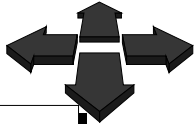


Einführung in die Softwaretechnik

Was ist Softwaretechnik?

Einführung in die Softwaretechnik AB Softwaretechnik

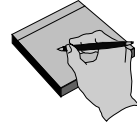
Was ist Softwaretechnik?

Einordnung der Softwaretechnik 

- **Überblick**
 - Software Engineering als Forschungsprogramm
 - Entstehung von Software Engineering
 - Aktivitäten (Phasen) der Softwareentwicklung
 - Forschungsrichtungen und Ergebnisse
 - Softwaretechnik ist mehr als Software Engineering
- **Einordnung**
 - Was machen wir in der Lehrveranstaltung?
 - Wo sind Einschränkungen und Spezialisierungen?
 - Spezielle Fachauffassung
- **Lernziele**
 - die Erwartungen an die Lehrveranstaltung klären
 - Grundbegriffe einführen

Einführung in die Softwaretechnik AB Softwaretechnik

Das Forschungsprogramm Software Engineering



- Ein **Forschungsprogramm** ist eine ausdrücklich ins Leben gerufene wissenschaftliche Anstrengung, mit
 - einem Gegenstandsbereich,
 - einer Art, diesen Gegenstandsbereich zu sehen (Sichtweise),
 - definierten Fragen, die zur Klärung anstehen,
 - zugelassenen wissenschaftlichen Methoden.
- **Software Engineering** wurde 1968 ins Leben gerufen
 - Gegenstandsbereich: Softwareentwicklung,
 - Sichtweise: Software als Produkt, Softwareentwicklung als Herstellung des Produktes,
 - Fragen: Produktivität, Qualität von Softwareprodukten,
 - ingenieurwissenschaftlich (mathematisch, technisch).

Einführung in die Softwaretechnik

AB Softwaretechnik

Die Entstehung von Software Engineering

- **circa 1950** - erste Rechner für Anwendungen verfügbar
- **1955-1960** - erste höhere Sprachen (Fortran, Cobol, Algol)
 - Rechner dringen in die Wirtschaft ein
 - Programmierung wird Tätigkeitsfeld
- **1960-1965** - zahlreiche Problemfelder werden erschlossen
 - erste Betriebssysteme
 - Zahl der ProgrammiererInnen steigt
 - das Wort "Software" wird gebräuchlich
- **ab 1965** - Rechner der dritten Generation



Die Softwarekrise

Einführung in die Softwaretechnik

AB Softwaretechnik

Die Softwarekrise

- **Software war**
 - fehlerhaft
 - nicht wartbar
 - nicht termingerecht fertig
 - nicht den Anforderungen entsprechend . . .
- **Es gab keine lehrbaren Grundlagen für:**
 - Programmiermethodik,
 - Gliederung der Softwareentwicklung,
 - Arbeitsteilung,
 - Projektorganisation,
 - technische Unterstützung . . .
- **Heute: es gibt lehrbare Grundlagen, aber . . .**
- **Ist die Softwarekrise überwunden?**



Einführung in die Softwaretechnik AB Softwaretechnik

Die Gründungskonferenzen 1968/69

- **Garmisch-Partenkirchen (1968) & Rom (1969)**
 - Regie: Nato (militärische Bedeutung von Software!)
 - Teilnehmende aus Wirtschaft, Wissenschaft und Militär
- **Naur, Randell 69:**

The phrase 'software engineering' was deliberately chosen as being provocative, in implying the need for the types of theoretical foundations and practical disciplines, that are traditional in the established branches of engineering.
- **Forschungslinien:**
 - Konzepte,
 - Methoden,
 - Werkzeuge,
 - Organisationsformen
- **Entstehung der "Produktionssicht" der Softwareentwicklung**

