

# 15. Projektplanung

1

Prof. Dr. rer. nat. Uwe Aßmann  
Lehrstuhl Softwaretechnologie  
Fakultät Informatik  
TU Dresden  
Version 11-0.2, 01.05.11

## 1. Projektstruktur

1. Einführung
2. Projektstrukturplanung

## 2. Ablaufplanung

## 3. Aufwandsschätzung

1. Delphi
2. Function Point
3. CoCoMo

## 4. Terminplanung

## 5. Ressourcenplanung

## 6. Kostenplanung

Softwaremanagement, © Prof. Uwe Aßmann



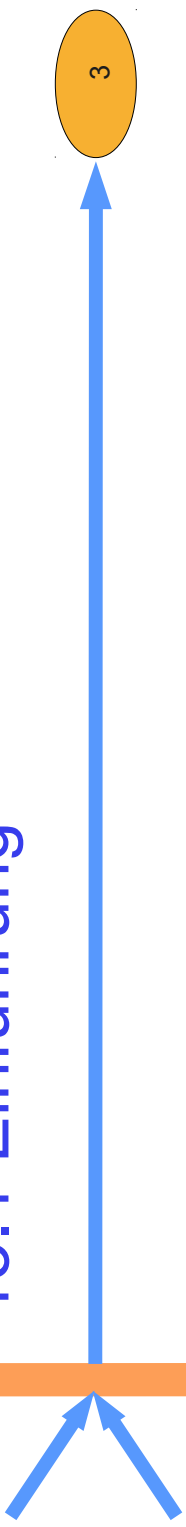
# Referenzierte Literatur

2

- ▶ Mayr, H.: Projekt Engineering - Ingenieurmäßige Softwareentwicklung in Projektgruppen; Fachbuchverlag Leipzig 2001
- ▶ Zuser, W., Grechenig, T., Köhle, M.: Software-Engineering mit UML und dem Unified Process (2. Auflage); Pearson Studium 2004
- ▶ Burghardt, M.: Projektmanagement. Leitfaden für die Planung, Überwachung und Steuerung von Entwicklungsprojekten; Publicis MCD Verlag 1997
- ▶ Poensgen, B., Bock, B.: Function-Point-Analyse; dpunkt.verlag 2005

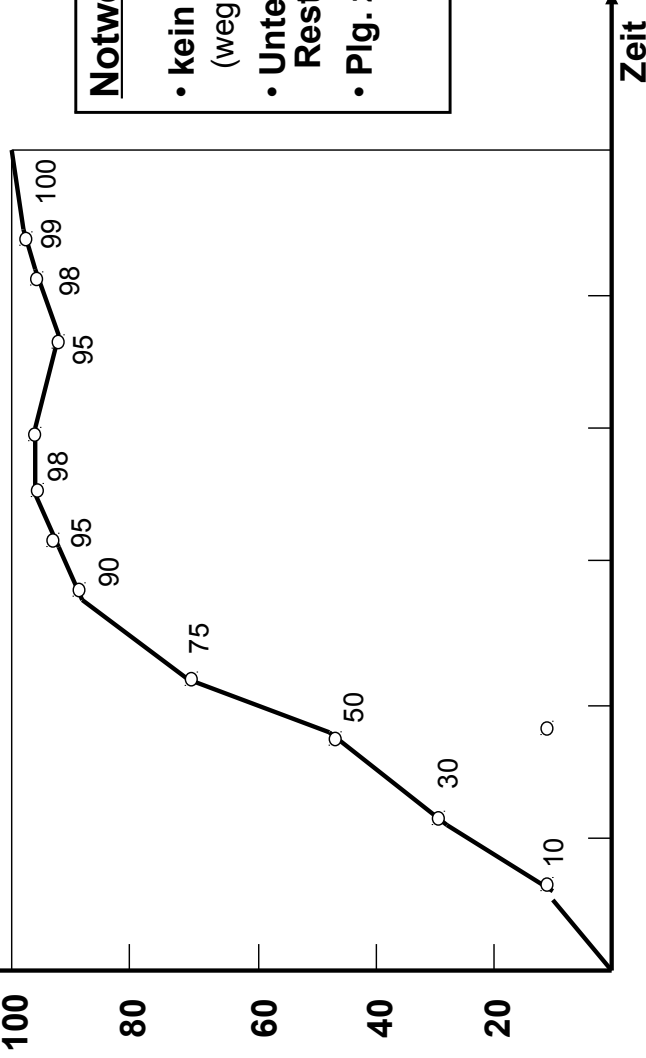


# 15.1 Einführung



Softwaremanagement, © Prof. Uwe Alßmann

# Das 90%-Syndrom nach Boehm (subjektive Einschätzung der Fertigstellung)



## Notwendigkeit der Plg.:

- kein Überblick (wegen Komplexität)
- Unterschätzung des Restaufwandes
- Plg. zu optimistisch

Der Fertigstellungsgrad wird während der Hälfte der Projektlaufzeit größer als 95% eingeschätzt!

Quelle: Deutsche Informatik Akademie



# Aufgaben der Projektplanung

Die **Projektplanung** muss im Projektplan die vorhandenen Ressourcen an Personal, Zeit, Geld, Maschinen, Räume so einteilen, dass auf Änderungen der Arbeitsabläufe rasch und kosteneffizient reagiert werden kann.

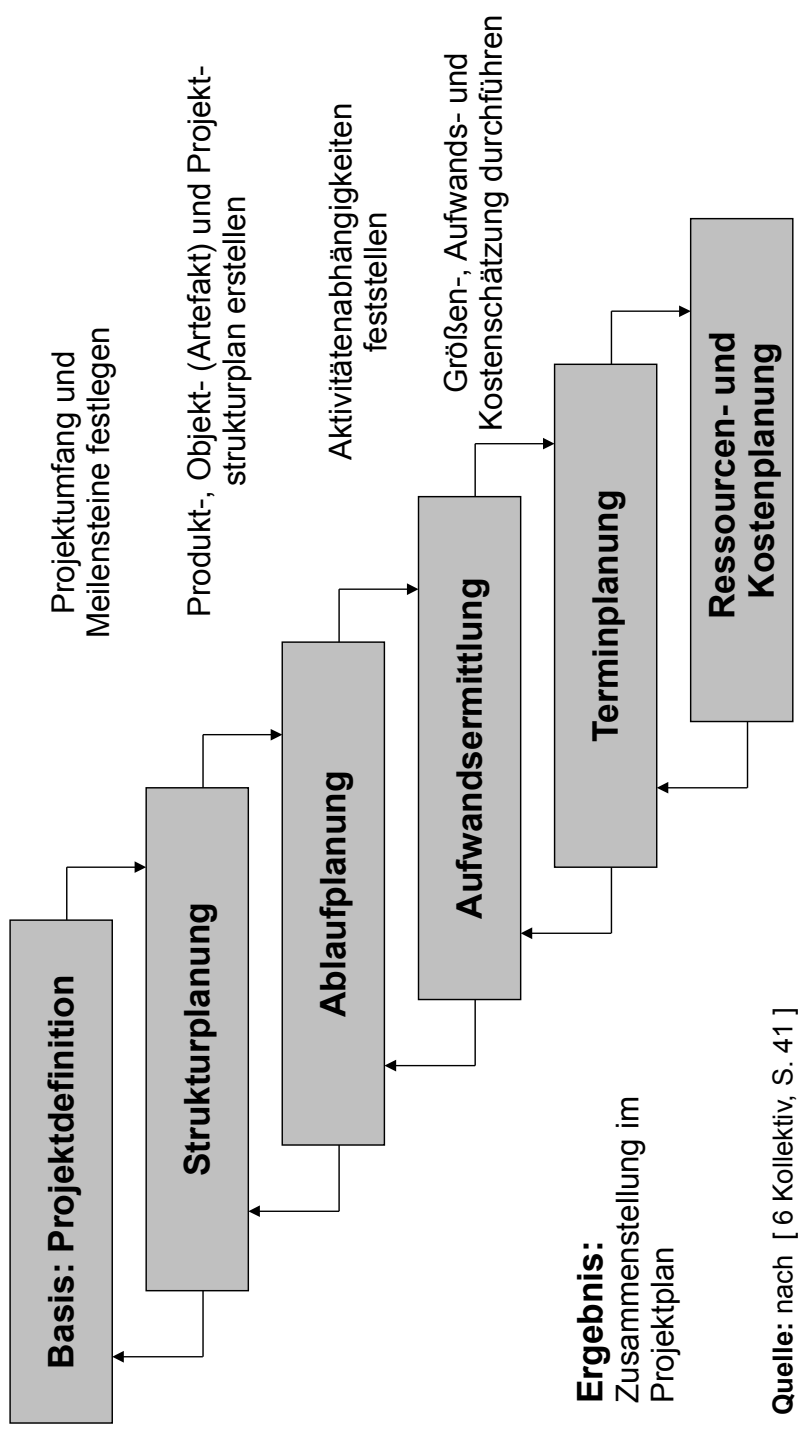
- ▶ Technische Planung:
  - Auswahl eines Modells der Ablauforganisation, nach dem alle zu erstellenden Zwischen- und Endprodukte für das Projekt bestimmt werden. Ziel ist es, einen (idealen) Plan zur Minimierung der Prozessrisiken zu finden, auf dessen Basis mit dem AG(Kunden) verhandelt werden kann.
- ▶ Qualitätsplanung:
  - Planung der Maßnahmen für jedes Qualitätskriterium. Überprüfung der Methoden des technischen Plans auf Brauchbarkeit.
- ▶ Wirtschaftliche Planung (Kostenplanung):
  - Planung von Personal, Ressourcen und der Finanzierung der dabei anfallenden Kosten. Projektrisiken können speziell abgesichert werden.

