

Fakultät Informatik

Professur Softwaretechnologie

SOFTWAREMANAGEMENT

15_PROJEKTPLANUNG I

Prof. Dr. Uwe Aßmann
Dr.-Ing. Birgit Demuth
Sommersemester 2017

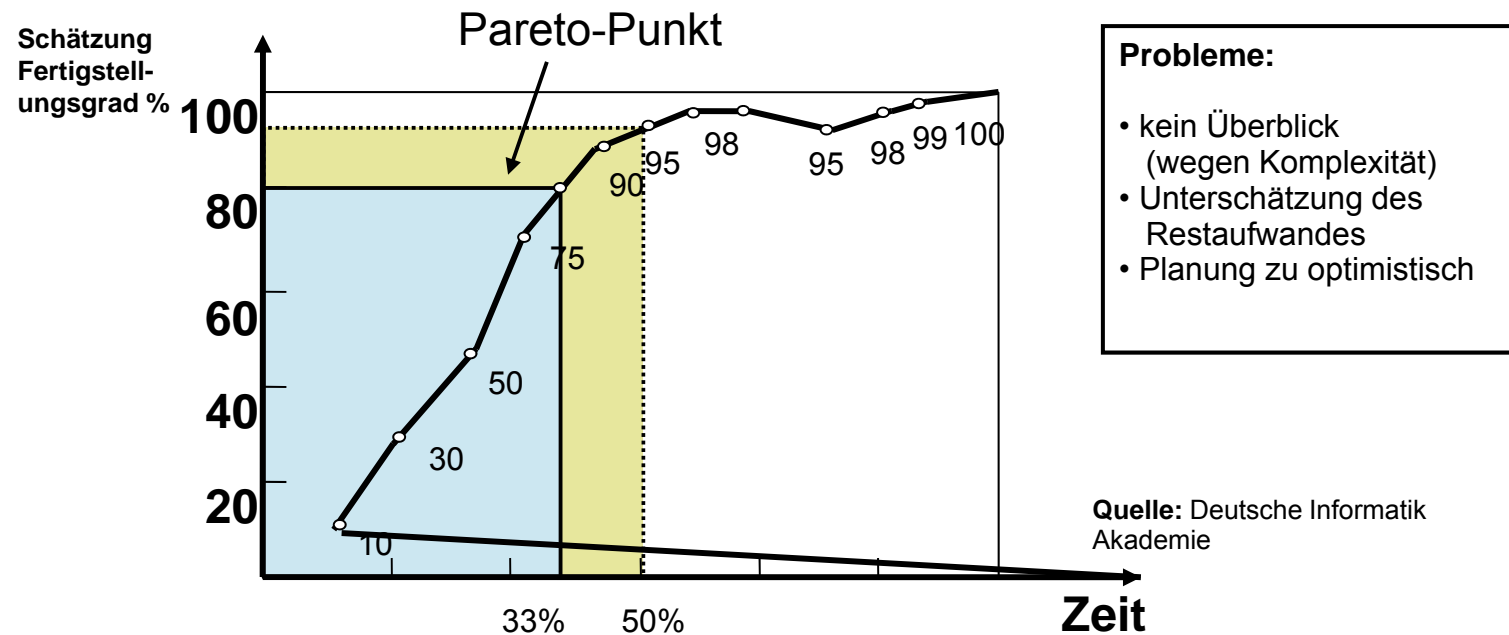
Überblick

- Einführung
- Projektstrukturplanung
- Ablaufplanung
- Aufwandsschätzung

Einführung

Das 90%-Syndrom (Subjektive Fehleinschätzung der Fertigstellung)

Etwa zur Hälfte der Projektlaufzeit glaubt man, bereits 90% bis 95% des Projektergebnisses zu haben.



Aufgaben der Projektplanung (1)

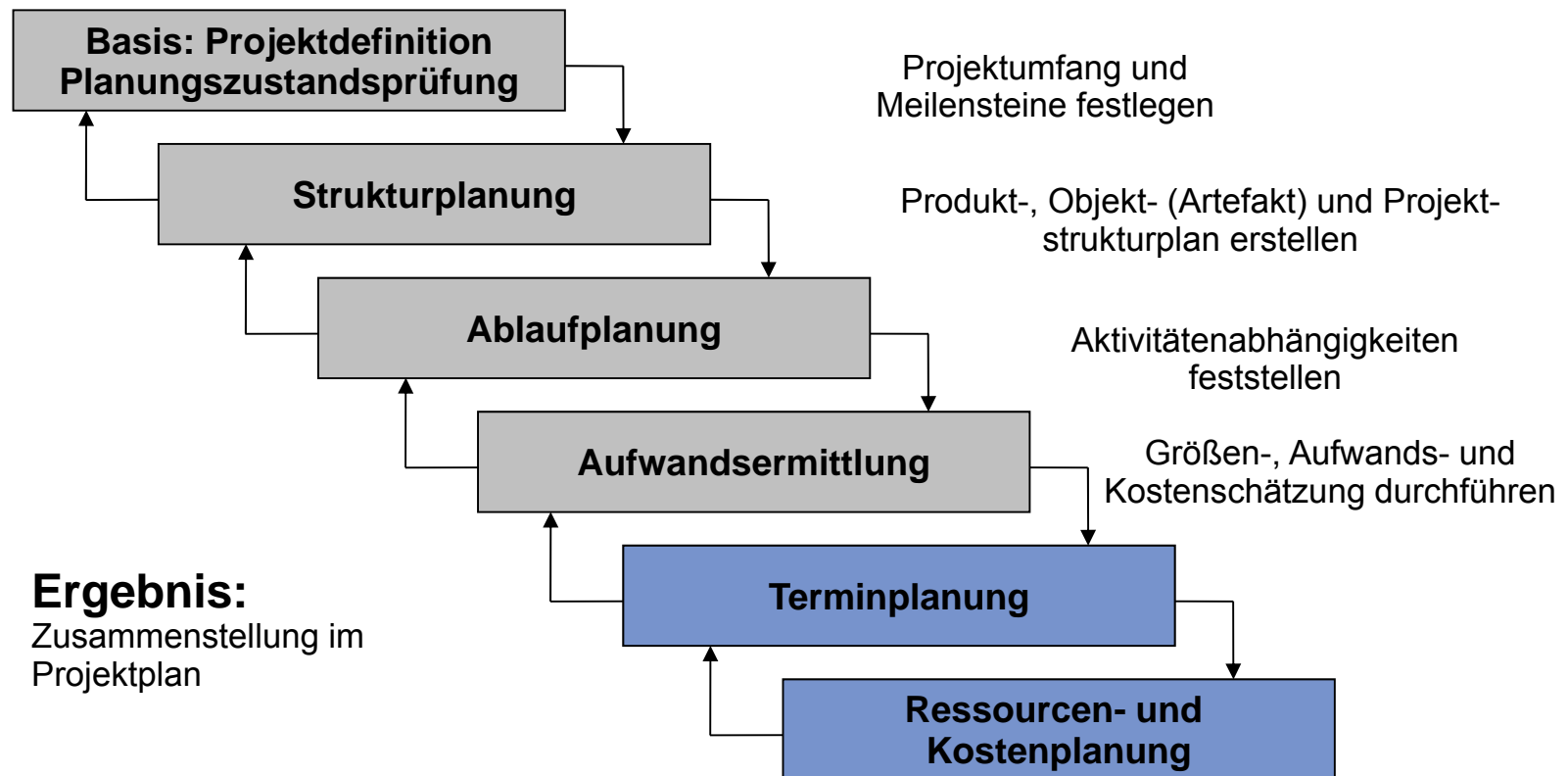
Die **Projektplanung** muss im Projektplan die vorhandenen Ressourcen an Personal, Zeit, Geld, Maschinen, Räume so einteilen, dass auf Änderungen der Arbeitsabläufe rasch und kosteneffizient reagiert werden kann.

