

Einführung in die Softwaretechnik

**Software-Entwicklungsprojekte
und
Projektmodelle**

Einführung in die Softwaretechnik AB Softwaretechnik

Einführung in die Softwaretechnik
Software-Entwicklungsprojekte und Projektmodelle

- **Überblick**
 - Projekte, Projektsituationen, Projektszenarien
 - Projektmodelle
 - logisches Phasenmodell
 - evolutionäre Sichtweise
 - zyklisches Projektmodell
- **Lernziele**
 - Grundbegriffe über Projekte lernen
 - Bedeutung von Projektmodellen einschätzen
 - lineare und evolutionäre Sichtweise unterscheiden

Einführung in die Softwaretechnik AB Softwaretechnik

Software-Entwicklungsprojekte

Ein Software-Projekt ist ein organisiertes Vorhaben zur Herstellung, Anpassung und / oder Weiterentwicklung von Softwareprodukten.

Jedes Projekt ist einmalig:

- Es findet in der Projektsituation statt,
- es hat distinkte Ziele, Beteiligte, Randbedingungen, etc.,
- es gibt Anlaß zu einer Fülle verschiedener Aktivitäten,
- der situierte Prozeß entfaltet sich in der Zeit.

Projektszenarien

- charakterisieren Projektsituationen,
- typisieren Ziele, Beteiligte, Randbedingungen, etc.

Projektmodelle

- dienen zum Verständnis vor der Situation (Theorie),
- und zur Anleitung in der Situation (Methode).

Einführung in die Softwaretechnik

AB Softwaretechnik

Projektszenarien

Unterscheidungsdimensionen:

- **Projektziel: Standard- oder Individualsoftware**
- **Art der Entwicklung:**
 - Neu- oder Weiterentwicklung?
 - eigene oder Anpassung fremder Software?
 - Betriebliche Verankerung?
 - Verhältnis Auftraggeber und Auftragnehmer?
- **Art des Produktes:**
 - Anwendungsorientierung (Domäne, Aufgabenstruktur),
 - Einbettung (technische oder in Arbeitsprozesse),
 - Größe und Lebensdauer,
 - technische Beschaffenheit und äußere Architektur.
- **Art der technischen Umgebung:**
 - Entwicklungs- und Einsatzrechner,
 - Basissoftware: Datenbank, Fensterverwaltungssysteme ...
 - Programmiersprache,
 - Softwareentwicklungswerkzeuge und Programmierumgebung.
- **Beeinflussen: Welche Methodik taugt für welches Projekt?**

Einführung in die Softwaretechnik

AB Softwaretechnik

Projektmodelle als Landkarten

Wir können ein Projekt mit einem Territorium vergleichen,
in dem wir uns zurechtfinden müssen.

Ein Projektmodell liefert eine Landkarte für das
Territorium Software-Entwicklungsprojekt.

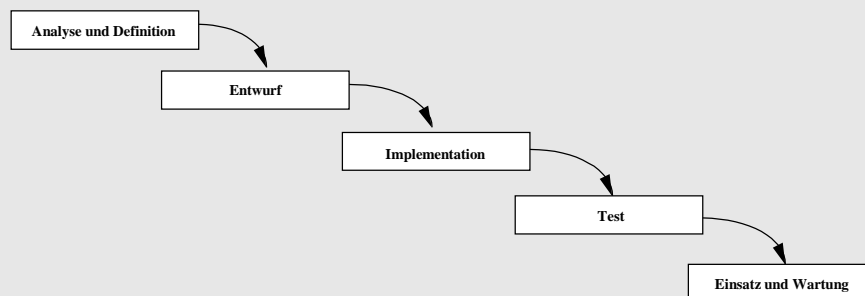
- **Eine Landkarte verdeutlicht interessierende Aspekte eines Territoriums:**
 - z.B. Verkehrswege, Höhenunterschiede.
- **Verschiedene Landkarten zeigen verschiedene Aspekte desselben Territoriums:**
 - z.B. Straßenkarte, Wanderkarte oder Wetterkarte.
- **Die Landkarte ist nicht das Territorium:**
 - z.B. ein Projekt "besteht" nicht aus Phasen, eine Einteilung in Phasen dient vielmehr zur Orientierung.
- **Die Landkarte ist das Territorium:**
 - z.B. wir sehen Projekte "durch die Brille" des Modells, wir beeinflussen Projekte durch das Modell.

Einführung in die Softwaretechnik

AB Softwaretechnik

Projektmodelle

Wasserfall- oder Phasenmodell



Nach Boehm 1976 – Vorbild für eine Klasse vergleichbarer Modelle

Einführung in die Softwaretechnik

AB Softwaretechnik

Diskussion von Wasserfall- oder Phasenmodellen

Charakteristika

- Herstellung von Software als Folge von Aktivitäten (Phasen).
- Unterschiedliche benannte und standardisierte Entwicklungsschritte, die zeitlich sequentiell in den Phase durchlaufen werden sollen.
- Ergebnissen der Phasen sind jeweils festgelegte Dokumente
- Rückgriffe auf vorangegangene Phasen sind beschränkt erlaubt, in der Praxis die Regel, gelten aber als Folge von "Entwicklungsfehlern".

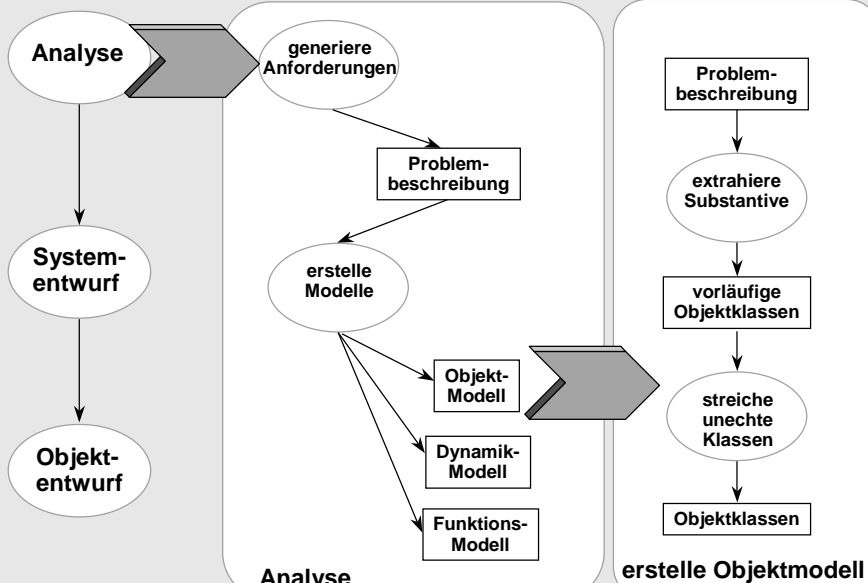
Probleme

- Einhalten der linearen Vorgehensweise.
- Anforderungen können nur teilweise vorab ermittelt werden und ändern sich ständig.
- Dokumente sind keine zuverlässigen Zwischenergebnisse, weil ihre Konsequenzen schwer abzusehen sind.
- Lernprozeß der Beteiligten bei der Entwicklung wird ignoriert.