# Ziele der Projektplanung

- ▶ Ermittlung realistischer Sollvorgaben
  - für zu erbringende Arbeitsleistung, die Termine, den Ressourceneinsatz, die Kosten sowie den Einzelschritten der Projektdurchführung
- ▶ Vorbereiten des Controlling (Steuerung)
  - Verbesserung der Effizienz der Projektentwicklung
  - Frühe Korrektur
  - Fehlererkennung
  - Dokumentation der Vorgaben

# Übersicht Planungsschritte

7

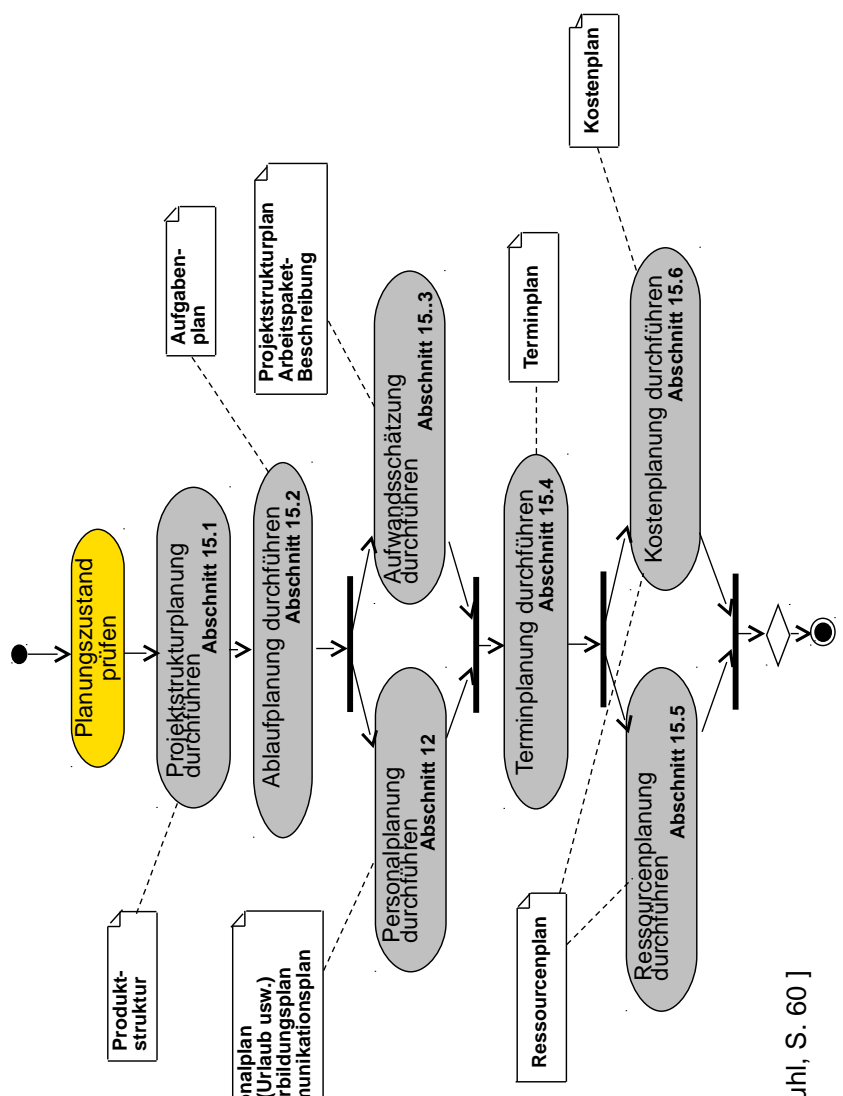


Quelle: nach [ 6 Kollektiv, S. 41 ]



# Aktivitätsdiagramm der Planungsphase

8



Quelle: [ 4 Buhl, S. 60 ]



# Planungszustand prüfen:

## Verschiedene Planungshorizonte

- Perspektivplanung (>1 Jahr)
- Langfristige Planung (ca.1 Jahr)
- Mittelfristige Planung (ca. 6 Monate)
- Kurzfristige Planung (ca. 1 Monat)
- Operative Planung während Projektsteuerung/-Überwachung

Anstoß	Studie	Projektierung	Design	Implementierung	Syst.Integrat/-test	Abnahme
--------	--------	---------------	--------	-----------------	---------------------	---------

Projektplan 0

Phasenplan 1

Projektplan 1

Phasenplan 2

Projektplan 2

### Legende:

▲ Verabschiedung

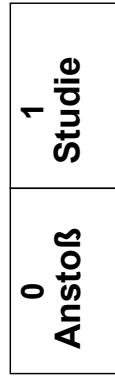
Quelle: Deutsche Informatik Akademie



Genauigkeit wächst

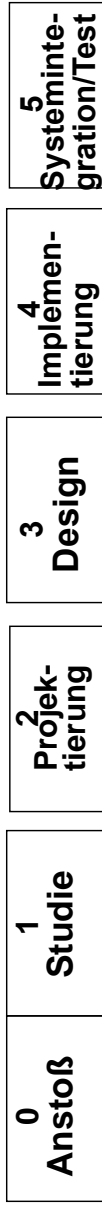
## Reichweite der Aussagen der Planung

Projekte mit geringer Innovation und geringem Risiko



◆ Ab Pflichtenheft ist das Projekt ausreichend genau zu überblicken

Projekte mit hoher Innovation und hohem Risiko



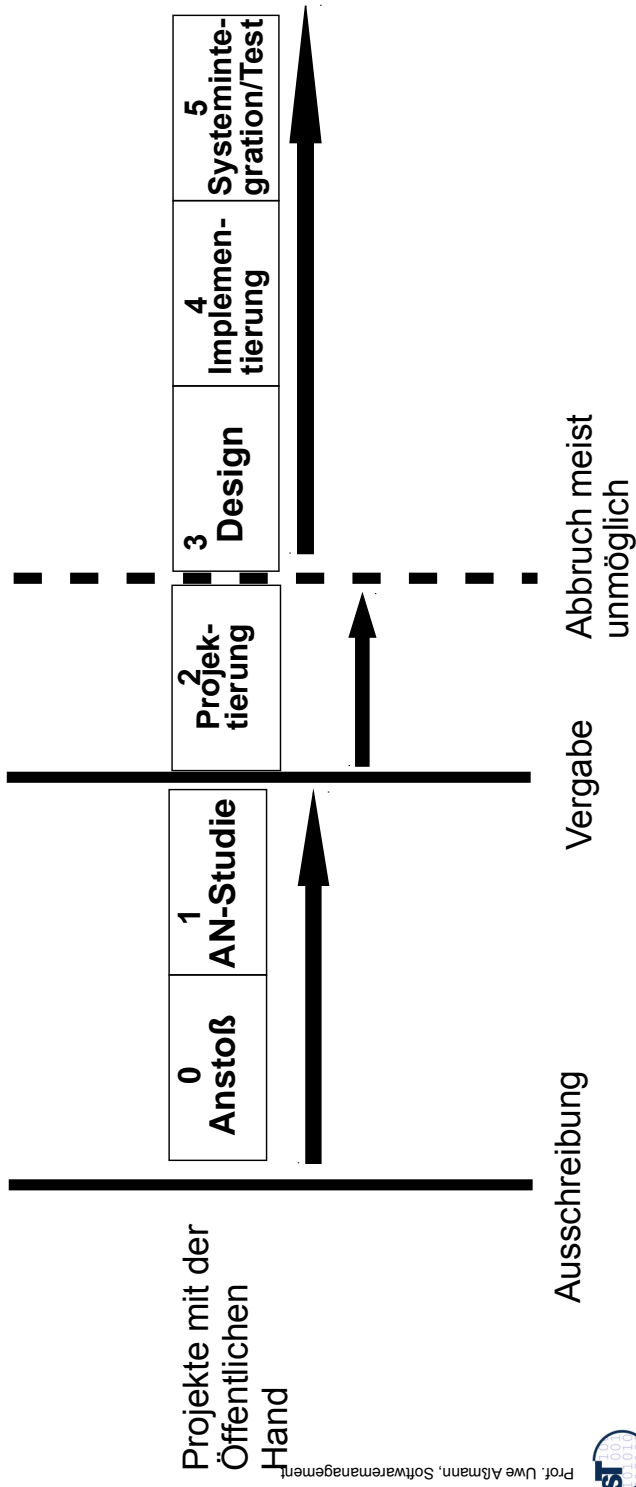
- ◆ Es kann nur jeweils ein Horizont von 10% bis 20% des Gesamtaufwandes ausreichend genau überblickt werden
- ▶ evtl. **Teilverträge, Teilprojektierung**



Quelle: Deutsche Informatik Akademie

# Projekte mit der öff. Hand

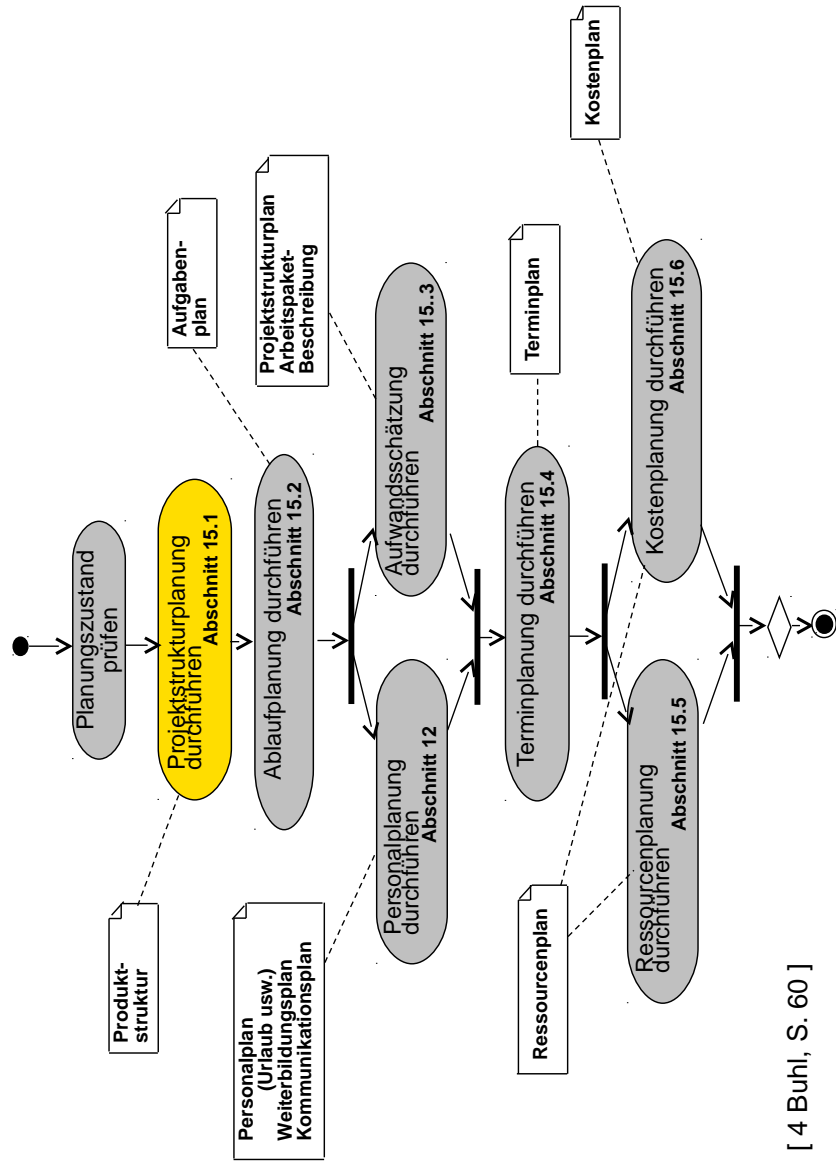
- ▶ Projektierungsphase mit separatem Teilvertrag enorm wichtig
  - meist aber nicht durchgeführt...
  - Ausschreibung verlangt Festpreiskalkulation