- ▶ Operative Planung:
 - Auswahl eines Modells der Ablauforganisation, nach dem alle zu erstellenden Zwischen- und Endprodukte für das Projekt bestimmt werden.
 - Ziel: idealer Plan zur Minimierung der Prozessrisiken, auf dessen Basis mit dem AG (Kunden) verhandelt werden kann.
- ▶ Qualitätsplanung:
 - Planung der Maßnahmen für jedes Qualitätskriterium
- ▶ Wirtschaftliche Planung (Kostenplanung):
 - Planung von Personal, Ressourcen und der Finanzierung
 - Projektrisikobehandlungsplanung

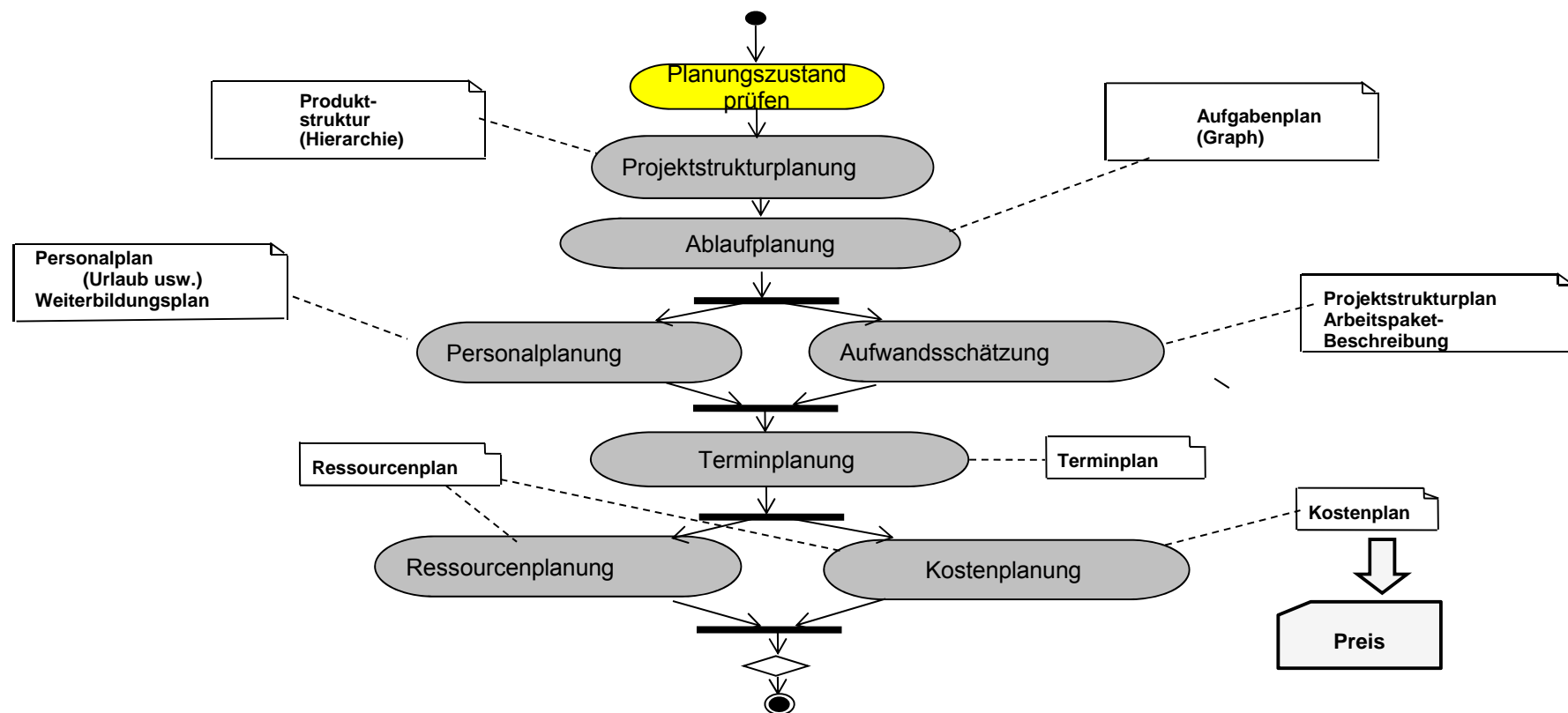
Aufgaben der Projektplanung (2) [Vorl. Prof. H. Schmidt]

- ▶ Planung und Vorbereiten des **Controlling**
 - wie messen und verbessern wir die Prozesse?
 - Ermittlung realistischer Sollvorgaben
 - Verbesserung der Effizienz der Projektabwicklung
 - Frühe Fehlererkennung und -Korrektur
 - Dokumentation der Vorgaben
- ▶ Planung der **Prozessverbesserung**
 - Retrospektive (Nachstudie)
 - Wie gestalten wir unsere Prozesse besser?
 - Wie verbessern wir den Feedback im PDCA?

