

# 14 Ablauforganisation

Prof. Dr. rer. nat. Uwe Aßmann  
Lehrstuhl Softwaretechnologie  
Fakultät Informatik  
TU Dresden  
Version 11-0.2, 21.04.11

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

Softwaremanagement, © Prof. Uwe Aßmann

1

# Literatur

- ▶ [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
- ▶ [Lippert] Lippert, M., Roock, S., Wolf, H.: Software entwickeln mit eXtreme Programming – Erfahrungen aus der Praxis; dpunkt.verlag 2002
- ▶ [Hruschka] Hruschka, P., Rupp, Ch.: Agile Softwareentwicklung für Embedded Real-Time Systems mit der UML; Hanser Verlag 2002

2

# Ablauforganisation

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

- ▶ Arbeitsabläufe dienen dazu,
  - den Ablauf von Arbeitsvorgängen/Prozessen innerhalb eines Projekts zu organisieren
  - Verantwortlichkeiten innerhalb der Vorgänge zu formalisieren
  - zu rationalisieren, vereinfachen und standardisieren
- ▶ Methodiken des Vorgehens sind *Prozessmodelle*, die das Vorgehen schematisch vorgeben

3

# Prozessmodelle als Mittel der Ablauforganisation

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

- ▶ **Phasenmodelle:** Phasen sind erkennbar
  - Spiralmodell
  - V-Modell (nach Boehm)
- ▶ **Vorgehensmodelle:** Ohne Phasen, aber aus Schablonen/Mustern zusammengesetzt
  - V-Modell des Bundes oder RUP
  - Oft zyklisch:
    - Modelle zur inkrementellen SWE
    - Evolutionäre Modelle
    - Prototypisches Vorgehen

4

## 14.1 Prozessmodelle

5

### 14.1.1 Phasenmodelle

zunächst betrachtet: Phasenmodelle

Softwaremanagement, © Prof. Uwe Aßmann

6

## Phasenorganisation

7

Eine **Projektphase** ist ein zeitlicher Abschnitt in einem Projektablauf, der sachlich getrennt von einem anderen abläuft. Die Projektphase wird durch eine Vernehmlassung 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 trägt zur Reduzierung des technischen wirtschaftlichen und terminlichen Risikos bei durch:
  - schrittweises Vorgehen
  - Vorgabe und Überwachung von Zwischenergebnissen (Meilensteinen)
  - Transparenz über den Projektstand (Projektkontrolle, -regelung)

## Vorteile der Projektphasen

8

- ▶ Phasenweises Vorgehen
  - hilft, den **Überblick** zu behalten und sich nicht im Detail zu verlieren
  - zwingt zur **periodischen Stellungnahme**
  - hilft bei der Herstellung einer **fortlaufenden Dokumentation** zu den Phasen; kann durch Checklisten sichergestellt werden
- ▶ Verringerung des Risikos einer Fehlentwicklung durch **bessere Überwachung** und Steuerung innerhalb einer abgegrenzten Phase
  - Der Inhalt und damit das Ende jeder Phase ist definiert und bekannt
- ▶ **Entscheidungsfreiheit** bleibt gewahrt durch Beeinflussung der Entwicklung an den vordefinierten Ergebniszeitpunkten
  - Aufgrund der klaren Arbeitszuweisung ist der Entwickler(Auftragnehmer) für die Qualität und Funktionalität seiner Arbeit gegenüber dem Auftraggeber voll **verantwortlich**

[ 1, S. 68 ]

# Meilensteine in der Planung von Phasenorganisation

- ▶ Meilensteine sind typisch für die Planung einer Phasenorganisation
  - sind Eckpunkte der Planung und Durchführung
  - bezeichnen definierte Sachergebnisse zu einem Termin
  - führen zu Entscheidungen (sind Entscheidungspunkte)
- ▶ Meilenstein-Inhalte sind
  - SMART, insb. überprüfbar (z. B. Voraussetzung für nächste Phase)
  - Meilensteinüberprüfung im Controlling durch Meilenstein-Trendanalyse
  - CCC
  - übergebbar (z. B. in die Produktbibliothek, dem AG, ...)

9

# Reihenfolge der Meilensteine (Beispiel)

(Zahl und Inhalt je nach Phasenmodell)

<b>MS 1:</b> Projektziele/ Pflichtenheft ⇨ <u>Anforderungskatalog</u>	<b>Was soll erreicht werden?</b>
<b>MS 2:</b> Fachliches Konzept ⇨ <u>Leistungsbeschreibung</u>	<b>Wie sieht die Lösung aus?</b>
<b>MS 3:</b> Technisches Konzept ⇨ <u>Design-Spezifikation</u>	<b>Wie wird die Lösung technisch realisiert?</b>
<b>MS 4:</b> Realisierung ⇨ <u>Komponenten</u>	<b>Erstellung der Komponenten</b>
<b>MS 5:</b> Integration ⇨ <u>System</u>	<b>Arbeiten die Komponenten zusammen?</b>
<b>MS 6:</b> Getestetes System ⇨ <u>Produkt</u>	<b>Werden die Funktionen erfüllt?</b>
<b>MS 7:</b> Eingeführtes System	<b>Einführung und Anwendung des Systems</b>

10

# Meilenstein-Inhalte (Beispiel)

## **MS1: Pflichtenheft**

- priorisierter Requirementskatalog (realisierbare Funktionen, Qualität)
- Test-/ Prüfanforderungen
- Manualgrobplan (Umfang, Bearbeiter, Termine)
- Personal, Kosten, Termine
- Dokumentenverwaltung eingerichtet
- Konfigurationsmanagement eingerichtet

## **MS2: Leistungsbeschreibung**

- Feinentwurf der log. Funktionen, Daten, Schnittstellen
- DV-Grobentwurf (mit Lösungsalternativen)
- Festlegung der Q.-eigenschaften, der Dialogführung, -Masken
- Festlegung des Liefer-, Wartungs- und Pflegekonzepts
- Test- und Prüfkonzept (mit Testkonfiguration)
- Manualfeinplan (Struktur, Layout, Herstellung, ...)
- vorläufiger Projekt- und QS-Plan
- Vorkalkulation

11

# Meilensteinüberprüfung / -entscheidung beim Übergang zur nächsten Phase

## **Beteiligte:**

- Projektleiter
- Entwicklungsverantwortlicher
- Qualitätssicherung
- Wirtschaftliches Controlling
- Auftraggeber

⇨ Jeder Meilenstein wird durch eine explizite Entscheidung abgeschlossen (**Entscheidungspunkt**).

