

# 16. Projektplanung

Prof. Dr. rer. nat. Uwe Aßmann  
Lehrstuhl Softwaretechnologie  
Fakultät Informatik  
TU Dresden  
Version 13-1.3, 05.06.13

## 1. Projektstruktur

1. Einführung
2. Projektstrukturplanung

## 2. Ablaufplanung

## 3. Aufwandsschätzung

1. Delphi
2. Function Point
3. CoCoMo

## 4. Terminplanung

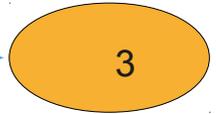
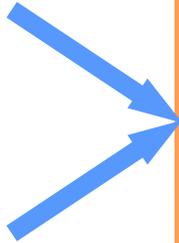
## 5. Ressourcenplanung

## 6. Kostenplanung

# Referenzierte Literatur

- ▶ 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

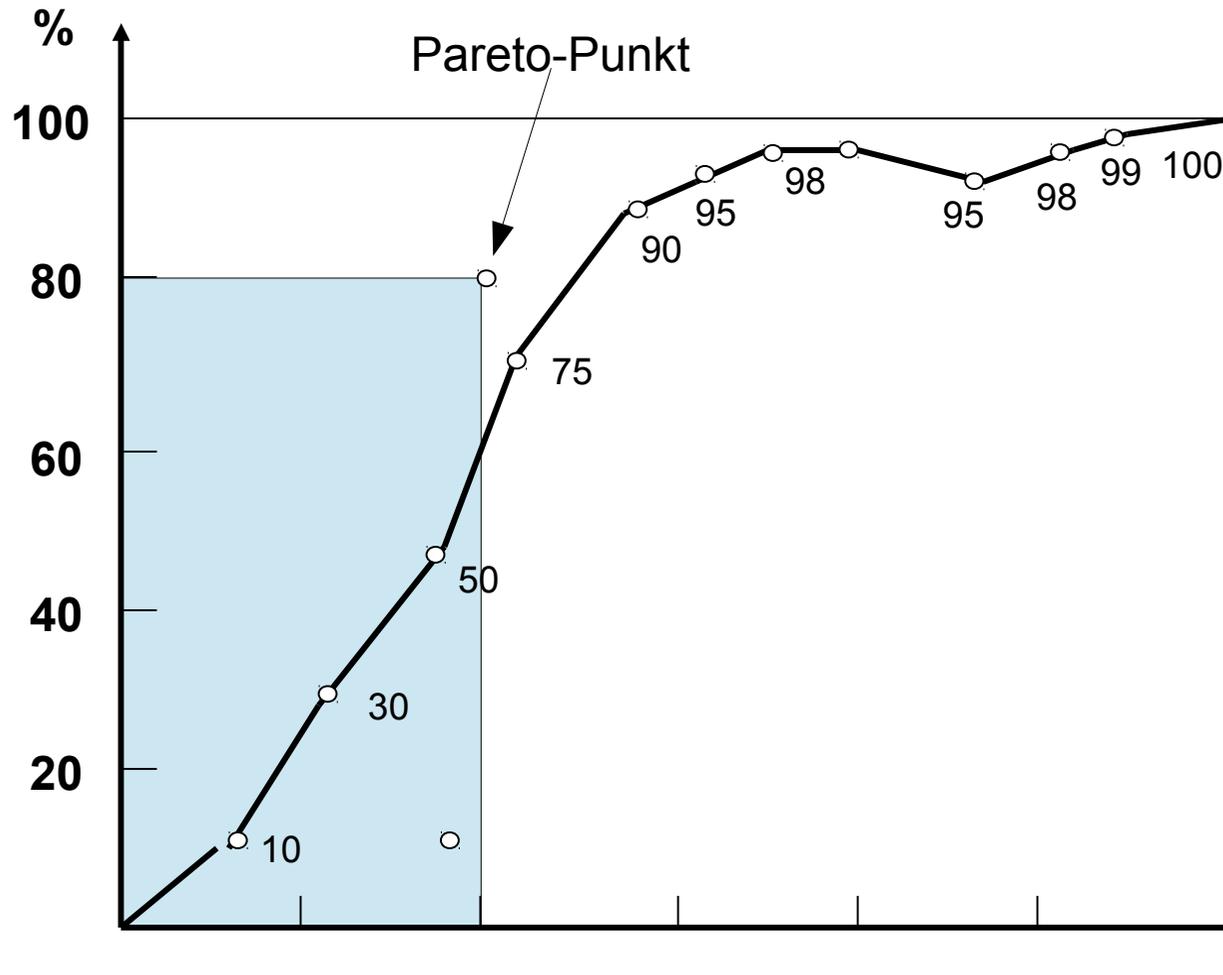
# 16.1 Einführung



# Das 90%-Syndrom nach Boehm

(subjektive Einschätzung der Fertigstellung)