Einführung in die Softwaretechnik

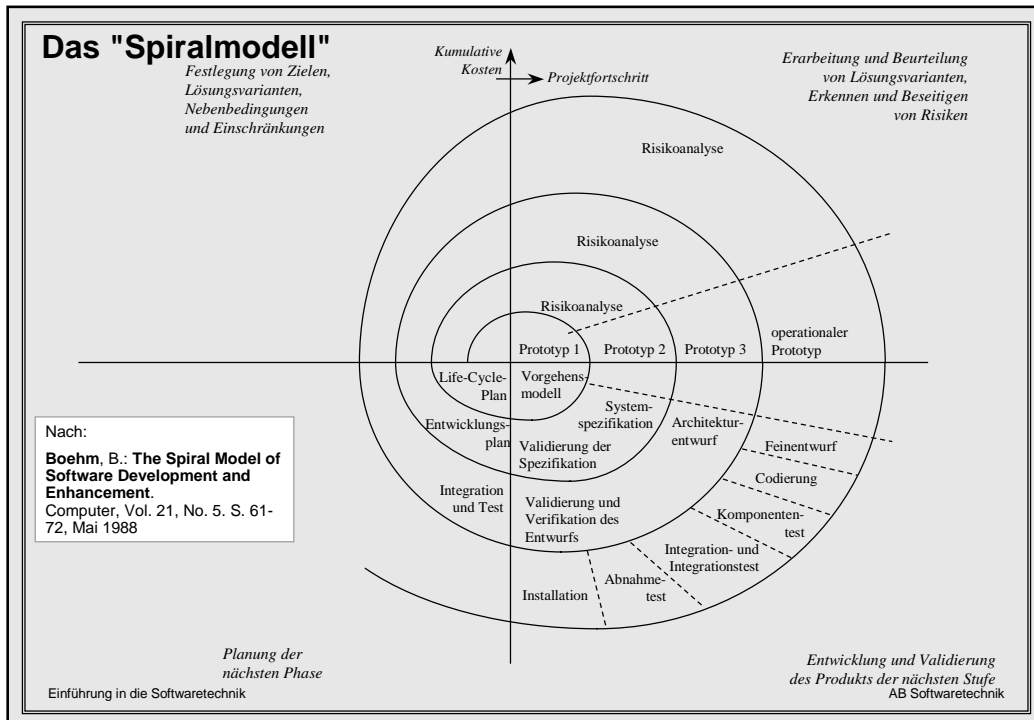
AB Softwaretechnik

Ein objektorientiertes Wasserfallmodell



Einführung in die Softwaretechnik

AB Softwaretechnik



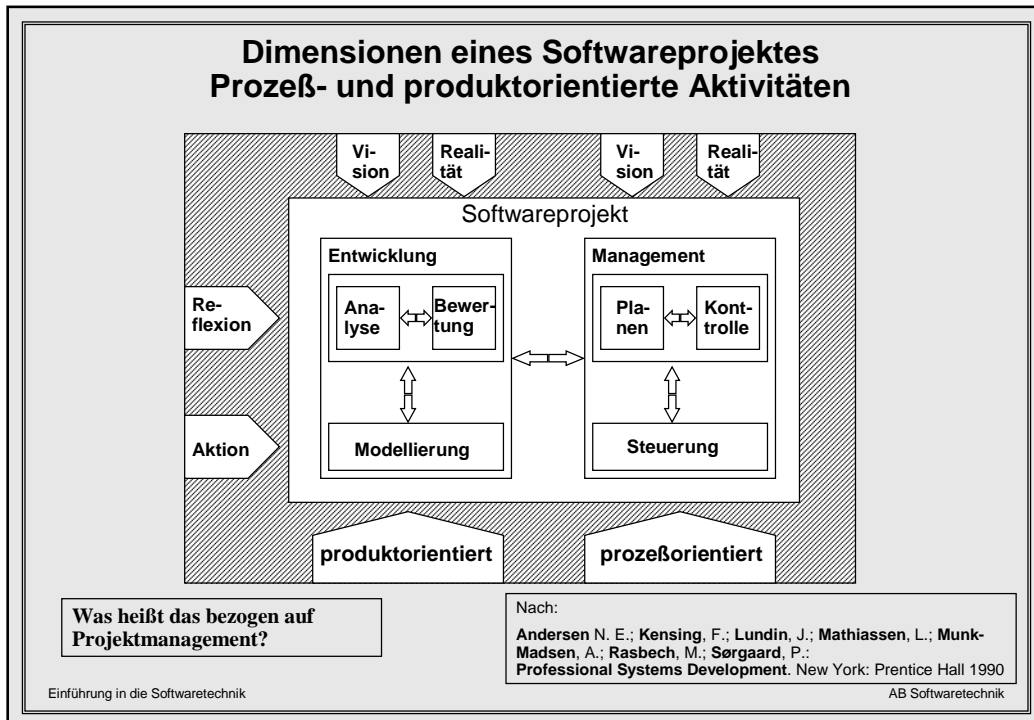
Diskussion des Spiralmodells

Charakteristika

- Benannte und standardisierte Entwicklungsschritte wie beim Wasserfallmodell.
- Phasen werden zyklisch mehrfach durchlaufen, bis das Produkt fertig gestellt ist.
- Schwierigkeiten der initialen Anforderungsermittlung und Lernprozesse werden berücksichtigt.

Probleme

- Ziel ist nach wie vor ein festes Produkt.
- Trennung von Herstellung und Einsatz bzw. Wartung bleibt erhalten.
- Die Aktivitäten sind technikzentriert.
- Versionen und Ausbaustufen werden nicht berücksichtigt.



