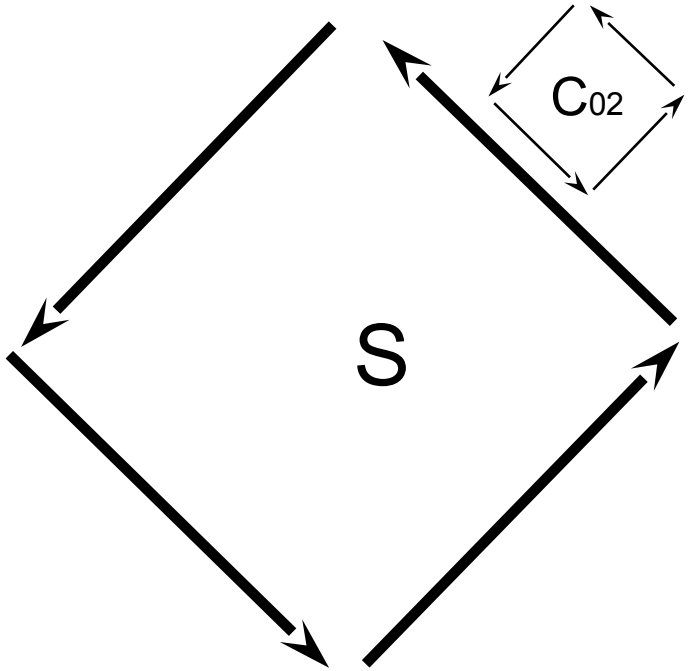
Typical EOS-like Process Structure

- EOS blends the phases
- k parallel development threads, resp. state tokens
- Development cycles intertwined in time
- If an obstacle appears, thread continues elsewhere
  - E.g., when dependencies to other components appear which were not known beforehand
- Parallel wavefront algorithm over the 3-level tree (bush)