⇨ **Entscheidung** über

- Technische Alternativen
- Freigabe der Ergebnisse
- Fortführung des Projektes unter Abwägung des Risikos

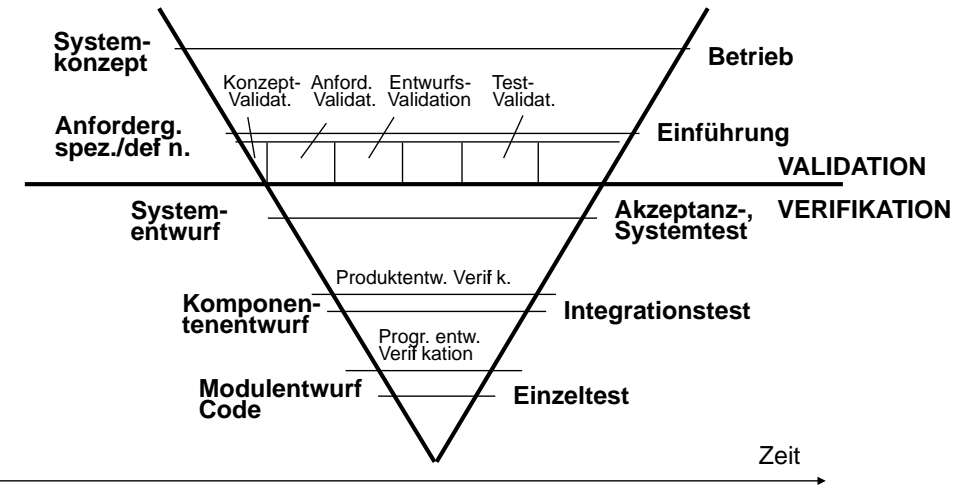
⇨ Der Projektleiter stützt sich bei seiner Entscheidung auf die Ergebnisse der QS und ggf. des wirtschaftlichen Controllings

⇨ Die Entscheidung ist für alle am Projekt Beteiligten verbindlich.

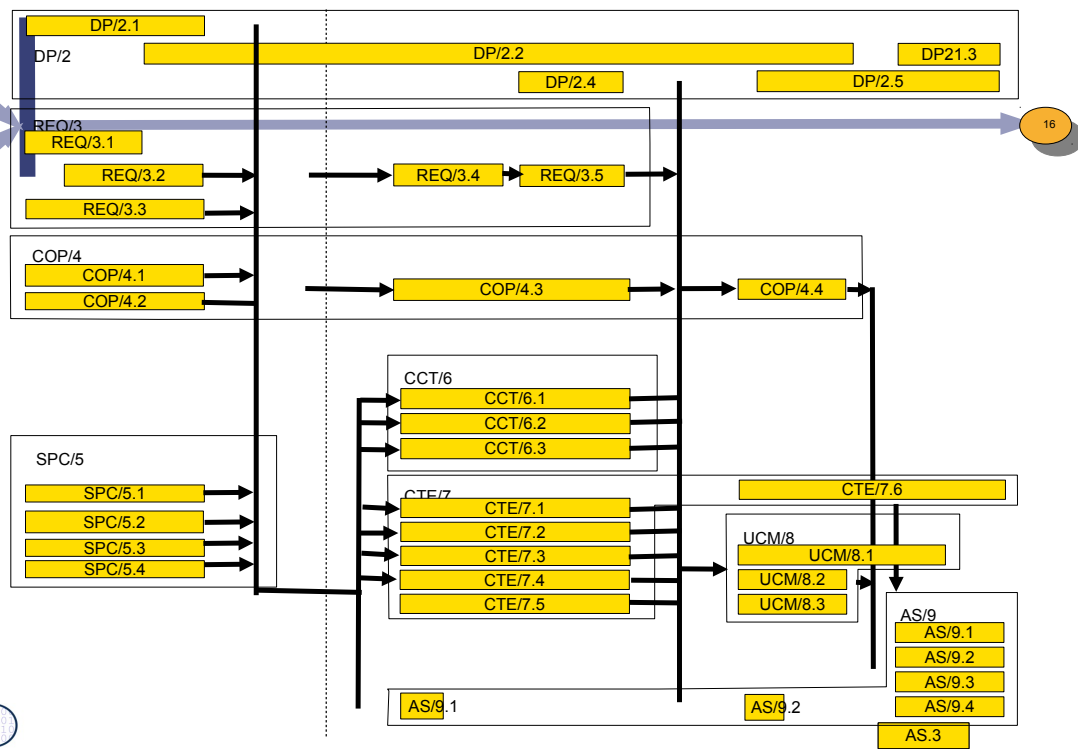
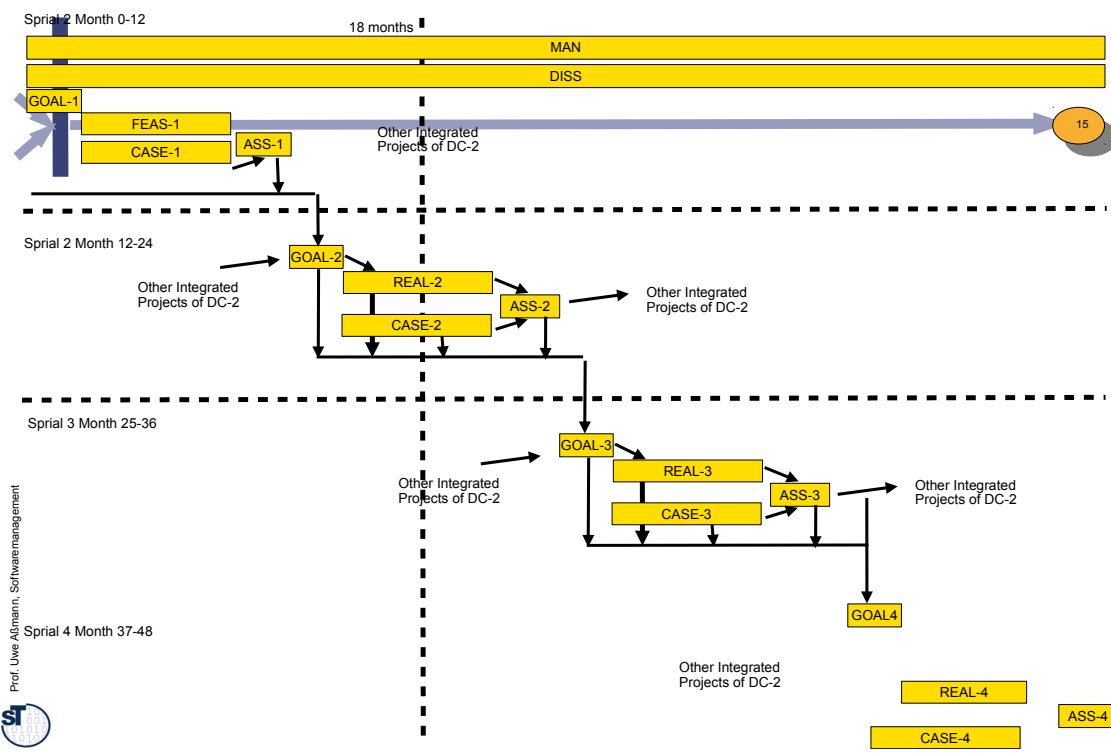
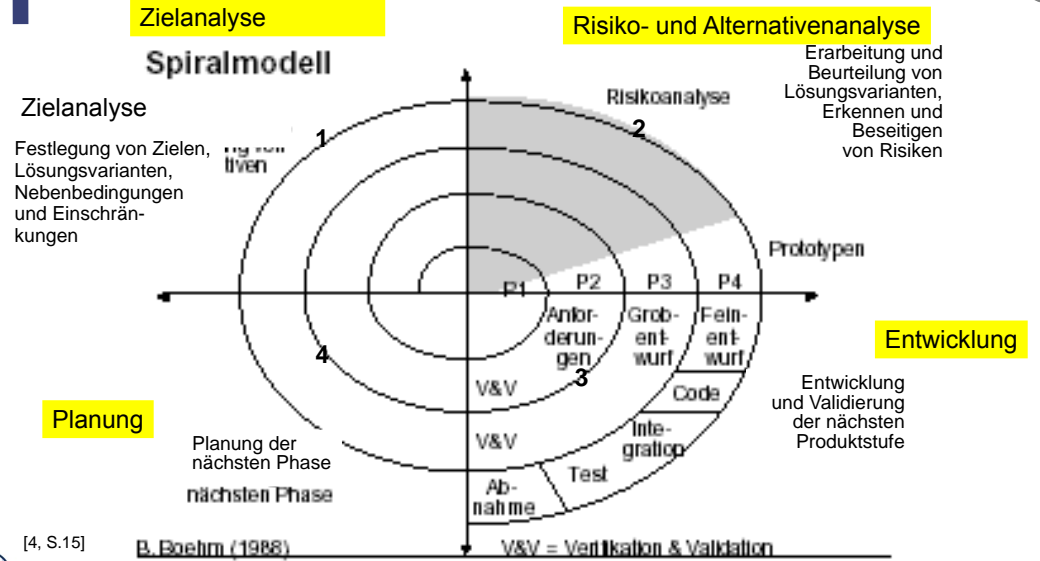
12