Übersicht Schritte der operativen und wirtschaftl. Planung



Aktivitätsdiagramm der Planungsphase



Planungszustand prüfen: Verschiedene Planungshorizonte

- ▶ An jedem Meilenstein der Phasengliederung setzt man eine Verfeinerung der Planung an

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

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; Teilverträge, Teilprojektierung einsetzen

Projekte mit der *öffentlichen* Hand

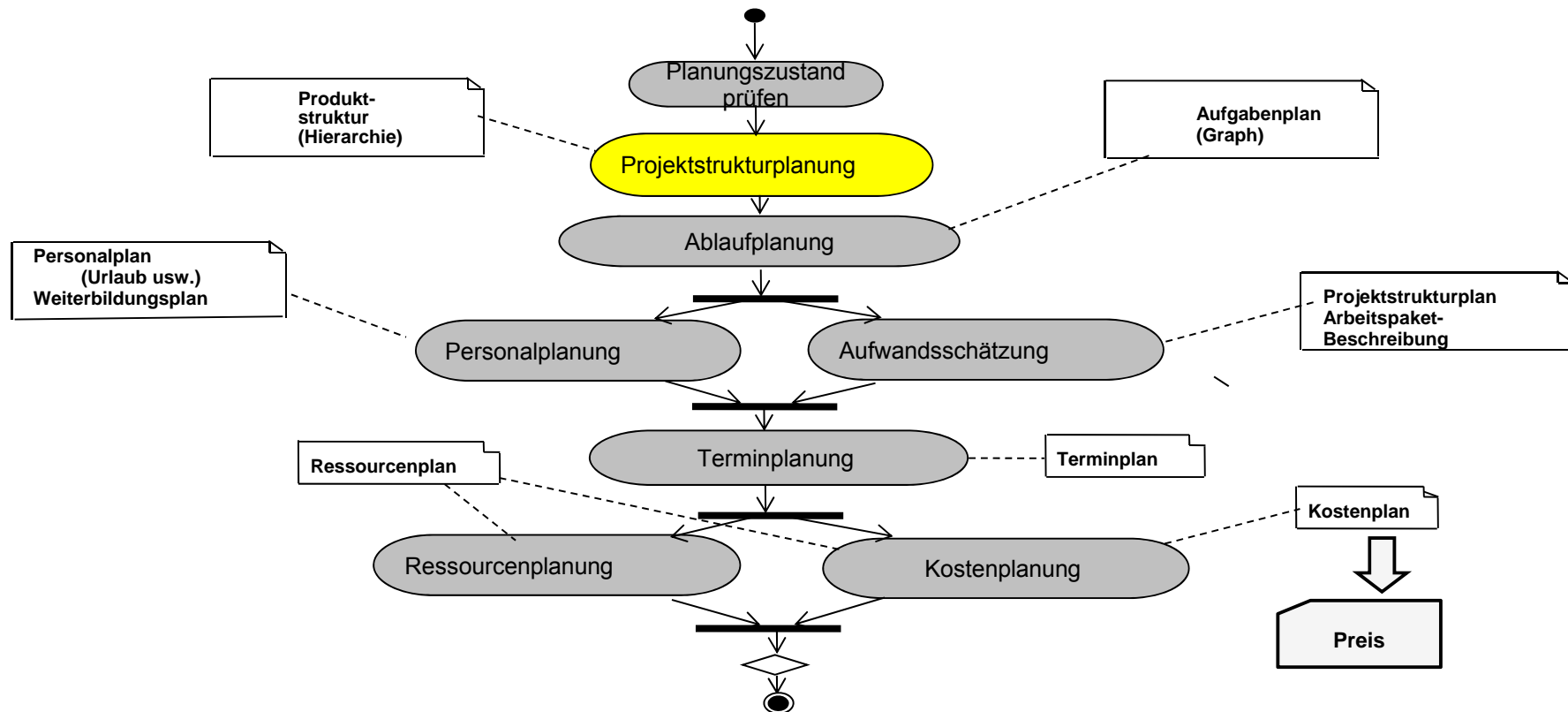
- Ausschreibung verlangt Festpreiskalkulation. Sehr großes Problem in Deutschland.

Projektstrukturplanung

Projektstrukturplan (**PSP**)/Project Breakdown Structure (**ProBS**) bestehend aus

- Produktstrukturplan/Product Breakdown Structure (**PBS**)
- Artefaktstrukturplan/Artefact Breakdown Structure (**ABS**)
- Arbeitsstrukturplan/Aktivitätenstrukturplan/Work Breakdown Structure (**WBS**)

Aktivitätsdiagramm der Planungsphase



Projektstrukturplanung

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

**(End-)Produktstruktur
(Product Breakdown Structure)**

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

**Artefaktstruktur (Objektstruktur)
(Artefact Breakdown Structure ABS)**

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

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

**Arbeitsstruktur
(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?

**Projektstruktur
(Project Breakdown Structure ProBS)**

Quelle: Deutsche Informatik Akademie

Komponenten der Projektstrukturplanung

| Inhalt | Strukturplan |
|--|---|
| 1. Das zu liefernde End-Produkt | 1 = Produktstrukturplan |
| 2. Zur Erstellung des End-Produktes notw. Artefakte (Arbeitsergebnisse, Zwischenergebnisse) | 1 + 2 = Artefaktstrukturplan (Objektstrukturplan) |
| 3. Für die Abwicklung des Projektes nötige Aufgaben | 3 = Arbeitsstrukturplan (Arbeitspakete, Aktivitäten) |
| 1 + 2 + 3 = Projektstrukturplan | |

