

# 14 Ablauforganisation mit Vorgehensmodellen

1

Prof. Dr. rer. nat. Uwe Aßmann  
Lehrstuhl Softwaretechnologie  
Fakultät Informatik  
TU Dresden  
Version 13-0.4, 25.05.13

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

Softwaremanagement, © Prof. Uwe Aßmann, Technische Universität Dresden, Fakultät Informatik

# Ablauforganisation

3

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)** dienen dazu,
  - den Ablauf von Arbeitsvorgängen (Aktivitäten) innerhalb eines Projekts mit Hilfe von Prozessen zu organisieren
  - Verantwortlichkeiten innerhalb der Vorgänge zu formalisieren (Rollen)
  - zu rationalisieren, vereinfachen und standardisieren
- ▶ **Prozessmodelle** beschreiben schematisch die Methodik des Vorgehens, *geben also* Arbeitsabläufe schematisch vor
  - ad-hoc, automatisiert oder formalisiert
  - sequentiell oder iterativ

# Literatur

2

- ▶ [Pichler] Roman Pichler. SCRUM- - Agiles Projektmanagement erfolgreich einsetzen. dpunkt-Verlag.
- ▶ [Lippert] Lippert, M., Rook, 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](http://vmxt.fraunhofer.de/plaintext/downloads/erfolgsfaktoren034.07.pdf)
- ▶ [www.v-modell-xt.de](http://www.v-modell-xt.de)

Prof. Uwe Aßmann, Softwaremanagement



Prof. Uwe Aßmann, Softwaremanagement



# 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
  - Spiralmodell
  - V-Modell
  - Inception/
- ▶ **Vorgehensmodelle:** Ohne Phasen, aber aus Schablonen/Mustern zusammengesetzt
  - V-Modell des Bundes oder RUP
  - Oft zyklisch:
    - Modelle zur inkrementellen SWE
    - Evolutionäre Modelle
    - Prototypisches Vorgehen
  - Rückkopplung ist wichtig!
- ▶ **Leichtgewichtige, agile Vorgehensmodelle:** XP, SCRUM, EOS

# 14.1 Phasenmodelle

Softwaremanagement, © Prof. Uwe Alßmann, Technische Universität Dresden, Fakultät Informatik

# Phasenorganisation

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

- ▶ Phasenweises Vorgehen ist ein einfaches Planungs- und Controlling-Instrument
  - 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** des Projektleiters 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**

[ Jenny, S. 68 ]

# Meilensteine in der Planung von Phasenorganisation

- ▶ **Meilensteine** sind typisch für die Planung einer Phasenorganisation
  - sind Eckpunkte der Planung und Durchführung
  - an ihnen wird der Stand des Projekts evaluiert
  - befinden sich am Ende einer Phase
  - Basis der Bewertung sind: definierte Arbeitsprodukte an einem Termin
- ▶ Ein Meilenstein, der für eine Gruppe von Aktivitäten gilt, und an dem der Steuerfluss verzweigen kann, heisst **Entscheidungspunkt (EP)**
  - Entscheidungspunkte kennzeichnen MiniMeilensteine im Projekt
  - 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
  - SMART, insb. überprüfbar (als Voraussetzung für nächste Phase)
  - Meilensteinüberprüfung im Controlling durch Meilenstein-Trendanalyse
  - CCC
  - übergebbar (z. B. in die Produktbibliothek, dem AG, ...)

9

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

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

10

# Meilensteinüberprüfung / -entscheidung beim Übergang zur nächsten Phase

## Beteiligte:

- Projektleiter
- Entwicklungsverantwortlicher
- Qualitätssicherung
- Wirtschaftliches Controlling
- Auftraggeber

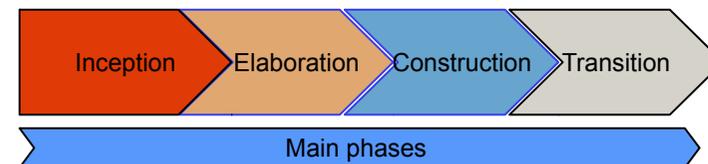
- ⇒ Jeder Meilenstein wird durch eine explizite Entscheidung abgeschlossen (**Entscheidungspunkt**).
- ⇒ **Entscheidung** über
  - Technische Alternativen
  - Freigabe der Ergebnisse
  - Fortführung des Projektes unter Abwägung des Risikos
- ⇒ Der Projektleiter stützt sich bei seiner Entscheidung auf die Ergebnisse der QS und ggf. des wirtschaftlichen Controllings
- ⇒ Die Entscheidung ist für alle am Projekt Beteiligten verbindlich.

11

# Phasenmodell des Durchführungsprozesses INECT

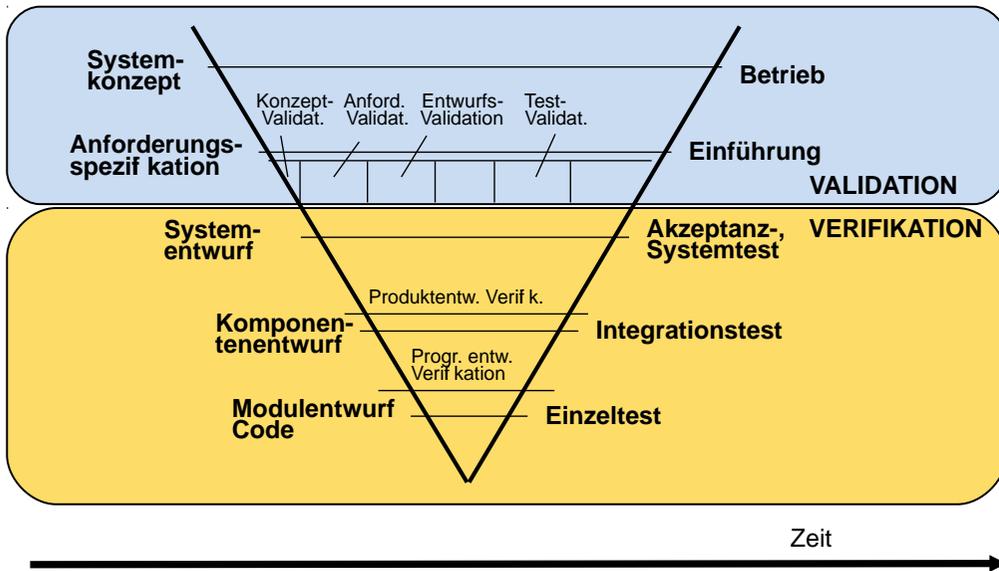
Die Phasengliederung INECT des schwergewichtigen Vorgehensmodells RUP ist allgemein für Planung und Controlling verwendbar:

- ▶ **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



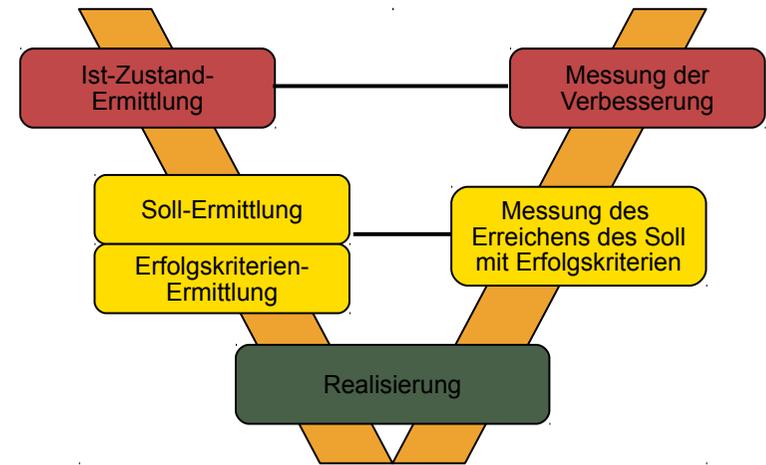
12

# Das V-Modell (nach B. Boehm)



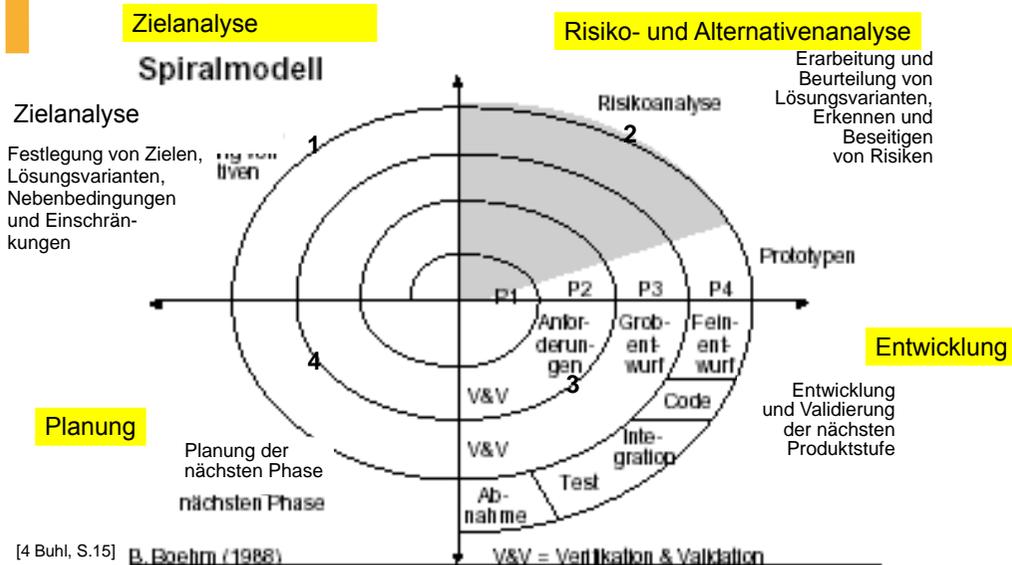
13

# Das V-Modell ist ein generisches Phasenmodell zum Problemlösen

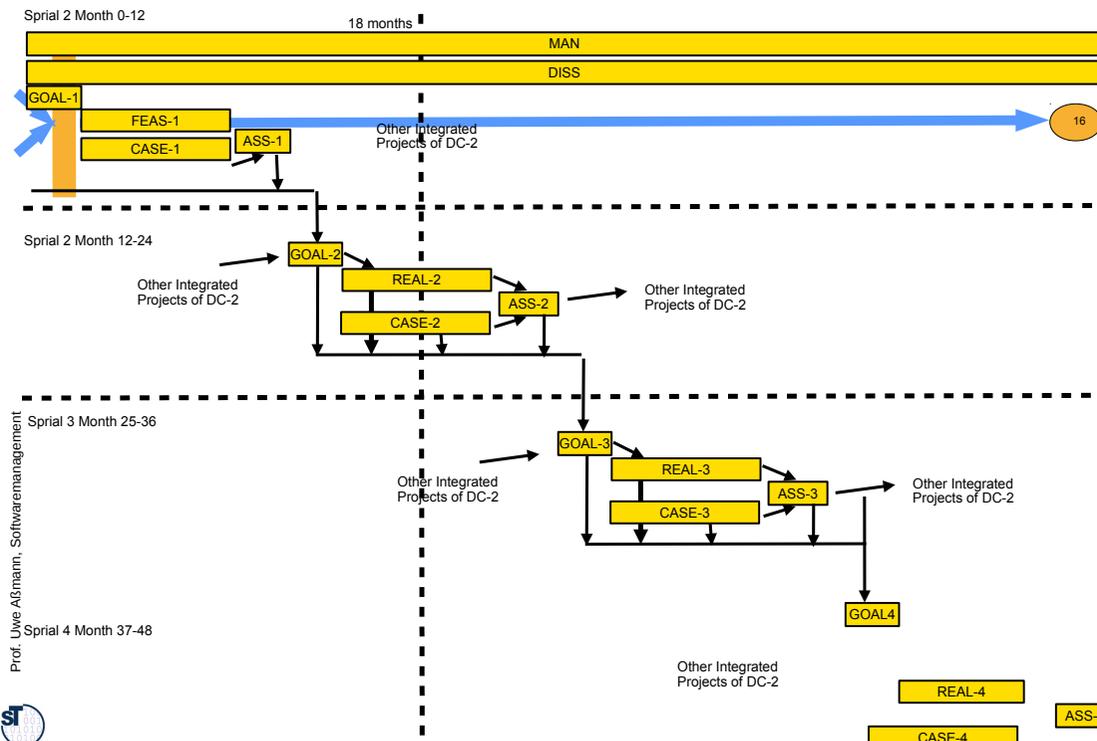


14

# Ein Phasenmodell mit Rückkopplung: Das Spiralmodell nach Boehm



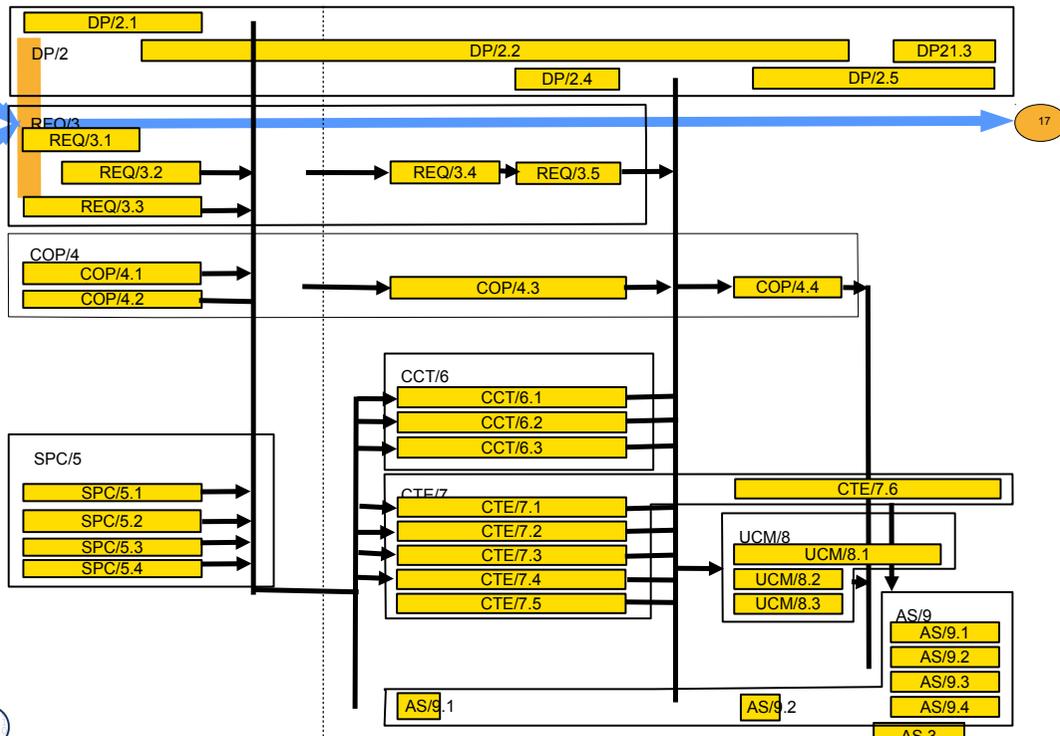
15



16

[4 Buhl, S.15] B. Boehm (1988)





## 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:

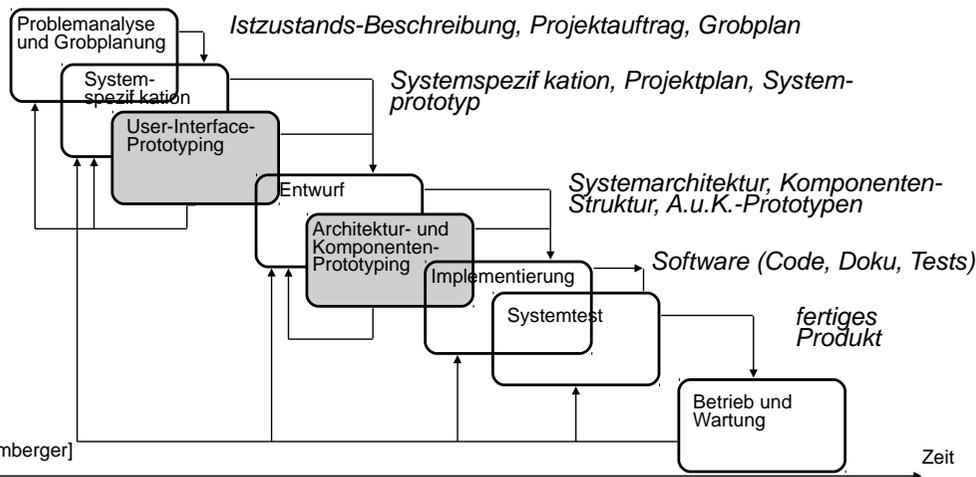