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



## Probleme:

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

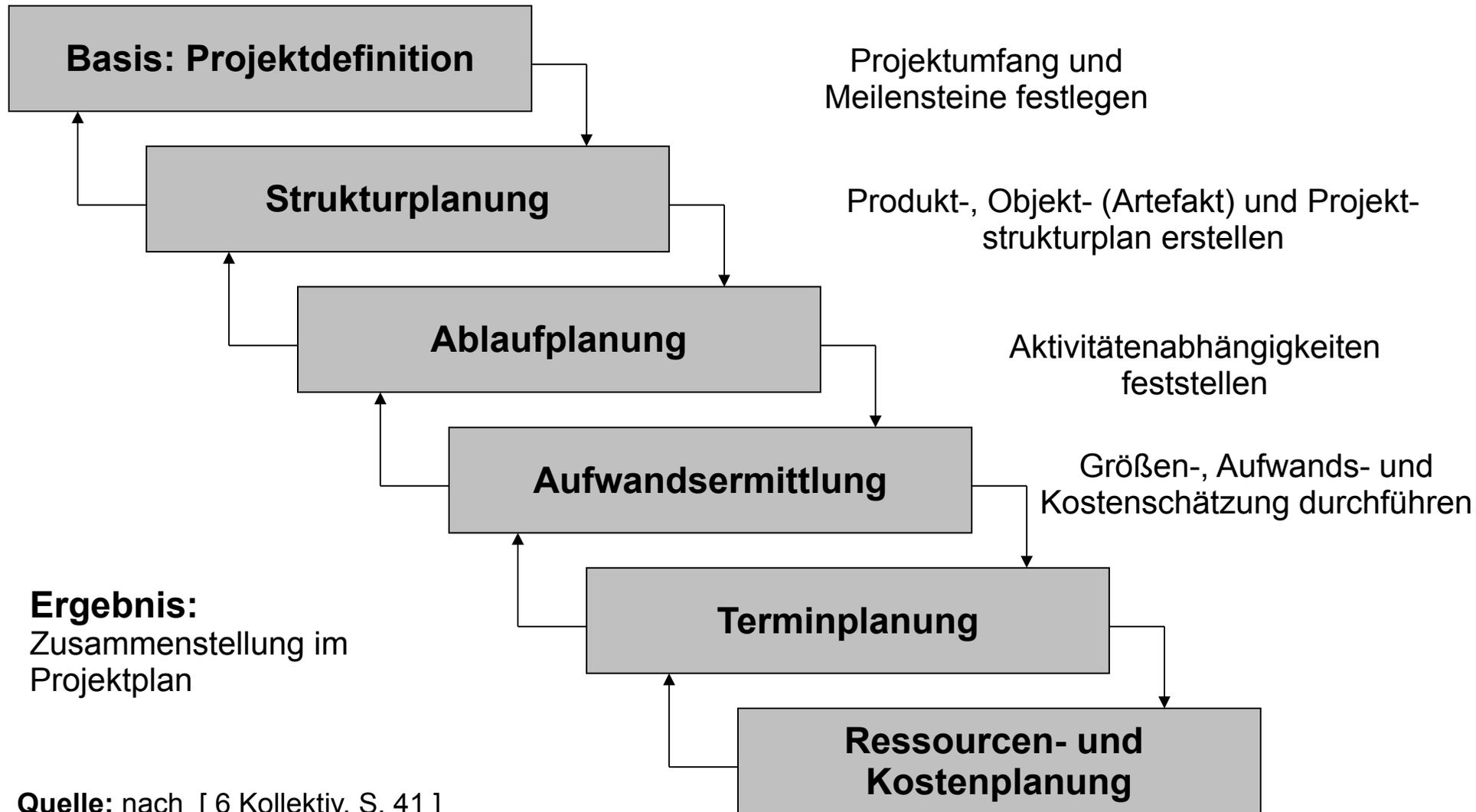
Quelle: Deutsche Informatik Akademie

# 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.

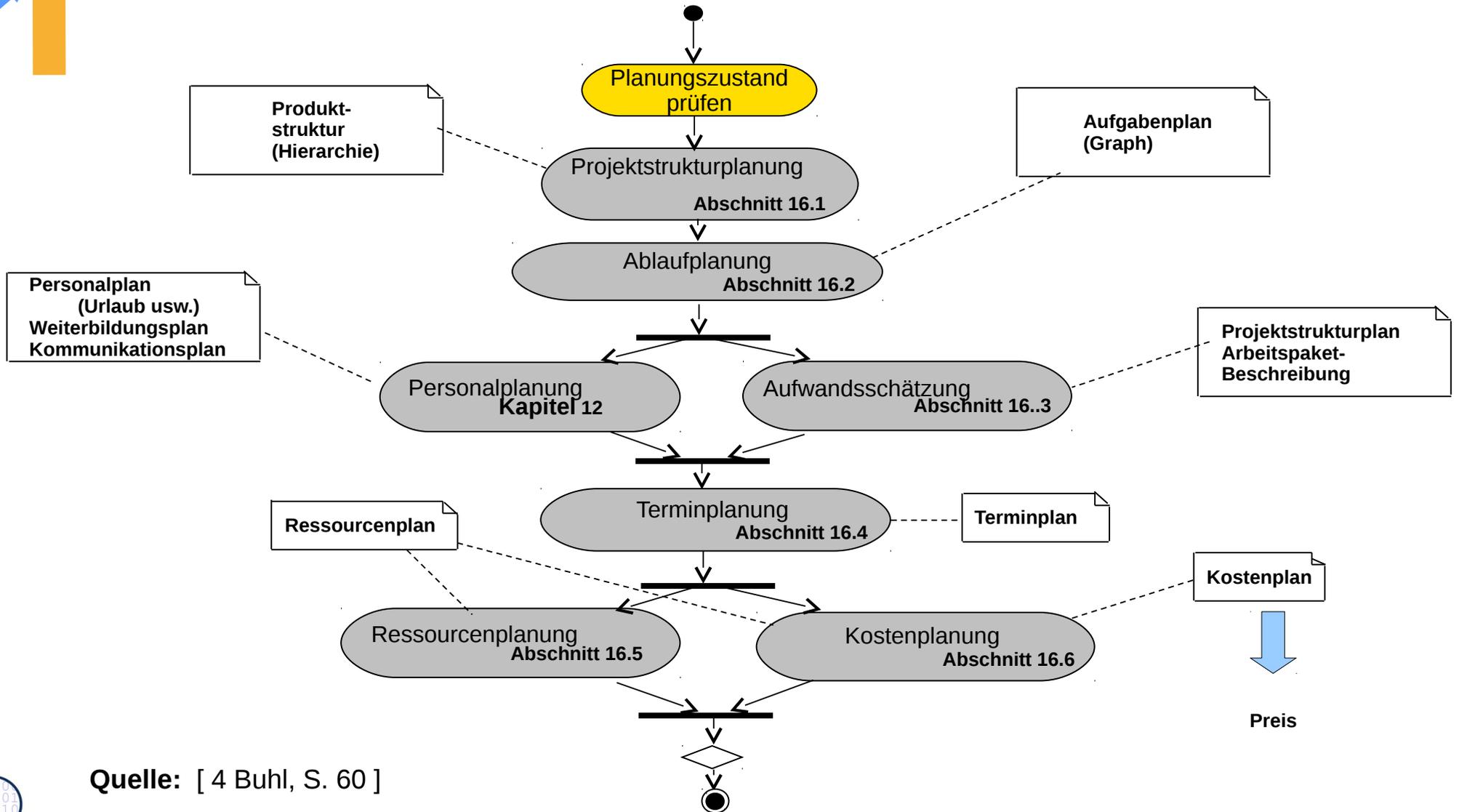
- ▶ Technische bzw. Operative Planung:
  - Auswahl eines Modells der Ablauforganisation, nach dem alle zu erstellenden Zwischen- und Endprodukte für das Projekt bestimmt werden. Ziel ist es, einen (idealen) Plan zur Minimierung der Prozessrisiken zu finden, auf dessen Basis mit dem AG(Kunden) verhandelt 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.

# Übersicht Planungsschritte



Quelle: nach [ 6 Kollektiv, S. 41 ]

# Aktivitätendiagramm der Planungsphase



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	Projek- tierung	Design	Implemen- tierung	Integrat.- test	Abnahme
--------	--------	--------------------	--------	----------------------	--------------------	---------

Projektplan 0

Phasenplan 1

Projektplan 1

Phasenplan 2

Projektplan 2

**Legende:**

▲ Verabschiedung

usw.

Genauigkeit wächst

# Reichweite der Aussagen der Planung

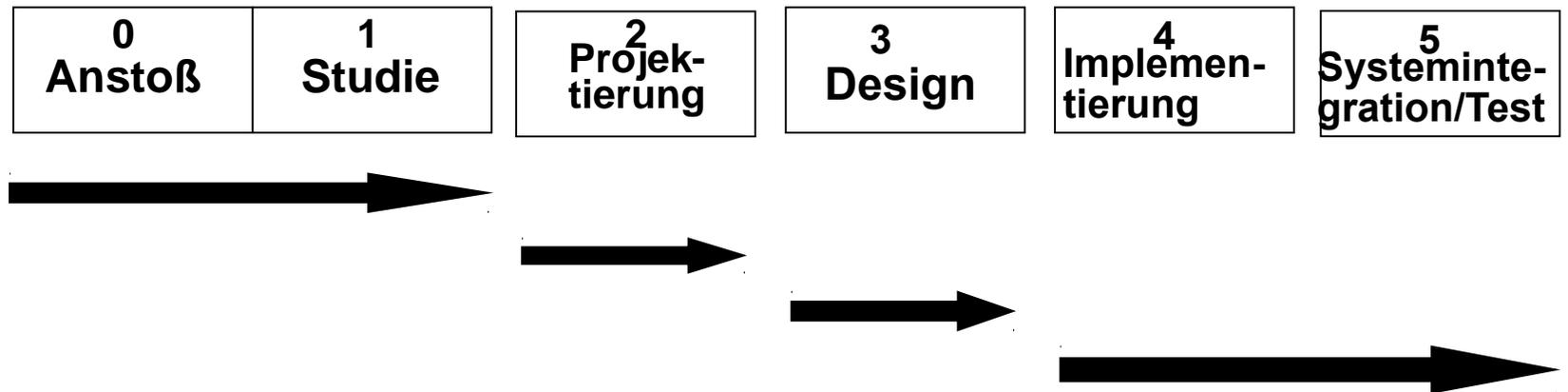
10

Projekte mit  
*geringer  
Innovation und  
geringem Risiko*



- ◆ Ab Pflichtenheft ist das Projekt ausreichend genau zu überblicken

Projekte mit  
*hoher  
Innovation und  
hohem Risiko*



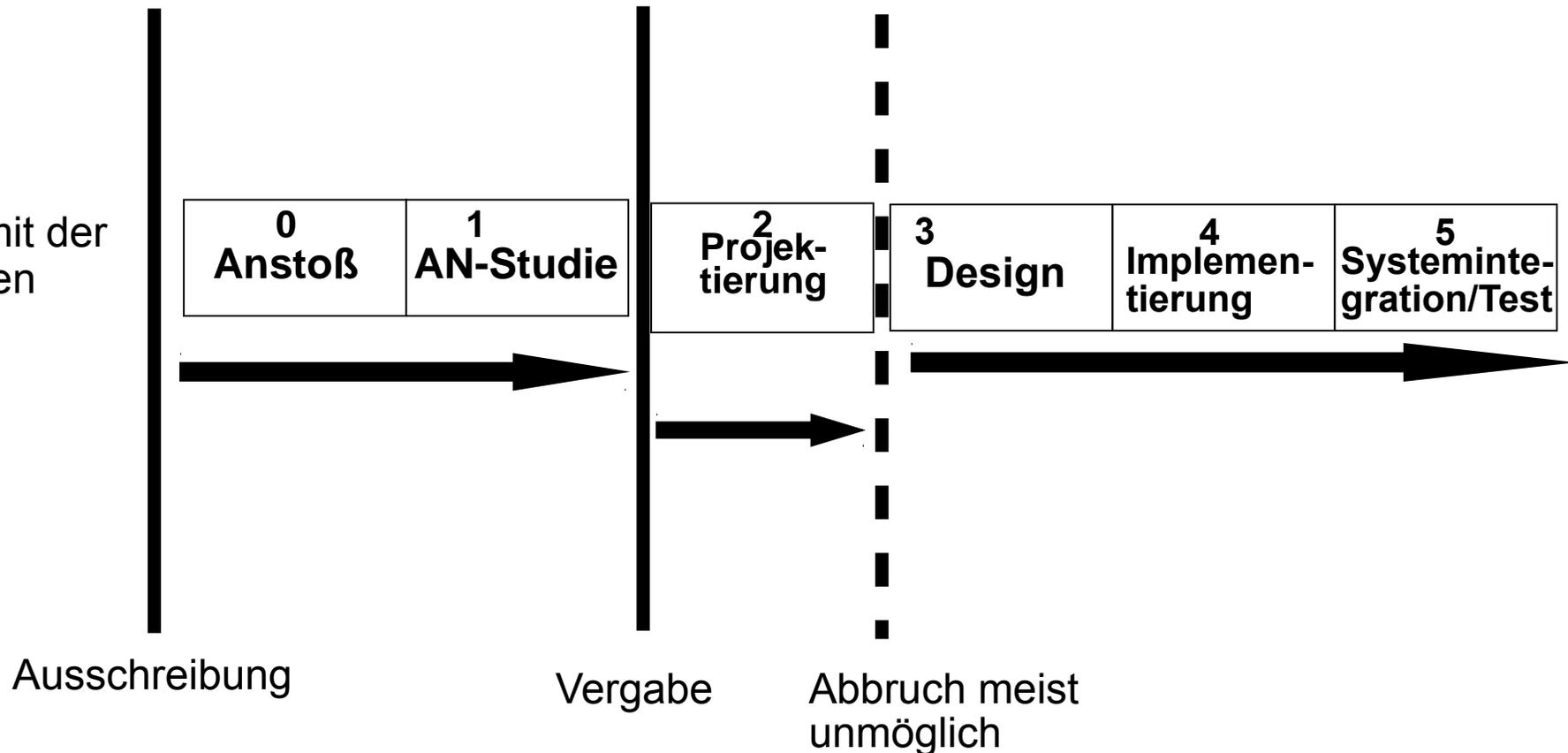
- ◆ Es kann nur jeweils ein Horizont von 10% bis 20% des Gesamtaufwandes ausreichend genau überblickt werden
- ▶ *evtl. Teilverträge, Teilprojektierung*