Feature Extension During Evolution

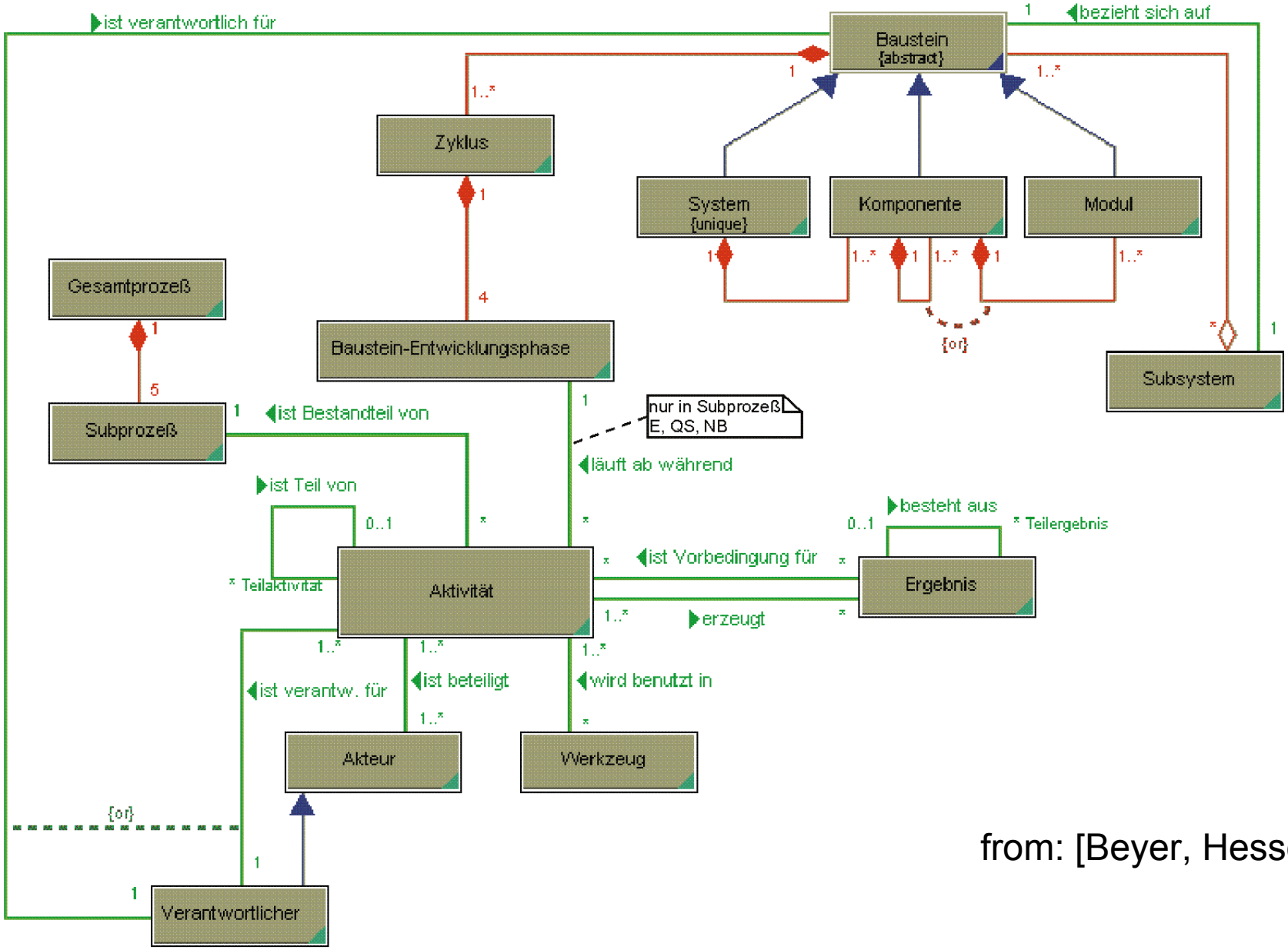
- With a new class C_{02}



EOS is Agile with Backlogs

- As in SCRUM, there is a backlog of prioritized next activities
- k backlogs with $l<k$ parallel development threads
- At the completion of an activity (small or large),
  - EOS allows for replanning and reprioritization of the activities to perform (agile development)
  - Costs can be estimated anew (agile cost estimation)
- Very flexible
- Customer can be involved, but need not