- Hohe Flexibilität des Vorgehens durch rückgekoppelten Prozess (wie PDCA)
- Integrierte Risikoanalyse
- Qualitätssicherungsmaßnahmen gut integrierbar
- Auch für Wartungsprojekte geeignet

### Nachteile:

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

## Prototyping-Orientiertes Phasenmodell mit Rückkopplung

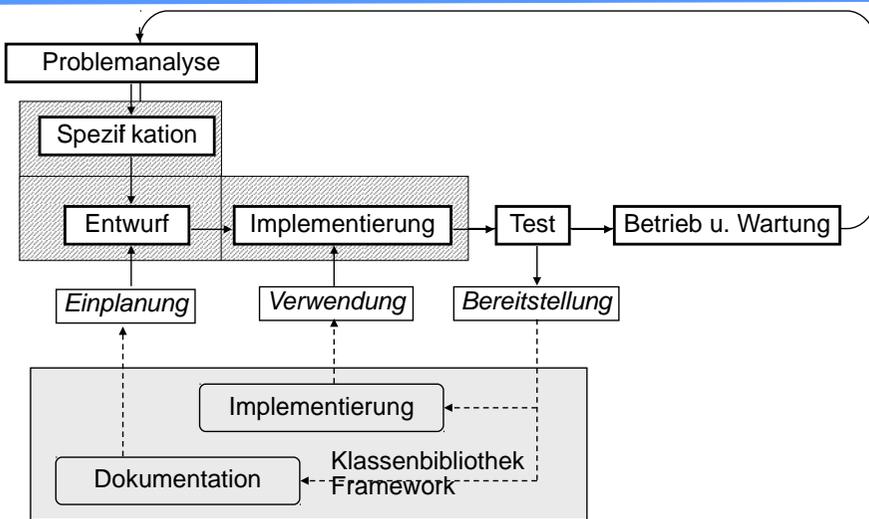
- ▶ Verfeinerung des iterativen Wasserfall-Modells oder des V-Modells
- ▶ Teile eines Produktes grob entwickeln, so, dass sie **ausführbar** werden
- ▶ **Minimal viable product (MVP)**: welches ist das wichtigste Feature?