Produkt- und prozeßbezogene Aufgaben

Komplementäre Sichtweisen:

- **produktorientiert:**
 - Software als Produkt aus Programmen und Dokumenten.
 - Herstellung und Einsatz getrennt.
 - Softwareentwicklung = Herstellung des Produktes.
- **prozeßorientiert:**
 - Software im Kontext von Lern-, Arbeits- und Kommunikationsprozessen.
 - Herstellung und Einsatz von Software miteinander verschränkt.
 - Softwareentwicklung = Gestaltung von Produkt, Prozeß und Einsatz.

Produktbezogene Aufgaben

Modellierung
Spezifikation
Konstruktion
Programmierung
Test

↔

Prozeßbezogene Aufgaben

Projektkommunikation
Koordination
Qualitätssicherung
Verwaltung
Management

... Kritik der Produktionssicht (2): ist nicht adäquat für prozeßbezogene Aufgaben ...

Einführung in die Softwaretechnik
AB Softwaretechnik

Softwareentwicklung im Kontext

- Software ist nur im Einsatzkontext auswertbar
- Organisatorisches Umfeld wandelt sich
- Softwareeinsatz wirkt auf die Arbeit zurück

Kritik der Produktionssicht (3): berücksichtigt nicht Wechselwirkungen zwischen Herstellung und Einsatz

Einführung in die Softwaretechnik
AB Softwaretechnik

Softwareentwicklung ist evolutionär



M.M. Lehman führt aus:

- **In konventionellen Projekten findet Evolution in der Wartung statt:**

„For software the term maintenance is generally used to describe all changes made to a program after its first installation... Programs do not suffer from wear, tear, corrosion, or pollution. They do not change unless people change them, and this is done whenever the current behavior of a program in execution is found to be wrong, inappropriate, or too restricted.“

- **Qualität von Software wird evolutionär entwickelt:**

„... performance, capability, quality in general, cannot at present be designed and built into a program ab initio. Rather they are gradually achieved by evolutionary change and refinement.“

- **Software und der Entwicklungsprozeß müssen auf Änderung ausgerichtet sein:**

„... the need for continuing change is intrinsic to the nature of computer usage. Thus the question raised by the high cost of maintenance is not exclusively how to control and reduce the cost by avoiding errors or by detecting them earlier in the development and usage cycle... Programs must be made more alterable, and the alterability maintained throughout their lifetime. The change process itself must be planned and controlled.“

Einführung in die Softwaretechnik

AB Softwaretechnik

Evolutionäre Systementwicklung

Probleme:

- Bei der Softwareentwicklung lernen wir.
- Der Kontext von Software ändert sich.

Widerspruch:

Phasenmodell oder Durch-Wursteln?



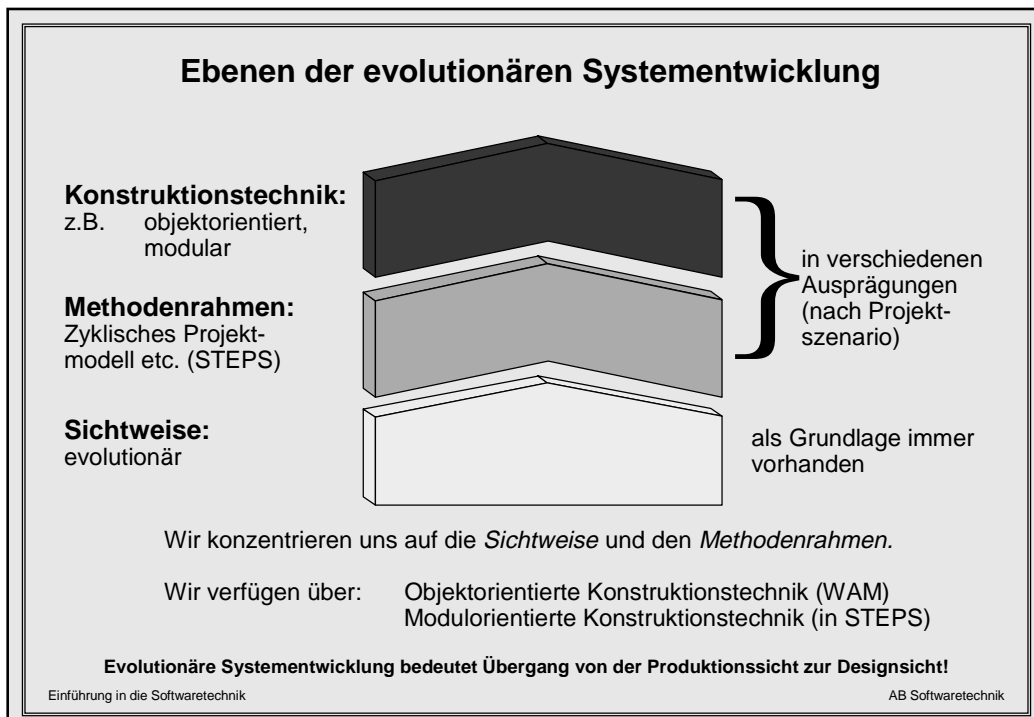
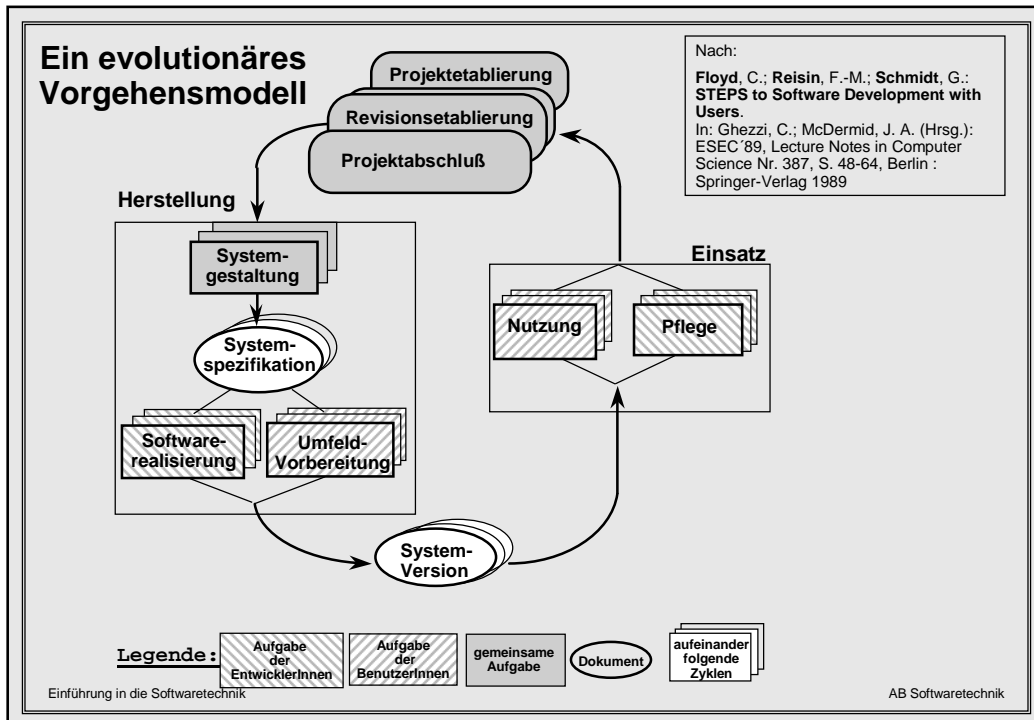
Lösungsansatz STEPS:

SoftwareTechnik für Evolutionäre, Partizipative Systemgestaltung

- Zyklisches Projektmodell,
- Softwareentwicklung als mehrstufiger Lernprozeß mit Prototyping,
- Softwareprodukt als Geflecht von Dokumenten-Versionen,
- Koordination durch Etablierung, Referenzlinien und Meilensteine,
- Projekttechniken zur Förderung der Projektkommunikation.

Einführung in die Softwaretechnik

AB Softwaretechnik



Lineare versus evolutionäre Sichtweise

Was bedeutet "evolutionär"?

- Nicht Software evolviert, sondern unser Verständnis über Software.
- Evolution findet in jedem Fall statt.
- Evolutionäre Sichtweise bedeutet bewußtes Vorgehen.

Lineare Sichtweise

Anforderungen sind fest,
sind vorweg definierbar,

sequentielle Anordnung der Schritte,
Rückgriffe nur im Sonderfall,
Fehler stören,

definiert "Soll-Praxis".

Evolutionäre Sichtweise

Anforderungen wandeln sich,
werden im Prozeß erschlossen,

zyklische Anordnung der Schritte,
Rückgriffe werden eingeplant,
Fehler sind Lernereignisse,

orientiert sich an "Ist-Praxis"

Zeitliche Phasen versus logische Stufen

In der Produktionssicht:

- Phasenmodell wird zeitlich interpretiert.
- Phasenmodell definiert
 - eine lineare Folge zeitlicher Phasen,
 - zugehörige Dokumente
- Reihenfolge der Entwicklungsschritte vordefiniert: entsprechen Phasen,
- ein Ergebnis-Dokument pro Entwicklungsschritt,
- nur vordefinierte Dokumente als Zwischenergebnisse.

In der Designsicht:

- Zeitliche Interpretation abgelehnt.
- Phasenmodell definiert
 - logisch abhängige Stufen,
 - zugehörige Diskursbereiche
- Entwicklungsschritte im situierten Prozeß vereinbart, feingranular,
- Entwicklungsschritte schreiben alle Dokumente fort,
- auch situationsbedingte Dokumente sowie auswertbare Prototypen.

Partizipation und Antizipation

Partizipation

- bedeutet Zusammenarbeit zwischen Entwickler/innen und Benutzer/innen;
- wird hier im Rahmen der Softwareentwicklung betrachtet;
- betriebliche Voraussetzungen müssen vorhanden sein.

Fragen

- Wie lernen Entwickler/innen die Benutzer/innen verstehen?
- Wie lernen Benutzer/innen ihre Anforderungen artikulieren?
- Wie wird Software in Arbeitsprozesse eingebettet?
- Wie gehen Entwickler/innen auf veränderte Anforderungen ein?

Antizipation

- bedeutet konstruktives Vorausdenken;
- mit, ggf. auch ohne Benutzer/innen

Fragen

- Wie stellen sich die Beteiligten das System im Einsatz vor?
- Wie sollen die neuen Arbeitsprozesse sein?
- Wie verändern sich Aufgaben und Arbeitsprozesse durch Software?
- Welche Wechselwirkungen von Menschen und Rechner sind erwünscht?