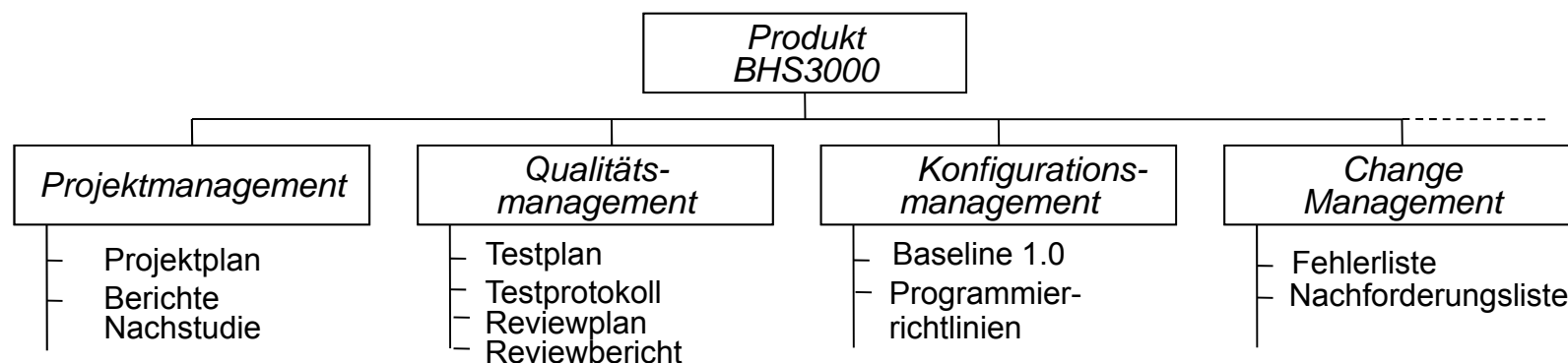
Quelle: Deutsche Informatik Akademie

Dekomposition von Produktstrukturplänen

- ▶ **(Projekt-)Funktionsorientierter Produktstrukturplan** dekomponiert das Produkt anhand der Projektfunktionen (siehe VMXT, EOS)
- ▶ **Anforderungsorientierter (Produkt-funktionsorientiert, Feature-oriented) Produktstrukturplan** dekomponiert das Produkt anhand der *Produktfunktionen* (Features, Anforderungen, **Funktionsbaum** des Projekts)
- ▶ **Komponentenorientierter Produktstrukturplan** dekomponiert anhand von Systemkomponenten (System, Subsysteme, Klassen, siehe EOS)

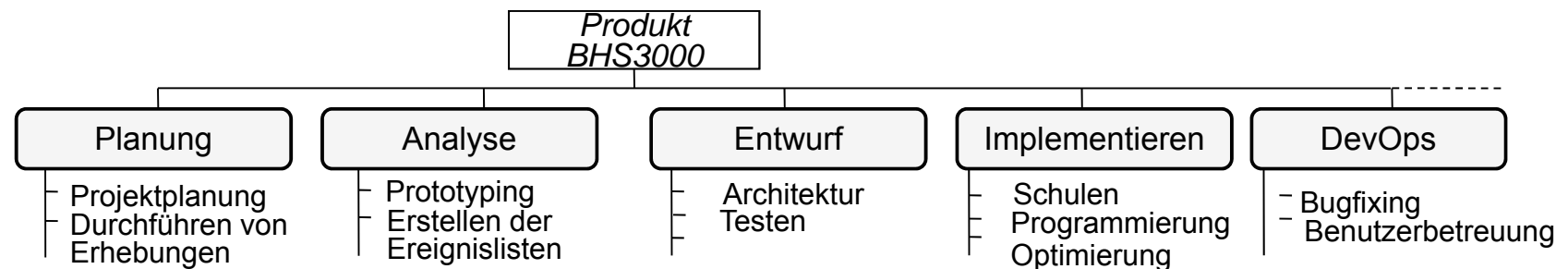
Artefaktstrukturplan (Artifact Breakdown Structure, ABS)

- ▶ Die zur Erstellung der Produktkomponenten nötigen Artefakte (Zwischenprodukte) werden I.d.R. aus den "Produktgruppen" eines Prozessmodell wie VMXT entnommen
- ▶ Dabei können auch gleich Aktivitäten aus den zugehörigen "Aktivitätengruppen" ermittelt werden und in die Work Breakdown Structure eingetragen werden



Arbeitsstrukturplan (Aktivitätenstrukturplan, WBS)

- ▶ Der **Arbeitsstrukturplan (Aktivitätenstruktur, Work Breakdown Structure, WBS)** ist ein Aktivitätenbaum mit allen Aktivitäten, die zu tun sind, um die Projektziele zu erreichen
- ▶ Darstellung als strukturierte Aktivitätenliste mit 3 Ebenen:
 - **1. Ebene:** Projektbezeichnung
 - **2. Ebene:** Hauptaktivitäten
 - **3. Ebene:** Unter-Arbeitspakete



[anlehnend B. C. Schreckeneder]

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]

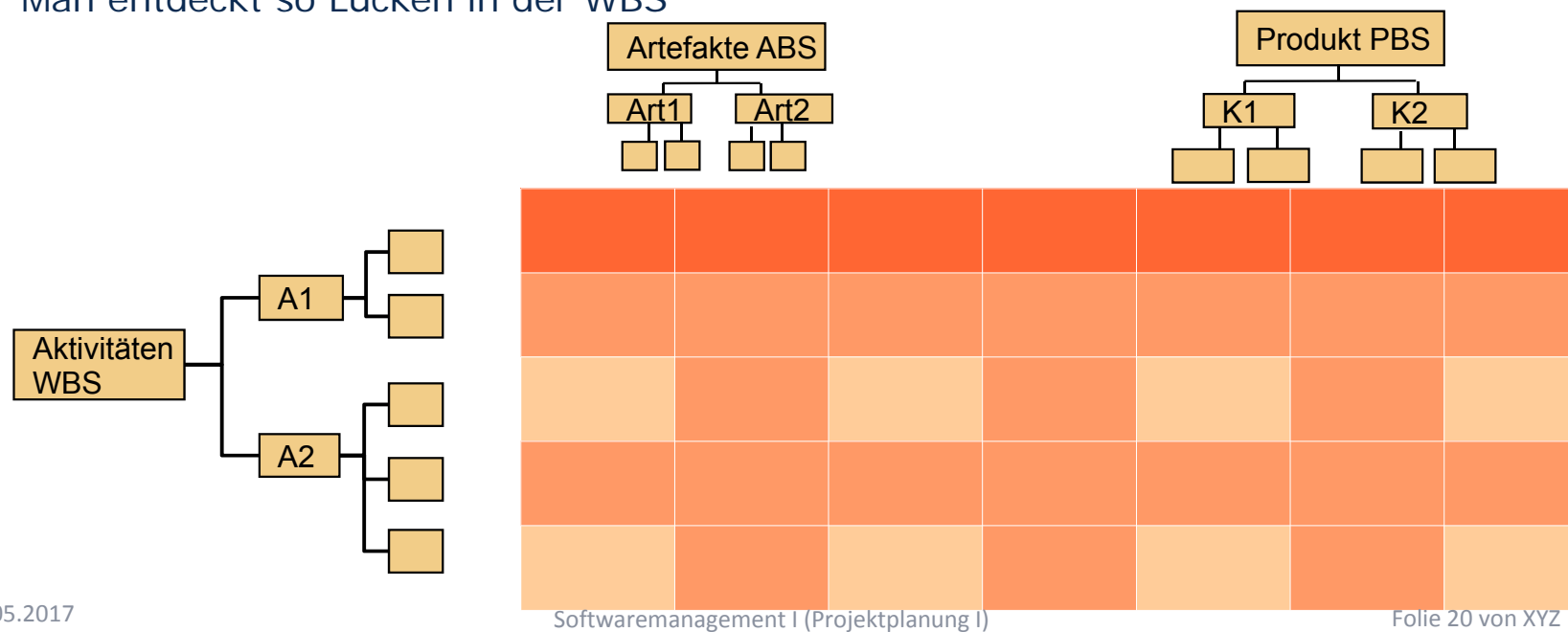
- ▶ Ziele
 - planbare Arbeitsvolumen (überschaubar, abrechenbar)
 - eigenverantwortliche Durchführung durch organisatorische Einheit oder Person
 - Projektverfolgung
- ▶ Arbeitspakete (AP) sind Grundlage für Aufwandsermittlung und Erstellung des Netzplanes
- ▶ Klare Arbeitspakete: CCC (checkable, consistent, complete)
 - Disjunkt: Arbeitspakete müssen klar voneinander abgegrenzt sein