## Prototyporientiertes Vorgehen

- ▶ Prototyping ist ein Vorgehensmodell, das schneller realitätsnahe Funktionstests ermöglicht
  - Das zugrundeliegende Modell ist ein iteratives Spiralmodell
  - Es wird so früh 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 und Kunden:
  - Durch unmittelbaren Beteiligung der Anwender erhöht sich die **Akzeptanz** des späteren Produktes
  - **Größere Sicherheit** von Anforderungsanalyse und Entwurf ist Basis für eine bessere Planbarkeit, Aufwandsabschätzung und Termintreue.
  - Die frühzeitige Auseinandersetzung mit dem Prototypen führt zu einer vorgezogenen **Einsatzvorbereitung**, während der die Benutzer geschult und trainiert werden

# Objektorientiertes Phasenmodell mit Rückkopplung und Wiederverwendung



21

nach [Pomberger]

auf der Basis eines Komponentenmodells und Kompositionssystems [CBSE]

# 14.2 Vorgehensmodelle

22

# 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 Projektlauf 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**)

23

# Einsatz von schwergewichtigen Vorgehensmodellen

- ▶ Bei Festpreisprojekten
  - Schätzung und Planung spielt dann eine große Rolle
- ▶ 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

24

# Grundlegende Begriffe von Vorgehensmodellen

25

- ▶ **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.

# 14.2.1 (Rational) Unified Process

26

- Ursprünglich von der Firma Rational, dann IBM

# Eigenschaften des Rational Unified Process (RUP)

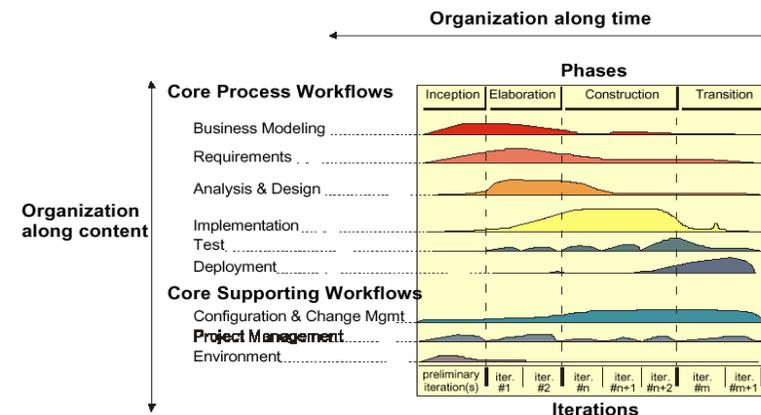
27

- ▶ Durch Anwendungsfälle (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)

28

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

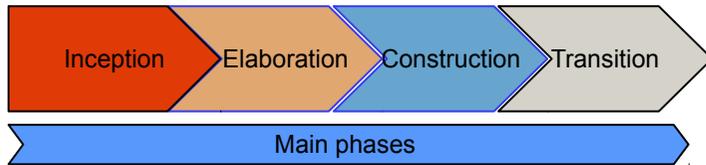


Softwaremanagement, © Prof. Uwe Aßmann

# 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



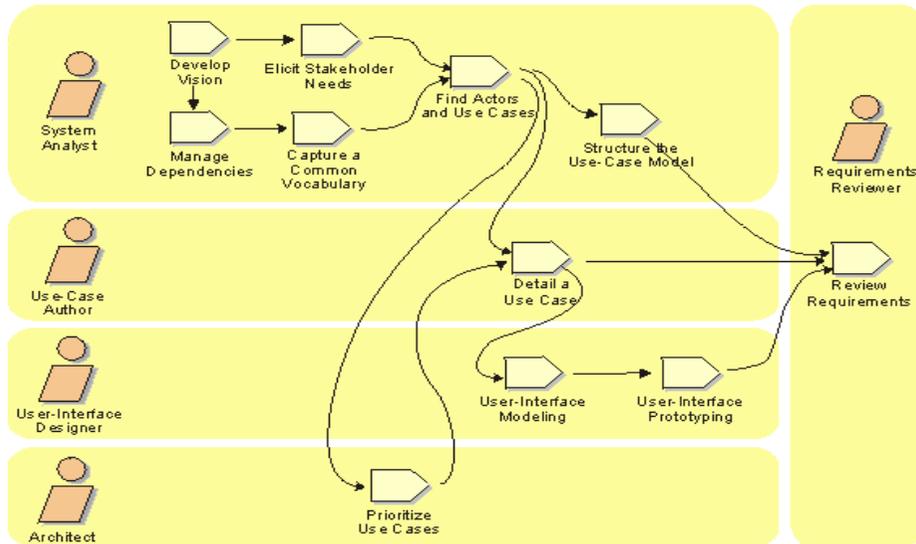
29

# 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.

30

# Bsp: RUP Workflow Requirements Analysis



31

# 14.2.2 V-Modell-XT des Bundes (VMXT)

32

## Historie

|              |   |
|--------------|---|
| 1986         | Das Bundesministerium der Verteidigung gibt ein Vorgehensmodell für die Softwareentwicklung auf Basis des V-Modells in Auftrag<br>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<br>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><br>Es wurde weiter an der Verbesserung gearbeitet (Schwerpunkt Erweiterbarkeit)   |
| August 2004  | Vorstellung des V-Modell XT (VMXT)  |
| Februar 2005 | Veröffentlichung des V-Modell XT  |
| 2009         | VMXT 1.3  |
| 2011         | VMXT 1.4  |