## 15.1.2 Strukturplanung

# Aktivitäten während der Planungsphase

13



Quelle: [ 4 Buhl, S. 60 ]

# Projektstrukturplanung

14

(End-)Produktstruktur

Objektstruktur

Projektstruktur

♦ Aus welchen **Komponenten** besteht das (End-)Produkt?

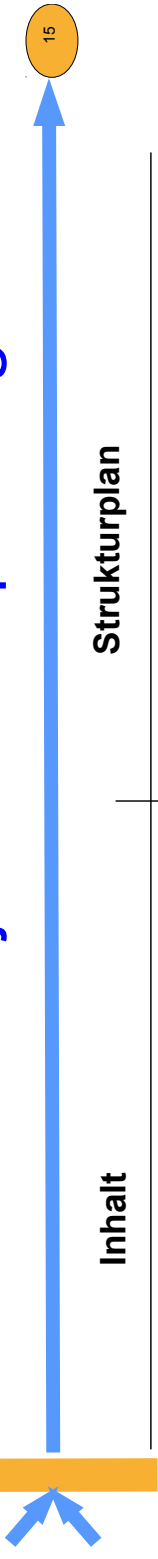
- Welche **Artefakte** (Objekte, Arbeitsergebnisse, Arbeitsprodukte) werden fürs Produkt gebaut?
  - Zwischenergebnisse (z. B. Prototypen)?
  - Entwicklungsdokumente?
  - Hilfsmittel, Tools, Vorrichtungen, Messgeräte?
  - Steuerungsergebnisse (Pläne, Berichte)?
- Welche generellen Untersuchungen?

- ♦ Welche **AP** zur Erstellung der Objekte?
- ♦ Welche AP der „Projektfunktionen“?
- ♦ Welche AP sind voneinander abhängig?
- ♦ Welche AP nebenläufig durchführbar?

Quelle: Deutsche Informatik Akademie

AP: Arbeitspaket

# Schritte der Projektstrukturplanung



## 1. Das zu liefernde **End-Produkt**

- + 2. Zur Erstellung des End-Produktes notw. **Artefakte (Hilfsmittel, Arbeitsergebnisse, Zwischenergebnisse)**
- + 3. Für die Abwicklung des Projektes nötige **Aufgaben**

## 1 = **Produktstrukturplan**

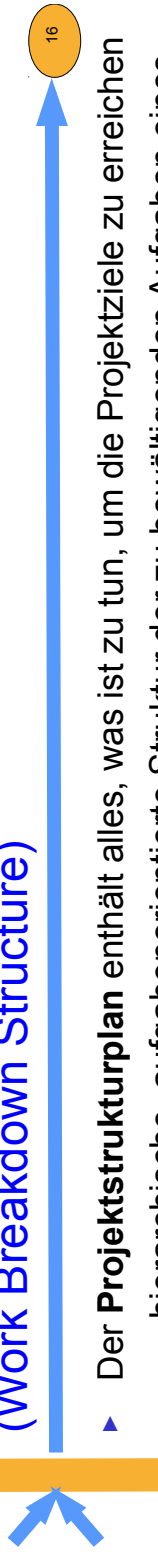
1 + 2 = **Objektstrukturplan**  
(**Artefaktstrukturplan**)

3 = **Arbeitspakete**

1 + 2 + 3 = **Projektstrukturplan**

**Hinweis:** Es kann von Größe und Art der Projekte abhängen, wie detailliert alle Schritte zu durchlaufen sind.

# Projektstrukturplan (PSP, WBS) (Work Breakdown Structure)



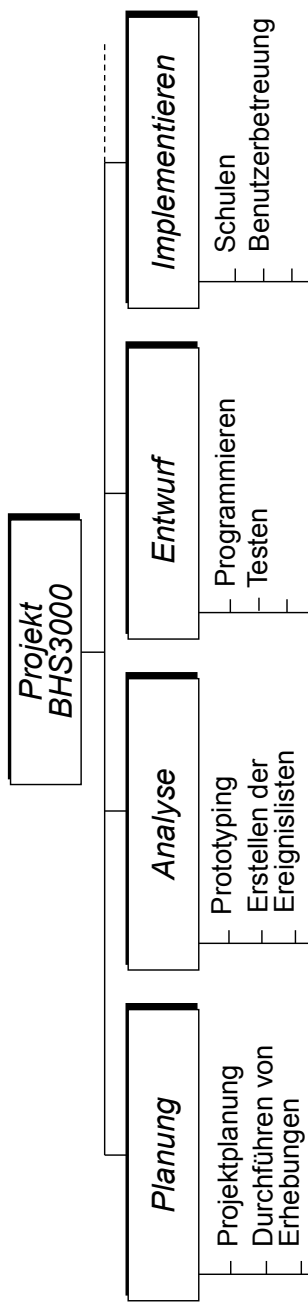
- ▶ Der **Projektstrukturplan** enthält alles, was ist zu tun, um die Projektziele zu erreichen
  - hierarchische, aufgabenorientierte Struktur der zu bewältigenden Aufgaben eines Projekts
  - Zentrales Kommunikationsinstrument
  - Basis für die Termin-, Ressourcen-, Kostenplanung, die Projektdokumentation, die Projektkontrolle und -steuerung
  - Relativ stabiles Planungsinstrument (Termin- und Kostenänderungen haben darauf keinen Einfluss)
- ▶ Darstellung als Strukturdiagramm (Baum) oder als Gliederungsliste. Der PSP gliedert sich in 3 Ebenen:
  - **1. Ebene:** Projektbezeichnung
  - **2. Ebene:** Strukturierung des Projektes nach verschiedenen Gliederungsebene-punkten (z.B. Funktionen, Phasen, Objekten ...)
  - **3. Ebene:** Arbeitspakete



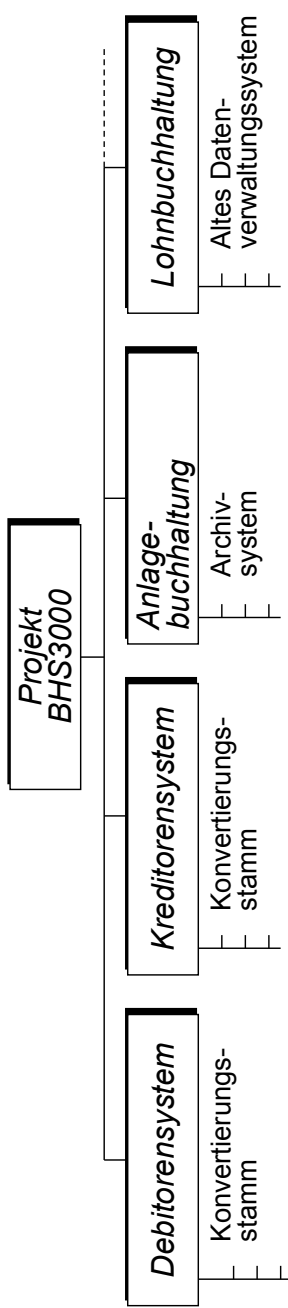
# Dekompositionskriterien von Projektstrukturplänen