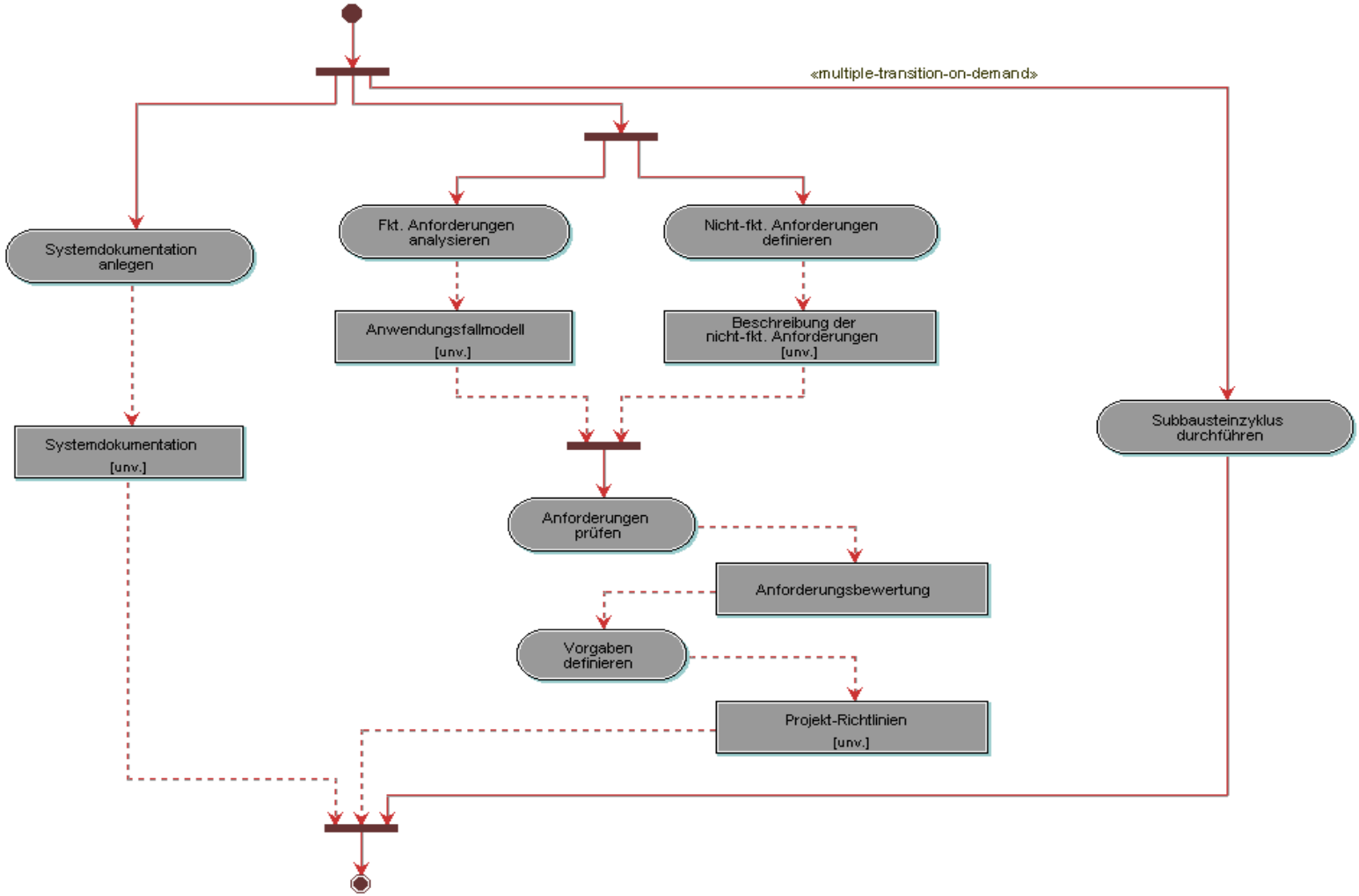
Metamodel for EOS process elements



from: [Beyer, Hesse 2002])

EOS is a Heterogeneous Process

- EOS allows for other process models in each of the four big phases
- Here: UML activity diagram for system analysis (SA) phase



14.3.3.2 Managing EOS Projects

Principles of Managing EOS Projects

- **Management structure follows system structure (PBS)**
  - Starting point: the EOS hierarchy levels
  - S-cycle: Global planning (project-wide)
  - X-cycles: Detailed steps (e.g. team work packages)
  - C-Cycles: Activities of single developers
- **Differenciated units of planning and control** (on each level)
  - 1st planning stage: plan the development cycle as a whole
  - 2nd planning stage: phases within cycle
- **Dynamic, situative planning (agile)**
  - Rather informal planning, "stand by"-management
  - Situation-driven adjustment of plans (backlogs)
  - Frequent plan revisions

Management principles (cont'd)