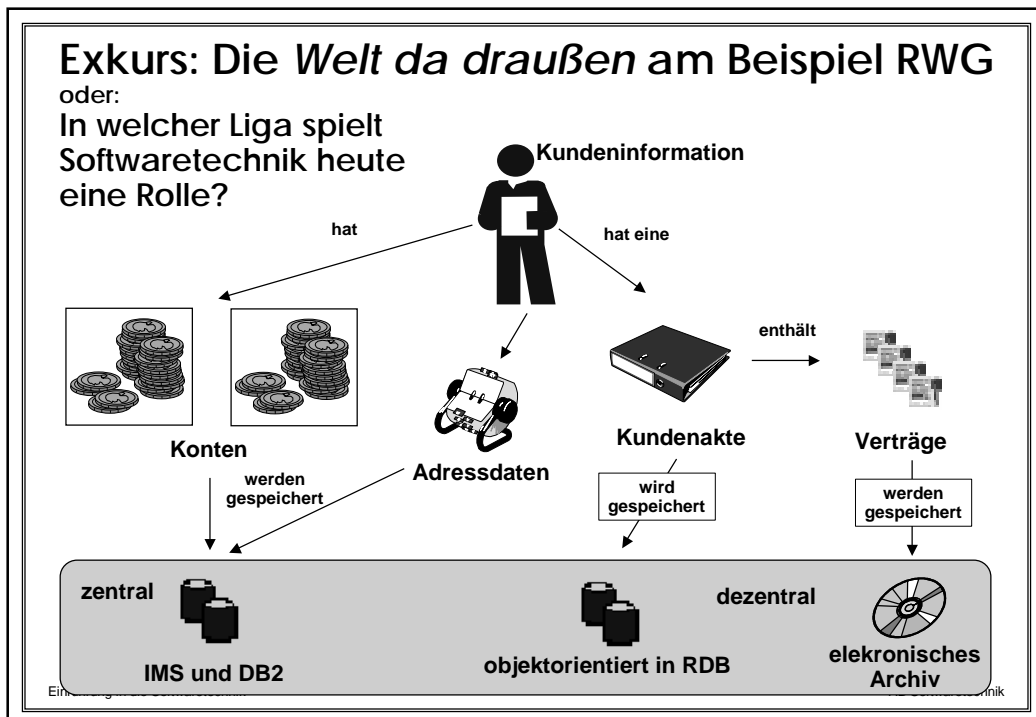
Einführung in die Softwaretechnik AB Softwaretechnik

**Softwaretechnik
heute**

**Die Anwendung der Softwaretechnik soll Softwarequalität
produzieren.**

**In EST werden Konzepte, Methoden und Techniken zur Verbesserung der
Softwarequalität vorgestellt.**

Einführung in die Softwaretechnik AB Softwaretechnik



Softwarequalität



Softwarequalität nach DIN 55350 : Die Gesamtheit der Eigenschaften oder Merkmale, die Software in Verwendung und (Weiter-) Entwicklung aufweist, um die an sie gestellten Anforderungen zu erfüllen.

Äußere und innere Qualitätsmerkmale

Gebrauchsqualität : ist für die Verwendung von Software von primärer Bedeutung. Sie wird anhand äußerer Qualitätsmerkmale von Benutzern und anderen Personengruppen im Einsatz bewertet.

Produktqualität : bezieht sich auf die Konstruktion des Softwareproduktes und wird anhand innerer Qualitätsmerkmale von Softwareentwicklern festgestellt.

Prozeßqualität : sichert Gebrauchs- und Produktqualität im Entwicklungsprozeß ab.

Einführung in die Softwaretechnik © Floyd, Züllighoven in Rechenberg, Pomberger
AB Softwaretechnik

Gebrauchsqualität von Anwendungssoftware



- Die **Funktionalität** des Softwaresystems orientiert sich an den *Aufgaben* im Anwendungsbereich.
- Die **Handhabung** des Softwaresystems ist *benutzergerecht*.
- Die im System festgelegten **Abläufe und Schritte** lassen sich je nach Anwendungssituation problemlos den tatsächlichen Erfordernissen *anpassen*.