17

► Funktionsorientierter Projektstrukturplan dekomponiert anhand Funktionen



# Komponentenorientierter Projektstrukturplan



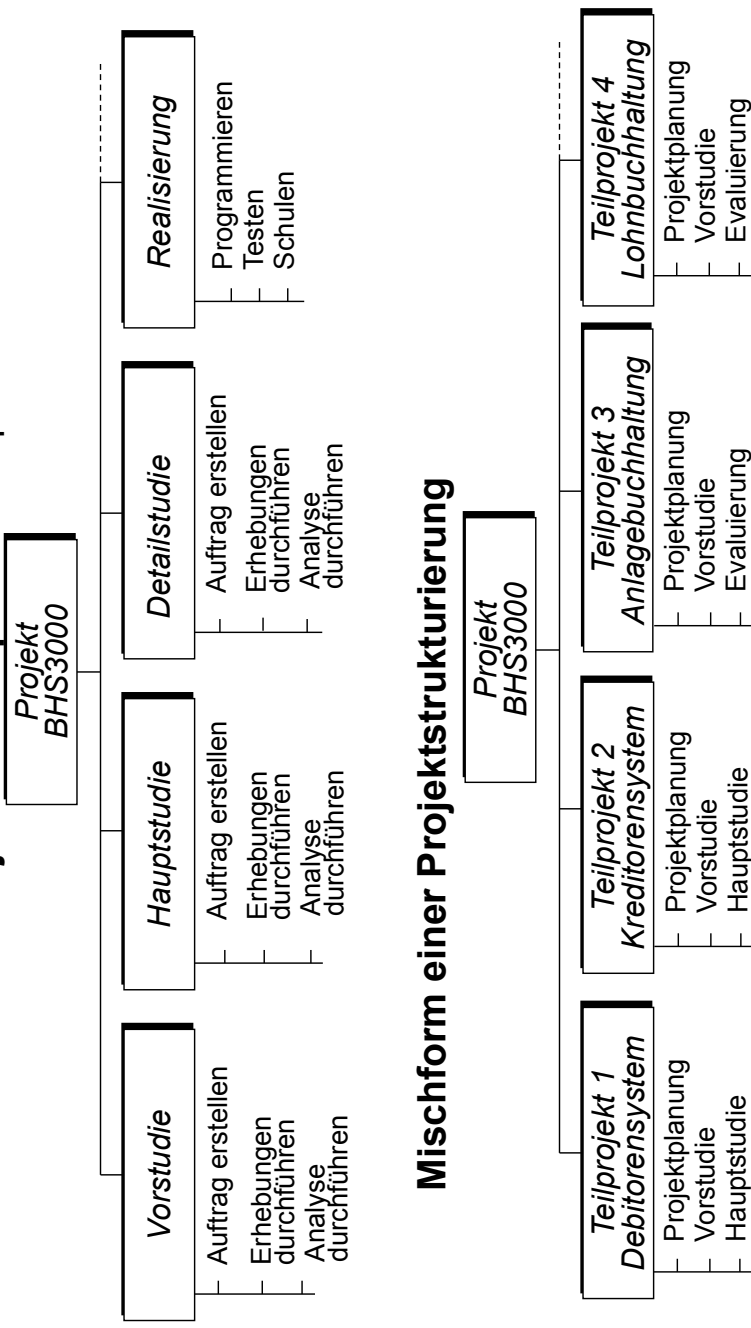
[1 Jenny, S. 194/195]



# Dekompositionskriterien von Projektstrukturplänen ctd.

18

► Ablauforientierter Projektstrukturplan dekomponiert anhand Aktivitäten



[1 Jenny, S. 213/214]



# Arbeitspakete

(Inhalt des Aktivitätenstrukturplanes)

Ein **Arbeitspaket (AP, Aktivität)** ist ein in sich geschlossene Aufgabenstellung innerhalb eines Projekts, die bis zu einem festgelegten Zeitpunkt mit definiertem Ergebnis und Aufwand vollbracht werden kann [DIN 69901-5]

19

- ▶ Randbedingungen eines AP:
  - selbständige Erledigung durch organisatorische Einheit oder Person
  - Ein Arbeitspaket kann zur besseren Strukturierung aufgliedert werden
- ▶ Ziel:
  - planbare Arbeitsvolumen (überschaubar, abrechenbar)
  - eigenverantwortliche Durchführung
  - Projektverfolgung
- ▶ Arbeitspakete (AP) sind Grundlage für:
  - Aufwandsermittlung
  - Erstellung des Netzplanes
  - Erteilung von internen Aufträgen
  - Ergebnisbeschreibung

# Definition eines Arbeitspaketes

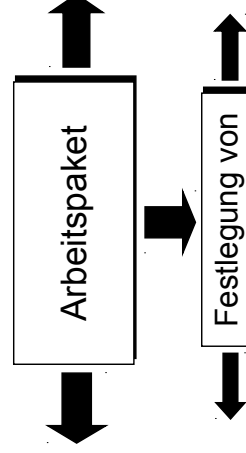
20

Zuordnung zu:

- Produkt oder Baugruppe
- Entwicklungsphase
- Netzplan

Zuordnung zu:

- Objekt(Artefakt)struktur
- Projektstruktur
- Kostenstruktur



- Aufwand
- Termin
- Dauer
- Teilprojektteam

- Meilensteinkennzeichen
- Teilprojektkennzeichen
- Einsatzmittel
- Anordnungsbeziehung

# Eigenschaften von Arbeitspaketen

- ▶ Disjunkt
  - Arbeitspakete müssen klar voneinander abgegrenzt sein
  - Es darf keine Überschneidungen geben
- ▶ Jedes Arbeitspaket hat genau einen Verantwortlichen
- ▶ CCC (checkable, consistent, complete)
- ▶ Abhängigkeiten
  - Arbeitspakete sind voneinander abhängig

[anlehnend Vorl. Prof. S. Seibert]



# Beispiel Arbeitspaket-Spezifikation

<b>Projektnummer:</b> <b>Projektname:</b>	Seite ..... von .....
<b>Arbeitspaket-Nummer:</b>	z.B.: 1.1.1 Projektstart durchführen im Organisationsprojekt
<b>Inhalt:</b>	Projektdefinition, Projektstrukturplan mit Definition der Arbeitspakete, Termin- und Kostenplanung, Zweitägigen Projektstartworkshop mit Projektteam, Zustimmung für Konzeptionsprojekt mit einem konkreten Projektauftrag durch den Projektauftraggeber, Festlegung eines Projektnamens und Projektlogos
<b>Nicht-Inhalt:</b>	Zusammenstellung Projektteam
<b>Ergebnisse:</b>	Klare Ziele, Erstsatz Projekthandbuch, Zustimmung von Projektauftraggeber
<b>Leistungsfortschrittsmessung:</b>	40% Zieldefinition, 40% Projekthandbuch, 20% Zustimmung Projektauftraggeber
<b>Verantwortlich:</b>	Frau Mayer X.
<b>Dauer und terminliche Lage:</b>	14.8. - 13.9. ...
<b>Zeitaufwand/Ressourcen:</b>	64 Std./P- Teammitglied ohne Projektleiter/in plus 80 Std. Projektleiter/in
<b>Kosten des vorliegenden AP:</b>	.....€ Personalkosten plus 1.200,-€ Tagungs- und Übernachtungskosten
<b>Abhängigkeit/Schnittstellen zu anderen Arbeitspaketen/ Projekten:</b>	[B.C. Schreckeneder]



# 15.2 Ablaufplanung

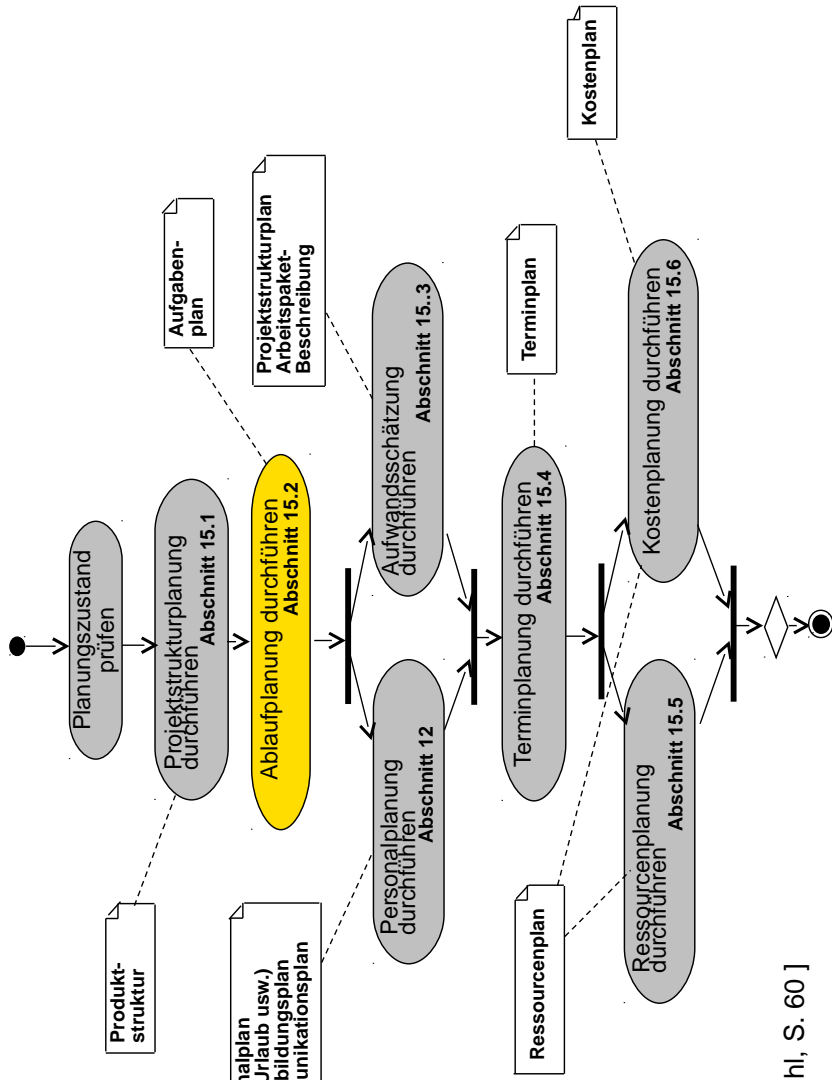
23



Softwaremanagement, © Prof. Uwe Alßmann

# Aktivitäten während der Planungsphase

24



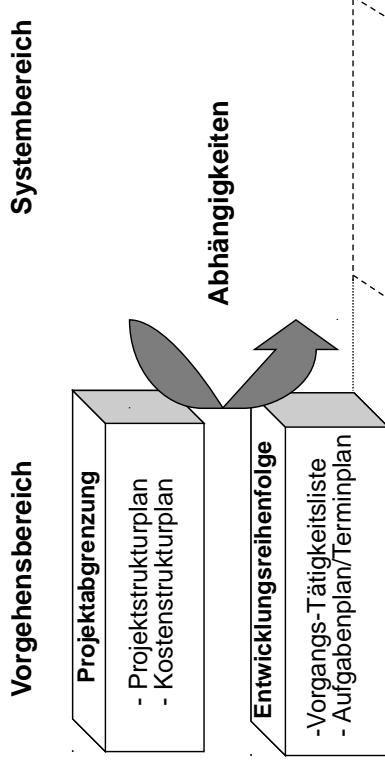
Quelle: [ 4 Buhl, S. 60 ]



# Ablaufplanung (Abhängigkeitsanalyse der Teilaufgaben)

25

- Die **Ablaufplanung** soll die **logischen Abhängigkeiten** der Projektteilaufgaben, d.h. die Ablaufreihenfolge logisch und verständlich präsentieren.
  - Bausteine der Ablaufplanung sind die **Arbeitspakete**.
  - Aus den Arbeitspaketen wird eine **Vorgangsliste** abgeleitet.
  - Auf dieser Grundlage wird der **Aufgabenplan/Terminplan** erstellt.