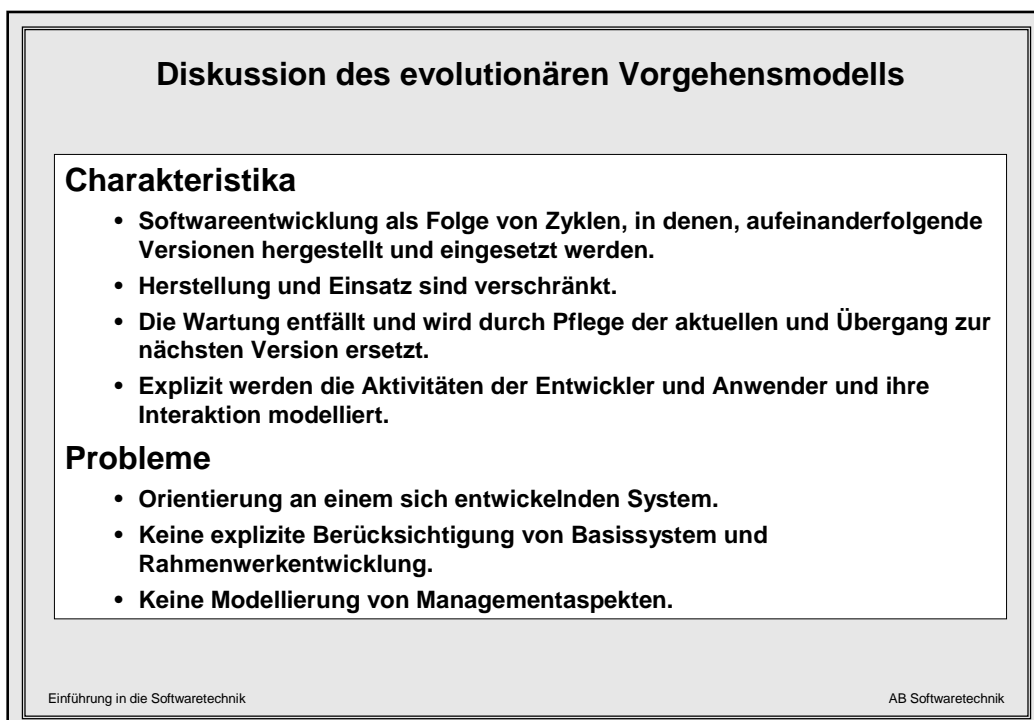
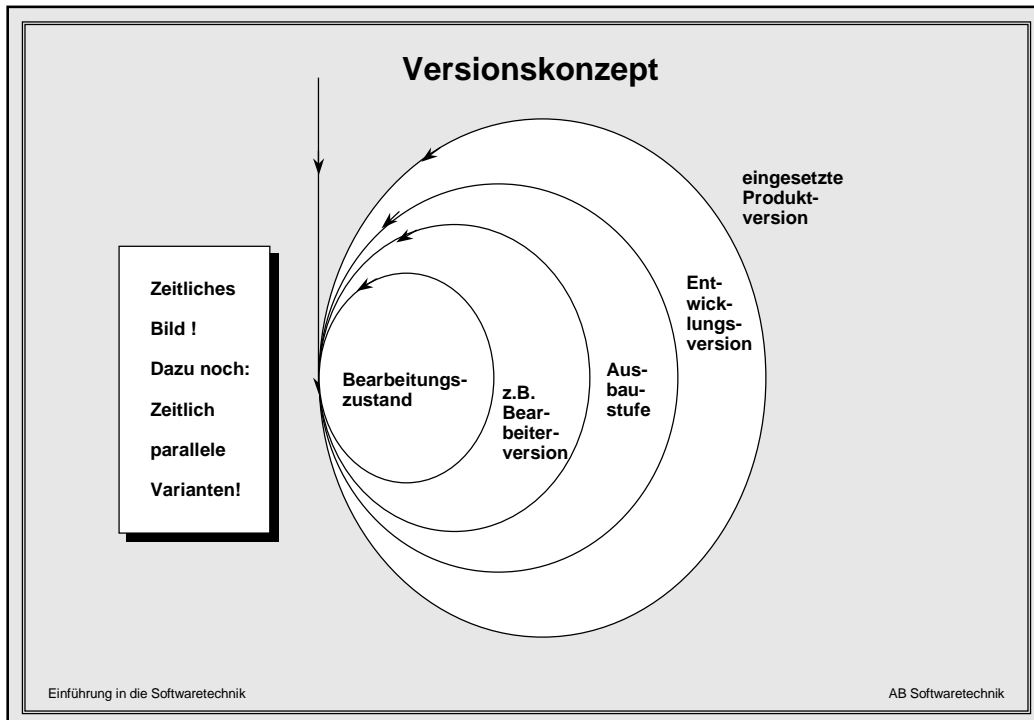
Einführung in die Softwaretechnik

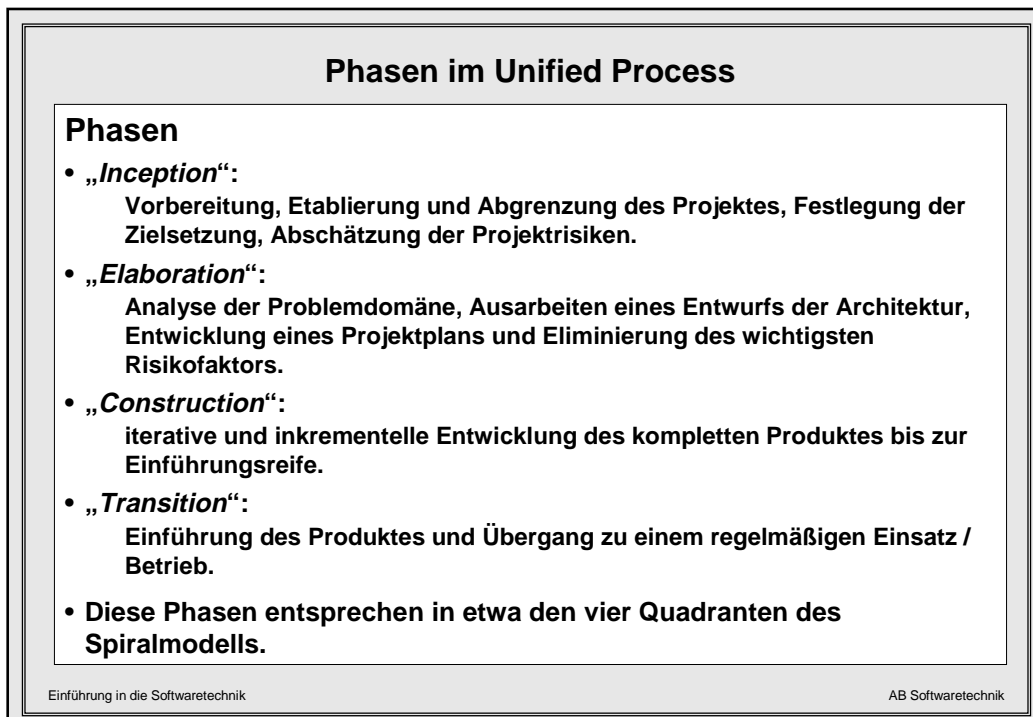
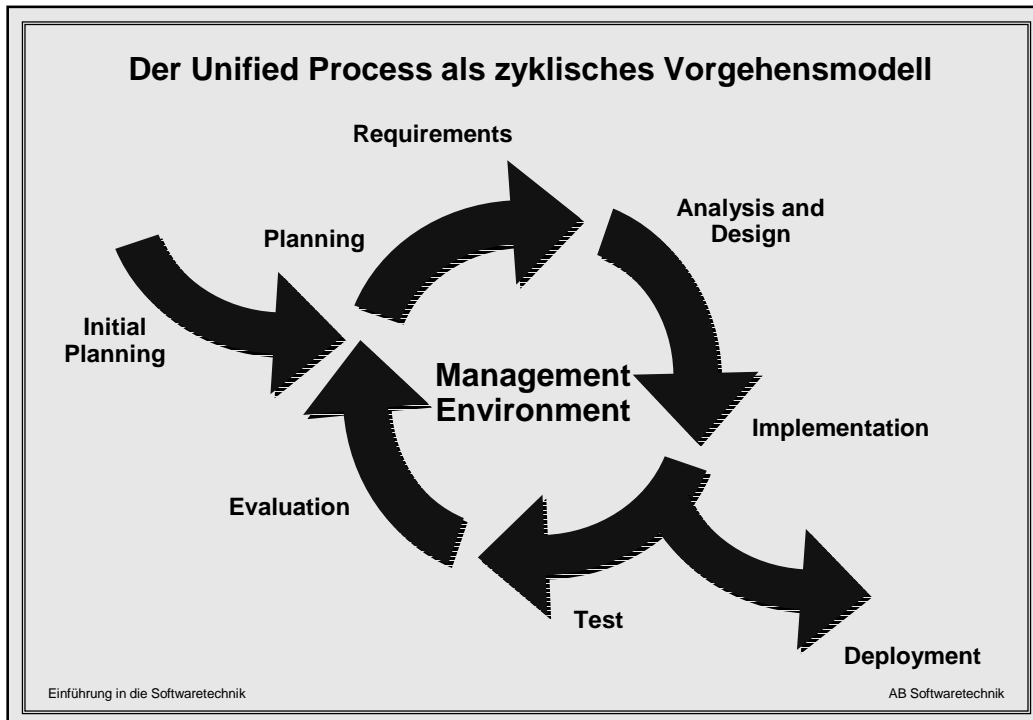
AB Softwaretechnik