Einführung in die Softwaretechnik © H. Züllighoven AB Softwaretechnik

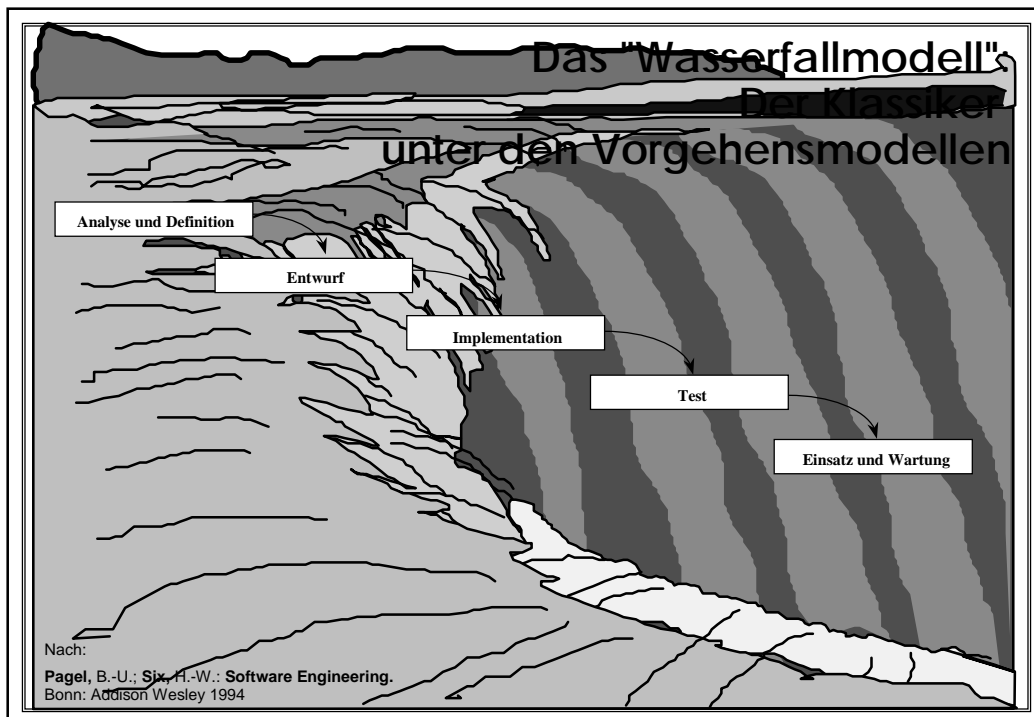
Gebrauchsqualität

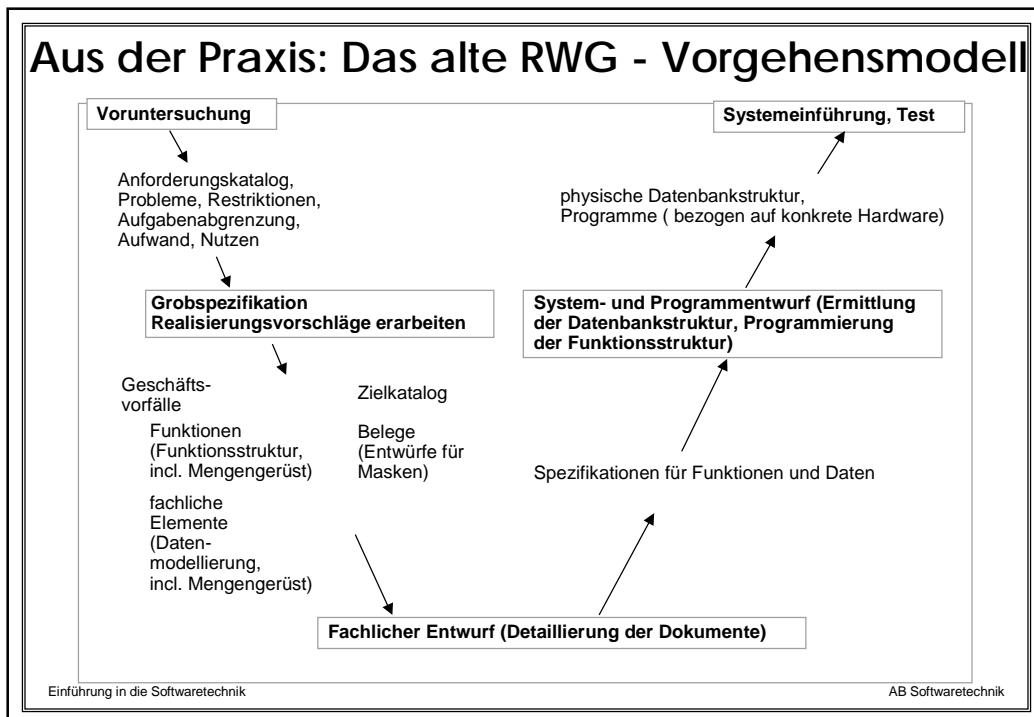
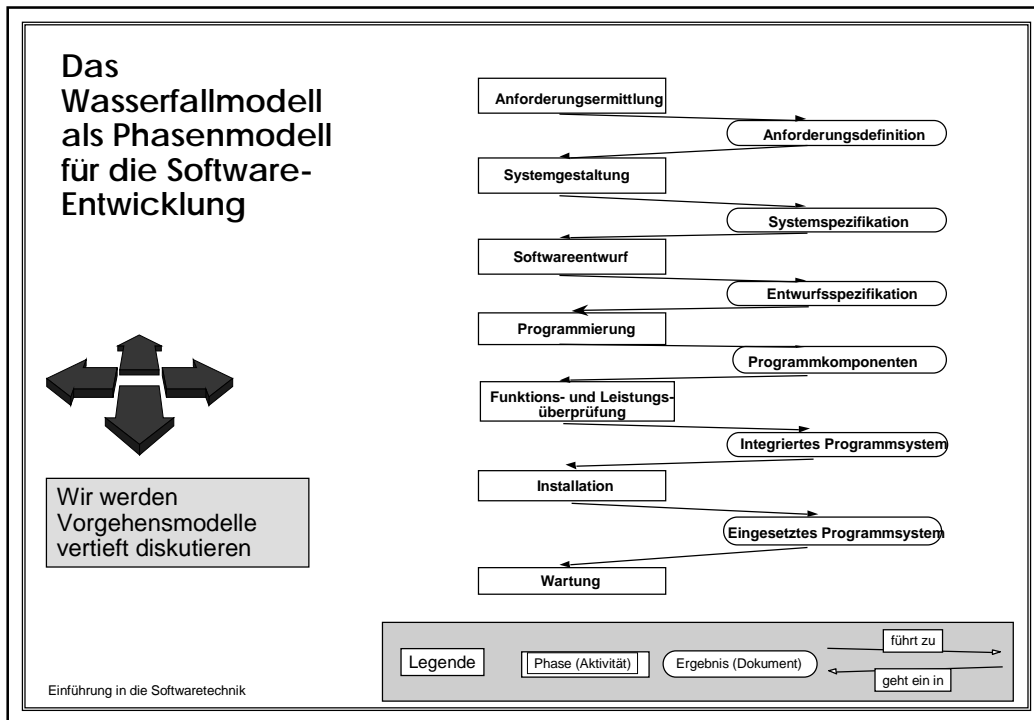
Zur Charakterisierung von Gebrauchsqualität dienen nach DIN 66234 die Kriterien

- Aufgabenangemessenheit,
- Transparenz,
- Steuerbarkeit,
- Fehlertoleranz,
- Selbstbeschreibungsfähigkeit,
- Erwartungskonformität,
- Fehlerrobustheit

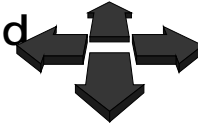
Einführung in die Softwaretechnik

AB Softwaretechnik





Konzepte für Programmierung und Spezifikation (1)



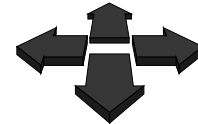
- **Abstrakte Konzepte für die imperative Programmierung**

- ursprünglich:
 - » direkt auf Von-Neumann-Rechner aufbauend
- dann:
 - » Zustandskonzept,
 - » abstrakte Steuerkonstrukte,
 - » Datentypen,
 - » Variable,
 - » parametrisierte Prozeduren,
 - » zustandsorientierte Programmierung.