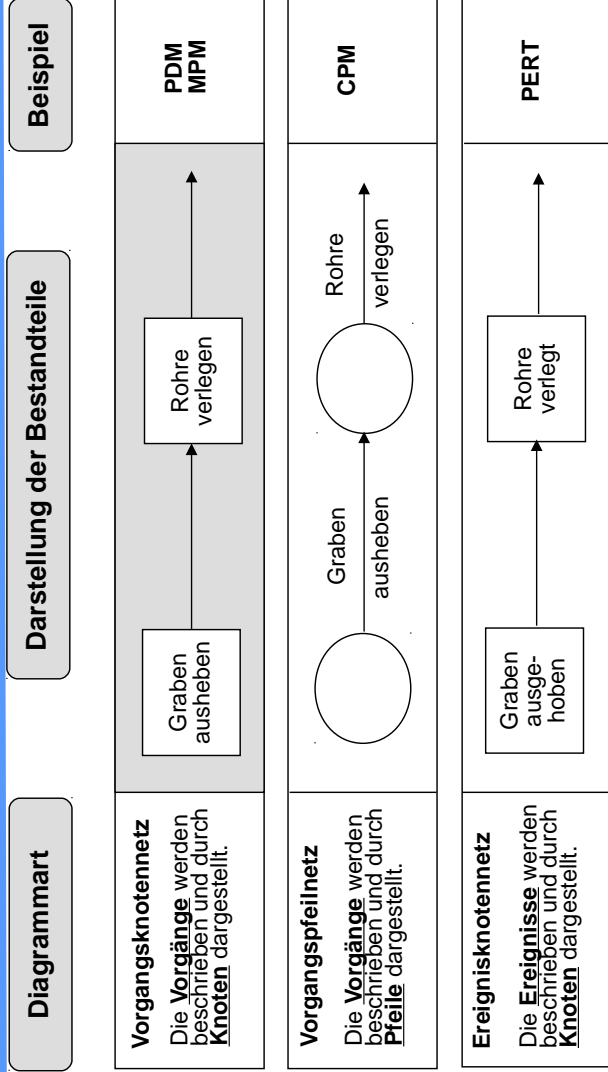


[ 1 Jenny ]



# Darstellungen von Aktivitäten

26



- Legende:
- PDM** : Precedence Diagramm Method (auch MS Project)
  - MPM** : Metra Potential Method
  - CPM** : Critical Path Method
  - PERT** : Program Evaluation and Review Technique

Quelle: [ 1 Jenny ]



# Abhängigkeitsdiagramm (Abhängigkeitsgraph)

27

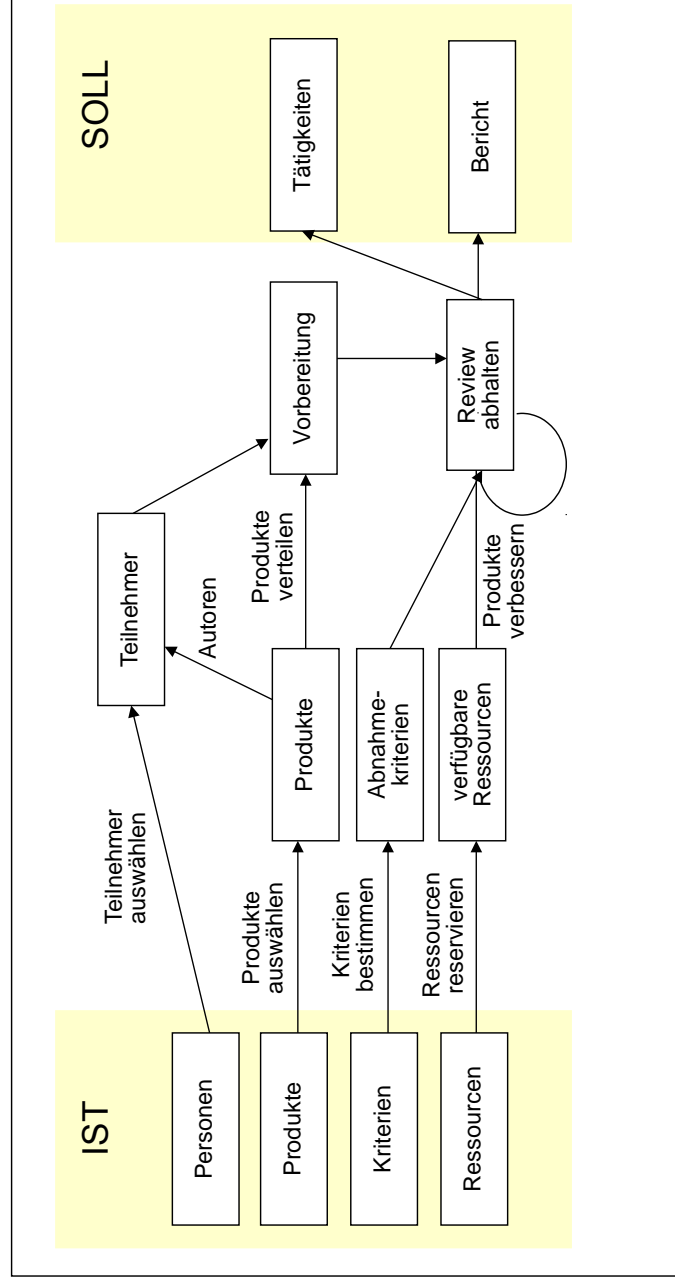
Das **Abhängigkeitsdiagramm** veranschaulicht die Abhängigkeiten von Aktivitäten, indem Zustände und Ressourcen als **Knoten** eines Graphen dargestellt werden.

- ▶ **A** → **B** heißt, Aktivität **A** ist Voraussetzung für Aktivität **B** oder auch Ressource **A** ist notwendig für Aktivität **B**
- ▶ Auf der linken Seite die Ressourcen im **Ist-Zustand**
- ▶ Auf der rechten Seite die Aktivitäten als **Soll-Zustand**
- ▶ Dazwischen stehen die nötigen **Teilprodukte** (Artefakte) und **Aktivitäten**, die für den Übergang vom Ist-Zustand in den Ziel-Zustand notwendig sind.
- ▶ Zweck:
  - Ordnen von Gedanken und Handlungsoptionen
  - Finden von Teilprodukten und Aktivitäten
- ▶ Die Überprüfung der Abhängigkeitsdiagramme erfolgt in der Regel durch **Reviews**.
- ▶ Es existieren keine wesentlichen syntaktischen Einschränkungen bei der Erstellung von Abhängigkeitsdiagrammen
- ▶ Genauigkeit ist von den momentanen Erfordernissen abhängig
- ▶ Für eine Machbarkeitsstudie genügt ein Übersichtsplan

Quelle: [ Zuser, W. S. 120ff. ]

# Bsp.: Abhängigkeitsdiagramm(-graph) eines Review als Vorgangspfeil-Netz

28



Quelle: [ Zuser, W. S. 122 ]

# Vorgangsliste

29

- ▶ Besteht Klarheit über die benötigten konkreten Aktivitäten (AP, Vorgänge), wird eine **Vorgangsliste (Aktivitätenliste)** in Form einer Tabelle aufgestellt.
  - Die Tabelle enthält alle Aktivitäten (Vorgänge) des Projektes inkl. Verantwortlichen und zugeordneten Mitarbeitern
  - Für jede Aktivität ist die Dauer bzw. der Aufwand (Tage geplant, bisher getan, noch zu tun) zu schätzen und die benötigten Ressourcen sind zuzuordnen
  - Welche Aktivitäten sind unmittelbare Voraussetzung, welche können unmittelbar auf die betrachtete Aktivität folgen?
  - Welche Aktivitäten können unabhängig voneinander ausgeführt werden?
  - Entspricht der Feinheitgrad der Aktivitäten den Anforderungen?
- ▶ Jede Aktivität ist mit einer Priorität (ABC) versehen, und sie werden hierarchisch nummeriert
- ▶ Das notwendige Ausmaß an Training für Mitarbeiter ist im Zeitrahmen und Projektbudget berücksichtigt
- ▶ Als praktisches Tool zur Erstellung und Verwaltung der Vorgangsliste bietet sich z.B. ein Textverarbeitungsprogramm mit Gliederungsansicht an



Quelle: [ 1 Jenny, S. 241]

# Bsp.: Vorgangsliste (0)

30

Vorgangsliste		Projekt: Aussteller:		Nr.: Datum:		Seite:								
Projekt-tätigkeit	Nr.	Vorgangszeitpunkte			Vorgang Dauer	Direkter Vorläufer	direkter Nachfolger	Pufferzeiten		Bedarf				
		FA	SA	FE				SE	GP	FP	UP	MA	SM	
Arbeitspaket 01	A						B,C,D							
Arbeitspaket 02	B					A	E							
Arbeitspaket 03	C					A	E							
Arbeitspaket 04	D					A	E							
Arbeitspaket 05	E					B,C,D	G							
Arbeitspaket 06	F					F	I							
Arbeitspaket 07	G													
Arbeitspaket 08	H					H	K							
Arbeitspaket 09	I					I								
Arbeitspaket 10	K													

FA = frühestmöglicher Anfang des Vorgangs  
 SA = spätestzulässiger Anfang des Vorgangs  
 SE = spätestzulässiges Ende des Vorgangs  
 FE = frühestmögliches Ende des Vorgangs

GP = Gesamte Pufferzeit  
 FP = Freie Pufferzeit  
 UP = Unabhängige Pufferzeit

MA = Personal (Mitarbeiter/Mitarbeiterin)  
 SM = Sachmittel (pro Vorgang)