33

## 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

34

## Aufbau und Auslieferungsstruktur des V-Modell XT

**Vorgehensbausteine** (Moduln) beinhalten schematische Workflows, 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

35

## 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.

36

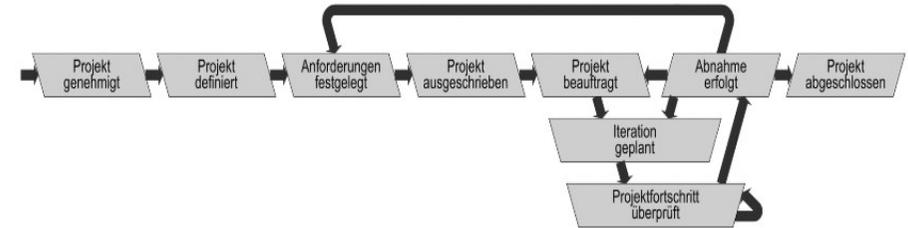
## Projekttypvarianten

- ▶ 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

37

## Beispiele für PDS (1): Systementwicklungsprojekt eines Auftraggebers

- ▶ 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



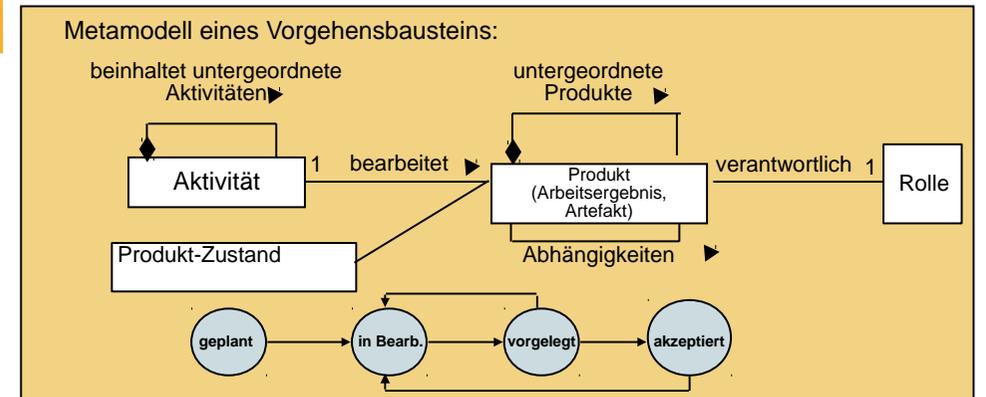
<http://v-modell.iabg.de/v-modell-xt-html/index.html>

## Beispiele für PDS (2): Agile Systementwicklung

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

39

## Vorgehensbausteine im VMXT



- ▶ Vorgehensbausteine sind die modularen Bausteine von Workflows, aus denen das V-Modell XT aufgebaut ist. Ein Vorgehensbaustein (workflow component)
  - kapselt zusammengehörnde Aktivitäten, Arbeitsergebnisse (Produkte), Rollen und Zustände
  - Ordnet Produktzustände zu
  - ist eine Einheit, die eigenständig verwendet, verändert und weiterentwickelt werden kann

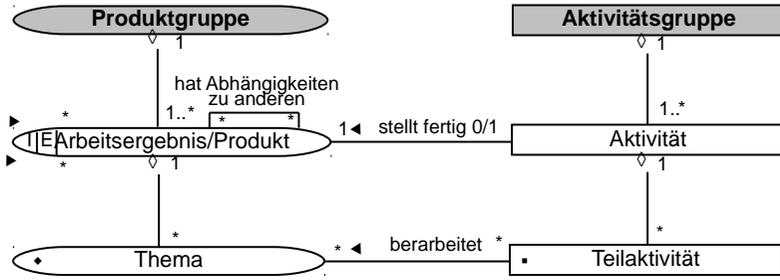
<http://v-modell.iabg.de/v-modell-xt-html/index.html>

# Vorgehensbausteine und ihre Bestandteile

41

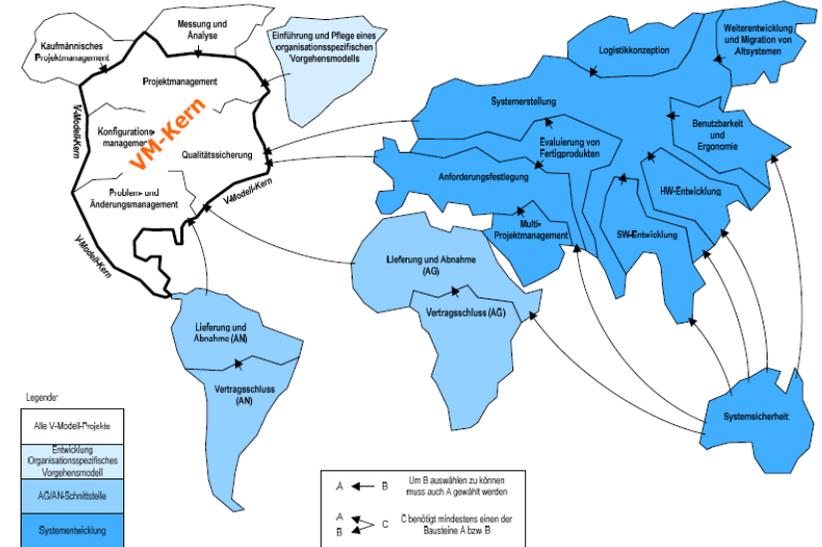
- ▶ Eine **Produktgruppe** ist eine Sammlung von einzelnen Produkten, welche einem Vorgehensbaustein zugeordnet sind.
- ▶ **Arbeitsergebnisse (Produkte)** sind bestimmte Dokumentations- und Teilergebnisse der Entwicklung. (Produkt == Artefakt == Objekt == Ergebnis)
- ▶ Der **Produkttyp** beschreibt auf 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