- **Sprachen: vor allem Pascal**

- **sprachunabhängig: Strukturierte Programmierung**

Konzepte für Programmierung und Spezifikation (2)



- **Problem der zustandsorientierten Programmierung**

- keine *referentielle Transparenz*
(durch Zuweisung ändert sich der Wert von Variablen, ein Name kann für unterschiedliche Werte stehen)

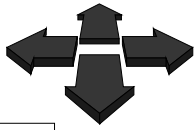
- **Lösungen**

- Alternative applikative Programmierung
- Vorstufe formale Spezifikation
- logische Zusicherungen über die Vor- und Nachbedingungen von Programm(-teilen)

- **Trennung**

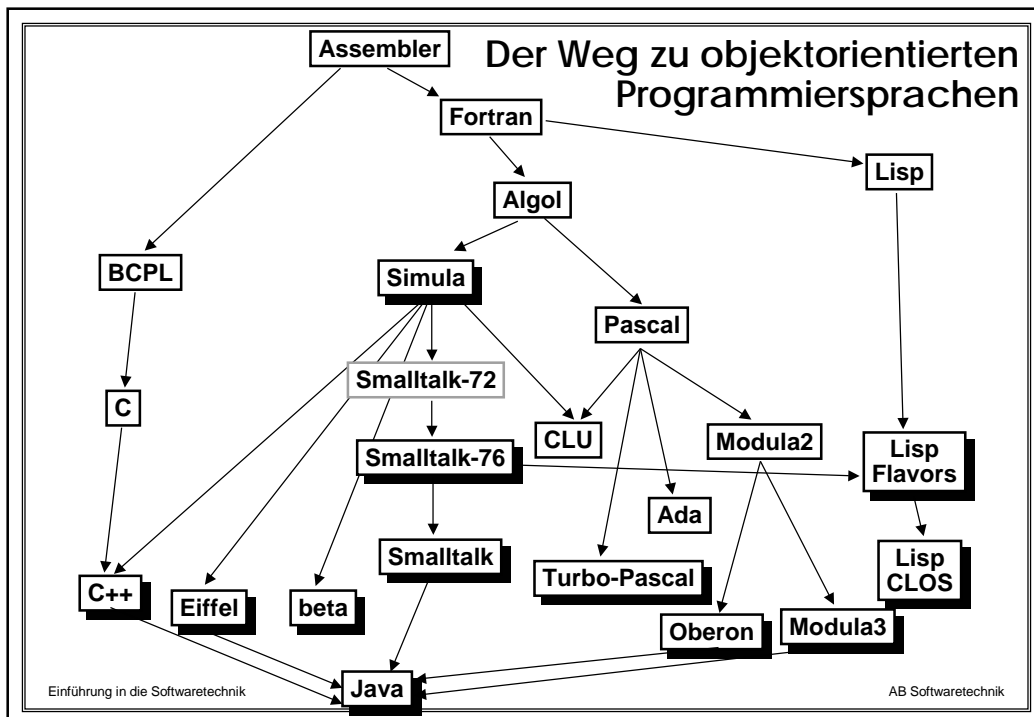
- Spezifikation \Leftrightarrow Programmierung
- Programm \Leftrightarrow Programmbeweis