Quelle: Deutsche Informatik Akademie

# Projekte mit der öff. Hand

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

Projekte mit der  
Öffentlichen  
Hand



## 16.1.2 Strukturplanung

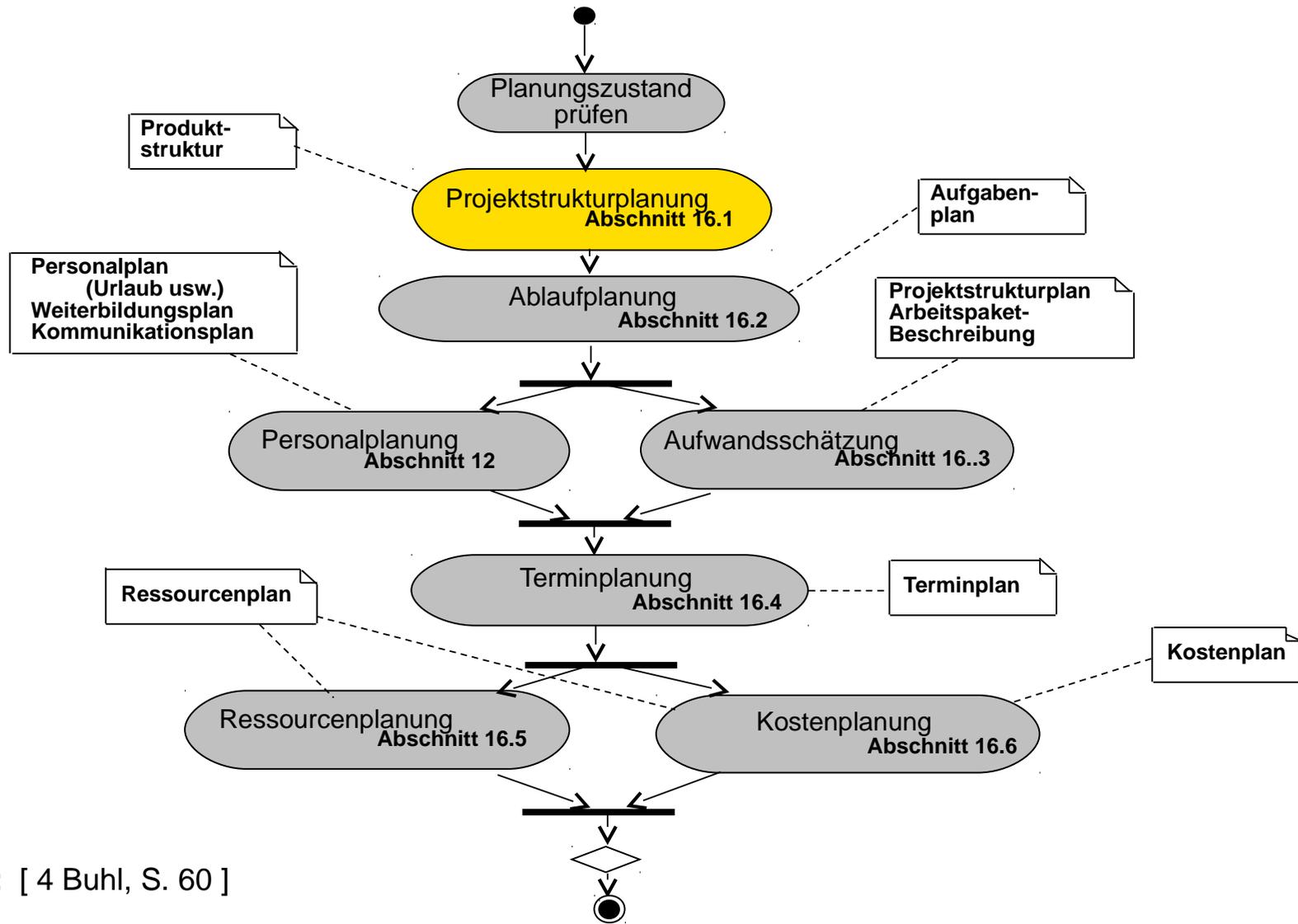


12

[http://en.wikipedia.org/wiki/Product\\_breakdown\\_structure](http://en.wikipedia.org/wiki/Product_breakdown_structure)



# Aktivitäten während der Planungsphase



Quelle: [ 4 Buhl, S. 60 ]



# Projektstrukturplanung

Die **Projektstrukturplanung** plant die *Komponenten-* und *Ablauf-Struktur* des Projektes.

(End-)Produktstruktur  
(product breakdown structure)

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

Objektstruktur  
(project breakdown structure PBS)

Welche **Artefakte** (Objekte, Arbeitsergebnisse, Arbeitsprodukte) werden fürs Produkt gebaut?

- Zwischenergebnisse (z. B. Prototypen)?
- Entwicklungsdokumente?
- Hilfsmittel, Tools, Vorrichtungen, Messgeräte?
- Steuerungsergebnisse (Pläne, Berichte)?

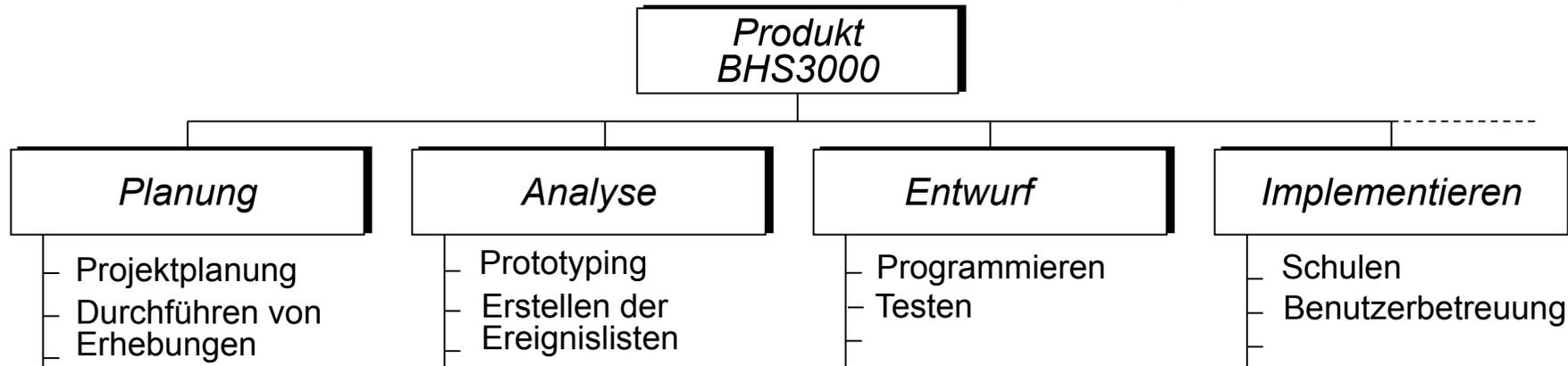
Projektstruktur  
(work breakdown structure WBS)

Einteilung der **Arbeitspakete (AP)**

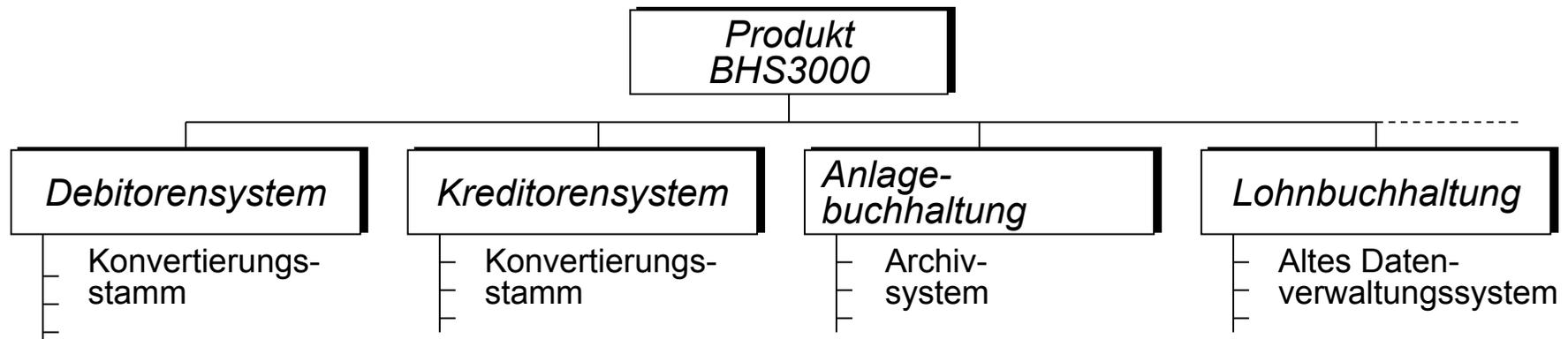
- ◆ Welche AP zur Erstellung der Objekte?
- ◆ Welche AP der „Projektfunktionen“?
- ◆ Welche AP sind voneinander abhängig?
- ◆ Welche AP nebenläufig durchführbar?

# Dekompositionskriterien von Produktstrukturplänen

- ▶ **Funktionsorientierter Produktstrukturplan** dekomponiert das Produkt anhand seiner Funktionen (Features, Funktionsbaum)



- ▶ **Komponentenorientierter Produktstrukturplan** dekomponiert anhand von Systemkomponenten (siehe EOS)