# V-Modell (nach B. Boehm)



# Evolutionärer Lebenszyklus nach Boehm (Spiralmodell)



# 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
- Integrierte Risikoanalyse
- Qualitätssicherungsmaßnahmen gut integrierbar
- Auch für Wartungsprojekte geeignet

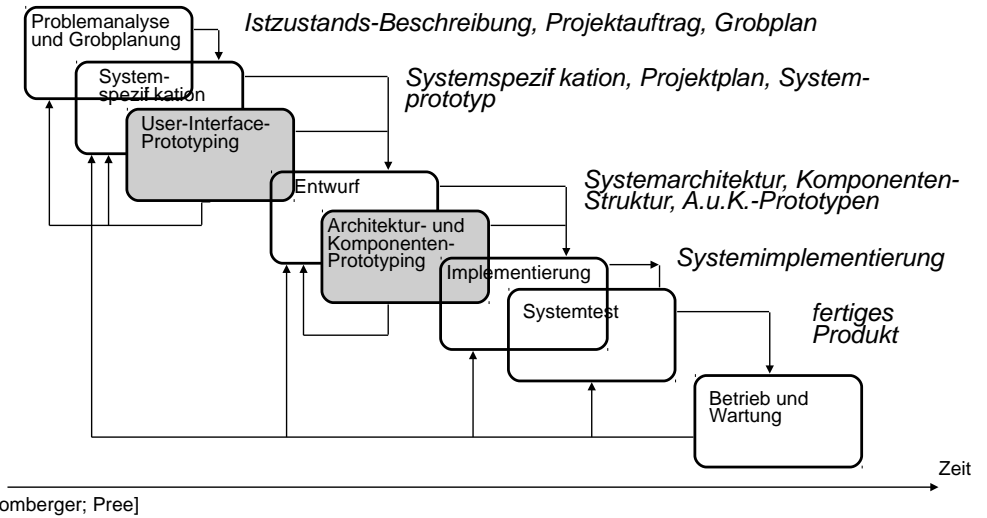
## Nachteile:

- Hoher Managementaufwand
- Geringe Eignung für kleinere Projekte

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

# Prototyping-orientiertes Phasenmodell

- Verfeinerung des iterativen Wasserfall-Modells oder des V-Modells
- Teile eines Produktes grob entwickeln, so, dass sie **ausführbar** werden

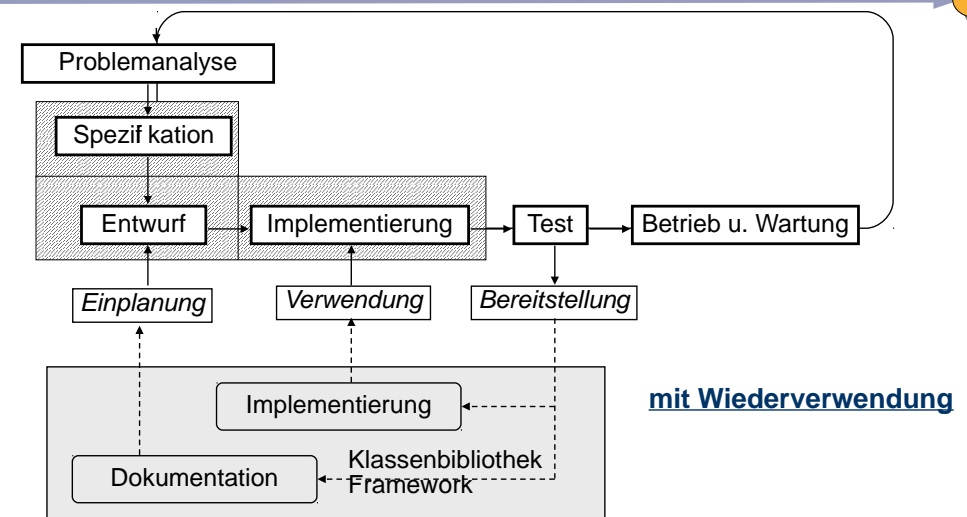


[Pomberger; Pree]

# Prototyporientiertes Vorgehen

- Prototyping ist eine topdown-orientierte Vorgehensmethode, die schneller realitätsnahe Funktionstests ermöglicht
  - Das zugrundeliegende Modell ist ein iteratives (Spiral-)Modell
  - Es wird so früh wie möglich, nicht so spät wie möglich implementiert ("release-often")
- Folgen für Produkt:
  - Frühe **Vervollständigung** der Zielerforschung, Veranschaulichung des Design und frühzeitige Überprüfung der Implementierung
  - Frühzeitige **Fehlererkennung** erleichtert die Qualitätssicherung.
  - Qualitätsmerkmale** Benutzerfreundlichkeit, funktionale Adäquatheit, Änder- und Erweiterbarkeit, Korrektheit und Zuverlässigkeit werden gesteigert.
- Folgen für Stakeholder:
  - Größere Sicherheit** von Anforderungsanalyse und Entwurf ist Basis für eine bessere Planbarkeit, Aufwandsabschätzung und Termintreue.
  - Durch unmittelbare Beteiligung der Anwender erhöht sich die **Akzeptanz** des späteren Produktes
  - Die frühzeitige Auseinandersetzung mit dem Prototypen führt zu einer vorgezogenen **Einsatzvorbereitung**, während der die Benutzer geschult und trainiert werden

# Objektorientiertes Phasenmodell



nach [Pomberger, Pree]

## 14.2 Vorgehensmodelle

## Vorgehensmodellierung

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

Ziele:

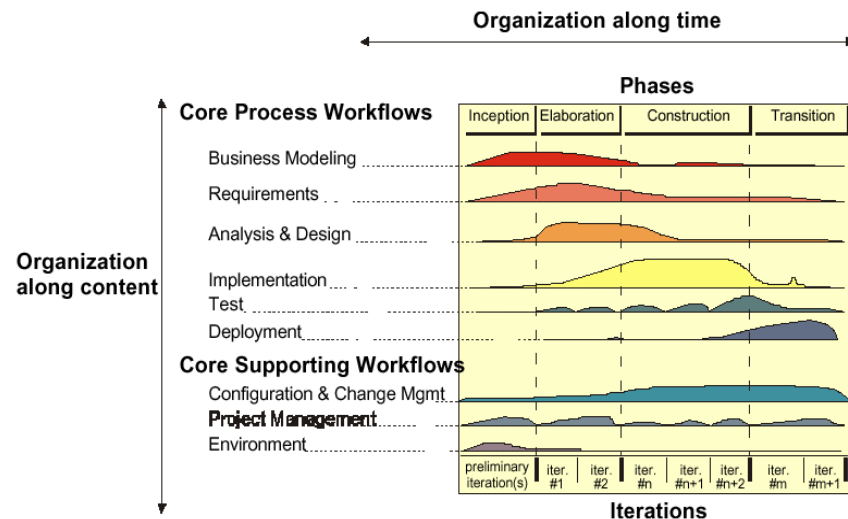
- **Modellierung als System** ineinandergreifender Bestandteile soll sicherstellen, dass die vom Kunden gewünschte Leistung **planmäßig** in hervorragender Qualität erbracht wird
- **Modellierung in ihrer Gesamtheit** aus Aktivitäten, Artefakten (Produkten), Zuständen und Rollen und Ressourcen
- **Modellierung des Produktfluss (Workflow)** unter Einschluss der beteiligten Rollen zur Entwicklung begleitende Tätigkeiten, wie Qualitätsmanagement (QM, QS), Konfigurationsmanagement (KM), Projektmanagement (PM)
- **Modellierung der inkrementellen Softwareentwicklung** durch stufenweises Vorgehen auf der Basis insgesamt gefestigter Anwenderforderungen
- organisationsneutrale Anwendung und Anpassung an spezielle Anwendungsfälle (**Tailorisierung**)

## Eigenschaften des Unified Process

### 14.2.1 (Rational) Unified Process

- ▶ Durch Anwendungsfälle gesteuert
  - Vom Lastenheft ausgehend wird das gesamte Vorgehen durch die Ableitung von Anwendungsfällen (use cases) gesteuert.
    - 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.

## Rational Unified Process (1)



## Rational Unified Process (2)

Phasen treten in jedem Arbeitsschritt auf:

- ▶ **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.
- ▶ Alle Arbeitsschritte werden in jedem Zyklus durchlaufen

## Grundlegende Elemente des RUP

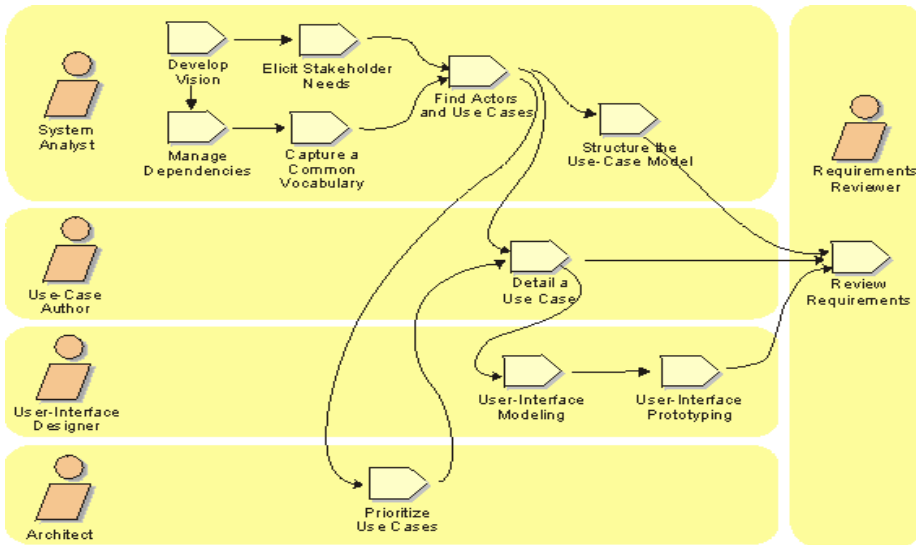
- ▶ **Worker/Rollen:** „wer“ - Beschreiben, wie sich Personen im Prozess verhalten sollen und welche Verantwortlichkeiten sie besitzen.
  - Ein Worker kann durch eine Person oder ein ganzes Team realisiert werden
  - Eine Person kann im Laufe des Lebenszyklus verschiedene Rollen einnehmen.
- ▶ **Aktivitäten:** „wie“ - Sind Teile von Tätigkeiten, die ein Worker durchführen soll
  - Üblicherweise geht es dabei um die Erstellung oder Überarbeitung von Artefakten.
- ▶ **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.
- ▶ **Workflows:** „wann“ - Beschreiben aussagekräftig die Abfolge von Aktivitäten, damit es zu einem sinnvollen Ergebnis kommt.
  - Außerdem werden die Interaktionen zwischen den Workern aufgezeigt.

## Supporting Workflows

Neben den Modellierungselementen (s. vorher. Folie) sind zu beschreiben

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

# Bsp: RUP Workflow Requirements Analysis



# 14.2.2 V-Modell des Bundes

## Historie

1986	Das Bundesministerium der Verteidigung gibt ein Vorgehensmodell für die Softwareentwicklung auf Basis des V-Modells in Auftrag Erstes Modell <b>V-Modell 92</b> vom Bundesamt für Wehrtechnik und Beschaffung (BWB) zusammen mit einem CASE-Tool-Hersteller entwickelt
1992	Erste Version wurde für die Bundesbehörden des Verteidigungsministeriums verbindlich Es folgten zahlreiche Änderungs- und Verbesserungsvorschläge
1994	Es wird die Nutzergemeinschaft ANSSTAND e.V. (Anwender des Softwareentwicklungsstandards der öffentlichen Verwaltung) gebildet, die regelmäßig Erfahrungsaustausche organisiert.
1997	Veröffentlichung des verbesserten <b>V-Modell'97</b> (mit Objektorientierung) als <b>IT-Standard der Bundes</b> Es wurde weiter an der Verbesserung gearbeitet (Schwerpunkt Erweiterbarkeit)
August 2004	Vorstellung des V-Modell XT
Februar 2005	Veröffentlichung des V-Modell XT

## V-Modell

Entwicklungsstandard für IT-Systeme des Bundes (*Vorgehensmodell*)