Beispiel Attribute eines Arbeitspakets [B.C. Schreckeneder]

| Projektnummer: Projektname: | Seite von |
|---|---|
| Arbeitspaket-Nummer | 1.1.1 Projektstart durchführen im Organisationsprojekt |
| Inhalt | 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 |
| Nicht-Inhalt | Zusammenstellung Projektteam |
| Ergebnisse | Klare Ziele, Erstansatz Projekthandbuch, Zustimmung von Projektauftraggeber |
| Leistungsfortschrittmessung | 40% Zieldefinition, 40% Projekthandbuch, 20% Zustimmung Projektauftraggeber |
| Verantwortlich | Frau M. Mayer |
| Dauer und Termine | 14.8. - 13.9. ... |
| Zeitaufwand/Ressourcen | 64 Std./P-Teammitglied ohne Projektleiter/in plus 80 Std. Projektleiter/in |
| Kosten |€ Personalkosten plus 1.200,-€ Tagungs- und Übernachtungskosten |
| Abhängigkeit/Schnittstellen zu anderen Arbeitspaketen/Projekten | |

Ableitung von Arbeitspaketen aus PBS und ABS

- ▶ Aus der PBS und der ABS kann die WBS abgeleitet werden
- ▶ Die PBS und die Artefaktstruktur können mit der WBS kreuzgeprüft werden (Kreuzmatrix)
- ▶ Man entdeckt so Lücken in der WBS



Projektstrukturplan [anlehnend B. C. Schreckeneder]

- ▶ Der **Projektstrukturplan (PSP, Project Breakdown Structure (ProBS))** ist ein Baum mit
 - allen Aktivitäten, die zu tun sind, um die Projektziele zu erreichen
 - allen Artefakten, die zu erstellen sind
- ▶ Zentrales Kommunikationsinstrument
 - Stabiles Planungsinstrument (Termin- und Kostenänderungen haben darauf keinen Einfluss)
- ▶ Darstellung als strukturierter Aktivitäten- oder Strukturbaum mit 3 Ebenen:
 - **1. Ebene:** Projektbezeichnung
 - **2. Ebene:** Strukturierung des Projektes nach verschiedenen Gliederungsgesichtspunkten (z.B. Funktionen, Phasen, Artefakten ...)
 - **3. Ebene:** Arbeitspakete

Ablaufplanung

- Vorgangsknotennetz
- Abhängigkeitsdiagramm (-graph), später Netzplan

Ablaufplanung (Abhängigkeitsanalyse der Teilaufgaben)

Die **Ablaufplanung** soll die **logischen Abhängigkeiten** der Arbeitspakete, d.h. die Ablaufreihenfolge logisch und verständlich präsentieren.

- Aus den Arbeitspaketen wird eine **Vorgangsliste** abgeleitet
- Deren **Abhängigkeiten** werden ermittelt
- Auf dieser Grundlage wird der **Abhängigkeitsdiagramm/-graph** und der **Netzplan** erstellt

Vorgangsliste

- ▶ Eine strukturierte **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)
 - benötigte Ressourcen
 - Abhängigkeiten (Welche Aktivitäten können unabhängig voneinander ausgeführt werden?)
 - Priorität (ABC)
 - hierarchische Nummerung
- ▶ Als praktisches Tool zur Erstellung und Verwaltung der Vorgangsliste bietet sich an
 - Projektmanagement-Werkzeug wie MS Project, OpenProj
 - Zur Not Textverarbeitungsprogramm mit Gliederungsansicht, Spreadsheet

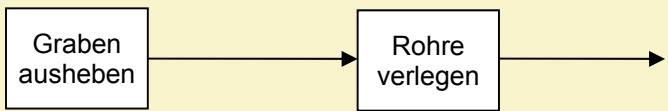
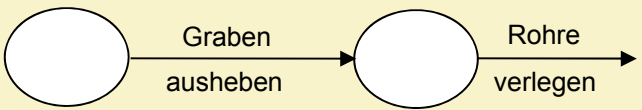
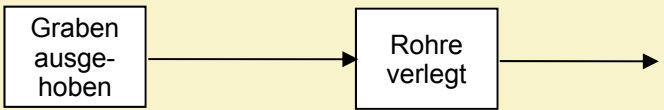
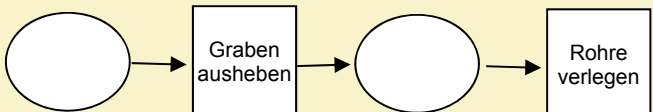
Beispiel Vorgangsliste mit Abhängigkeiten

| Vorgangsliste | | | | | Projekt: Aussteller: | | Nr.: Datum: | | Seite: | | | | |
|---------------|---|--------------------|----|----|-------------------------|------------------|-----------------------|------------------------|--------------|----|----|--------|----|
| Nr. | Projekt-tätigkeit 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 | | | | | | | I | | | | | |
| I | Arbeitspaket 09 | | | | | | H | K | | | | | |