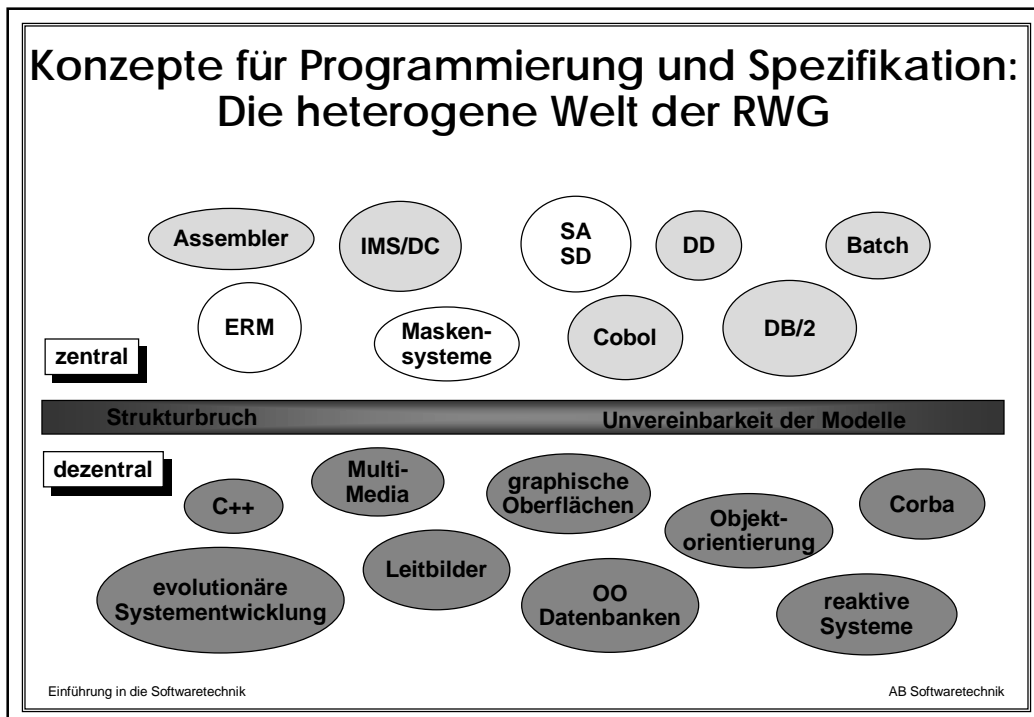
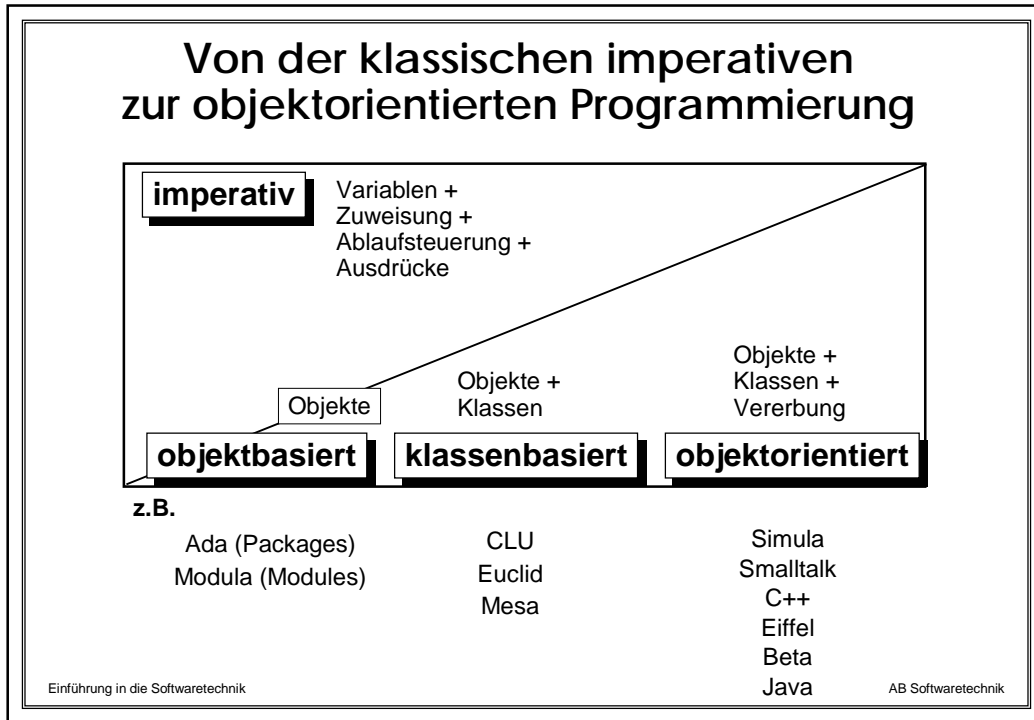
Konzepte für Programmierung und Spezifikation (3)