# Schritte der Projektstrukturplanung

16

Inhalt

Strukturplan

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

1 = **Produktstrukturplan**

2. Zur Erstellung des End-Produktes  
notw. **Artefakte (Arbeitsergebnisse,  
Zwischenergebnisse)**

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

3. Für die Abwicklung des  
Projekt es nötige **Aufgaben**

3 = **Arbeitspakete (Aktvitäten)**

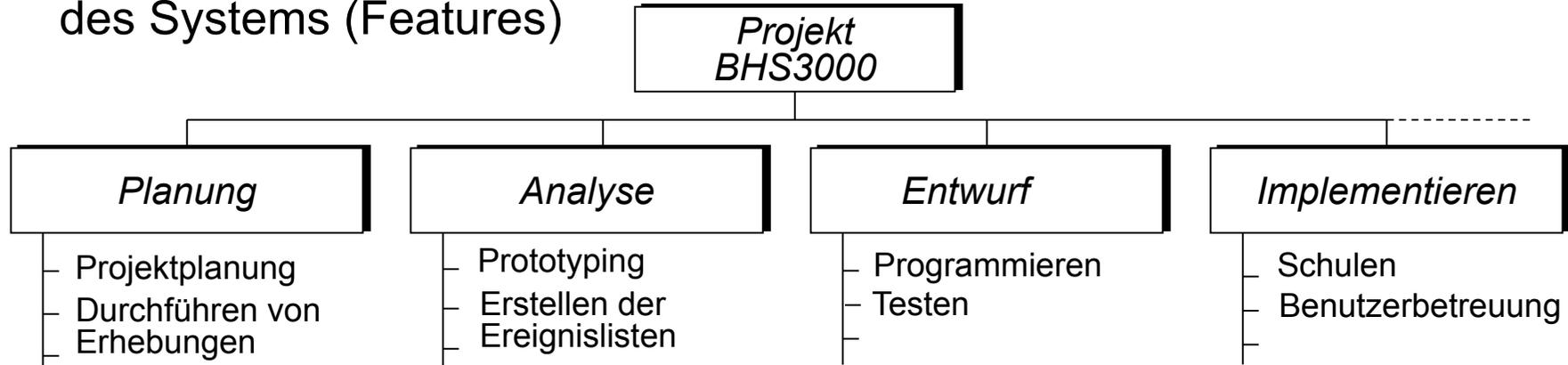
1 + 2 + 3 = **Projektstrukturplan**

# Projektstrukturplan (WBS)

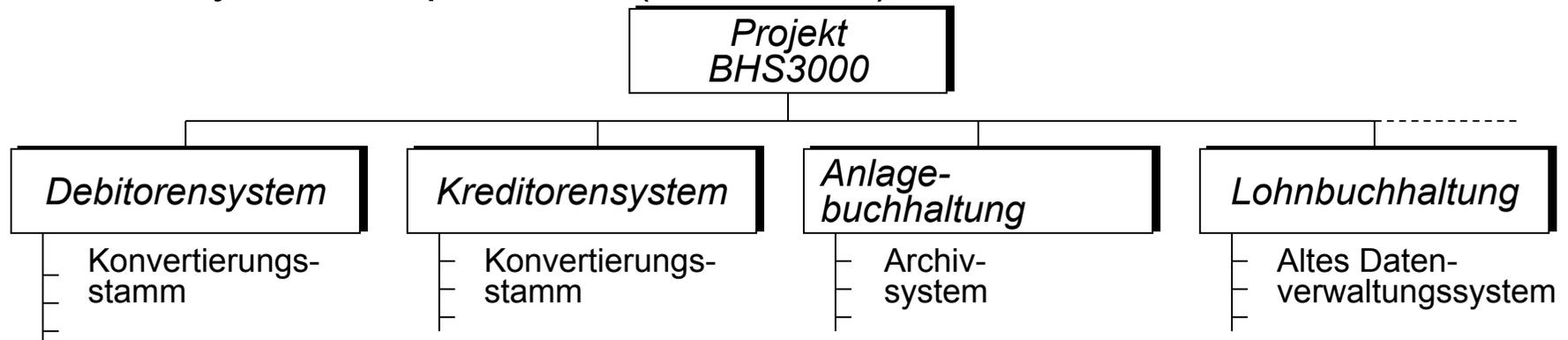
- ▶ Der **Projektstrukturplan (PSP, Aktivitätenstruktur, Work Breakdown Structure, WBS)** enthält alles, was ist zu tun, um die Projektziele zu erreichen
  - hierarchische Struktur (Baum) der zu bewältigenden Aufgaben eines Projekts
  - aufgabenorientierte Struktur
  - Zentrales Kommunikationsinstrument
  - Stabiles Planungsinstrument (Termin- und Kostenänderungen haben darauf keinen Einfluss)
- ▶ Darstellung als strukturierte Aktivitätenliste oder Strukturdiagramm (Baum) mit 3 Ebenen:
  - **1. Ebene:** Projektbezeichnung
  - **2. Ebene:** Strukturierung des Projektes nach verschiedenen Gliederungsgesichtspunkten (z.B. Funktionen, Phasen, Objekten ...)
  - **3. Ebene:** Arbeitspakete

# Dekompositionskriterien von Projektstrukturplänen

- **Funktionsorientierter Projektstrukturplan** dekomponiert anhand Funktionen des Systems (Features)

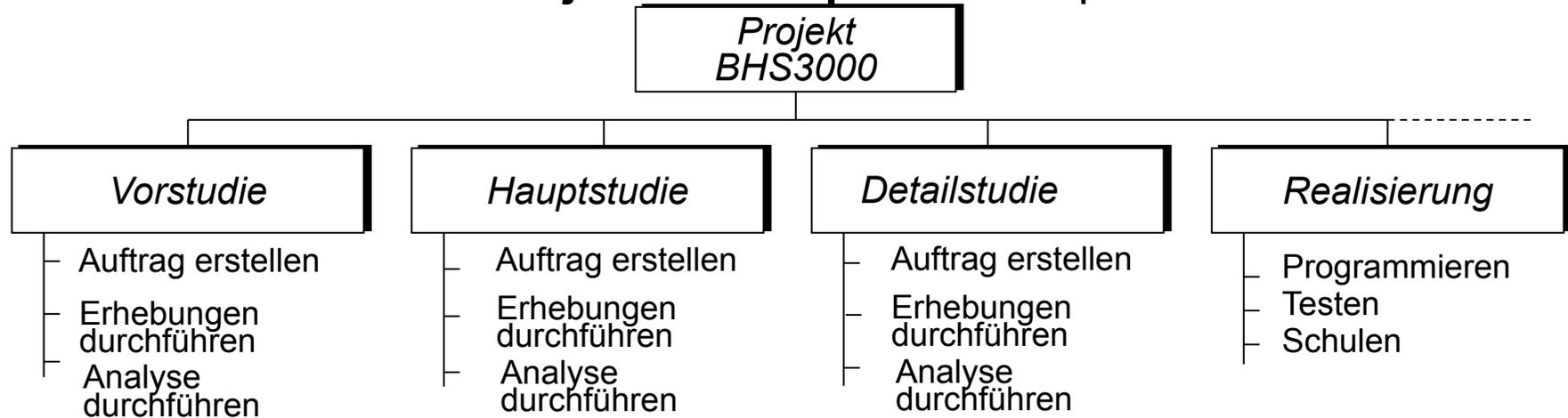


- **Komponentenorientierter Projektstrukturplan** dekomponiert anhand von Systemkomponenten (siehe EOS)

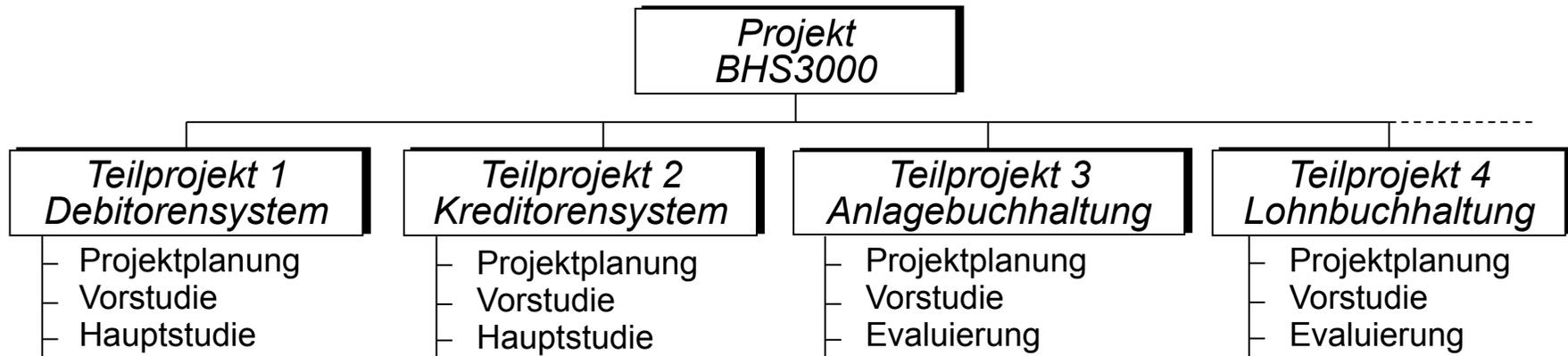


# Dekompositionskriterien von Projektstrukturplänen ctd.

- ▶ **Ablauforientierter Projektstrukturplan** dekomponiert anhand Aktivitäten



## Mischform einer Projektstrukturierung



# Arbeitspakete

(Inhalt des Aktivitätenstrukturplanes)

20

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]

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

# Eigenschaften von Arbeitspaketen

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

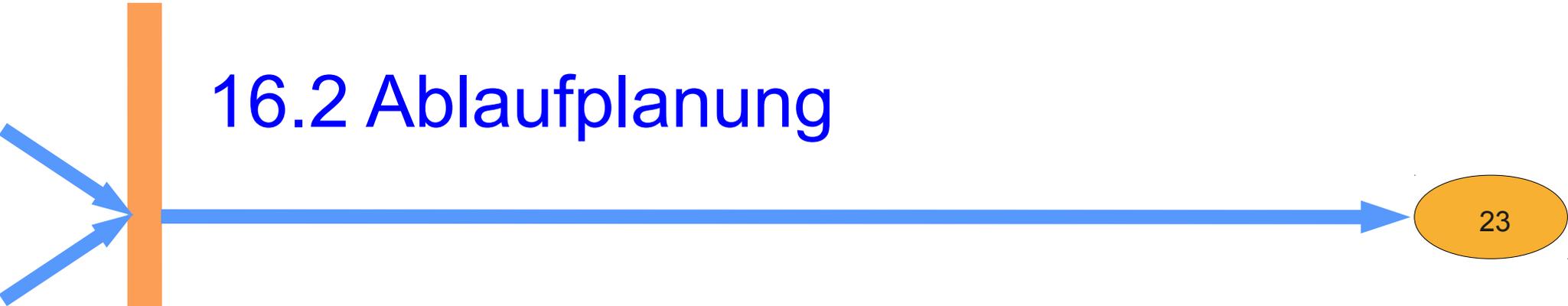
[anlehnend Vorl. Prof. S. Seibert]

# Beispiel Arbeitspaket-Spezifikation

<b>Projektnummer: 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, Erstansatz 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]