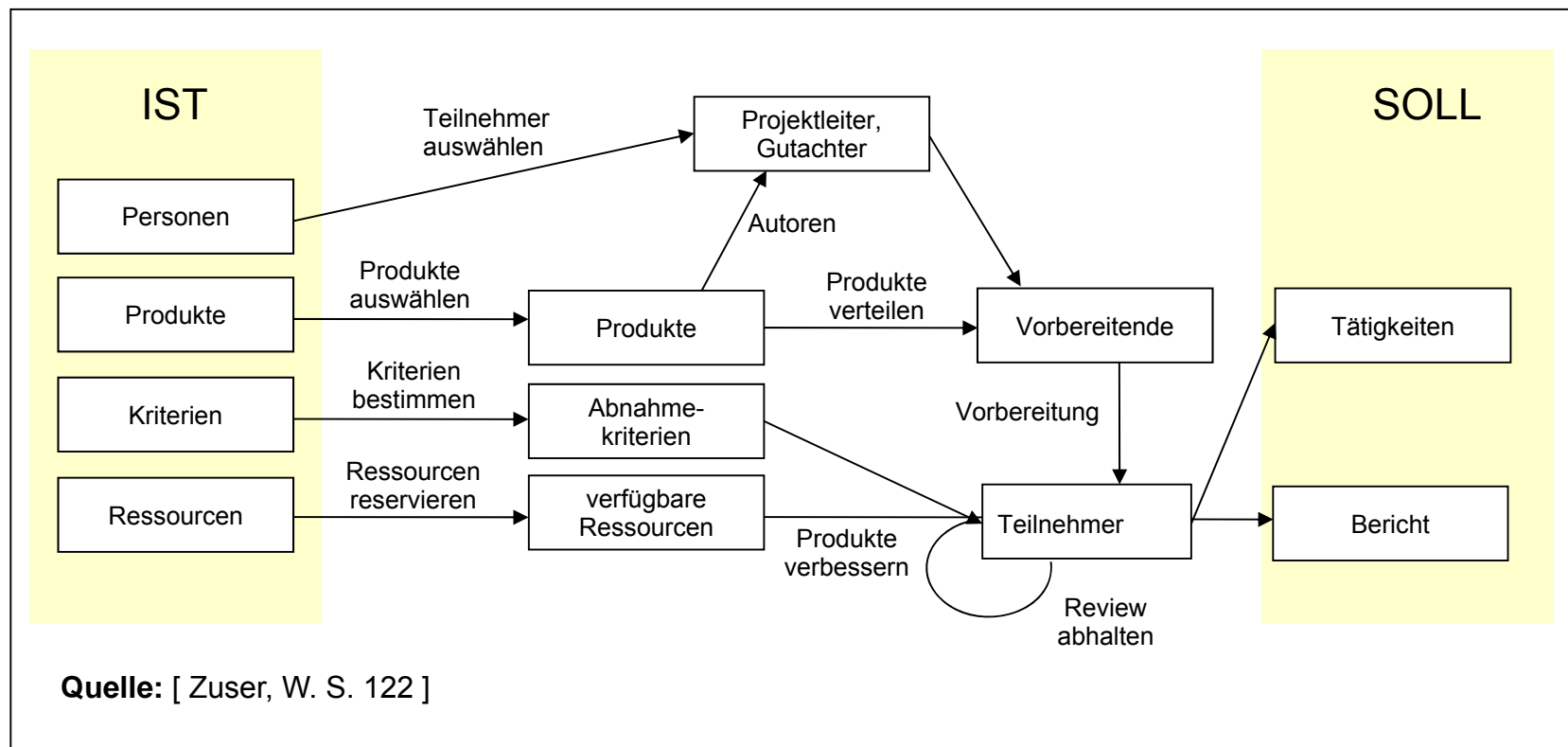
FA = frühestmöglicher Anfang des Vorgangs
 SA = spätestzulässiger Anfang des Vorgangs
 SE = spätestzulässiges Ende des Vorgangs
 FE = frühestmögliches Ende des Vorgangs
 GP = Gesamte Pufferzeit
 FP = Freie Pufferzeit
 UP = Unabhängige Pufferzeit
 MA = Personal (Mitarbeiter/Mitarbeiterin)
 SM = Sachmittel (pro Vorgang)

Quelle: [1 Jenny, S. 242]

Darstellungen von Aktivitäten in Abhängigkeitsdiagrammen

| Diagrammart | Darstellung der Bestandteile | Beispiel | |
|---|--|--|---|
| Vorgangsknotennetz (Aktivitätendiagramm, Datenflussdiagramm) Knoten: Vorgänge |  | PDM MPM UML-Activity Diagrams | PDM: Precedence Diagramm Method (auch MS Project) MPM: Metra Potential Method CPM: Critical Path Method PERT: Program Evaluation and Review Technique |
| Vorgangspfeilnetz Knoten: Zustände Pfeile: Vorgänge |  | CPM UML-Statecharts | |
| Ereignisknotennetz Knoten: Ereignisse |  | PERT VMXT | |
| Bipartite Netze Stellen: Zustände Rechtecke: Vorgänge (synchronisierend) |  | PetriNet WorkflowNets BPMN | |

Beispiel Abhängigkeitsdiagramm(-graph) für ein Review als Vorgangspfeil-Netz



Aufwandsschätzung

- Überblick
- Delphi-Verfahren
- Function-Point-Verfahren
- COCOMO-Verfahren

Parkinsons Gesetz zum Bürokratiewachstum (1955)