# “Landkarte” der Vorgehensbausteine: Modulbibliothek des VMXT

42



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

# Produktgruppen

43

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

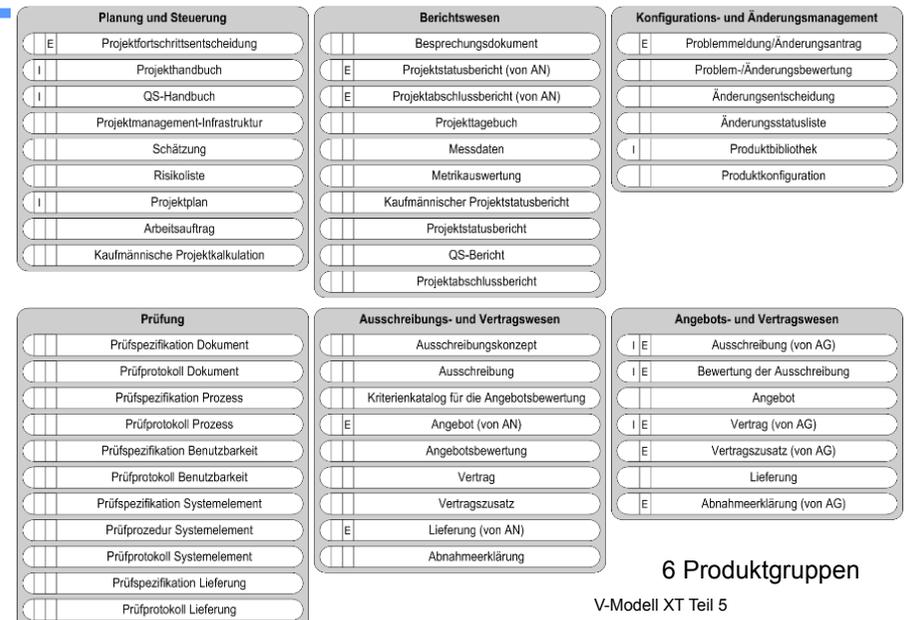
Im *Systementwurf* werden alle Produkte gesammelt, welche zur technischen Realisierung benötigt werden

### Bereich Organisation – 1 Produktgruppe

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

# Produktgruppe Bereich Projektmanagement

44



6 Produktgruppen

V-Modell XT Teil 5

# Rollen im VMXT

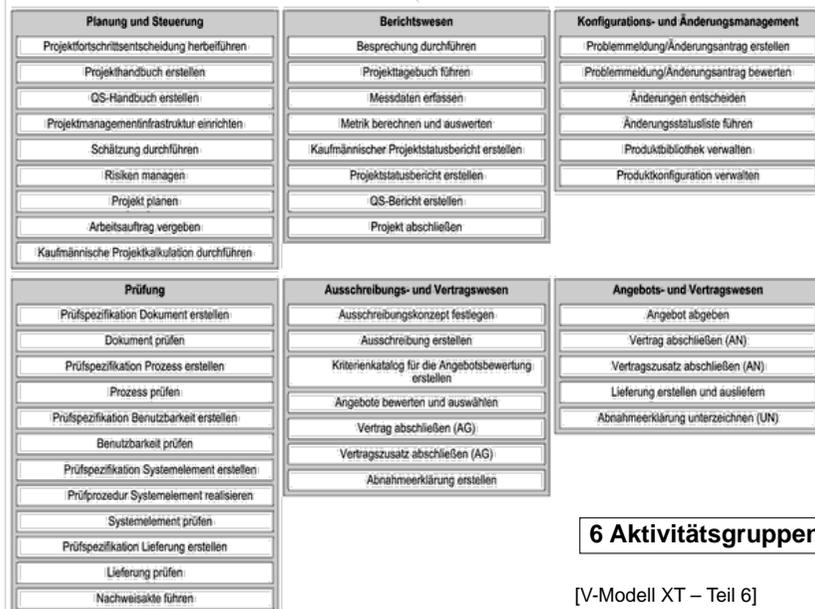
Jedem Arbeitsergebnis/Produkt wird eine eindeutige Rolle zugewiesen, der eine Person im Projekt entspricht. Eine Person kann auch mehrere Rollen einnehmen. Jede Rolle ist durch eine festgesetzte Struktur bestimmt, die besteht aus der Beschreibung, den Aufgaben und Befugnissen, dem Fähigkeitsprofil, der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

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

# Aktivitäten

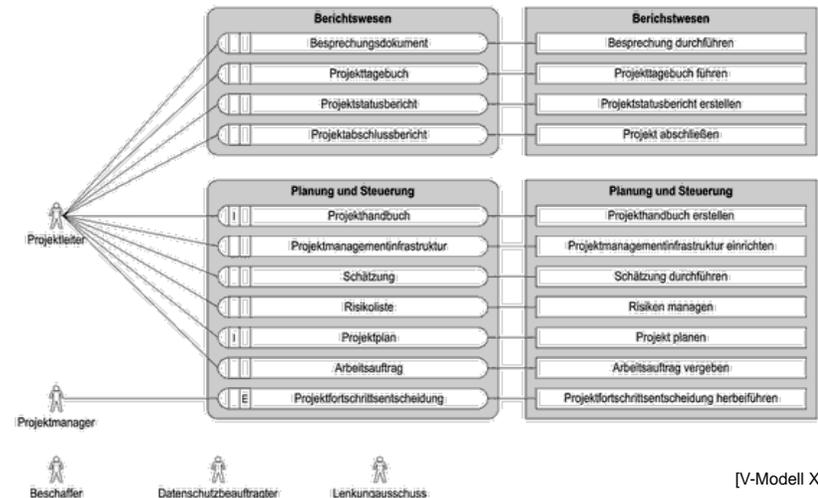
- ▶ In einer **Aktivität** wird festgelegt, welches Arbeitsergebnis (Produkt, Artefakt) von wem, wie erstellt wird.
- ▶ Meist wird unter einer Aktivität auch ein **Aktivitätstyp** verstanden.
- ▶ Aktivitäten können in **Aktivitätsgruppen** zusammengefasst werden.
  - Die Aktivitätsgruppen können analog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