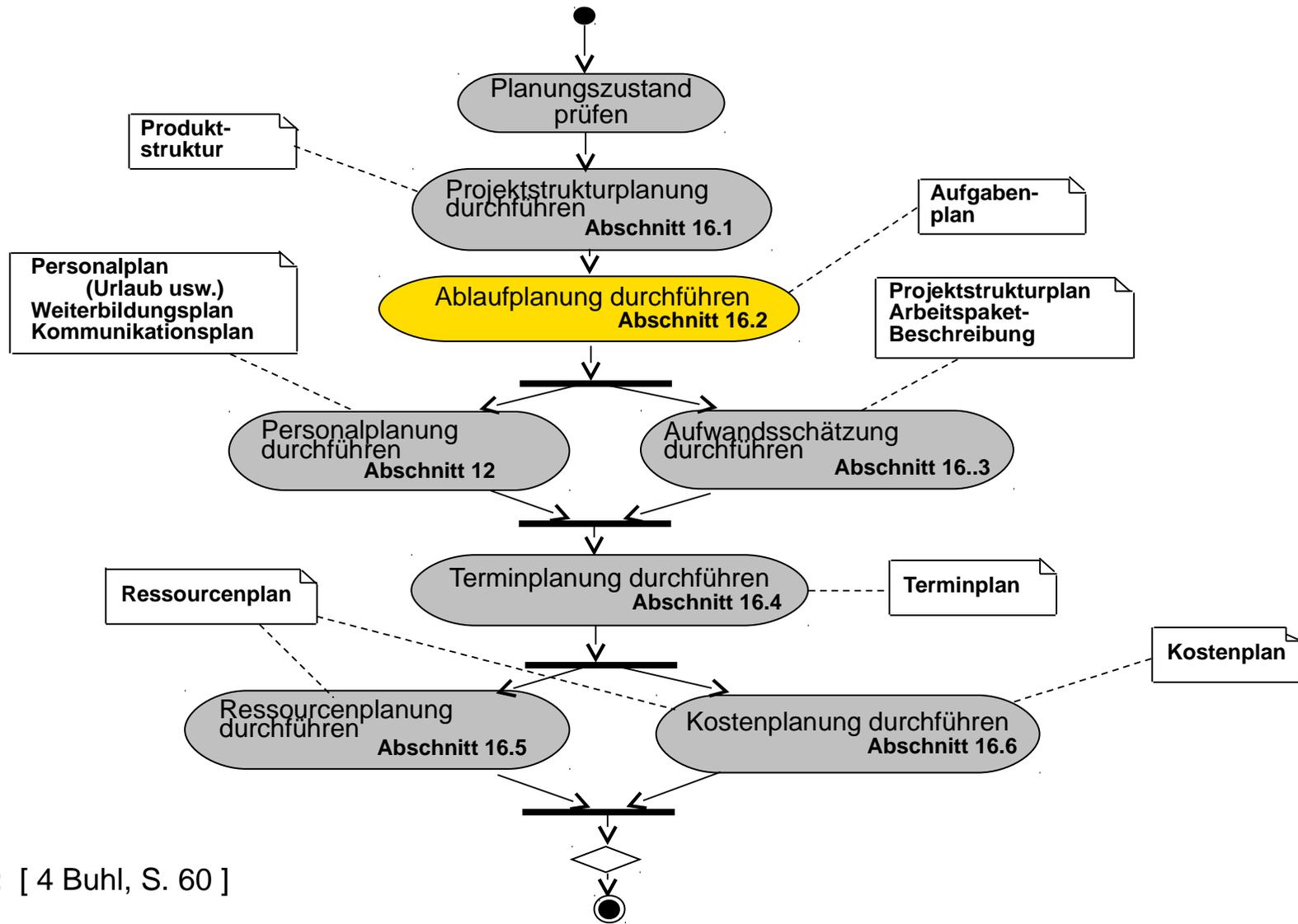


## 16.2 Ablaufplanung



23

# Aktivitäten während der Planungsphase

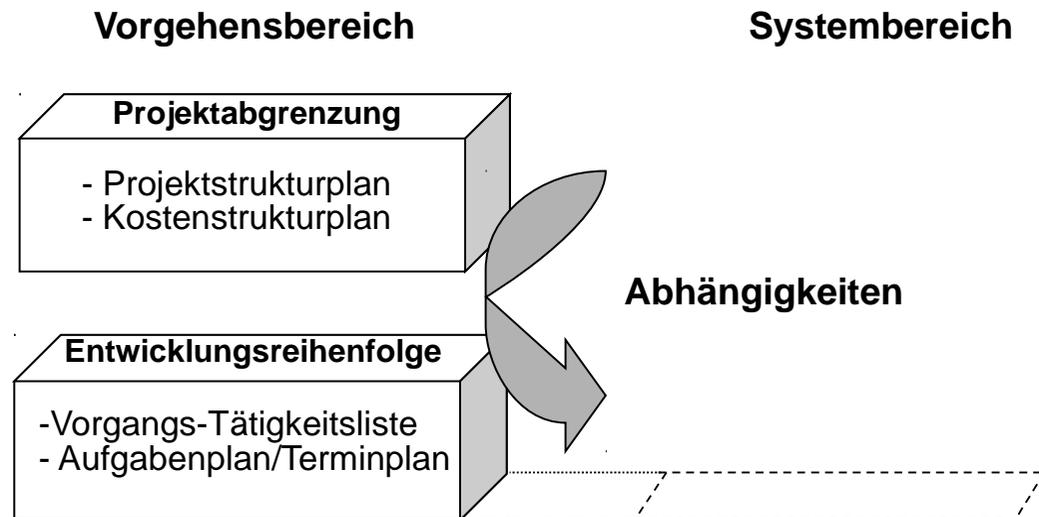


Quelle: [ 4 Buhl, S. 60 ]



# Ablaufplanung (Abhängigkeitsanalyse der Teilaufgaben)

- ▶ 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
  - Deren **Abhängigkeiten** werden ermittelt
  - Auf dieser Grundlage wird der **Aufgabenplan/Terminplan** erstellt



[ 1 Jenny ]

# 16.2.1 Vorgangsliste

- ▶ Eine strukturierten **Vorgangsliste (Aktivitätenliste)** beinhaltet die Aktivitäten (AP, Vorgänge) eines Projekts in Form einer Tabelle.
  - inkl. Verantwortlichen und zugeordneten Mitarbeitern
  - Dauer bzw. der Aufwand (Tage geplant, bisher getan, noch zu tun)
  - die benötigten Ressourcen sind zuzuordnen
  - Abhängigkeiten:
    - Welche Aktivitäten können unabhängig voneinander ausgeführt werden?
    - Entspricht der Feinheit der Aktivitäten den Anforderungen?
  - Priorität (ABC)
  - hierarchische Nummerung
- ▶ Als praktisches Tool zur Erstellung und Verwaltung der Vorgangsliste bietet sich an
  - Textverarbeitungsprogramm mit Gliederungsansicht
  - Spreadsheet

# Bsp.: Vorgangsliste (0)

<b>Vorgangsliste</b>	Projekt: Aussteller:	Nr.: Datum:	Seite:
----------------------	-------------------------	----------------	--------

Nr.	Projekt <span style="font-size: small;">tätigkeit</span> Arbeitspaket (Tätigkeit)	Vorgangszeitpunkte				Vorgang Dauer	Direkter Vorläufer	direkter Nachfolger	Pufferzeiten			Bedarf	
		FA	SA	FE	SE				GP	FP	UP	MA	SM
A	Arbeitspaket 01							B,C,D					
B	Arbeitspaket 02						A	E					
C	Arbeitspaket 03						A	E					
D	Arbeitspaket 04						A	E					
E	Arbeitspaket 05						B,C,D						
F	Arbeitspaket 06							G					
G	Arbeitspaket 07						F						
H	Arbeitspaket 08							L					
I	Arbeitspaket 09						H	K					
K	Arbeitspaket 10						I						

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 ]



# Einfache Vorgangsliste

- ▶ mit einfacher Aufzählung

Responsible		Workedout		Version		
Andy		Suny		0,2		
Due date	Milestone graph	Task with Milestone	Date	Report	Estimated Personweeks	Start
31.03.03		Design ready	20.03.03	Johnny		01.03.03
30.04.03		First prototype			4	01.04.03
10.05.03		Test first prototype			3	10.04.03
31.05.03		Second prototype			4	01.04.03
10.06.03		Test Second prototype			3	05.04.03
30.06.03		Acceptance test done			5	01.06.03

# Strukturierte Vorgangsliste

- ▶ Einteilung von Unteraktivitäten (Strukturierung)

Responsible		Workedout		Version		
Andy		Suny		0,2		
Due date	Milestone graph	Task with Milestone	Date	Report	Estimated	Start
					Personweeks	
31.03.03		Design ready	20.03.03	Johnny		01.03.03
		- first design				
		- design review				
		- final design				
		Prototyping				
30.04.03		- First prototype			4	01.04.03
10.05.03		--Test first prototype			3	10.04.03
31.05.03		- Second prototype			4	01.04.03
10.06.03		-- Test Second prototype			3	05.04.03
30.06.03		Acceptance test done			5	01.06.03

# 16.2.2 Darstellungen von Aktivitäten in Abhängigkeitsdiagrammen

Diagrammart	Darstellung der Bestandteile	Beispiel
<b>Vorgangsknotennetz (Aktivitätendiagramm, Datenflussdiagramm)</b> Die <b>Vorgänge</b> werden durch <b>Knoten</b> dargestellt.		<b>PDM</b> <b>MPM</b> <b>UML-AD</b>
<b>Vorgangspfeilnetz</b> Vorgänge: Pfeile Knoten: Zustände		<b>CPM</b> <b>Statecharts</b>
<b>Ereignisknotennetz</b> Knoten: Ereignisse		<b>PERT</b> <b>VMXT</b>
<b>Bipartite Netze</b> Stellen: Zustände Rechtecke: Vorgänge (synchronisierend)		<b>PetriNet</b> <b>Workflow</b> <b>BPMN</b>

**PDM:** Precedence Diagramm Method(auch MS Project

**MPM:** Metra Potential Method

**CPM:** Critical Path Method

**PERT:** Program Evaluation and Review Technique



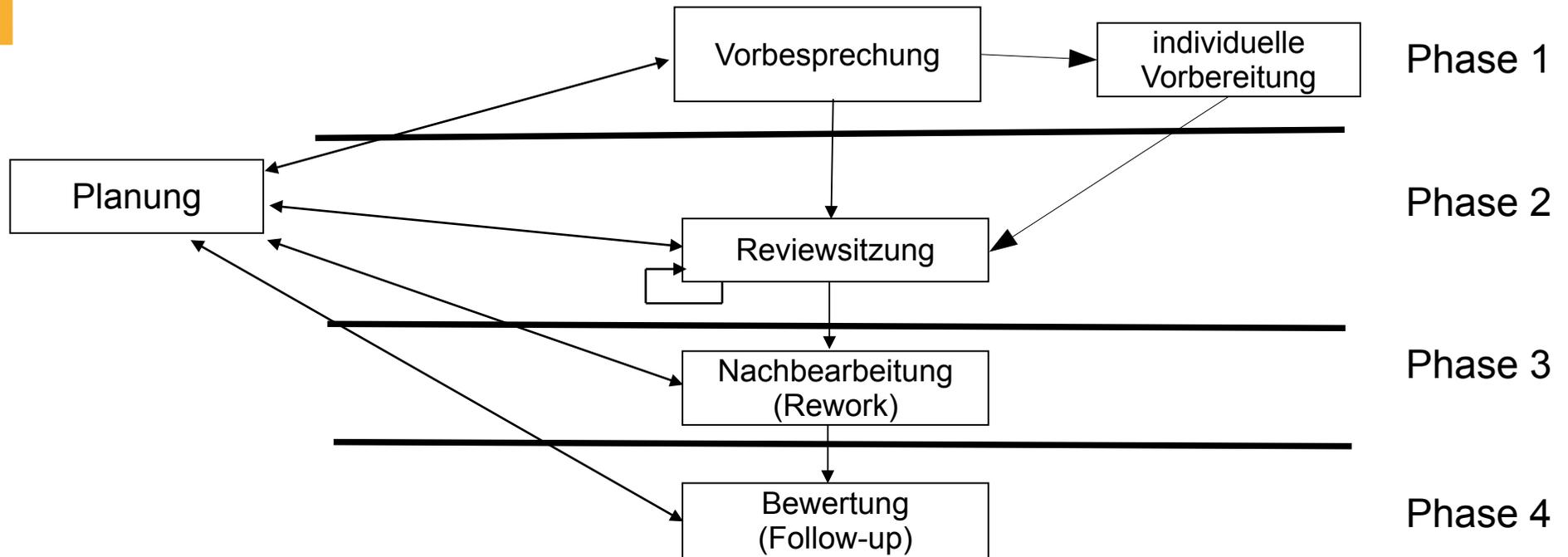
# Abhängigkeitsdiagramm (Abhängigkeitsgraph)