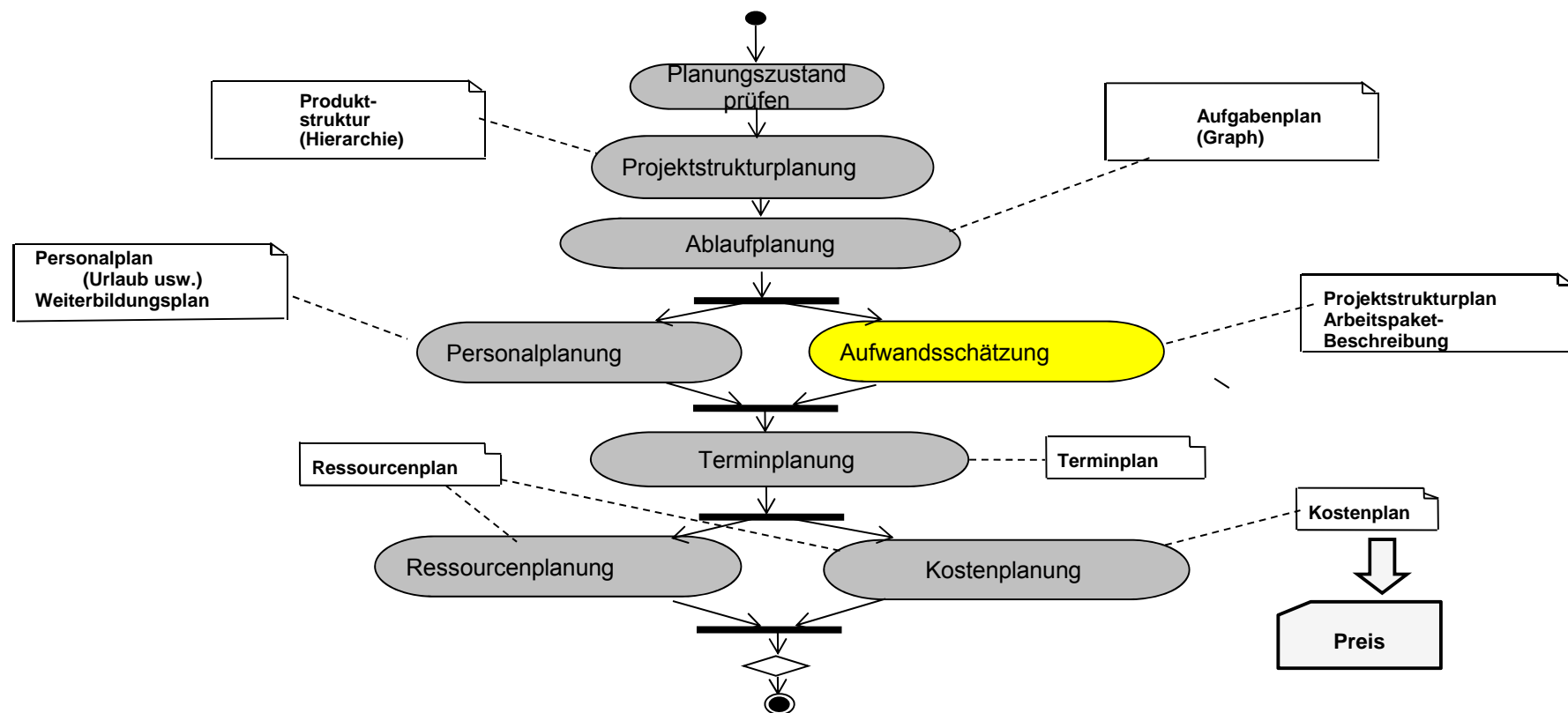
“Work expands to fill the time available for its completion”

bedeutet

Arbeit dehnt sich in genau dem Maß aus, wie Zeit für ihre Erledigung zur Verfügung steht ... und nicht in dem Maß, wie komplex sie tatsächlich ist.

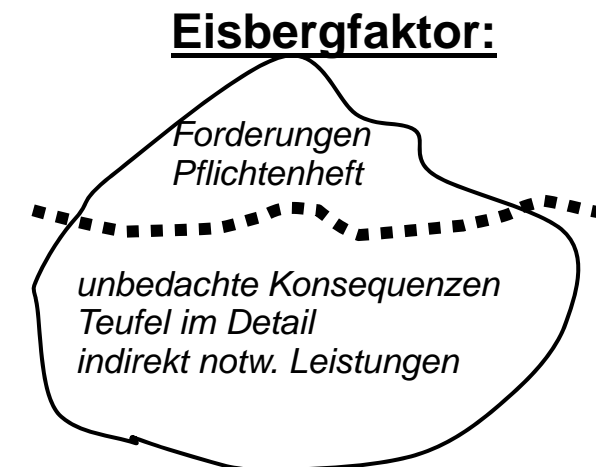
[https://de.wikipedia.org/wiki/Parkinsonsche_Gesetze]

Aktivitätsdiagramm der Planungsphase



Aufwandsschätzung und das Eisberg-Problem

- ▶ 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!



Ressourcen

Ein **Einsatzmittel (Resource)** 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]

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.)
- ▶ Bei signifikanten Änderungen neu schätzen
- ▶ Nachkalkulation zur Auswertung von Erfahrungen

Methoden der Aufwandschätzung (1)

- ▶ Einzelschätzung (Drei-Punkt-Schätzung): Sie wird von anerkannten Spezialisten/Experten durchgeführt.

Gesamtaufwand A ergibt sich als arithmetisches Mittel dreier Schätzwerte für

- den optimistischen Aufwand A_o ,
- den wahrscheinlichsten Aufwand A_w
- den pessimistischen Aufwand A_p

$$A = \frac{A_o + A_p + A_w}{3}$$

- ▶ Oft auch gewichteter Schätzwert (mit $4 * A_w$)
- ▶ Kann mit allen anderen Schätzverfahren orthogonal kombiniert werden!
- ▶ In einem Schätz-Excel-Sheet sehe man immer eine Drei-Punkt-Schätzung vor

Methoden der Aufwandschätzung (2)

- ▶ Expertenschätzungen mit interdisziplinär zusammengesetzter Gruppe von Experten
 - z.B. Delphi-Methode, kombiniert mit Drei-Punkt-Schätzung
 - z.B. SWOT oder andere 2-D-Attributanalysen
- ▶ Analogiemethoden (Prozentsatzmethode, Multiplikatormethode)
 - (Grobe) Schätzung im Vergleich zu abgeschlossenen ähnlichen Projekten (Voraussetzung ist aktives Sammeln von Projektdaten)
 - Prozentsatzmethode: Analogieschlüsse von einer Phase auf eine andere
 - Multiplikatormethode: Analogieschlüsse von einer Größe (z.B. LOC) auf eine andere (z.B. Kosten)
- ▶ Algorithmische Methoden
 - bedienen sich Formeln oder eines Formelgebildes, dessen Strukturen, Variablen und Konstanten mit mathematischen Modellen bestimmt werden

Delphi-Verfahren zur Expertenbefragung (1)

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

Top-Down-Methode, gut geeignet zu Beginn eines Projektes für grobe Schätzungen

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.
- Das Schätzergebnis ergibt sich aus dem Durchschnittswert der letzten Überarbeitung.