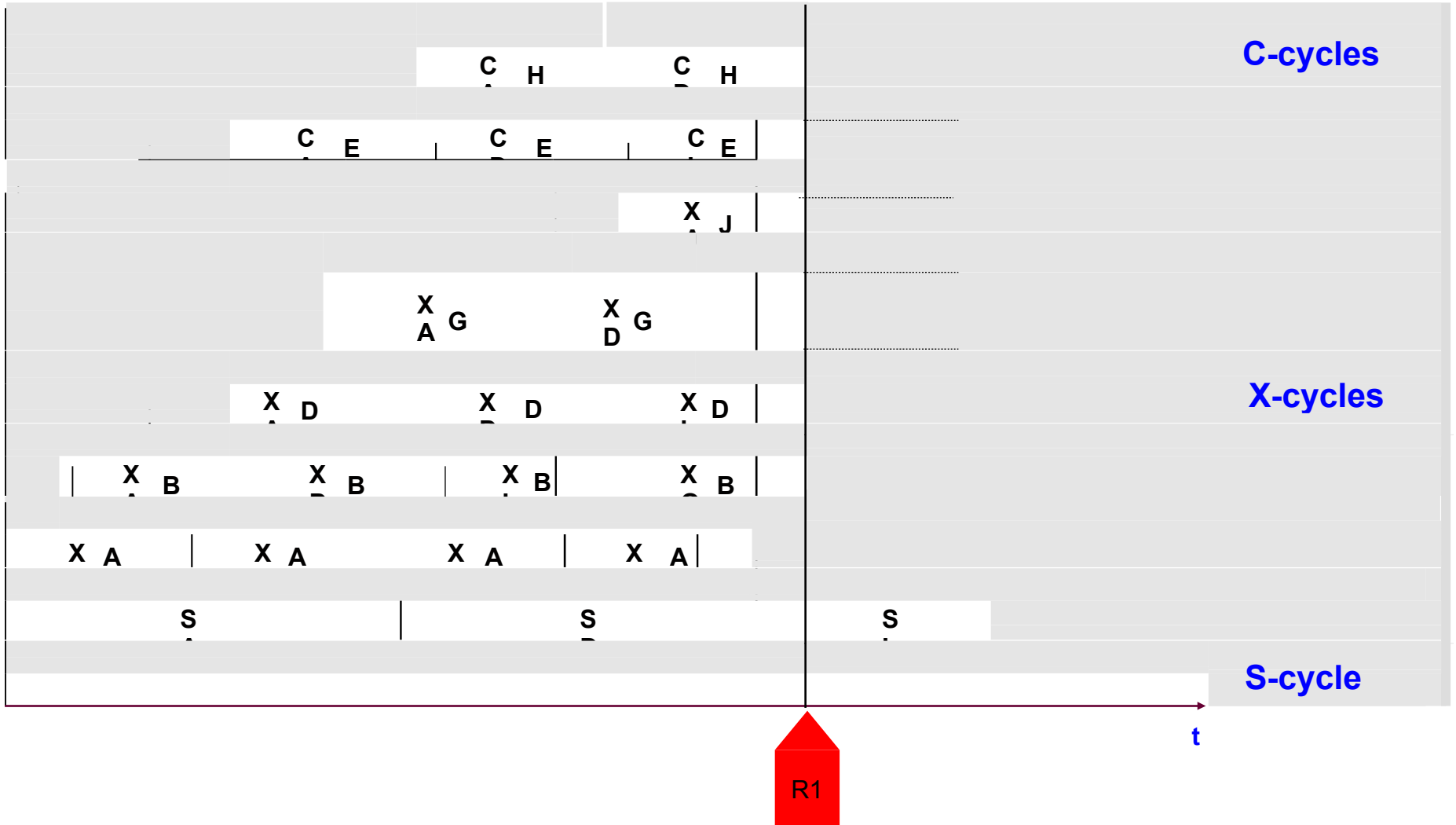
- “Object oriented” resp. “component-oriented” workpackages
  - Developers are primarily responsible for “objects” and “comnponents” - not for activities
    - Planning refers to objects rather than to activities:
    - on S- and X-level: by development (&support) teams (with users participating whereever necessary)
    - on C-level: by single developers or users
- Transparent planning, reliable plan control
  - Continuous information of teams on the project status
  - Plan revisions at defined points of time (revision points)
- Dynamic and adaptable cost and effort estimation
  - based on the EOS process structure, experience data and statistical regression methods [Sarferaz, Hesse 2000]

EOS and SCRUM

- The classical EOS is *not* time-boxed, but clearly structured along the PBS
  - If the PBS is stable, but it remains unclear, how long it takes to realize the activities, EOS is amenable
- **EOS can be combined with time-boxing (tEOS)**
  - k "component backlogs" for all components
  - k "sprint backlogs" for the current sprint in a component

EOS Revision Points

- A revision point is a special *milestone,* more differentiated and flexible, because lying between small or large activities
- Revision points allow for replanning and reestimation



Summary of EOS

- EOS combines the ideas of *evolutionary*, *agile*, *component-oriented*, and *object-oriented* software development
- The development process is structured along the PBS
  - by three hierarchy levels (system, component/subsystem, class)
  - by four phases (analyse, design, implement, operate)
- Cycles and phases are linked in a *systematic* and *orthogonal* manner
- Wavefront algorithm
- Development cycles are planned and executed *on demand* and in a *dynamic* way
- Project managers can plan and survey the project on every level of *detail* by means of *revision points*