Das **Abhängigkeitsdiagramm (-graph)** stellt die Abhängigkeiten von Aktivitäten, Zustände und Ressourcen als **Knoten** eines Graphen mit Hilfe eines Vorgangsknotennetzes (Aktivitätendiagramm) dar

- ▶ 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:
  - partielles Ordnen von Gedanken und Handlungsoptionen
  - Einteilen von Phasen
  - Finden von weiteren Teilprodukten und Aktivitäten
  - Genauigkeit ist von den momentanen Erfordernissen abhängig
- ▶ Erstellung aus dem Projektstrukturplan
- ▶ Überprüfung erfolgt in der Regel durch **Reviews**
- ▶ Für eine Machbarkeitsstudie genügt ein Übersichtsplan

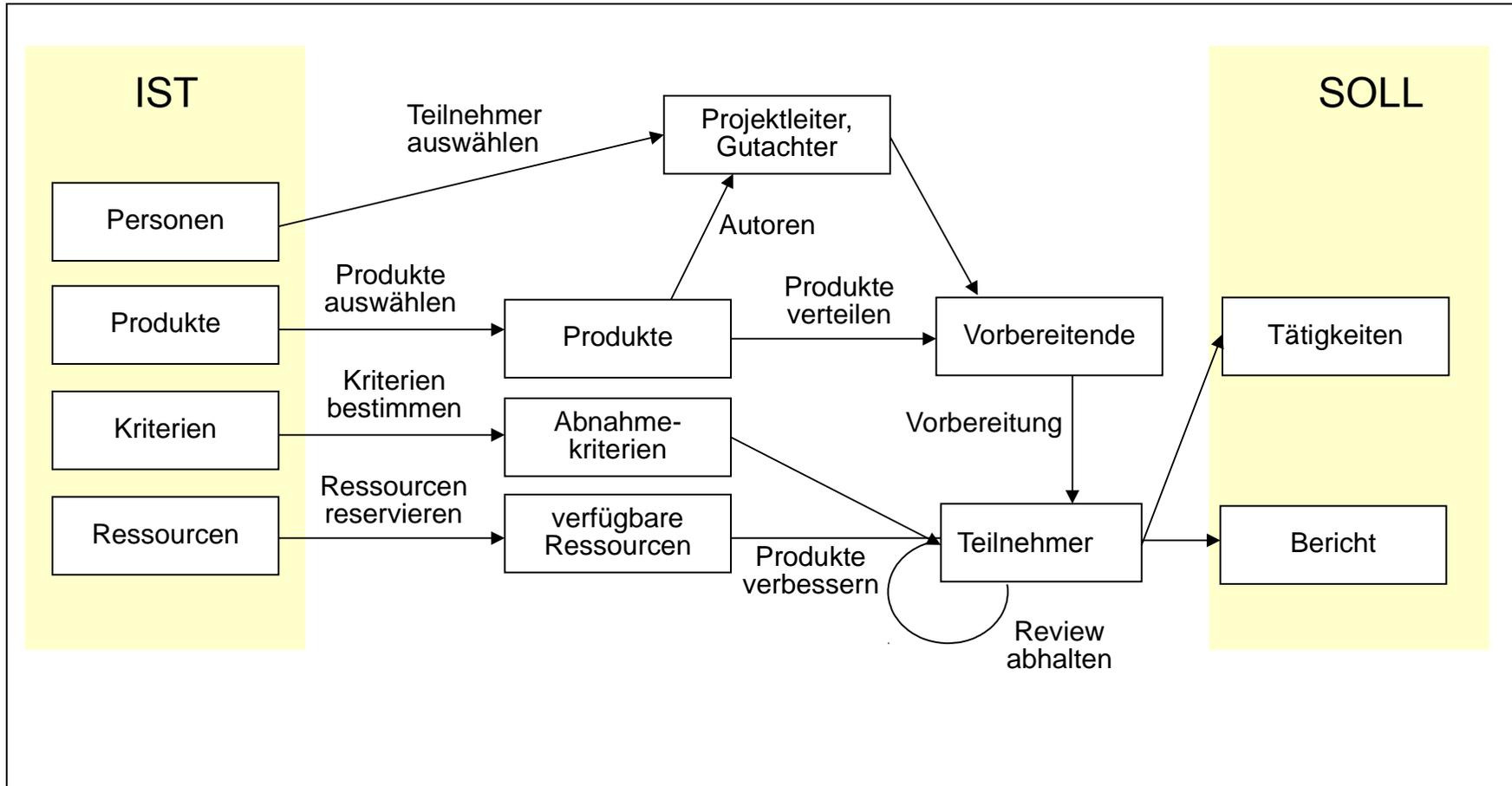
# Bsp.: Ablauf eines Reviews als Aktivitätendiagramm (Vorgangsknotennetz)

32



Quelle: [Wallmüller]

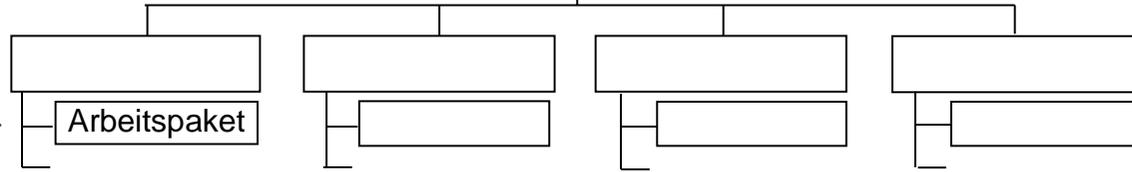
# Bsp.: Abhängigkeitsdiagramm(-graph) eines Review als Vorgangspfeil-Netz über Daten



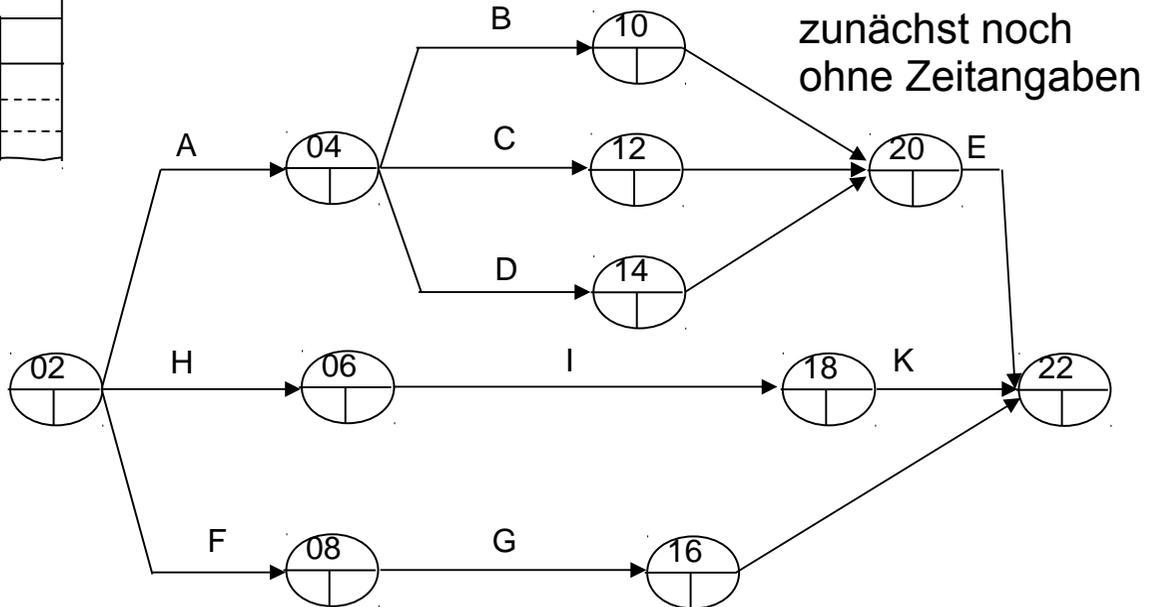
Quelle: [ Zuser, W. S. 122 ]