Hier weiter!

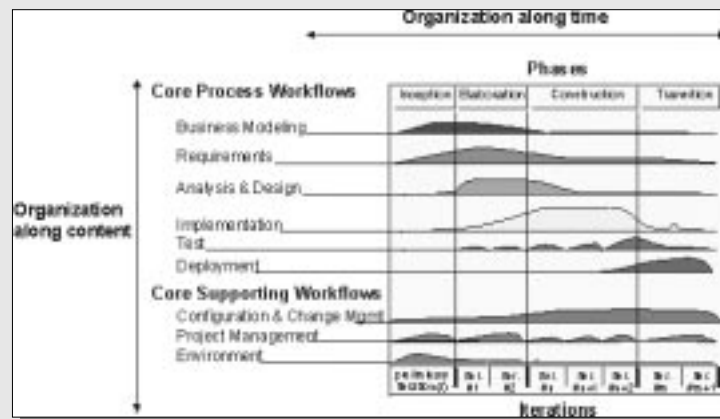
Einführung in die Softwaretechnik

AB Softwaretechnik





Zeitliche Organisation des Unified Process



Copyright © 1995-1999 by Rational Software Corporation

Einführung in die Softwaretechnik

AB Softwaretechnik

Diskussion des Unified Process Vorgehensmodells

Charakteristika

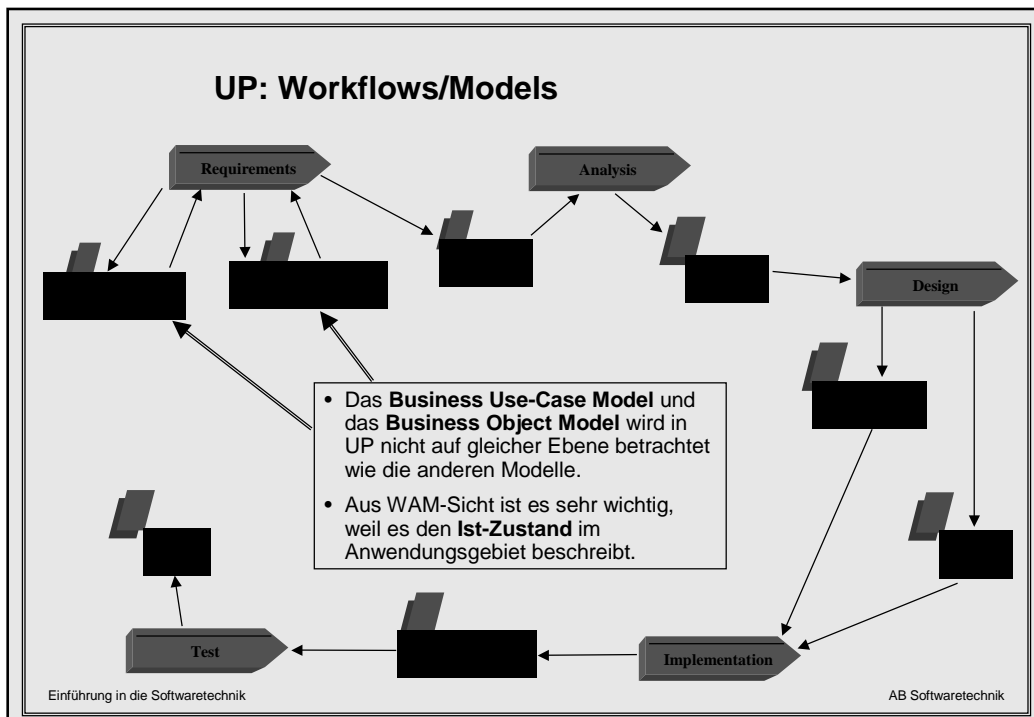
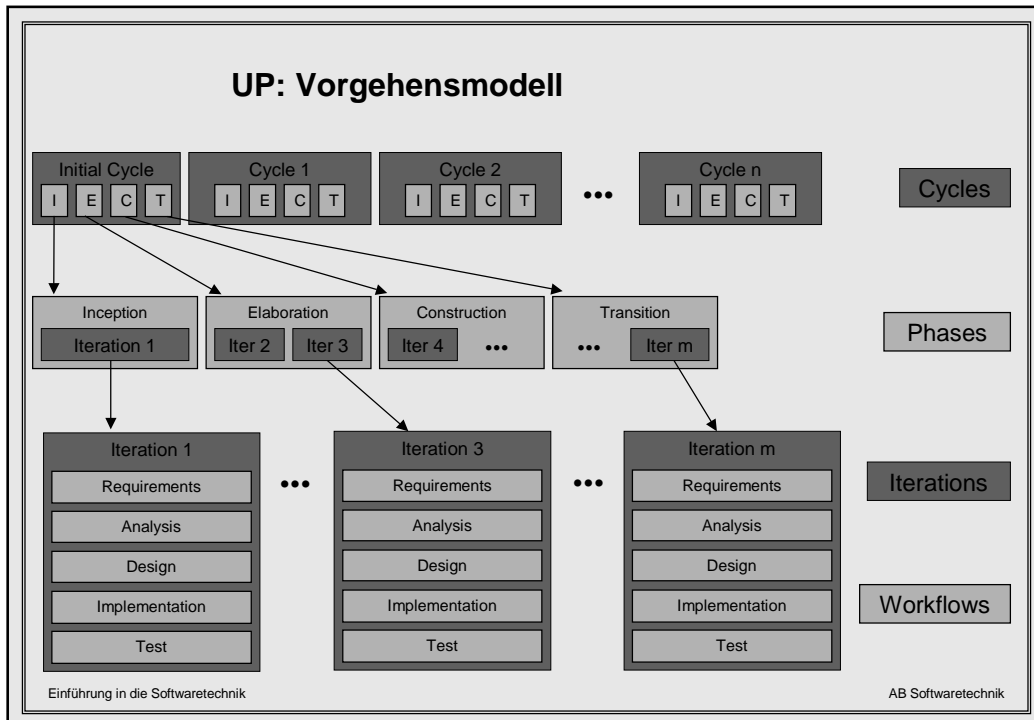
- „*Use-case-driven*“: Konkrete Anwendungsfälle bilden Ausgangspunkt der Modellierung.
- „*Architecture-centric*“: Im Zentrum des Entwicklungsprozesses steht die Architektur des zu entwickelnden Softwaresystems; dabei beeinflussen sich Anwendungsfälle und Architektur wechselseitig.
- „*Iterative and incremental*“: Im Verlauf des Entwicklungsprozesses wird das Softwaresystem in kleinen kontrollierten und sich wiederholenden Arbeitsschritten hergestellt.
- Weist Ähnlichkeiten zum Spiralmodell auf, unterscheidet sich jedoch von diesem durch den hohen Stellenwert der Anwendungsorientierung.