- vom Interministeriellen Koordinierungsausschuss (IMKA) am 05.06.96 zur Anw. empfohlen
- **Standardisierung** der
  - **Vorgehensweise**
  - anzuwendenden **Methoden**
  - funktionalen **Anforderungen an Werkzeuge**
- **Ziele:**
  - Verbesserung und Gewährleistung der **SW-Qualität** durch
    - standardisiertes Vorgehen ==> Vollständigkeit der Ergebnisse
    - hierarchische Erzeugnisstruktur ==> von System- über Segment-Ebene ==> SW-(od. HW-)Einheit ==> SW-Komponente ==> SW-(od. HW-)Modul
    - standardisierte Zwischenergebnisse ==> frühzeitige Prüfungen
    - einheitliche Produktinhalte ==> bessere Lesbarkeit und Prüfungen
    - vorgeschriebene Dokumente oder Produktmuster mit Dokumentendurchlauf durch Aktivitäten
  - Eindämmung der **SW-Kosten** über den Life-Cycle durch
    - standardisiertes Vorgehen ==> Kalkulation leichter, Risiko geringer
    - bessere Verständigung zwischen AG und AN
    - universelle Lösungsvarianten leichter erkennbar

Quelle: [ 14 ]



## Ziele des V-Modell XT

### ► Prozessverbesserung:

- 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

Es besteht aus einzelnen Vorgehensbausteinen (Moduln), welche Aktivitäten, Produkte und Rollen in sich vereinen. Sie beschreiben mittels der Aktivitäten und der zugeordneten Rollen, welche Produkte zu erstellen sind.

Das Modell beinhaltet darüber hinaus Projektdurchführungsstrategien, welche Entscheidungspunkte beinhalten, die den Weg der Durchführung eines Projektes beschreiben.

[www.v-modell.iabg.de]

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

## Projekttypen

### Systementwicklungsprojekt

#### 1) eines Auftraggebers

Der Projekttyp vereinigt die Projektdurchführungsstrategie des Auftraggebers

#### 2) eines Auftragnehmers

Der Projekttyp vereinigt die Projektdurchführungsstrategie des Auftragnehmers

#### 3) eines Auftragnehmers mit dem Auftraggeber

Projekt ist in der gleichen Organisation oder in mehreren Organisationen, die bewusst miteinander arbeiten

#### 4) 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 Projekttypvarianten legen die Projektdurchführungsstrategie fest

Beispiel:

Projekttyp: Systementwicklungsprojekt (AG)

Projekttypvarianten

- AG-Projekt mit einem Auftragnehmer
- AG-Projekt mit mehreren Auftragnehmern

# Projekt-Durchführungsstrategien(PDS)

## PDS

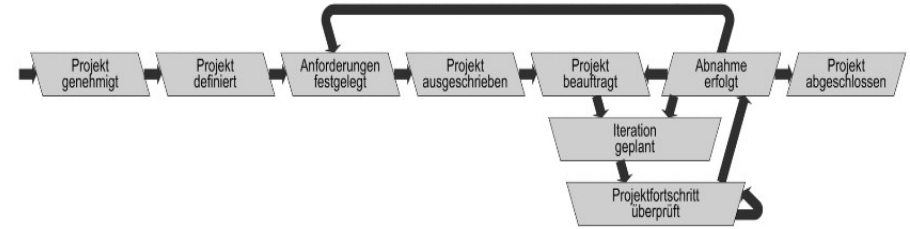
- ▶ Repräsentieren einen geordneten Projektablauf (inhaltlich, zeitlich)
  - Festlegung der Reihenfolge, in der die für das Projekt relevanten **Entscheidungspunkte** durchlaufen werden müssen
  - Unterteilung des Projektes in einzelne Abschnitte
- ▶ Eine Projekttypvariante legt die PDS fest

## Entscheidungspunkte (EP)

- ▶ dienen der Strukturierung im Projekt
  - kennzeichnen Meilensteine im Projekt
  - befinden sich am Ende eines Projekt-Abschnitts
- ▶ an ihnen wird der Stand des Projekts evaluiert
  - Basis der Bewertung sind: vorzulegende Produkte
- ▶ dienen der Entscheidung über weiteren Projektablauf
  - Bestimmung von Korrekturmaßnahmen
  - Freigabe des nächsten Abschnitts
  - Entscheid über den Abbruch des Projekts

# Beispiele für PDS (1): Systementwicklungsprojekt eines Auftraggebers

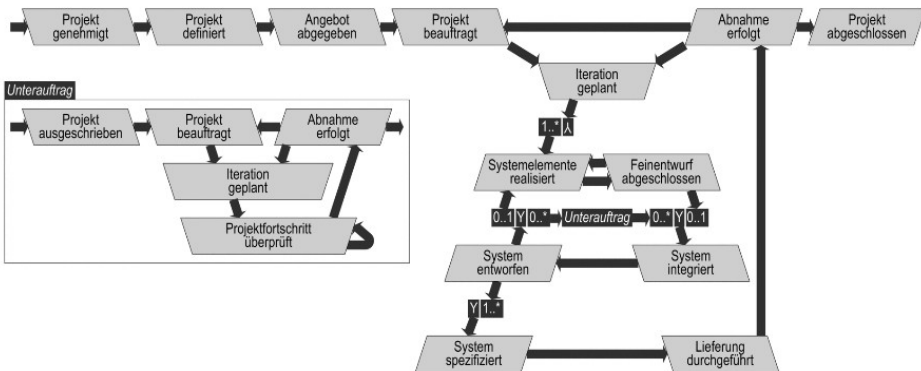
- ▶ Vergabe und Durchführung von Systementwicklungsprojekten (Schritte des Auftraggebers während des Projektes – System wird nicht selbst entwickelt) mit Abfolge der zugehörigen Entscheidungspunkte



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

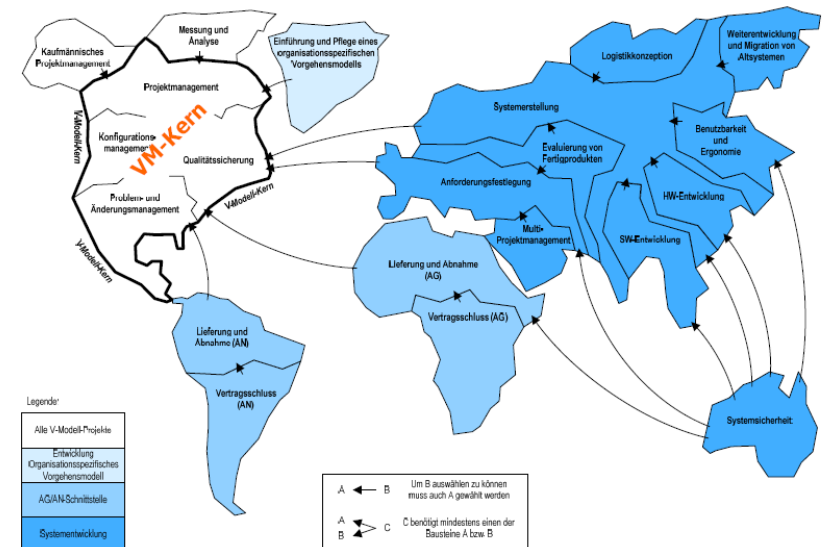
# Beispiele für PDS (2): Agile Systementwicklung