# Ergebnisse aus der Ablaufplanung

## Projektstruktur



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



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

## 16.3 Aufwandsschätzung (-ermittlung)

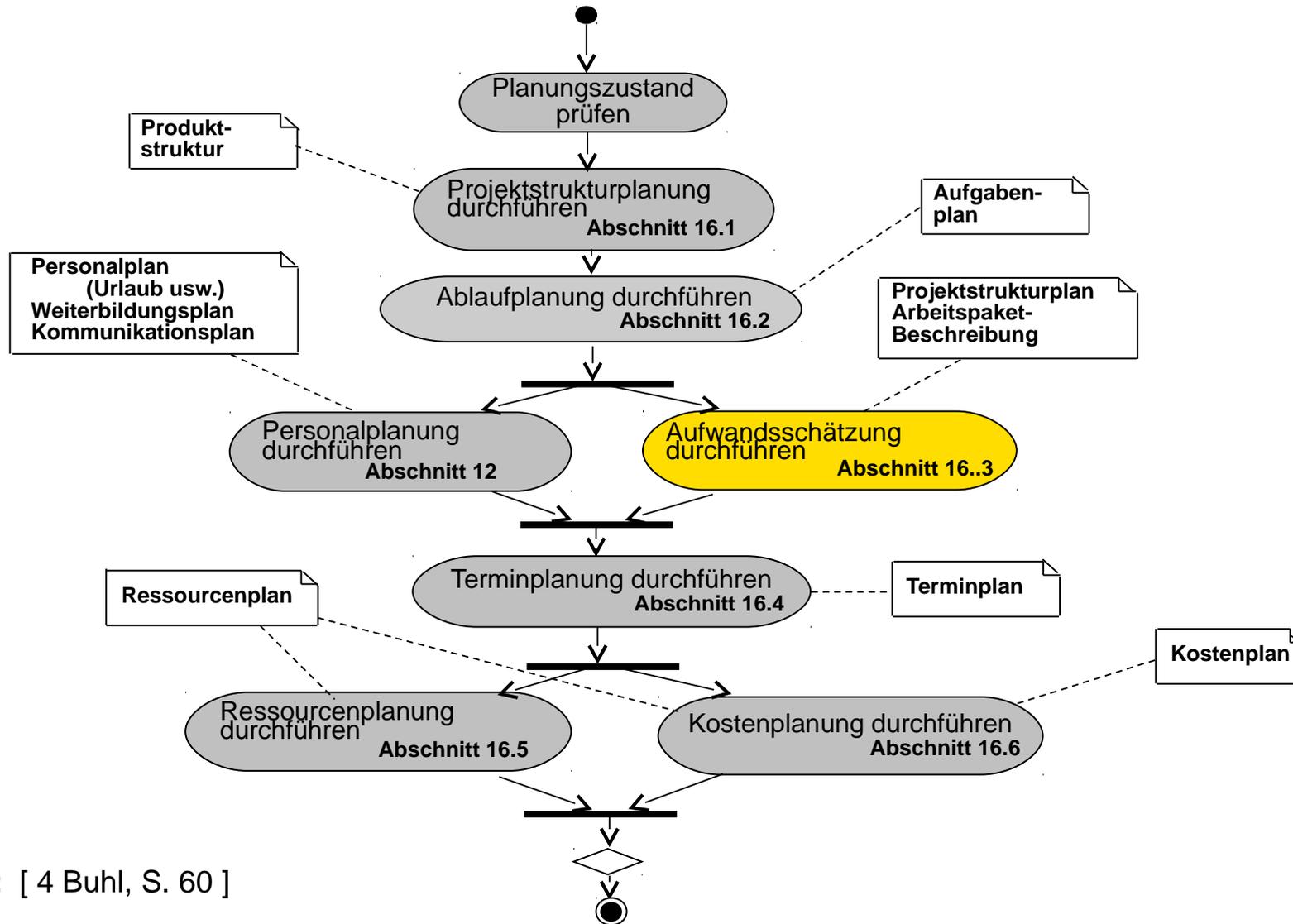


35

## Parkinson's Law

Work expands  
to fill the  
available  
volume...

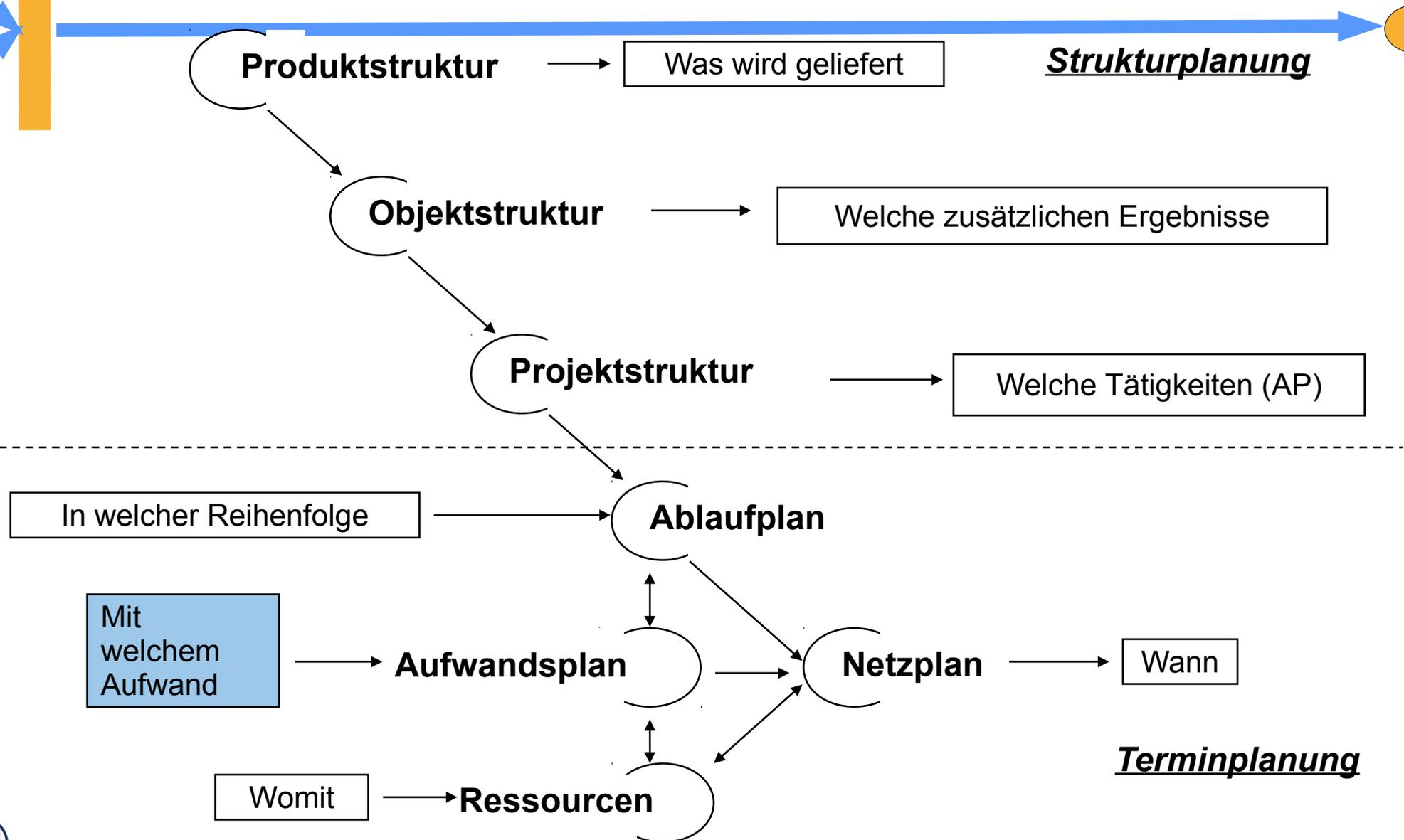
# Aufwandsschätzung



Quelle: [ 4 Buhl, S. 60 ]



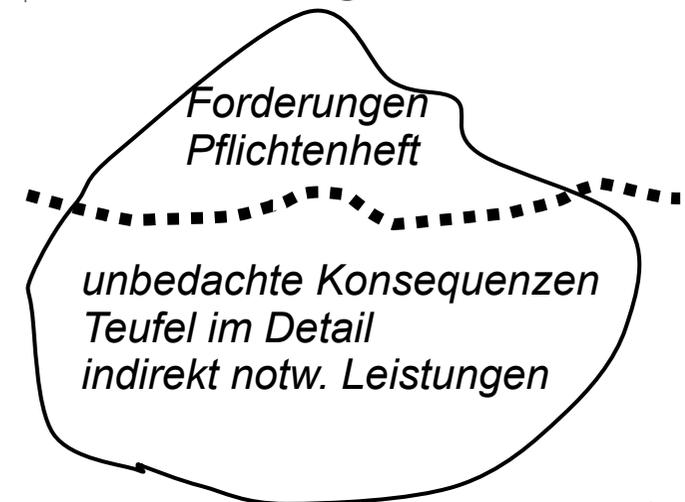
# Planungsablauf



# Aufwandsschätzung

- ▶ Die Aufwandsschätzung schätzt nicht den Preis, sondern die Kosten in der Einheit der Kostenkategorien
- ▶ 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

Ein **Einsatzmittel (Ressource)** ist eine abgrenzbare Gattung bzw. Einheit von Personal, Finanzmitteln, Sachmitteln, Informationen, Naturgegebenheiten, Hilfs- und Unterstützungsmöglichkeiten, die zur Durchführung oder Förderung von Vorgängen, Arbeitspaketen oder Projekten herangezogen werden können. [DIN 69901-5]

- ▶ 3 Ansätze der Schätzung:
  - Personenzeit (Tage, Monate)
  - Anzahl weiterer Einsatzmittel
  - Projekt-Zeitdauer, z. B. in Tagen

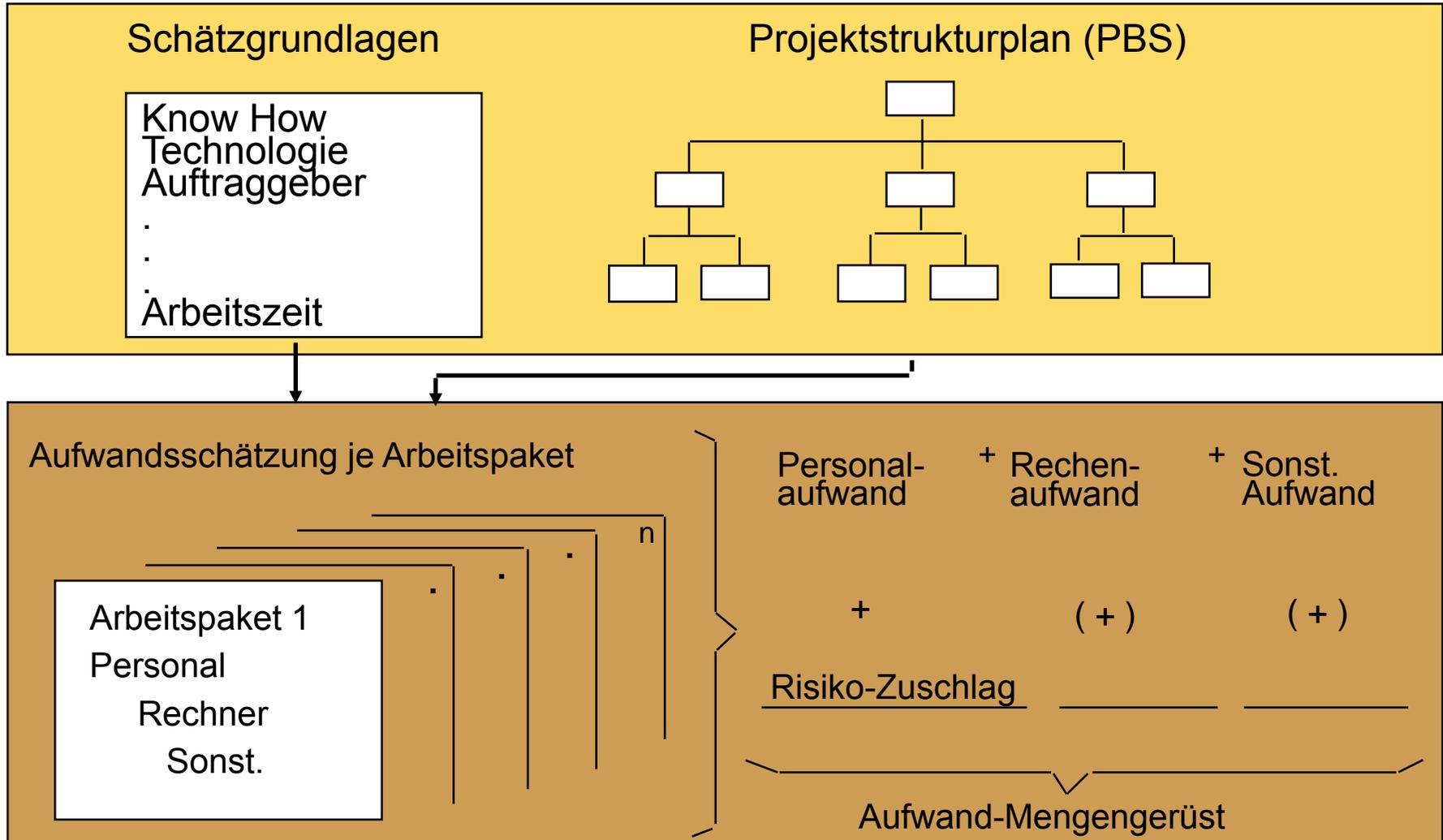
# Aufwandsschätzung

- ▶ Einsatzmittel- und Kostenkategorien einer Schätzung:
  - Personal -Zeit (und 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