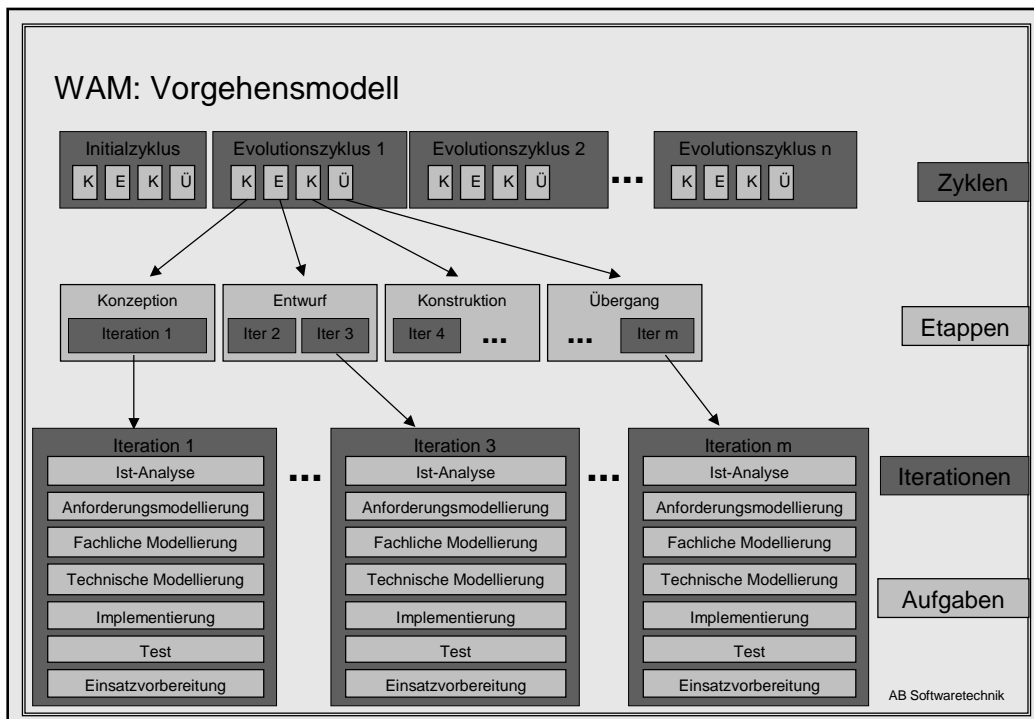
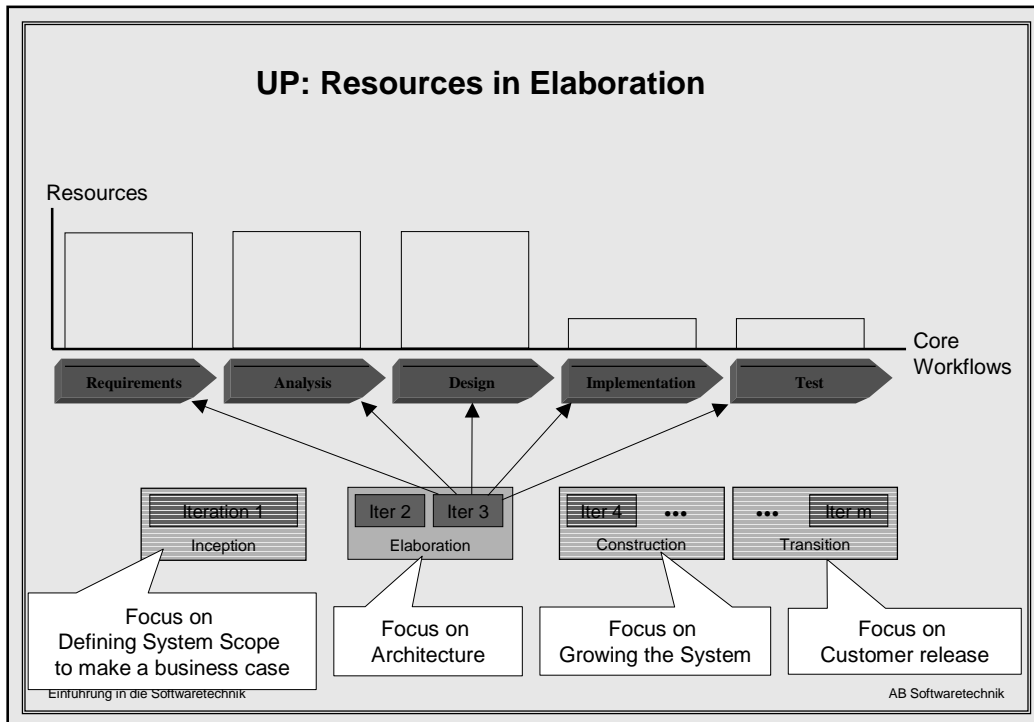
Probleme