- ▶ 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>

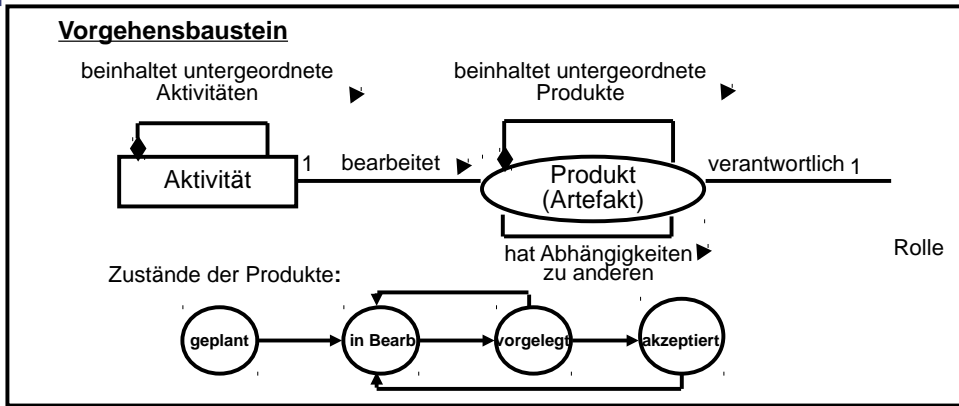
# Landkarte der Vorgehensbausteine



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

# Grundelemente des V-Modells XT

■ Vorgehensbausteine sind die modularen Bausteine, aus denen das V-Modell XT aufgebaut ist



■ Ein Vorgehensbaustein

- kapselt inhaltlich zusammengehörende Aktivitäten, Produkte, Rollen und Zustände
- ist eine Einheit, die eigenständig verwendet werden kann
- ist eine Einheit, die unabhängig veränder- und weiterentwickelbar ist

# Produktgruppen

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

Beispiele dafür wären:

**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

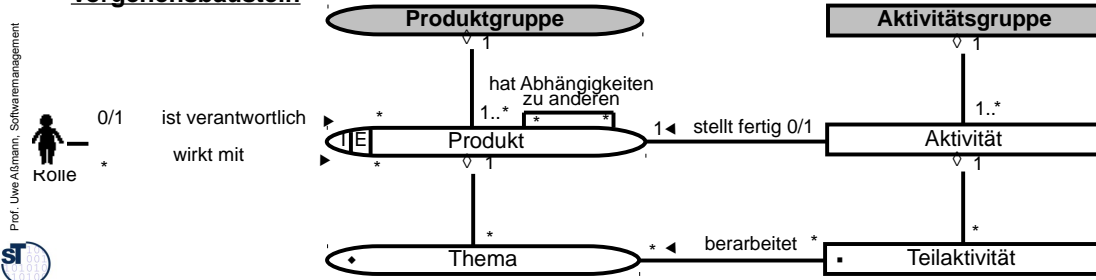
**Bereich Organisation** – 1 Produktgruppe

*Prozessverbesserung* wird für die Einführung und Pflege eines spezifischen Vorgehensmodell verwendet

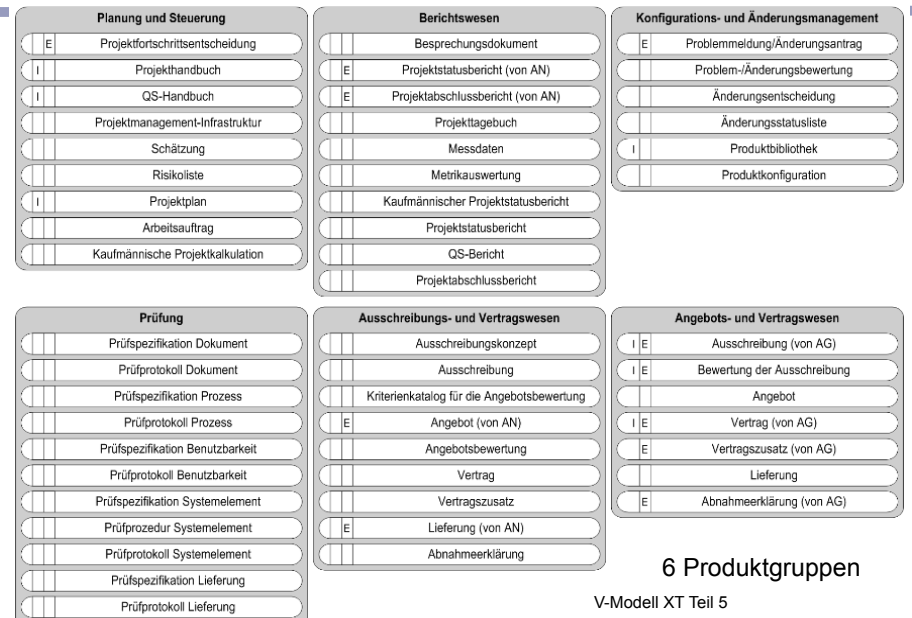
# Vorgehensbausteine und ihre Bestandteile

- ▶ Eine **Produktgruppe** ist eine Sammlung von einzelnen Produkten, welche einem Vorgehensbaustein zugeordnet sind.
- ▶ **Produkte** 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.
- ▶ **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

## Vorgehensbaustein



# Bereich Projekt



6 Produktgruppen

V-Modell XT Teil 5

# Rollen

Jedem Produkt wird eine eindeutige Rolle zugewiesen, der eine Person im Projekt entspricht. Eine Person kann auch mehrere Rollen einnehmen. Jede Rolle ist durch eine festgesetzte Struktur bestimmt, die besteht aus der Beschreibung, den Aufgaben und Befugnissen, dem Fähigkeitsprofil, der Verantwortung oder Mitwirkung. Beispiele für Rollen sind:

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

# Aktivität

In einer Aktivität wird festgelegt, welches Produkt von wem, wie erstellt wird.

## Aktivität:

Eine Aktivität wird allgemein als Aktivitätstyp verstanden. Sie können in Aktivitätsgruppen zusammengefasst werden.

## Aktivitätsstruktur:

Menge aller Aktivitätsexemplare eines Projekts und deren Beziehungen zueinander.

Die Aktivitätsgruppen können analog den Produktgruppen den Bereichen

- Projekt,
- Entwicklung
- Organisation

zugeordnet werden.