# Methoden der Aufwandschätzung

- ▶ **Einzelschätzung (Drei-Punkt-Schätzung):** Sie wird von anerkannten Spezialisten/Experten durchgeführt.  
$$A = \frac{A_o + A_p + A_m}{3}$$
  - Gesamtaufwand A ergibt sich als arithmetisches Mittel dreier Schätzwerte für den optimistischen Aufwand  $A_o$ , den wahrscheinl. Aufw.  $A_w$  und den pessimistischen Aufw.  $A_p$
- ▶ **Mehrfachbefragung:** Interdisziplinär zusammengesetzte Gruppe von Experten schätzt den Aufwand nach einer bestimmten Vorgehensweise
  - z.B. Delphi-Methode, kombiniert mit Drei-Punkt-Schätzung
  - z.B. SWOT oder andere 2-D-Attributanalysen
- ▶ **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 Formelgebildes, dessen Strukturen, Variablen und Konstanten mit mathematischen Modellen bestimmt werden

# Aufsummation der Schätzelemente



Quelle: Deutsche Informatik Akademie



# 16.3.1 Delphi-Verfahren

- ▶ Das **Delphi-Verfahren** führt eine systematische Befragung mehrerer kompetenter Personen („Experten“) über den Zeitbedarf der einzelnen Aktivitäten durch

## a) Standard-Delphi-Verfahren (*anonym* und *ohne Anwesenheit*, meist per Web)

- 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.

## b) Breitband-Delphi-Verfahren (mit Anwesenheit)

- 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; Iteration von vorne.

# Breitband-Delphi mit 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 Schätzwertbildung (Entscheidungsregel nötig!)
- offenes Protokoll
- Dokumentation der Schätzergebnisse

**Quelle:** Deutsche Informatik Akademie

# Ablauf der Delphi-Schätzklausur

## Vorbereiten

- Schätzer auswählen und einladen
- Unterlagen zur Verfügung stellen

## Durchführen

- Ablauf Schätzklausur erläutern
- Basisinformation geben
- Projektstruktur durchsprechen
- Schätzeinheiten bilden
- Annahmen festhalten

## Einzelschätzung

- Arbeitspaket erläutern
- Mengen und Aufwand schätzen
- Schätzwerte diskutieren
- Ergebnisse festhalten
- Risiko abschätzen

- Schätzgenauigkeit ermitteln

## Nachbereiten

- Ergebnisse dokumentieren
- Zuschlagsrechnung
- Netto / Brutto-Umrechnung
- Plausibilitätsprüfung
- Ergebnisse und Dank an Schätzer

# 16.3.2 Standards der Function-Point-Methode

Die Function-Point-Methode ist eine Kennzahlenmethode.

Sie 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/IEC 14143-1** 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.ukσμα.co.uk](http://www.ukσμα.co.uk))

**Quelle:** Tagungsband ISWM/MetriKon 2004; Shaker Verlag 2004

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

48

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

## Schritte:

- 1) Ermitteln der Nutzfälle (function points)
- 2) Bewerten der Nutzfälle
- 3) Ermitteln der globalen Einflussgrößen (Einflussfaktoren)
- 4) Ermittlung der „Total Function Points“ (TFP)
- 5) Normierung mit Erfahrungstabelle

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

zu 1.: Ermitteln der Komponenten des Systems (aus Produktstrukturplan)

- ▶ „**Nutzfälle**“ (Use Cases aus dem Pflichtenheft und Entwurf) bzgl.
  - **Eingabedaten** (Formulare, BS-Masken, Daten von anderen S.)
  - **Ausgabedaten** (GUI-Masken, Reports, Listen, Daten für andere Systeme)
  - **Abfragen** (queries, 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)

## Bewerten der Nutzfälle

- ▶ **1. Schritt:**
- ▶ **Vergabe von 3 bis 15 Funktionspunkten (FP)** für die Nutzfälle, je nach ihrer Komplexität

Funktionsart	einfach mittel komplex		
	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 Zahl der einfachen, **unjustierten Funktionspunkte S1**

# Function-Point-Verfahren (3)

## Ermitteln der Einflusspunkte

51

### ▶ 2. Schritt:

#### ▶ Bewertung der **Einflusspunkte** für die Nutzfälle:

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

#### ▶ Maximal können 60 Punkte vergeben werden (**Summe der Einflusspunkte S2**).

#### Mögliche Einflussfaktoren:

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

# Function-Point-Verfahren (4)

## Globale Einflussfaktoren

52

- ▶ **3. Schritt:**
- ▶ Der globale **Einflussfaktor S3** kann **maximal 30%** des errechneten Wertes S2 betragen

$$S3 = 0,70 + (0,01 * S2)$$

- ▶ **4. Schritt:** Berechnung der „Total Function Points“

$$TFP = S1 * S3$$

# Function-Point-Verfahren (5)

## Einbeziehung von Erfahrungen

### 5. Schritt:

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

### Beispiel:

<b>Function Point</b>	<b>PM</b>	<b>Function Point</b>	<b>PM</b>	<b>Function Point</b>	<b>PM</b>
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.	

# 16.3.3 COCOMO-Verfahren

(CO<sup>n</sup>structive CO<sup>s</sup>t MO<sup>d</sup>el nach Barry Boehm)

- ▶ **Eingabe:** Systemgröße in DSI (Delivered Source Instructions) bzw. LOC (Lines of Code)
- ▶ **Resultat:** Personenmonate (PM) und Time for development (TDEV)

## Verfahren:

1. **Ermittlung der Codezeilenanzahl** in KLOC (Kilo Lines of Code), Summe der Schätzung je Modul/ Komponente)
2. **Berechnung der Personenmonate  $PM_0$**
3. **Korrektur mit den Einflussfaktoren/Kostentreibern**
4. **Ermittlung der Entwicklungszeit TDEV**
5. **Ermittlung Anzahl der Mitarbeiter**

# Projekt-Schwierigkeitsklassen in COCOMO

- ▶ Es gibt 3 Projektklassen im COCOMO:
- ▶ **Organic projects (einfache Softwareprojekte)**
  - kleine Teams, SW innerhalb des Hauses, mit Erfahrung mit ähnlichen Projekten
  - gute Sachkenntnis, klare Ziele, kein Termindruck
  - Produktgröße kleiner als 50 KDSI (Kilo Delivered Source Instructions)
- ▶ **Semi-detached projects (mittelschwere SW-Projekte)**
  - Team mit erfahrenen und weniger erfahrenen Mitarbeitern, Erfahrungen auf Teilgebieten des Projektes
  - Produktgröße kleiner als 300 KDSI
- ▶ **Embedded projects (komplexe SW-Produkte)**
  - größere Innovation, hohe Anforderungen an das Team
  - starker Kosten- und Termindruck
  - umfangreiches, komplexes SW-Produkt mit integrierten Elementen
  - Produktgröße: jede

# COCOMO-Varianten 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 KLOC)
  - 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 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: [ Jenny ]

# Formeln des COCOMO-Verfahren für Personenzeit-Bedarf und Projektdauer

	<u>BASIC-COCOMO</u>	<u>INTERMEDIATE-COCOMO</u>
<b>Organic:</b>	$PM = 2.4 * (KDSI)^{1.05}$ $TDEV = 2.5 * (PM)^{0.38}$	$PM = 3.2 * (KDSI)^{1.05}$
<b>Semi-detached:</b>	$PM = 3.0 * (KDSI)^{1.12}$ $TDEV = 2.5 * (PM)^{0.35}$	$PM = 3.0 * (KDSI)^{1.12}$
<b>Embedded:</b>	$PM = 3.6 * (KDSI)^{1.20}$ $TDEV = 2.5 * (PM)^{0.32}$	$PM = 2.8 * (KDSI)^{1.20}$

PM = Personenmonate

KDSI = Kilo Delivered Source Instructions (in KLOC)

TDEV = Time for Development (optimale Projektdauer ==> daraus Personenzahl abschätzen)

## Beispiel-Werte, auf qualifizierte Informatiker umgerechnet:

Quelle: [ Jenny ]

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

# COCOMO-Verfahren: Einflussfaktoren

Boehm unterscheidet 15 **Einflussfaktoren/ Kostentreiber** in 4 Klassen, aufgeteilt auf einzelne 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)

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

<b>PCAP</b>	<b>PD</b>	<b>DD</b>	<b>CUT</b>	<b>IT</b>
<b>very high</b>	1.00	0.65	0.65	0.65
<b>high</b>	1.00	0.83	0.83	0.83
<b>nominal</b>	1.00	1.00	1.00	1.00
<b>low</b>	1.00	1.20	1.20	1.20
<b>very low</b>	1.00	1.50	1.50	1.50

# COCOMO-Verfahren (5)

## Schritte:

1. Ermittlung der KLOC (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:** Semidetached-und Intermediate-Projekt mit 20 KLOC

$$PM_0 = 3.0 * 20^{1.12} = 86 \text{ notwendige Personenmonate (Basismodell)}$$

Annahme: alle Kostentreiber haben den Wert „nominal“ (0) außer CPLX und LEXP:

- CPLX (Komplexität): 1.15
- LEXP (Erfahrung in der Progr.-Sprache): 1.10

Dann sind:

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

$$N = PM / TDEV = 109 / 12,9 = \underline{8,5} \text{ Mitarbeiter}$$

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

# Praxis Cocomo/FPM

- ▶ Erstelle ein Spreadsheet mit den Formeln der Schätzmethoden
  - Für unterschiedliche Schwierigkeitsgrade
  - Für unterschiedliche Zeitpunkte
- ▶ Wähle die Parameter
- ▶ Bestimme die Einflussfaktoren
- ▶ Bestimme die Produktivität
- ▶ Berechne Schätzung.

# The End