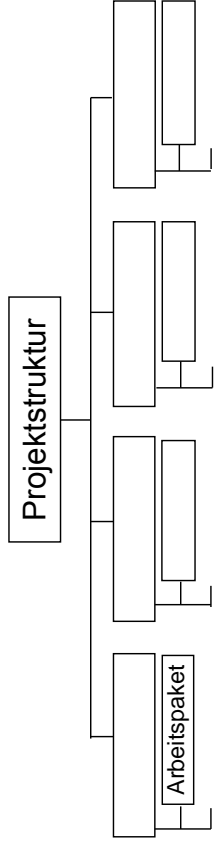
Quelle: [ 1 Jenny, S. 242 ]





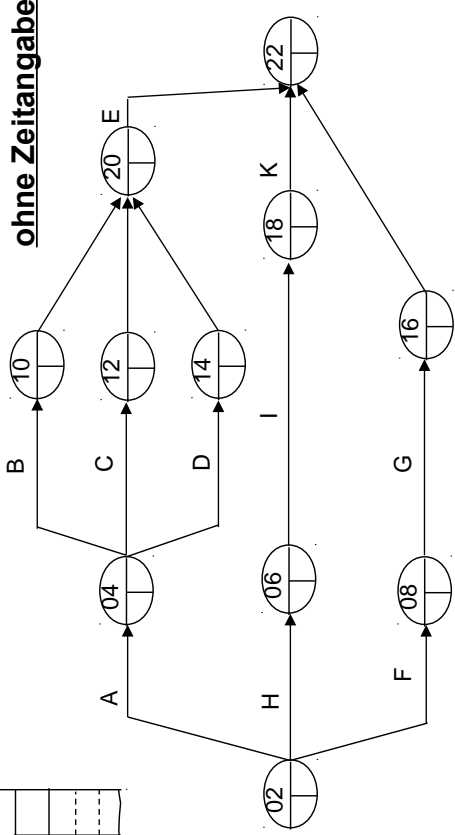
# Ergebnisse aus der Ablaufplanung

31



Vorgangs-Liste	Projekt-Aussteller:	Nr.:	Datum:	Seite:
Projektaktivitäten				
Arbeitspakete				

## ohne Zeitangaben



Quelle: [Burghardt, M., S. 227]

# 15.3 Aufwandsschätzung (-ermittlung)

32



Parkinson's Law

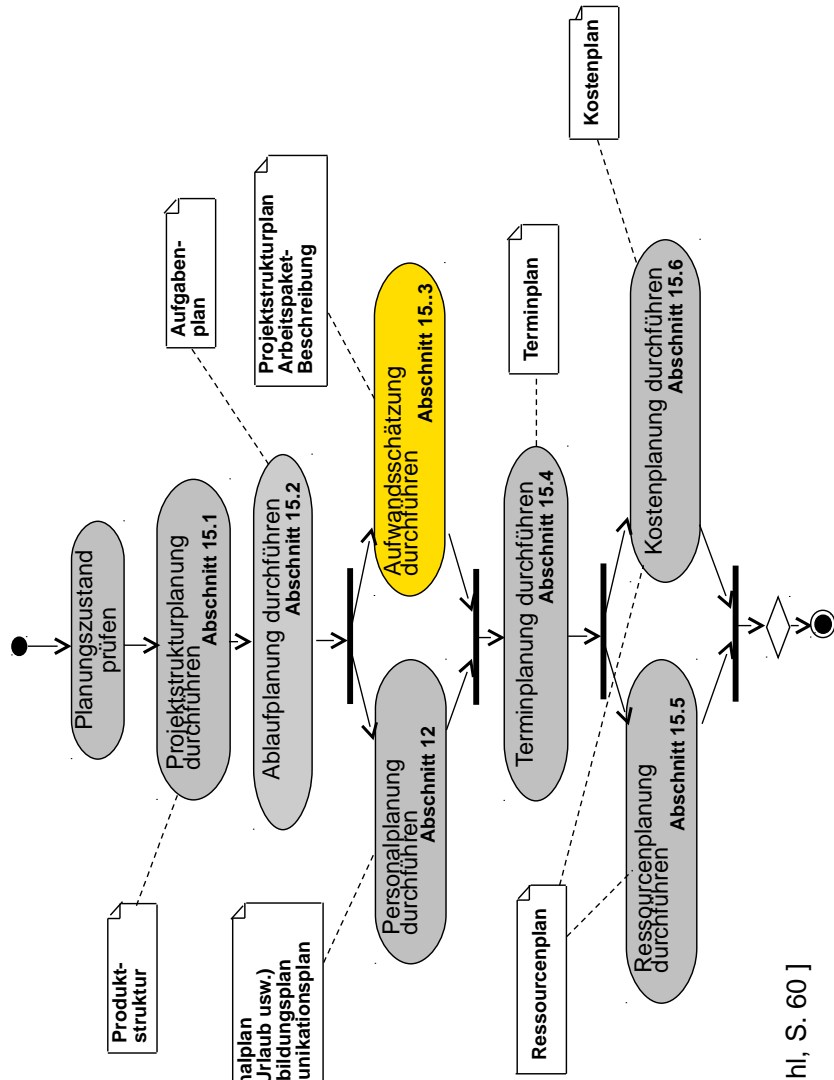
# Work expands

# to fill the

# available

# volume...

Aufwandsschätzung



Quelle: [ 4 Buhl, S. 60 ]

# Planungsablauf

35

## Strukturplanung

Was wird geliefert

Produktstruktur

Objektstruktur

Welche zusätzlichen Ergebnisse

Projektstruktur

Welche Tätigkeiten (AP)

In welcher Reihenfolge

Ablaufplan

Mit welchem Aufwand

Aufwandsplan

Netzplan

Wann

## Terminplanung

Womit

Ressourcen



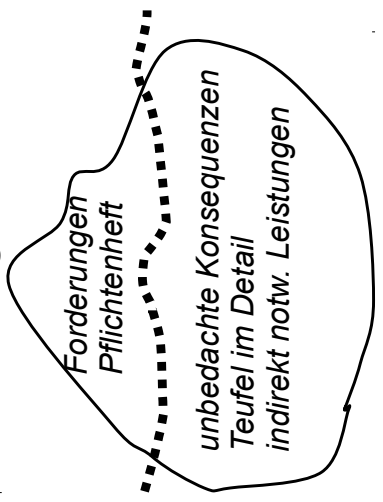
Quelle: Deutsche Informatik Akademie

# Aufwandsschätzung (Einführung 1)

36

- Die Aufwandsschätzung schätzt nicht den Preis, sondern die Kosten in der Einheit der Kostenkategorie
- Zeitpunkt: möglichst früh (und genau!) für
  - Angebotserstellung
  - Pflichtenheft
- Schätzen heißt nicht Raten
  - Erfahrungen sammeln und verwerten
  - Randbedingungen beachten
  - Aufgaben strukturieren und detaillieren
  - ständig aktualisieren
- Schätzung als solche ausweisen!

## Eisbergfaktor:



## Aufwandsschätzung (Einführung 2)

37

- ▶ Komponenten einer Schätzung:
  - Personal (Kosten)
  - Reisen
  - Computerzeit
  - Einrichtungen (Kosten für Computer, Netze, Testgeräte)
  - Dienstleistungen/ Aufwendungen (Unteraufträge, Beratung, Ausbildung, Druck, Büro, ...)
  - Gemeinkosten (nicht direkt nachweisbare Kosten, wie Heizungskostenanteil, Wasseranteil, etc.)
  - Nutzensschätzung ausweisen: Phasen/ Zyklen (enthaltene Kosten, auch für Dok., Datenerfassung, ...)
- ▶ Bei signifikanten Änderungen neu schätzen
- ▶ Nachkalkulation zur Auswertung von Erfahrungen

## Aufwandsschätzung (Einführung 3)

38

- ▶ Def. Einsatzmittel (Ressource): [DIN 69901-5] „abgrenzbare Gattung bzw. Einheit von Personal, Finanzmitteln, Sachmitteln, Informationen, Naturegebenheiten, Hilfs- und Unterstützungsmöglichkeiten, die zur Durchführung oder Förderung von Vorgängen, Arbeitspaketen oder Projekten herangezogen werden können“
- ▶ 3 Ansätze:
  - Personenzeit (Tage, Monate)
  - Anzahl Einsatzmittel
  - Projekt-Zeitdauer, z. B. in Tagen