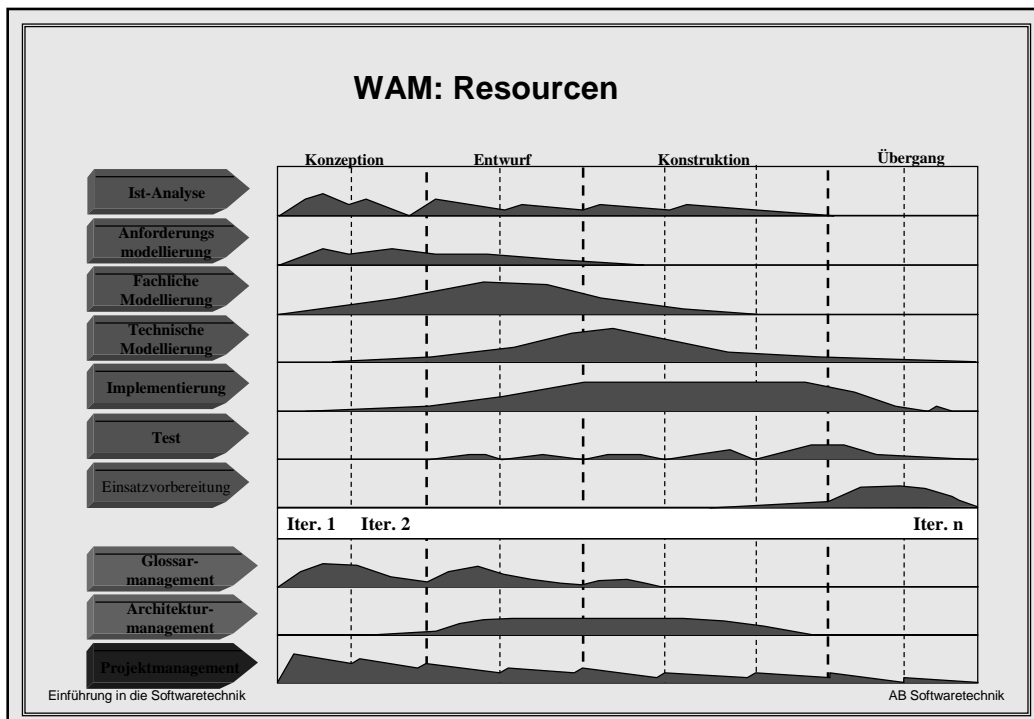
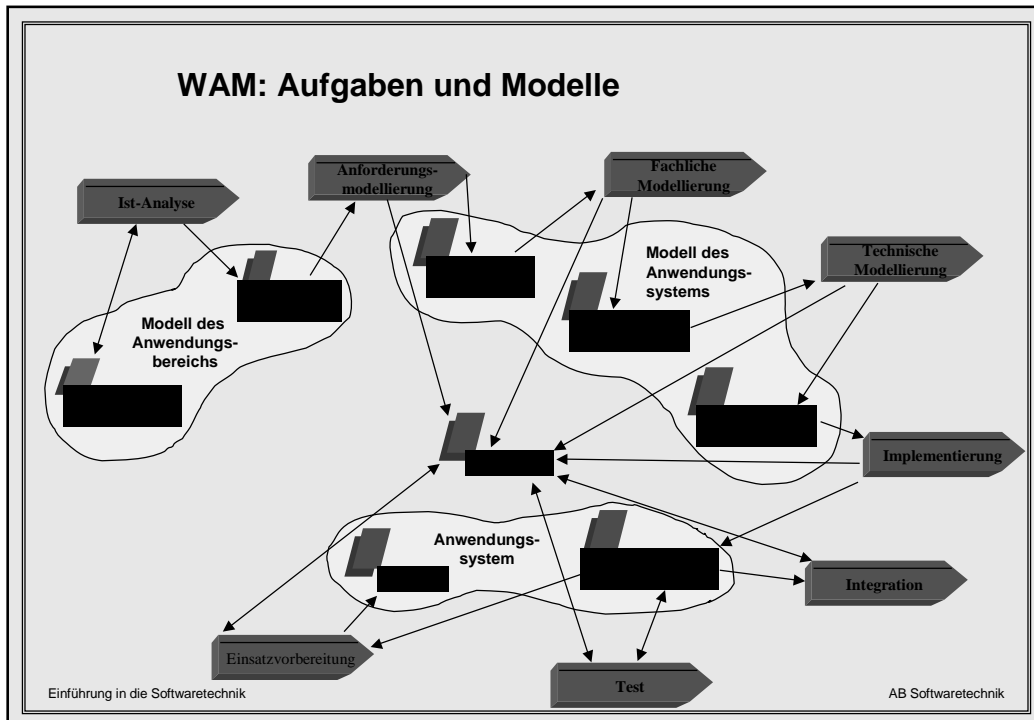
- Ziel ist nach wie vor ein festes Produkt.
- Trennung von Herstellung und Einsatz bzw. Wartung bleibt erhalten.
- Die Aktivitäten sind technikzentriert.

Einführung in die Softwaretechnik

AB Softwaretechnik







WAM: Projektorganisation

- Projektkalibrierung
- Projektplanung und Durchführung
 - **Projektetablierung**
 - **Strukturierung der Projektplanung durch Systemdekomposition**
 - **Etappen und Etappenplan**
 - **Iterationspläne oder Feinplanung über Referenzlinien**
- Projektsteuerung und Projektkontrolle
 - **Projektziele und Projektprinzipien**
 - **Risiko-Management**
 - **Ressourcen, Qualität, Zeit und Umfang**
 - **Aufwandsschätzung**
 - **Projektkontrolle durch Metriken**
 - **Führungsstil**
- Aufbauorganisation
 - **Rollen im Entwicklungsprozess/Organisatorische Einheiten**
 - **Verteilte Entwicklung**
- Zusammenspiel von Framework- und Anwendungsentwicklung

- **UP hat zu diesen Themen keine expliziten Aussagen.**
- **RUP hat hier sehr viel zu bieten, dass WAM sehr nahe kommt.**

Zusammenfassung

- **UP stellt einen allgemeinen Rahmen für jegliche Art von Softwareentwicklung zur Verfügung**
- **WAM ist eine konkrete Ausprägung von UP**
- **WAM bietet eine Reihe von praktischen Mitteln, um im Softwareentwicklungsprozess zu einem konkreten Produkt zu kommen.**