Delphi-Verfahren zur Expertenbefragung (2)



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.

- **Besetzung:** Moderator, 3-4 Schätzer = Experten aus Projektteam oder extern, 1-2 Berater aus Projektteam, Protokollführer
- **Arbeitstechnik:** verdeckte Einzelschätzung, gemeinsame Schätzwertbildung (Entscheidungsregel nötig!), offenes Protokoll, Dokumentation der Schätzergebnisse

Function-Point-Methode (1)

Die Function-Point-Methode ist eine weit verbreitete **Kennzahlenmethode**, ursprünglich von IBM (1979). 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 und davon abgeleitete Public Available Standards (PAS):
- **ISO/IEC 20926** standardisiert für eine spezifische **I**nternational **F**unction **P**oint **U**ser **G**roup die Methode, die unter der Bezeichnung **IFPUG Function Point Methode Version 4.1** bekannt geworden ist. [www.ifpug.org]
- **ISO/IEC 19761** nach diesem Standard nutzt die **COSMIC-FFP** (**C**ommon **S**oftware **M**easurement **I**nternational **C**onsortium - Full Function Points) die Methode [www.cosmicon.com]
- Deutschsprachige Anwendergruppe für Software-Metrik und Aufwandschätzung e.V. (DASMA) [<http://www.dasma.org/>]

Function-Point-Verfahren (2)

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

Schritte:

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

Function-Point-Verfahren (2)

Schritt (1): Ermitteln der Nutzfälle / Function Points (FP) bezüglich von

- **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 (3)

Schritt (2): Bewerten der Nutzfälle durch **Vergabe von 3 bis 15 FP** für die Nutzfälle, je nach ihrer 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 Zahl der einfachen, **unjustierten Funktionspunkte S1**

Function-Point-Verfahren (4)

Schritt (3a): Ermitteln der
Einflussgrößen (Einflussfaktoren) :

- 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 (5)

Schritt (3b): Ermitteln des globalen **Einflussfaktors S3**

- kann **maximal 70%** des errechneten Wertes der Einflusspunkte S2 betragen

$$\mathbf{S3 = 0,70 + (S2 * 0,01)}$$

Schritt (4): Ermittlung der „Total Function Points“ (TFP)

$$\mathbf{TFP = S1 * S3}$$

Function-Point-Verfahren (6)

Schritt (5): Normierung mit Erfahrungstabelle

- anhand der ermittelten TFP 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 einer Erfahrungstabelle

| TFP | 150 | 200 | 250 | 300 | 350 | 400 | 450 | 500 | 550 |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| PM | 5 | 9 | 13 | 17 | 21 | 25 | 29 | 33 | 37 |

COCOMO-Verfahren (1)

(CONstructive COSt MOdel nach Barry Boehm, 1981)

[http://sunset.usc.edu/csse/research/COCOMOII/cocomo_main.html]

Kennzahlenmethode

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

Schritte:

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

Projekt-Schwierigkeitsklassen in COCOMO (2)

Es gibt 3 Projekt-Schwierigkeitsklassen 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 (3)

Die COCOMO-Varianten unterscheiden sich in den Schätzformeln.

- ▶ **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

COCOMO-Verfahren (4)

Beispiel-Werte auf qualifizierte Informatiker umgerechnet

Variablen für Schätzformeln

- PM: Personenmonate
- DSI: Delivered Source Instructions (in LOC)
- TDEV = Time for Development (optimale Projektdauer → daraus Personenzahl abschätzen)

| Produktgröße | Notwendige PM | Produktivität in DSI/PM | Entwicklungszeit in Monaten (TDEV) | Personenzahl |
|--------------|---------------|-------------------------|------------------------------------|--------------|
| Small | 5.0 | 400 | 4.6 | 1.1 |
| Intermediate | 21.3 | 376 | 8.0 | 2.7 |
| Medium | 91.0 | 352 | 14.0 | 6.5 |
| Large | 392.0 | 327 | 24.0 | 16.0 |

COCOMO-Verfahren: Kostentreiber (5)

- 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

Tabelle

- Kostentreiber mit Erfahrungswerten
- [<http://www.software-kompetenz.de>]

| Name | Beschreibung | min | max |
|---|--|------|------|
| Produkt-abhängige Attribute | | | |
| RELY | Erforderliche Systemzuverlässigkeit | 0.75 | 1.40 |
| DATA | Größe der verwendeten Datenbank | 0.94 | 1.16 |
| CPLX | Komplexität des Produkts | 0.70 | 1.60 |
| Rechner-abhängige Attribute | | | |
| TIME | Ausführungszeit Beschränkung | 1.00 | 1.60 |
| STOR | Speicherbeschränkungen | 1.00 | 1.56 |
| VIRT | Software-Entwicklungsumgebung | 0.87 | 1.30 |
| TURN | Computer-Antwortzeiten/-Anforderungen | 0.87 | 1.30 |
| Personen-abhängige Attribute | | | |
| ACAP | Fähigkeit der Projektanalytiker | 0.71 | 1.46 |
| AEXP | Erfahrung mit der Anwendung | 0.82 | 1.29 |
| PCAP | Fähigkeiten der Programmierer | 0.70 | 1.42 |
| VEXP | Erfahrung mit Softwareentwicklungsumgebung | 0.90 | 1.21 |
| LEXP | Erfahrung mit der Programmiersprache | 0.95 | 1.14 |
| Projektumgebungs-abhängige Attribute | | | |
| MODP | Erfahrung mit modernen Programmiermethoden | 0.82 | 1.24 |
| TOOL | Verfügbare Software-Tools | 0.83 | 1.24 |
| SCED | „Härte“ der Zeitvorgabe für das Projekt | 1.00 | 1.23 |

Weitere Informationen zur Aufwandsschätzung

Ringvorlesung „Softwareentwicklung in der industriellen Praxis“ WS 2014/15

→ Harry Sneed zu „**Software-Projektaufwandsschätzung**“ am 15.12.2014

Produktivitätseinheiten für Software

- Anweisungen
- Data-Points
- Function Points
- UseCase Points
- Object Points

Referenzen

- 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
- G. Antoniol, R. Fiutem, C. Lokan. Object-Oriented Function Points: An Empirical Validation. Empirical Software Engineering, 8, 225–254, 2003. Kluwer Academic Publishers.

Ende