# Methoden der Aufwandschätzung

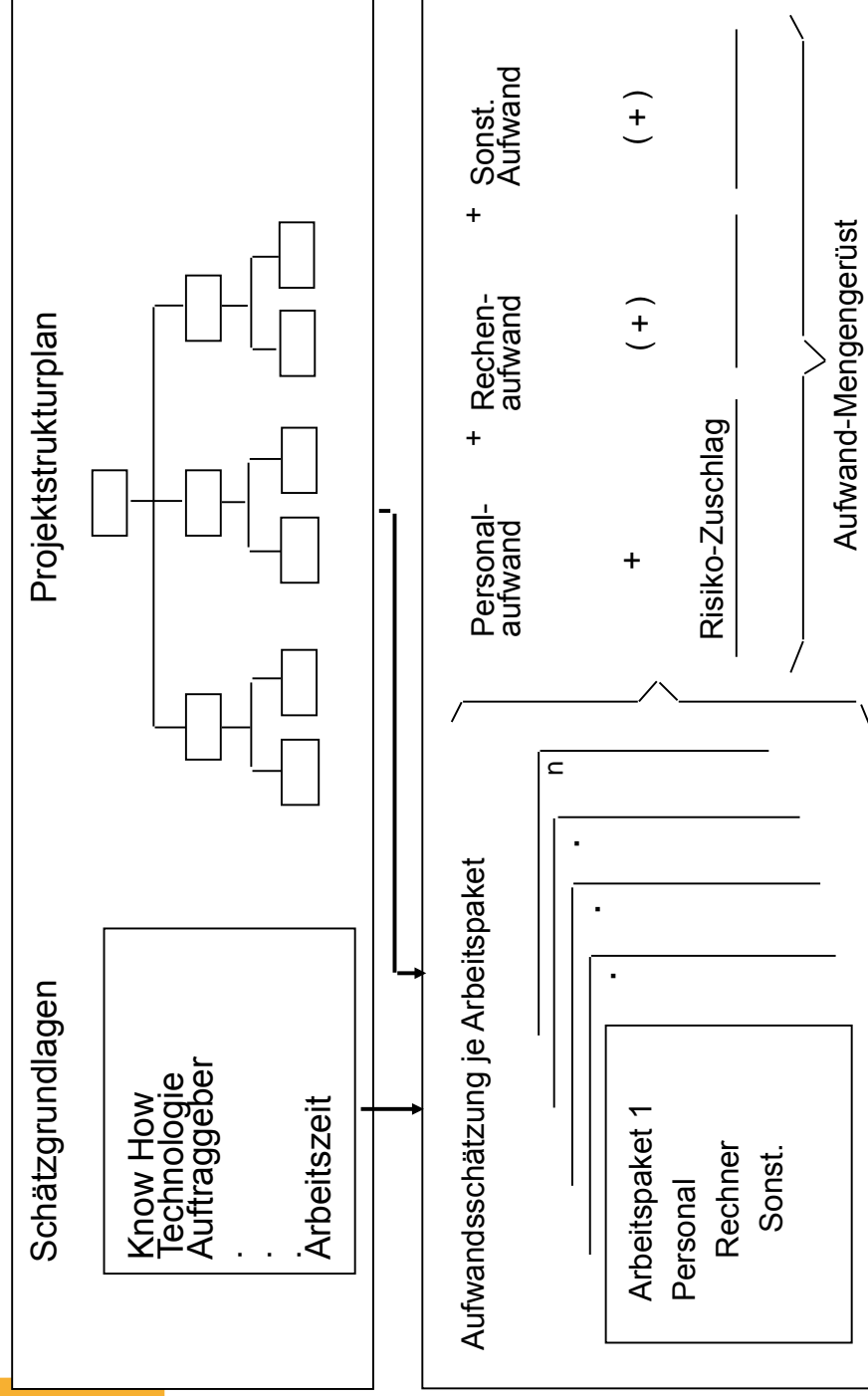
39

- ▶ **Einzelschätzung:** Sie wird von anerkannten Spezialisten/Experten durchgeführt. Sie bestimmen auf Basis dreier Schätzwerte für den optimistischen Aufwand  $A_o$ , den wahrscheinl. Aufw.  $A_w$  und den pessimistischen Aufw.  $A_p$  den Gesamtaufwand A.
- ▶ **Mehrfachbefragung:** Interdisziplinär zusammengesetzte Gruppe von Experten schätzt den Aufwand nach einer bestimmten Vorgehensweise (z.B. Delphi-Methode)
- ▶ **Analogiemethoden** (Prozentsatzmethoden, Vergleichsmethoden)  
Schätzung im Vergleich zu abgeschlossenen ähnlichen Projekten (Voraussetzung ist aktives Sammeln von Projektdaten)
- ▶ **Multiplikatormethoden** (Kennzahlenmethoden)  
Basis sind Multiplikationen für zu erbringende Leistungseinheiten (z.B. Lines of Code in Personenmonaten bei bekannten Programmieraufwand)
- ▶ **Algorithmische Methoden**  
bedienen sich Formeln oder eines Formelbildes, dessen Strukturen, Variablen und Konstanten mit mathematischen Modellen bestimmt werden

Quelle: [ 2 Fiedler ]

# Schätzelemente

40



# 15.3.1 Delphi-Verfahren

⇒ **systematische Befragung mehrerer kompetenter Personen** („Experten“) **über den Zeitbedarf der einzelnen Aktivitäten**

## a) Standard-Delphi-Verfahren

- Der Projektleiter schildert jedem Experten persönlich das Projektvorhaben und übergibt ihm ein Formular mit den Aufgabenpaketen.
- Jeder Experte füllt das Formular aus, ohne Kontakt zu anderen (außer zum PL).
- Der PL wertet die Formulare aus und verteilt ein neues Formular mit stark voneinander abweichenden Arbeitspaketen usw. usw.
- Das Schätzergebnis ergibt sich aus dem Durchschnittswert der letzten Überarbeitung.

⇒ **Die Schätzung erfolgt anonym und ohne Anwesenheit**

## b) Breitband-Delphi-Verfahren

- erster Schritt wie oben
- Der PL beruft eine Sitzung (Schätzklausur) ein, die Schätzung wird erläutert.
- Jeder Schätzer füllt das Formular selbständig aus (wie oben).
- Der PL sammelt die Formulare ein und wertet sie aus
- Pakete mit starken Abweichungen werden auf einem neuen Formular erfasst.
- Der PL beruft eine **neue Sitzung** ein usw. usw.

nach [1 Jenny]



# Delphi als Schätzklausur



### Besetzung

#### Moderator

3-4 **Schätzer** = Experten aus Projektteam oder extern

1-2 **Berater** aus Projektteam

#### Protokollführer

max 8 Personen  
Max 2 Tage

### Arbeitstechnik

- verdeckte Einzelschätzung
- gemeinsame Schätzwerbung (Entscheidungsregel!)
- offenes Protokoll
- Dokumentation der Schätzergebnisse



# Ablauf der Delphi-Schätzklausur

43

<b>Vorbereiten</b>	<ul style="list-style-type: none"><li>• Schätzer auswählen und einladen</li><li>• Unterlagen zur Verfügung stellen</li></ul>
<b>Durchführen</b>	<ul style="list-style-type: none"><li>• Ablauf Schätzklausur erläutern</li><li>• Basisinformation geben</li><li>• Projektstruktur durchsprechen</li><li>• Schätzeinheiten bilden</li><li>• Annahmen festhalten</li></ul> <div style="border: 1px solid black; padding: 5px;"><b>Einzelschätzung</b><ul style="list-style-type: none"><li>• Arbeitspaket erläutern</li><li>• Mengen und Aufwand schätzen</li><li>• Schätzwerte diskutieren</li><li>• Ergebnisse festhalten</li><li>• Risiko abschätzen</li></ul></div> <ul style="list-style-type: none"><li>• Schätzgenauigkeit ermitteln</li></ul>
<b>Nachbereiten</b>	<ul style="list-style-type: none"><li>• Ergebnisse dokumentieren</li><li>• Zuschlagsrechnung</li><li>• Netto / Brutto-Umrechnung</li><li>• Plausibilitätsprüfung</li><li>• Ergebnisse und Dank an Schätzer</li></ul>

Quelle: Deutsche Informatik Akademie



# 15.3.2 Standards der Function-Point-Methode

44

Die Function-Point-Methode wird international sehr erfolgreich eingesetzt und hat sich weit verbreitet. Folgende Standards der Methode sind bekannt:

**ISO 14143-1** ist seit 1999 Standard und beschreibt die grundlegenden Prinzipien einer funktionalen Größenmetrik FSM(Functional Size Metric) und enthält die dazugehörigen Definitionen

Zur Zeit sind nur abgeleitete Varianten der Function-Point-Methode nach **ISO/IEC14143-1** anerkannte Public Available Standards (PAS), wie folgende:

**ISO/IEC 20926** standardisiert für eine spezifische International Function Point User Group die Methode, die unter der Bezeichnung *IFPUG* Function Point Methode *Version 4.1* bekannt geworden ist. (URL: [www.ifpug.org](http://www.ifpug.org))

**ISO/IEC 19761** nach diesem Standard nutzt die *COSMIC-FFP* (Common Software Measurement International Consortium - Full Function Points) die Methode (URL: [www.cosmicon.com](http://www.cosmicon.com))

**ISO/IEC 24570** nach diesem Standard der Niederländische Metrik Organisation (*NESMA*) wird die Function-Point-Methode ebenfalls unterstützt. (URL: [www.nesma.org](http://www.nesma.org))

**ISO/IEC 20968** standardisiert die *Mark II* Function Point Methode (von Charles Symons in England für Anwendungen mit PSP der 4.Generation entwickelt)

(URL: [www.uksma.co.uk](http://www.uksma.co.uk))

Quelle: Tagungsband ISWM/MetriKon 2004; Shaker Verlag 2004



# Function-Point-Verfahren (1) (IBM)

- Eingabe (input): Funktionspunkte
- Resultat (output): Personenmonate

## Vier Schritte:

1. Ermitteln der Komponenten
2. Bewerten der Komponenten
3. Klassifizieren der Einflussgrößen (Einflussfaktoren)
4. Ermittlung der „Total Function Points“ (TFP), dann ⇒ Tabelle

## zu 1.: Ermitteln der Komponenten

### - „Geschäftsvorfälle“ (aus dem *Pflichtenheft und Entwurf*)

- **Eingabedaten** (Formulare, BS-Masken, Daten von anderen S.)
- **Ausgabedaten** (BS-Masken, Listen, Daten für andere Systeme)
- **Abfragen** (je Einheit von Online-Eingaben)

### - *Anwenderdateien* (Datenbestände)

- jede log. Datei, die gepflegt wird ( keine Zwischendateien)

### - *Referenzdateien*

- Dateien und Tabellen, die nur gelesen und nicht gepflegt werden

# Function-Point-Verfahren (2)

## zu 2.: Bewerten der Komponenten:

⇒ Vergabe von 3 bis 15 Funktionspunkten (FP) je nach Komplexität

Funktionsart	einfach	mittel	komplex
Eingabedaten	3	4	6
Ausgabedaten	4	5	7
Datenbestände	7	10	15
Referenzdaten	5	7	10
Abfragen	3	4	6

Die Anzahl der Funktionen wird mit den zugewiesenen Werten multipliziert und summiert.  
Das ergibt die erste

**Summe (S1)**

## zu 3.: Klassifizieren der Einflussgrößen

⇒ **Bewertung der Einflussfaktoren:**

(Einflussfaktoren s. nächste Folie)

- 0 = kein Einfluss
- 1 = gelegentlicher Einfluss
- 2 = mäßiger Einfluss
- 3 = mittlerer Einfluss
- 4 = bedeutender Einfluss
- 5 = starker Einfluss



# Function-Point-Verfahren (3)

## noch zu 3.: Einflussfaktoren

- Verflechtung mit anderen Systemen
- dezentrale Verarbeitung und Datenhaltung
- Transaktionsrate und Antwortzeitverhalten
- Verarbeitungskomplexität (hier Bewertungsspanne 0 - 30)
  - Rechenoperationen (0-10)
  - Umfang der Kontrollverfahren für die Datensicherstellung (0 - 5)
  - Anzahl der Ausnahmeregelungen (0 - 10)
  - Schwierigkeit und Komplexität der Logik (0 - 5)
- Wiederverwendbarkeit (Module, Routinen, ...)
- Datenbestand-Konvertierungen
- Benutzungs- und Änderungsfreundlichkeit



Maximal können 60 Punkte vergeben werden (Summe der Einflussfaktoren **S2**).  
 ⇒ Der Einflussfaktor **S3** kann maximal 30% des errechneten Wertes S2 betragen  
 $S3 = 0,70 + (0,01 * S2)$

⇒ Daraus werden im **4. Schritt** die „**Total Function Points**“ errechnet

$$TFP = S1 * S3$$

Quelle: nach [1]

# Function-Point-Verfahren (4)

## Fünfter Schritt:

- ⇒ anhand der ermittelten **Punkte** wird aus einer **Tabelle** der **Entwicklungsaufwand in Personenmonaten (PM)** abgelesen  
 (Die Wertetabelle muss entsprechend der Produktivität im Team/ Unternehmen auf Basis einer Nachkalkulation ständig aktualisiert werden).