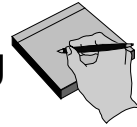
- **Abstraktionsmechanismen**
 - algorithmische (oder Prozeß-)Abstraktion: parametrisierte Prozeduren
 - Datenabstraktion: abstrakte Datentypen
- **Zerlegungskonzepte**
 - Blockstruktur: mangelhaft (globale Variablen!)
 - Modulkonzept: Kapselung, Lokalität
- **Objektorientierung**
 - Sprachen: Simula, Smalltalk, Eiffel, C++, **Java**
 - Klassen, Vererbung & Schnittstellen
 - Stufen der Objektorientierung
 - » objektbasiert / klassenbasiert / objektorientiert
 - » Objektbasierung in modularen Sprachen möglich!

Einführung in die Softwaretechnik
AB Softwaretechnik





Methoden zur Softwareentwicklung



- **Zentral ist für uns der Methodenbegriff der Softwaretechnik.**
- **Eine Methode**
 - ist eine Vorgabe für systematisches Vorgehen
 - geeignet für Softwareentwicklung oder Teilaufgabe davon
 - bietet Modellierungskonzepte
 - benennt und ordnet Teilaufgaben
 - verkörpert eine Sicht der Softwareentwicklung
 - bezieht sich auf einen Anwendungsbereich
 - gibt Richtlinien für
 - » Techniken
 - » Werkzeuge
 - » Organisationsformen

Einführung in die Softwaretechnik

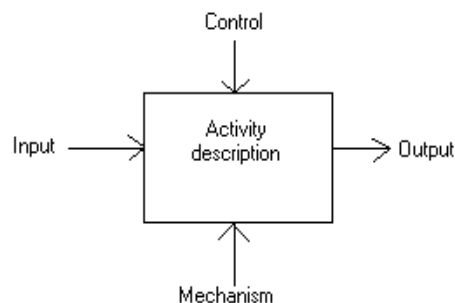
AB Softwaretechnik

Strukturierte Methoden

- **Strukturierte Methoden**
 - ablaufbezogene Funktionsmodellierung
 - Trennung von Funktions- und Datenmodellierung
 - Top-Down-Vorgehen
 - Verschiedene Modellebenen
 - Hierarchische Produktstruktur
 - Manchmal: Prozeß soll Produktstruktur entsprechen
- **Beispiele**
 - Structured Analysis/Structured Design (SA/SD)
 - Structured Analysis and Design Technique (SADT)
- **Programmierstil:** (strukturierte) zustandsorientierte Programmierung (heute veraltet)

Zum Beispiel: SADT

- **Inputs** are data items that are transformed to outputs.
- **Controls** are items such as budget and schedule that constrain the type of degree of the process being described;
- **Mechanisms** are external aids to the process, such as tools and techniques used to perform the transformation.




Einführung in die Softwaretechnik

© <http://ctr.umkc.edu/~dsparling/cs457/sadt.html>

Objektorientierte Methoden

- **Objektorientierte Methoden**
 - stellen die Gegenstände der Anwendungswelt in den Vordergrund
 - auf Gegenständen können Operationen durchgeführt werden
 - Daten und Operationen werden zusammen modelliert
 - keine starren Abläufe
 - einheitliche Modellbasis für alle Ebenen
 - Bottom-up-Vorgehen durch Bildung von Bibliotheken und Frameworks
- **Programmierstil:** objektbasiert oder objektorientiert
- **Beispiele:** OOA/OOD, **UML**, WAM



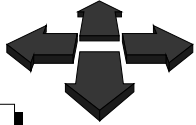
Einführung in die Softwaretechnik AB Softwaretechnik

Werkzeuge für die Softwareentwicklung


- **Einzelwerkzeuge**
 - zur Herstellung und Transformation von Dokumenten (Programmen und Texten)
 - zur Überprüfung von Dokumenten und Programmen
 - zur Unterstützung von Entwicklungsschritten
 - zur Umsetzung von Methoden
- **Entwicklungsumgebungen**
 - sprachbasiert,
 - betriebssystembasiert,
 - methodenbasiert (z.B. Rational Rose)
- **Produktverwaltung:**
 - Projektbibliothek,
 - Versions- und Konfigurationsverwaltung