# Aktivitätsgruppen im Bereich Projektmanagement



# Vorgehensbaustein Projektmanagement: Gruppierung von Aktivitäten und Produkten

- ▶ durch Kombination

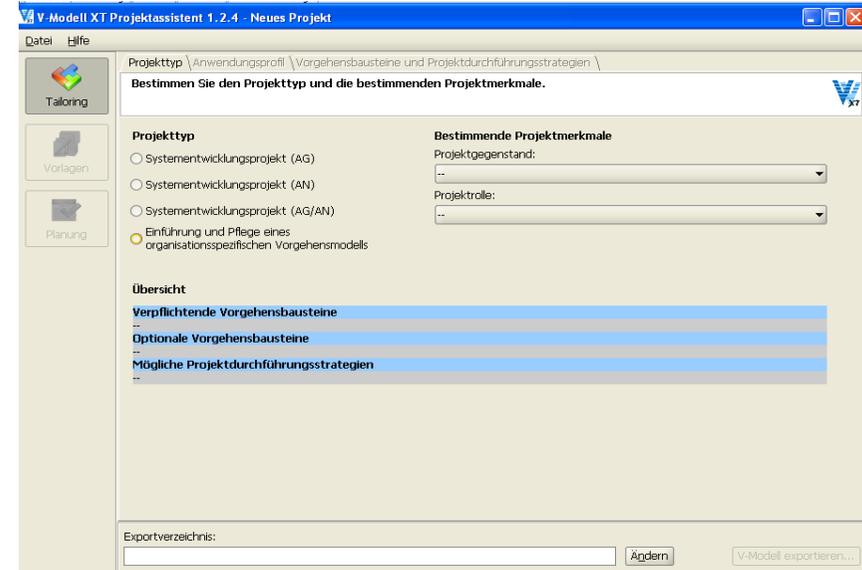


## Projektspezifische Anpassung - Tailoring

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

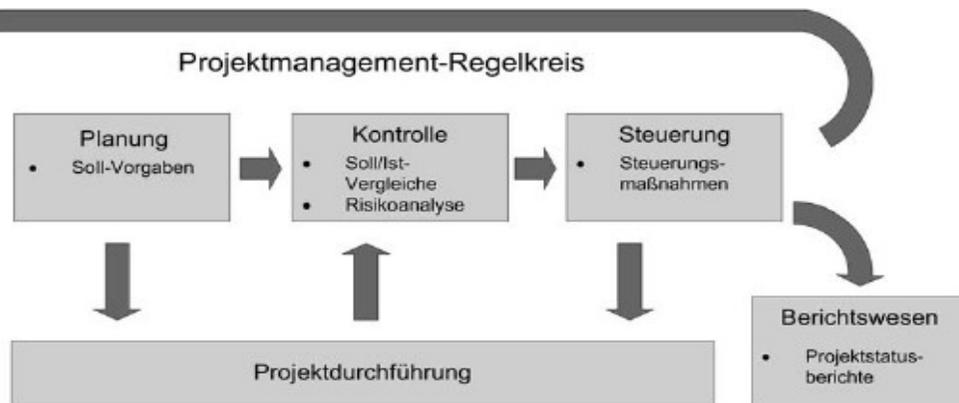
- ▶ Auswahl einer der vier unterstützten Projekttypen u. Projekttypvarianten
- ▶ Definition von *Streichbedingungen*, um die Anzahl der Aktivitäten und Produkte auf das notwendige Maß zu reduzieren
- ▶ Festlegung des Anwendungsprofils. Das Anwendungsprofil legt die Auswahl der zu verwendenden Vorgehensbausteine und die Projektdurchführungsstrategie fest.

## Tailoringwerkzeug „Projektassistent“



## VMXT Regelkreis als Ausprägung des PDCA

- ▶ Insgesamt werden Projekte im Regelkreis geführt



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

53

## Agile Softwareentwicklung

schwergewichtiges



leichtgewichtiges (agil)



„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 Rahmenbedingungen 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

55

|                                     |                                    |
|-------------------------------------|------------------------------------|
| 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          |

## 14.3.1 Extreme Programming

56

## 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:
  - 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

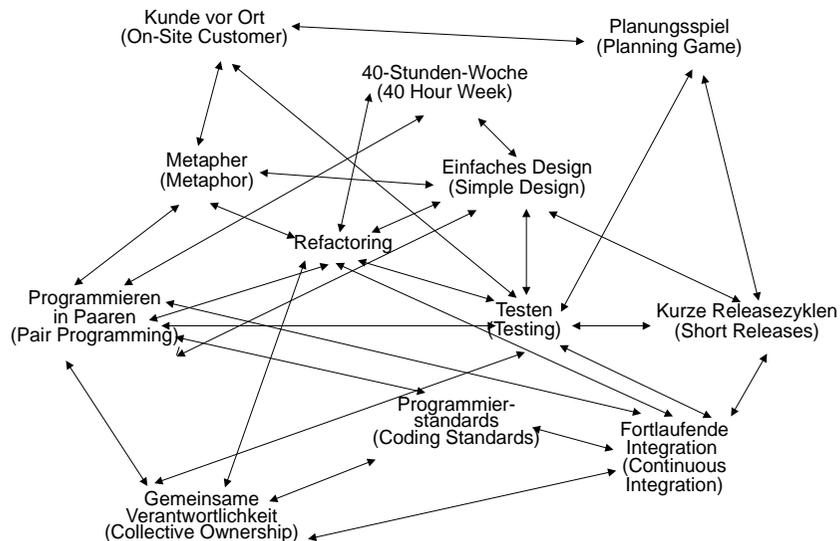
Westphal, F.: Extreme Programming; Copyright 2000-2004;  
URL: <http://www.frankwestphal.de/ExtremeProgramming.html>

## 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; 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
  - Gute Software befriedigt Softwareentwickler; sie möchten Qualitätsarbeit abliefern, dazu müssen die Qualitätsmaßstäbe vorgegeben werden.

[15 Balzert2]