## Beispiel:

Function Point	PM	Function Point	PM	Function Point	PM
150	5	500	33	850	61
200	9	550	37	900	65
250	13	600	41	950	70
300	17	650	45	1000	75
350	21	700	49	1050	84
400	25	750	53	1100	93
450	29	800	57	usw.	

Quelle: nach [1]





# 15.3.3 COCOMO-Verfahren (1)

(COConstructive COst MOdel nach Barry Boehm)

- **Eingabe (input):** Systemgröße in **DSI (Delivered Source Instructions)** bzw. **LOC (Lines of Code)**
- **Resultat (output):** **Personenmonate (PM)** und **Time for development (TDEV)**
- **3 Projektklassen:**
  - ♦ **Organic** (einfache Softwareprojekte)
    - kleine Teams, SW innerhalb des Hauses
    - Erfahrung mit ähnlichen Projekten
    - gute Sachkenntnis, klare Ziele, kein Termindruck
    - Produktgröße kleiner als **50 KDSI** (Kilo Delivered Source Instructions)
  - ♦ **Semi-detached** (mittelschwere SW-Projekte)
    - Team mit erfahrenen und weniger erfahrenen Mitarbeitern
    - Erfahrungen auf Teilgebieten des Projektes
    - Produktgröße kleiner als **300 KDSI**
  - ♦ **Embedded** (komplexe SW-Produkte)
    - größere Innovation, hohe Anforderungen an das Team
    - starker Kosten- und Termindruck
    - umfangreiches, komplexes SW-Produkt mit integrierten Elementen
    - Produktgröße: jede

Quelle: [ 1 Jenny, S. 366 ff ]



# COCOMO-Verfahren (2)

- 3 Modellvarianten für unterschiedliche Zeitpunkte:

- ♦ **Basis-Verfahren (BASIC-COCOMO ==> für frühe Schätzung)**
  - Detaillierung der Produkt- und Projektstruktur noch gering
  - Berechnung mit einer Grundgleichung (nur auf Basis von **LOC**)
  - der Schwierigkeitsgrad der Codierung ist über alles gleich hoch
- ♦ **Zwischenmodell (INTERMEDIATE-COCOMO)**
  - es werden Einflussparameter („Kostentreiber“) global mit einbezogen
  - es erfolgt noch keine Unterscheidung nach Entwicklungsphasen
- ♦ **Erweitertes Modell (DETAILED-COCOMO ==> Endmodell)**
  - zusätzlich zur Berücksichtigung der Einflussfaktoren noch Beachtung der anteiligen Aufwände für die einzelnen Phasen

## Projektprofile/ -größen:

<b>small</b>	Kleines Projektprofil	2000 loc
<b>intermediate</b>	Mittleres „	8000 loc
<b>medium</b>	Mittelgroßes „	32000 loc
<b>large</b>	Großes „	128000 loc
<b>very large</b>	Sehr großes „	512000 loc und mehr

Quelle: [ 1 ]



# COCOMO-Verfahren (3)

51

## BASIC-COCOMO

**Organic:**  
-Projekte  
 $PM = 2.4 * (KDSI)^{1.05}$   
 $TDEV = 2.5 * (PM)^{0.38}$

## INTERMEDIATE-COCOMO

$PM = 3.2 * (KDSI)^{1.05}$

**Semi-detached:**  
-Projekte  
 $PM = 3.0 * (KDSI)^{1.12}$   
 $TDEV = 2.5 * (PM)^{0.35}$

$PM = 3.0 * (KDSI)^{1.12}$

**Embedded:**  
-Projekte  
 $PM = 3.6 * (KDSI)^{1.20}$   
 $TDEV = 2.5 * (PM)^{0.32}$

$PM = 2.8 * (KDSI)^{1.20}$

PM = Personenmonate  
KDSI = Kilo Delivered Source Instructions  
TDEV = Time for Development (optimale Projektdauer ==> daraus Personenzahl)

### Werte, auf qualifizierte Informatiker umgerechnet:

Produktgröße	Notwendige Leistung	Produktivität	Entwicklungszeit	Anzahl einges. Personen
small	5.0 PM	400 DS/PM	4.6 Mon	1.1
intermediate	21.3 PM	376 DS/PM	8.0 Mon	2.7
medium	91.0 PM	352 DS/PM	14.0 Mon	6.5
large	392.0 PM	327 DS/PM	24.0 Mon	16.0

Quelle: [ 1 ]

# COCOMO-Verfahren (4): Einflussfaktoren

52

## Zu Einflussfaktoren/ Kostentreibern: (Beispiele)

- (Boehm unterscheidet 15 Faktoren in 4 Klassen, aufgeteilt auf einzelne Phasen.  
Zu Phasen: **PD** = Product design, **DD** = Detailed design, **CUT** = Code and unit test, **IT** = Integr. and Test)
- **Produktklasse: RELY** (Zuverlässigkeit), **DATA** (Größe der Datenbasis), **CPLX** (Komplexität)
  - **Computer-Klasse: TIME** (notw. Rechenzeit), **STOR** (Speichernutzg.), **VIRT** (Änderungshäufk.)  
**TURN** (Bearbeitungszyklus)
  - **Projekt-Klasse: MODP** (moderne Meth.), **TOOL** (Verwendung von), **SCED** (Anford. an E-Zeit)
  - **Personal-Klasse: ACAP** (Analysefähigkeit), **AEXP** (Sachkenntnis), **PCAP** (Programmierfähigkeit), **VEXP** (Erfahrung in der Systemumgeb.), **LEXP** (Erf. in der Programmiersprache)

CPLX	PD	DD	CUT	IT
extra high	1.65	1.65	1.65	1.65
very high	1.30	1.30	1.30	1.30
high	1.15	1.15	1.15	1.15
nominal	1.00	1.00	1.00	1.00
low	0.85	0.85	0.85	0.85
very low	0.70	0.70	0.70	0.70

Quelle: [ 1 ]

PCAP	PD	DD	CUT	IT
very high	1.00	0.65	0.65	0.65
high	1.00	0.83	0.83	0.83
nominal	1.00	1.00	1.00	1.00
low	1.00	1.20	1.20	1.20
very low	1.00	1.50	1.50	1.50

# COCOMO-Verfahren (5)

(Bsp. *Semidetached* und *Intermediate* (Kostentreiber global)

## Schritte:

1. Ermittlung der LOC (Summe der Schätzung je Modul/ Komponente)
2. Berechnung der Personenmonate  $PM_0$
3. Korrektur mit den Kostentreibern
4. Ermittlung der Entwicklungszeit TDEV
5. Ermittlung Anzahl der Mitarbeiter

**Beispiel:** Projekt mit 20.000 Befehlszeilen (20 KLOC )

$$PM_0 = 3.0 \cdot 20^{1.12} = \underline{86} \text{ notwendige Personenmonate lt. Basismodell}$$

unter Beachtung der Kostentreiber global:

- (Annahme: Kostentreiber haben den Wert „nominal“ (0) außer
- CPLX (Komplexität): 1.15
  - LEXP (Erfahrung in der Progr.-Sprache): 1.10

$$PM = 86 \cdot 1.15 \cdot 1.10 = \underline{109} \text{ (gerundet)} \quad TDEV = 2.5 \cdot 109^{0.35} = \underline{12.9} \text{ Monate}$$

$$N = PM / TDEV = \underline{109} / 12,9 = \underline{8.5} \text{ Mitarbeiter}$$

(bei BASIC-COCOMO ergibt sich  $PM = 56$  )

## Praxis Cocomo/FPM

- ▶ Erstelle ein Spreadsheet mit den Formeln der Schätzmethoden
- ▶ Wähle die Parameter
- ▶ Berechne Schätzung.

# Anwendung der Schätzmethoden

Schätzmethoden	Schätzmethoden während den Projektphasen		
	Grobe Schätzung (vor oder während Startphase)	Detaillierte Schätzung (Planungsphase)	Weitere Detaillierung (Durchführungsphase)
<b>1) Analogieschätzungen</b>			
Multiplikatormethode	X	(X)*	(X)*
Prozentsatzmethode	X	(X)*	(X)*
<b>2) Expertenschätzungen</b>			
Delphi-Methode (top-down)	X	(X)*	(X)*
Informelle Expertenschätzung (top-down und bottom-up)	X	X	X
Drei-Punkt-Schätzung (bottom-up)		X	X
<b>3) Fortgeschrittene Methoden</b>			
Cocomo		X	X
Function Point		X	X

Legende:

X: Methode kann in dieser Phase angewendet werden.

(X)\*: Kann für ausgewählte Module eingesetzt werden, ist aber für eine komplette Schätzung des Gesamtsystems zu aufwendig bzw. für eine komplette Schätzung liegen in der Regel nicht alle Erfahrungswerte vor.

Quelle: [ 6, S. 48 ]

# The End