Einführung in die Softwaretechnik AB Softwaretechnik

Organisationsformen

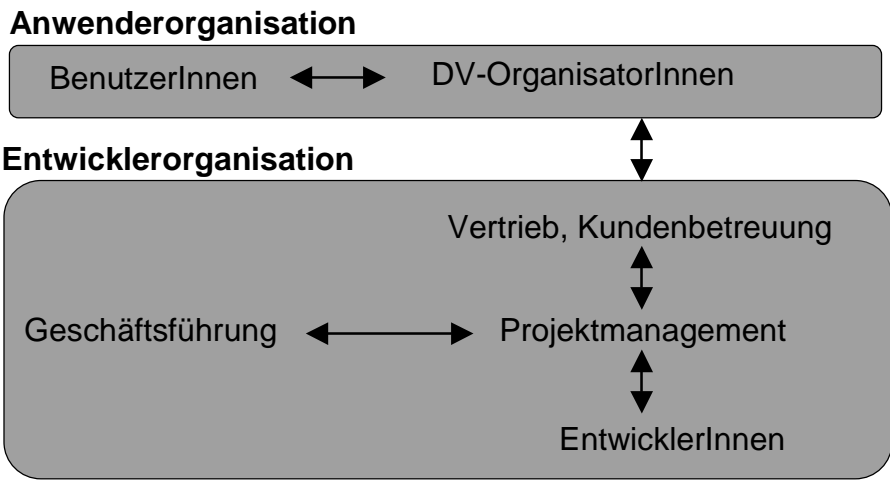


- **Projektmodelle**
 - Phasenmodell
 - Spiralmodell
 - zyklisches Modell
- **Team-Modelle**
 - Chef-Programmiererteam
 - Demokratisches Team
- **Prozeßkoordination**
 - Projektetablierung
 - Autor-Kritiker-Zyklus
 - Reviews
 - Meilensteine und Referenzlinien



Einführung in die Softwaretechnik AB Softwaretechnik

Exkurs: Traditionelle Organisations- und Kommunikationsstrukturen



Anwenderorganisation
BenutzerInnen ↔ DV-OrganisatorInnen

Entwicklerorganisation
Geschäftsführung ↔ Projektmanagement ↔ EntwicklerInnen
Vertrieb, Kundenbetreuung ↔ Projektmanagement

Einführung in die Softwaretechnik AB Softwaretechnik

Produktionssicht & Designsicht

- Software = Produkt aus Programmen und Dokumenten
- setzt feste Anforderungen um
- trennt Herstellung von Einsatz
- Herstellung = lineare Folge von Phasen
- Zwischenergebnisse: Dokumente
- Ergebnis = Produkt
- während des Einsatzes: "Wartung"



- Softwareprodukt im (Einsatz-)Kontext
- Anforderungen wandelbar
- Herstellung und Einsatz verschränkt
- Entwicklungszyklen mit Rückkopplung
- Zwischenergebnisse: auch Prototyping
- Ergebnis = Folge von Versionen
- statt "Wartung": versionsorientierte Weiterentwicklung



Einführung in die Softwaretechnik AB Softwaretechnik

Der Stoff der Lehrveranstaltung

- **Konzepte**
 - Abstrakte Datentypen
 - Modulare (objektbasierte), objektorientierte und komponentenbasierte Zerlegung
 - Vertragsmodell und Zusicherungen
- **Methoden**
 - Rahmen: Ansatz STEPS, evolutionäre, partizipative Systementwicklung
 - aufgabenbezogene Anforderungsermittlung
 - objektorientierte Methoden
- **Werkzeuge**
 - unix-basierte Programmierumgebung
 - Java-Entwicklungssystem
 - Fensterverwaltungssystem

Einführung in die Softwaretechnik AB Softwaretechnik