The End

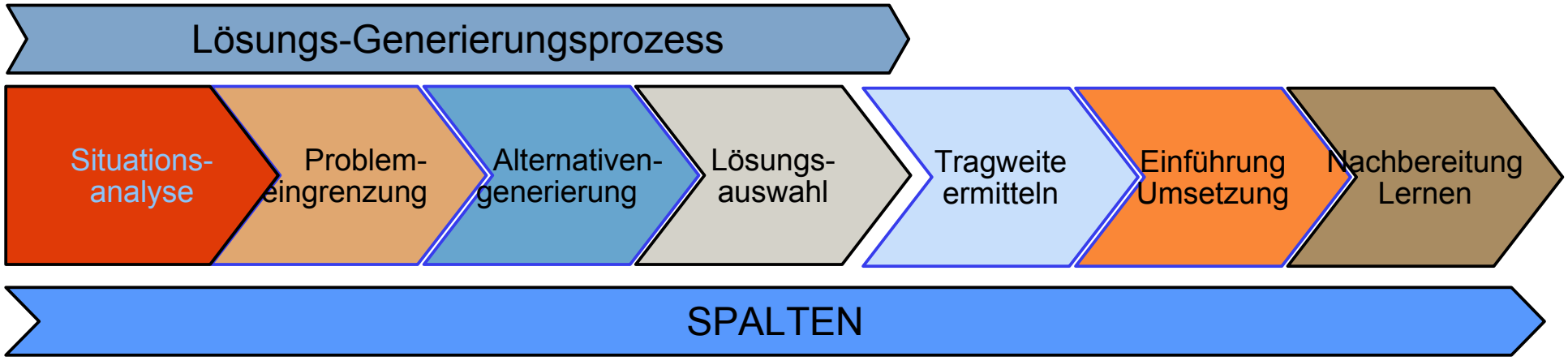
- Erklären Sie die grundlegenden Bestandteile des VMXT! Was sind Projekttypen? Was ist eine PDS?
  - Was sind die Kernmodule des VMXT?
  - Erklären sie das Metamodell des VMXT.
  - Wie funktioniert Tailoring?
- Wann würden Sie SCRUM, wann VMXT einsetzen?
- Welche Vorgehensmodelle sind für ein Projekt, in dem sicherheitskritische Software erstellt werden soll, geeignet?
- Warum ist Rückkopplung in einem Prozessmodell wichtig?
- Erklären Sie das Spiralmodell! das IneCT-Modell!
- Welche Vorteile haben Phasenmodelle?
- Vergleichen Sie EOS und SCRUM.
  - Wieso ist EOS ein gut geeigneter Prozess, wenn die Produktstruktur bekannt ist?

Der S.P.A.L.T.E.N. Prozess

- Der SPALTEN-Prozess ist ein allgemeiner Problemlöseprozess, bestehend aus einem Lösungs-Generierungsprozess und einem Realisationsprozess.

  Seine einzelnen Schritte sind: [Wikipedia/Problemlösen]

- Situationsanalyse (Ist-Analyse)
- Problemeingrenzung
- Alternativen aufzeigen (Lösungsgenerierung)
- Lösungsauswahl
- Tragweite analysieren - Chancen und Risiken abschätzen
- Einführung und Umsetzung - Maßnahmen und Prozesse
- Nachbearbeitung und Lernen



Appendix Spezielle betriebliche Vorgehen

- Große Firmen fassen ihre eigenen Vorgehensmodelle ab
  - Cap Gemini (SD+M) Quasar Enterprise www.openquasar.de
  - Accenture ADM

Glossary
Grundlegende Begriffe von Vorgehensmodellen

- **Worker/Rollen:** „wer“ - Beschreiben, wie sich Personen im Prozess verhalten sollen und welche Verantwortlichkeiten sie besitzen.
  - Ein Worker kann durch eine Person oder ein ganzes Team realisiert werden
  - Eine Person kann im Laufe des Lebenszyklus verschiedene Rollen einnehmen.
- **Aktivitäten:** „wie“ - Tätigkeiten, die ein Worker durchführen soll
  - Üblicherweise geht es dabei um die Erstellung oder Überarbeitung von Artefakten.
- **Arbeitsergebnisse/Produkte/Artefakte:** „was“ - Sind Informationsteile, welche durch einen Prozess erstellt, geändert oder genutzt werden.
  - Artefakte werden von Workern als Eingabe für Aktivitäten genutzt. Sie sind aber auch Ausgabe von Aktivitäten.
  - Weiterhin es es möglich, dass sich Artefakte aus anderen Artefakten zusammensetzen.
- **Arbeitsabläufe (Workflows):** „wann“ - Beschreiben aussagekräftig die Abfolge von Aktivitäten, damit es zu einem sinnvollen Ergebnis kommt.
  - Außerdem werden die Interaktionen zwischen den Workern aufgezeigt.