## XP-Techniken im Zusammenhang

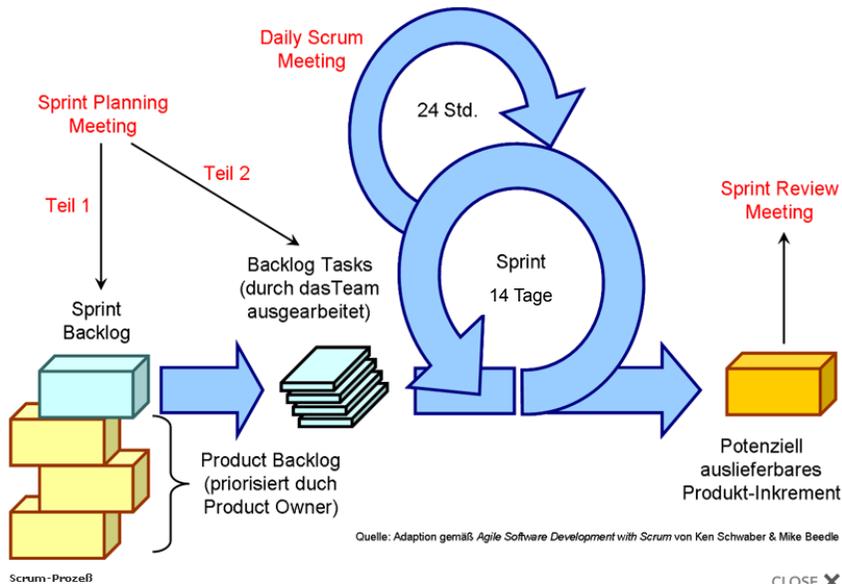


## 14.3.2 Scrum

- ▶ Leichtgewichtiges Vorgehensmodell im Rahmen der agilen SW-Entwicklung
- ▶ **Zeit-Schachtelung (time boxing)**
- ▶ Hauptmerkmale:
  - 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://www.scrum.org/Portals/0/Documents/Scrum%20Guides/Scrum_Guide.pdf)

<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, bzw. der einzelnen Sprint-Backlogs
- Aufwandschätzung 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 committed, but you`d only be involved!

## Daily SCRUM Meetings

- ▶ Berichten über den Zustand des Sprints und aufgetretene Probleme.
- ▶ KEIN Meeting zum Diskutieren der Architektur oder von Problemen!
- ▶ 3 Fragen soll jedes Team-Mitglied beantworten [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"

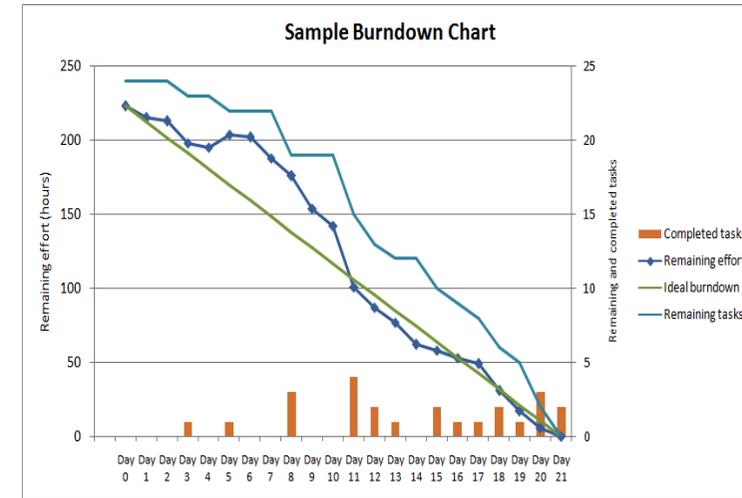


[http://en.wikipedia.org/wiki/Kanban\\_board](http://en.wikipedia.org/wiki/Kanban_board)

[http://en.wikipedia.org/wiki/Scrum\\_\(development\)](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 Sprints 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 geht nicht so einfach bei Festpreis-Projekten, z.B. auf Ausschreibungen hin

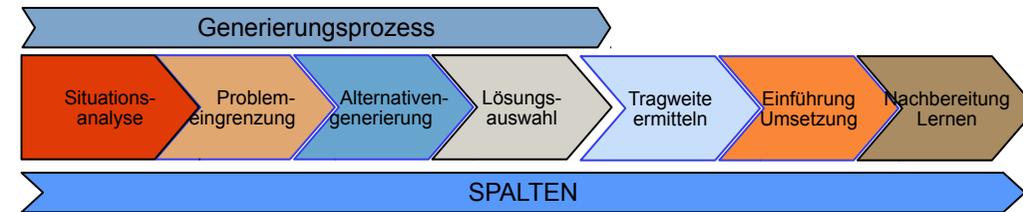
## The End

69

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

70

- ▶ Der SPALTEN-Prozess ist ein allgemeiner Problemlöseprozess, bestehend aus einem Lösungsgenerierungsprozess 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

71

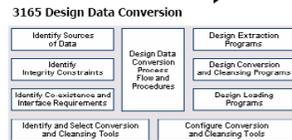
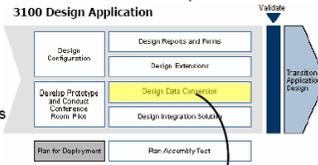
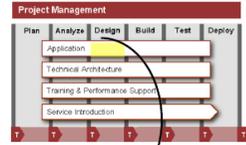
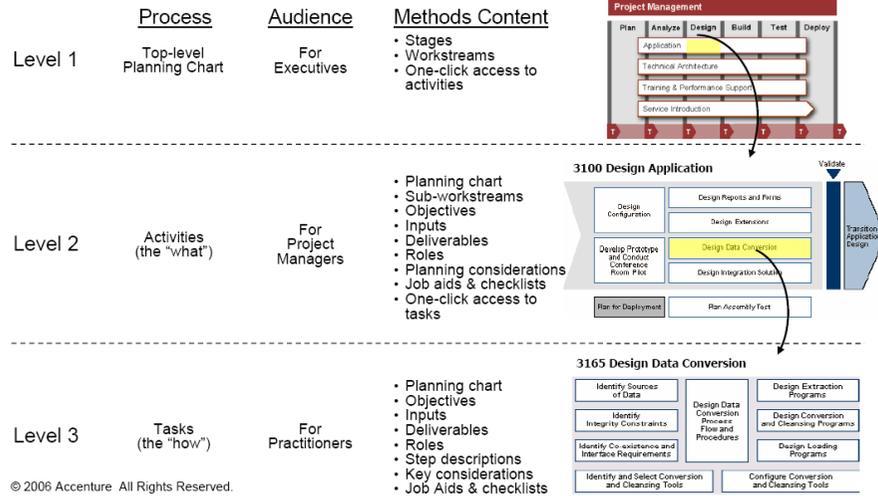
- ▶ Große Firmen fassen ihre eigenen Vorgehensmodelle ab
  - Cap Gemini (SD+M) Quasar Enterprise [www.openquasar.de](http://www.openquasar.de)
  - Accenture ADM

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

72

- 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.



© 2006 Accenture All Rights Reserved.