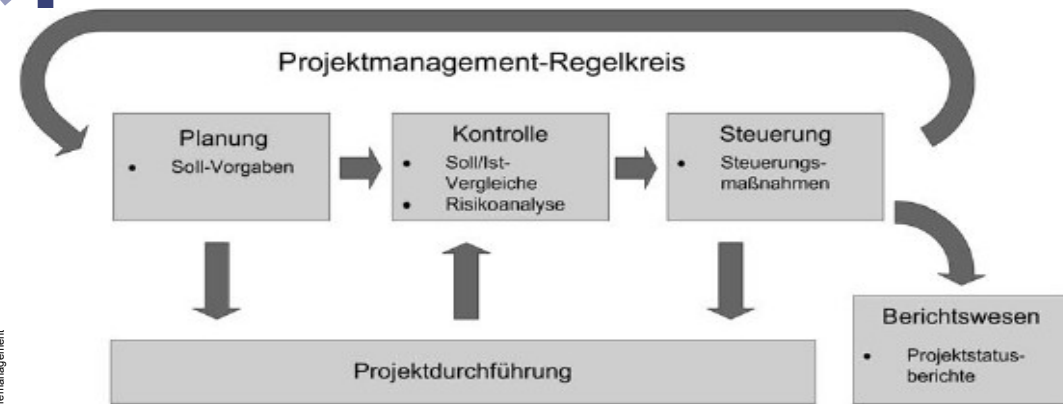
# Aktivitätsgruppen im Bereich Projekt



6 Aktivitätsgruppen

[V-Modell XT – Teil 6]

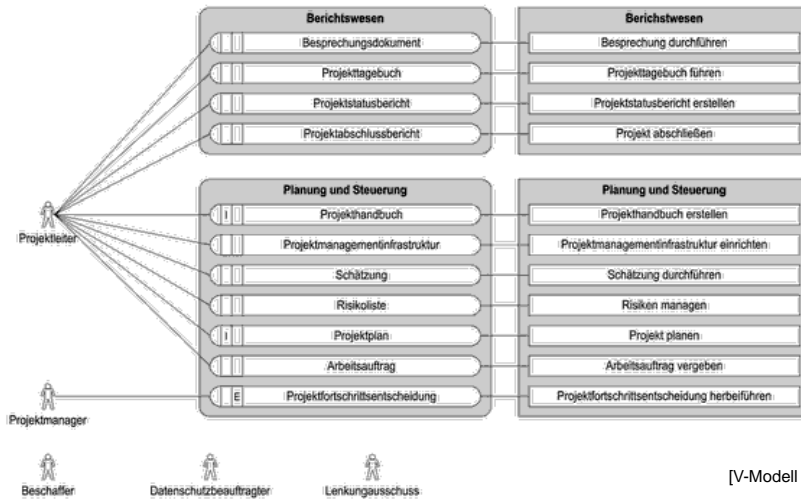
# Projektmanagement Regelkreis



V-Modell-Dokumentation (Teil 6)

# Vorgehensbaustein Projektmanagement: Gruppierung von Aktivitäten und Produkten

- durch Kombination



[V-Modell XT – Teil 3]

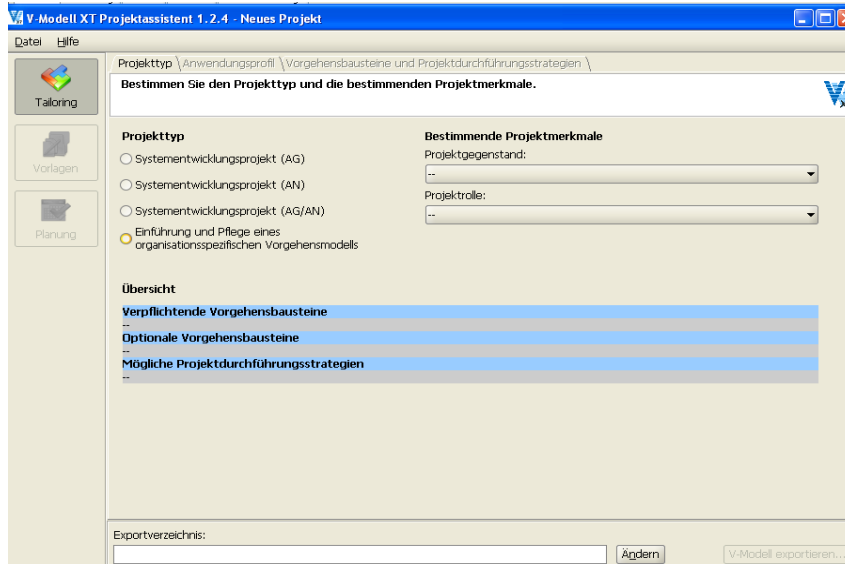
# Projektspezifische Anpassung - Tailoring

Die Anpassung des V-Modells an konkrete Projektbedingungen wird **projektspezifisches Tailoring** genannt. Es besteht die Möglichkeit Streichbedingungen zu definieren, um die Anzahl der Aktivitäten und Produkte auf das notwendige Maß zu reduzieren.

Unter **Tailoring** versteht man im V-Modell:

- Auswahl einer der vier **unterstützten Projekttypen u. Projekttypvarianten**
- Festlegung des **Anwendungsprofils**
- Das Anwendungsprofil legt die Auswahl der zu verwendenden **Vorgehensbausteine** und die **Projektdurchführungsstrategie** fest.

# Tailoringwerkzeug „Projektassistent“



# Bewertung des V-Modells des Bundes

## Vorteile:

- Integrierte Behandlung von Systementwicklung, Qualitätssicherung, Konfigurationsmanagement, Projektmanagement sowie anderen Prozessen
- Generisches Vorgehensmodell** mit definierten Möglichkeiten zum Maßschneidern auf projektspezifische Anforderungen
- Ermöglicht eine standardisierte Abwicklung von Systemerstellungs-Projekten
- Eine sehr gute Basis für die Prozessertifizierung 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

Softwaremanagement, © Prof. Uwe Aßmann

53

## 14.3.1 Extreme Programming

54

## Was heißt ‚Extreme Programming‘

- ein evolutionärer (agiler) Softwareentwicklungsprozess für kleine Teams in der Größe von zwei bis etwa zwölf Programmierern
- Werte:
  - Kommunikation, Einfachheit, Feed-back, Disziplin, Lernen, Qualität und Respekt
- Prinzipien:
  - offene Arbeitsumgebung in größerem Raum mit Flipcharts mit täglichem Standup-Meeting
  - kurze Iterationen in Perioden von ein bis drei Wochen, am Ende einer Periode steht ein getestetes System mit neuer Funktionalität
  - 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>

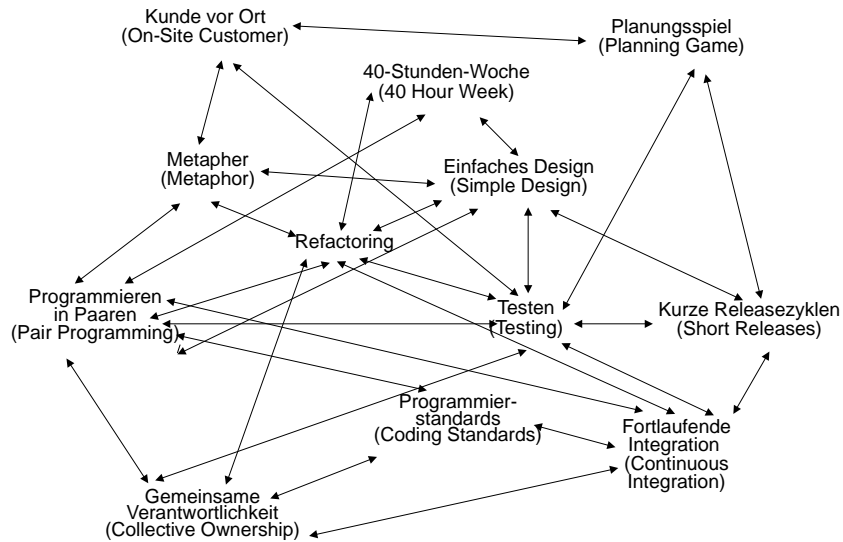
## 5 zentrale Prinzipien von XP

- 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 → 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 Wollen (embrace change)**
  - Änderungen auf Basis von Anforderungen, Entwicklungsprozessen, Systembestandteilen sind nichts Ungewolltes; sie führen in ihrer Gesamtheit zum neuen Produkt
- Qualitätsarbeit**
  - Gute Software befriedigt Softwareentwickler; sie möchten Qualitätsarbeit abliefern, dazu müssen die Qualitätsmaßstäbe vorgegeben werden.

[15]

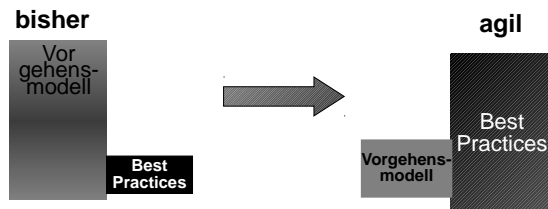
56

# XP-Techniken im Zusammenhang



# 14.3.2 Agile Software-Entwicklung

## Agile Softwareentwicklung



### „Best Practices“

- ▶ heißt kontinuierlich die Projektsituation beurteilen und entscheiden (kurze Feedback-Zyklen)
- ▶ so wenig wie möglich, aber soviel wie nötig
- ▶ von vornherein wird davon ausgegangen, dass man den Prozess an neue Rahmenbedingungen anpasst und ständig verbessert
- ▶ Einführung soll gut vorbereitet, ausgearbeitet, umgesetzt und angepasst erfolgen
- ▶ 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 Ermessensmaßnahme als Extremismus

## 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

### Zusammengefasst folgt daraus:

Wenn man den Sinn einsieht, kann man Regeln besser akzeptieren  
 Motivieren ist besser als Vorschreiben  
 Je komplexer das Projekt, desto mehr Vorschriften könnten notwendig sein. Es soll jedoch nichts geregelt werden, was nicht unbedingt geregelt werden muss

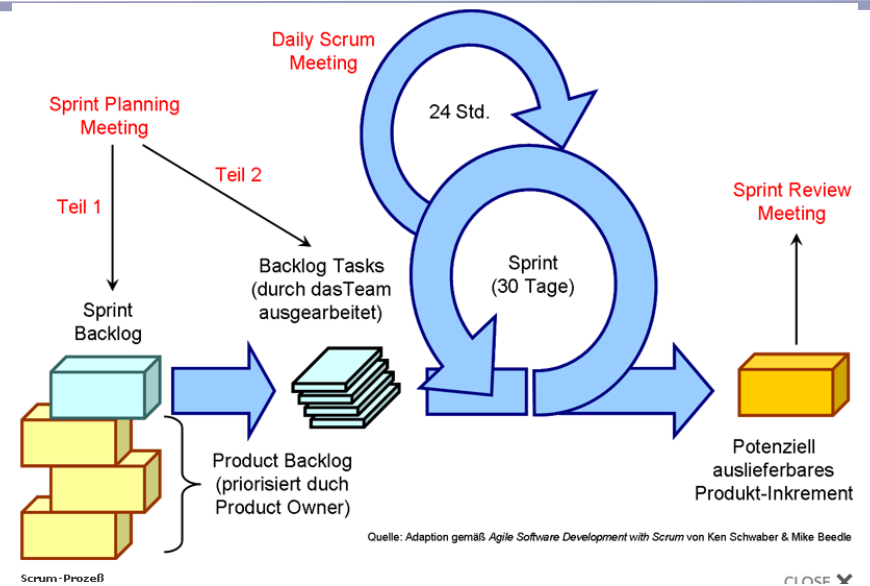
Software-Entwicklung ist ein kreativer Prozess, der (immer noch) von Menschen gestaltet wird, nicht von Robotern

## 14.3.3 Scrum

- ▶ Vorgehensmodell für das Projektmanagement im Rahmen der agilen SW-Entwicklung
- ▶ Hauptmerkmale:
  - für komplexe Projekte mit unklar definierten Anforderungen
  - iteratives Vorgehen
  - Zeitbegrenzt
  - wenig Rollen
  - selbstorganisierte Teams

<http://scrum-master.de>

## Scrum: Vorgehensweise



## 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
  - Aufwandsschätzung der einzelnen Backlog-Elemente
  - Team organisiert sich selbst

## Scrum-Rollen – Pigs & Chickens

- ▶ Direkt am Prozess beteiligte Personen 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 committed, but you'd only be involved!

[scrum-master.de](http://scrum-master.de)



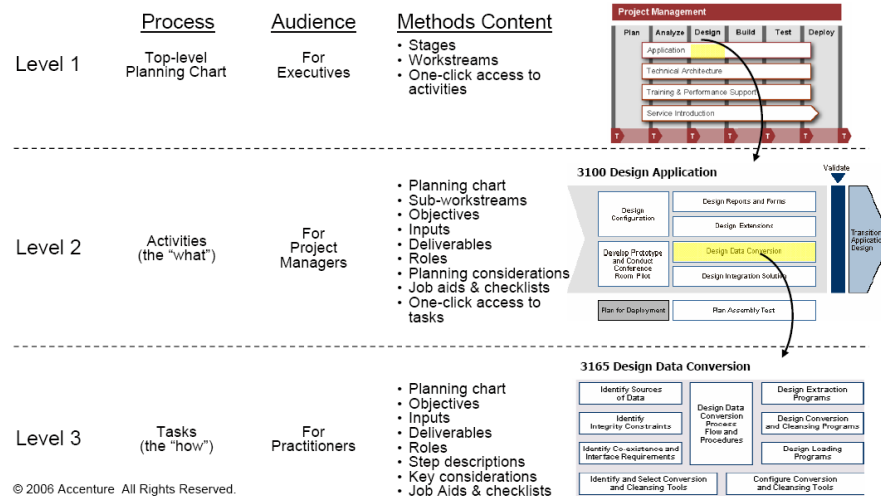
## 14.3.4 Spezielle betriebliche Vorgehen

- ▶ Große Firmen fassen ihre eigenen Vorgehensmodelle ab
  - SD&M Quasar Enterprise [www.openquasar.de](http://www.openquasar.de)
  - Accenture ADM

## Accenture's Business Integration Methodology und die Accenture Delivery Methods (ADM)

- Rahmen für umfassende Projekte (von der Strategie bis zur Einführung)
- Nicht nur Technologie (SW-Entwicklung), sondern auch Strategie, Prozesse und Mitarbeiter
- Vorschlag für Projekte-Prozesse und Ergebnismuster, aber sehr stark anpassbar und flexibel
- Hilfestellung für Projektmanager und –mitarbeiter, kein „Gängelband“
- Basiert auf bisherigen Vorgehensmodellen und modernen Konzepten aus der Forschung
- Extensives Training aller Mitarbeiter

## Die ADM umfaßt alle Detaillevel eines Projekts.



The End