

# Vorlesung "Software-Engineering"

---

Prof. Ralf Möller, TUHH, Arbeitsbereich STS

- Vorige Vorlesung
  - Projektphasen und Vorgehensmodelle
- Heute:
  - Lastenheft
  - Verfahren zur Aufwandsschätzung
  - Kombination von Schätzverfahren

# Produktplanung (1)

---

## ■ **Produktauswahl**

Trendstudien, Marktanalysen, Forschungsergebnisse, Kundenanfragen, Vorentwicklungen, ...

## ■ **Voruntersuchung des Produkts**

- u.U. gezielte Ist-Aufnahme, wenn bereits Vorgängerprodukt vorhanden;  
anschließend Ist-Analyse
- Festlegen der Hauptanforderungen
  - | Festlegen der Hauptfunktionen
  - | Festlegen der Hauptdaten
  - | Festlegen der Hauptleistungen
  - | Festlegen der wichtigsten Aspekte der Benutzungsschnittstelle
  - | Festlegen der wichtigsten Qualitätsmerkmale.

## ■ **Durchführbarkeitsuntersuchung**

- Prüfen der fachlichen Durchführbarkeit (softwaretechnische Realisierbarkeit, Verfügbarkeit von Entwicklungs- und Zielmaschinen, ...)
- Prüfen alternativer Lösungsvorschläge (Beispiel: Kauf und Anpassung von Standardsoftware vs. Individualentwicklung)
- Prüfen der personellen Durchführbarkeit:  
Verfügbarkeit qualifizierter Fachkräfte für die Entwicklung
- Prüfen der Risiken

## ■ **Prüfen der ökonomischen Durchführbarkeit**

- Aufwands- und Termschätzung
- Wirtschaftlichkeitsrechnung

# Problemanalyse und Planung

---

- Analyse des Ist-Zustandes (Aufgabenbereiche)
- Systemabgrenzung
  - Festlegung, welche Teile zum System gehören und damit Gegenstand der weiteren Untersuchung sind
  - Ermittlung der Umgebungsbedingungen des Systems (Schnittstellen)
- Systemerhebung
  - Sammeln und Strukturieren von Informationen über das System und seine Eigenschaften (insbes. Anforderungen u. Änderungswünsche)

# Lastenheft: Produktanforderungen

---

## Pflichtenheft: Systemanforderungen

- **Aufgabe:** Zusammenfassung aller fachlichen Basisanforderungen aus Sicht des Auftraggebers
- **Adressat:** Auftraggeber sowie Auftragnehmer (Projektleiter, Marketing, ...)
- **Inhalt:** Basisanforderungen („*Was?*“, nicht „*Wie?*“)
- **Form:** standardisiertes, nummeriertes Gliederungsschema (s. Beispiel)
- **Sprache:** verbale Beschreibung
- **Umfang:** wenige Seiten

# Beispiel für ein Lastenheft: Seminarorganisation (1)

Version	Autor	QS	Datum	Status	Kommentar
2.1	Schmidt	Hupe	2/03	akzeptiert	
2.2	Schmidt	Hupe	3/03	akzeptiert	/LF40/ gelöscht

Versions-  
historie

## ■ 1 Zielbestimmung

- Die Firma *Teachware* soll durch das Produkt in die Lage versetzt werden, die von ihr veranstalteten Seminare rechnerunterstützt zu verwalten.

*informell*

## ■ 2 Produkteinsatz

- Das Produkt dient zur Kunden- und Seminarverwaltung der Firma *Teachware*. Außerdem sollen verschiedene Anfragen beantwortet werden können.
- Zielgruppe: die Mitarbeiter der Firma *Teachware*.

*informell*

# Beispiel für ein Lastenheft: Seminarorganisation (2)

## ■ Produktfunktionen

### ■ /LF10/

Ersterfassung, Änderung und Löschung von Kunden (Teilnehmer, Interessenten)

### ■ /LF20/

Benachrichtigung der Kunden (Anmeldebestätigung, Abmeldebestätigung, Änderungsmitteilungen, Rechnung, Werbung)

### ■ /LF30/

Ersterfassung, Änderung und Löschung von Seminarveranstaltungen und Seminartypen

...

### ■ /LF70/

Erstellung verschiedener Listen  
(Teilnehmerliste, Umsatzliste, Teilnehmerbescheinigungen)

### ■ /LF80/

Anfragen der folgenden Art sollen möglich sein: Wann findet das nächste Seminar X statt? Welche Mitarbeiter der Firma Y haben das Seminar X besucht?

*Label /LF.../ zur  
Referenzierung  
von **Funktionen***

# Beispiel für ein Lastenheft: Seminarorganisation (3)

*Label /LD.../ zur Referenzierung von Daten*

## ■ 4 Produktdaten

### ■ /LD10/

Es sind relevante Daten über die Kunden zu speichern.

### ■ /LD20/

Falls ein Kunde zu einer Firma gehört, dann sind relevante Daten über die Firma zu speichern.

### ■ /LD30/

Es sind relevante Daten über Seminarveranstaltungen, Seminartypen und Dozenten zu speichern.

### ■ /LD40/

Bucht ein Kunde eine Seminarveranstaltung, dann sind entsprechende Buchungsdaten zu speichern.

# Beispiel für ein Lastenheft: Seminarorganisation (4)

---

## ■ 5 Produktleistungen (nicht-funktionale Anforderungen)

### ■ /LL10/

Die Funktion /LF80/ darf nicht länger als 15 Sekunden Interaktionszeit benötigen, alle anderen Reaktionszeiten müssen unter 2 Sekunden liegen.

### ■ /LL20/

Es müssen maximal 50.000 Teilnehmer und maximal 10.000 Seminare verwaltet werden können.

*Label /LL.../ zur  
Referenzierung  
von **Leistungen***

# Beispiel für ein Lastenheft: Seminarorganisation (5)

## ■ 6 Qualitätsanforderungen

Produktqualität	sehr gut	gut	normal	irrelevant
Funktionalität		x		
Zuverlässigkeit			x	
Benutzbarkeit		x		
Effizienz		x		
Änderbarkeit			x	
Übertragbarkeit			x	

## ■ 7 Ergänzungen

[keine]

# Methoden zur Kosten- und Termschätzung

---

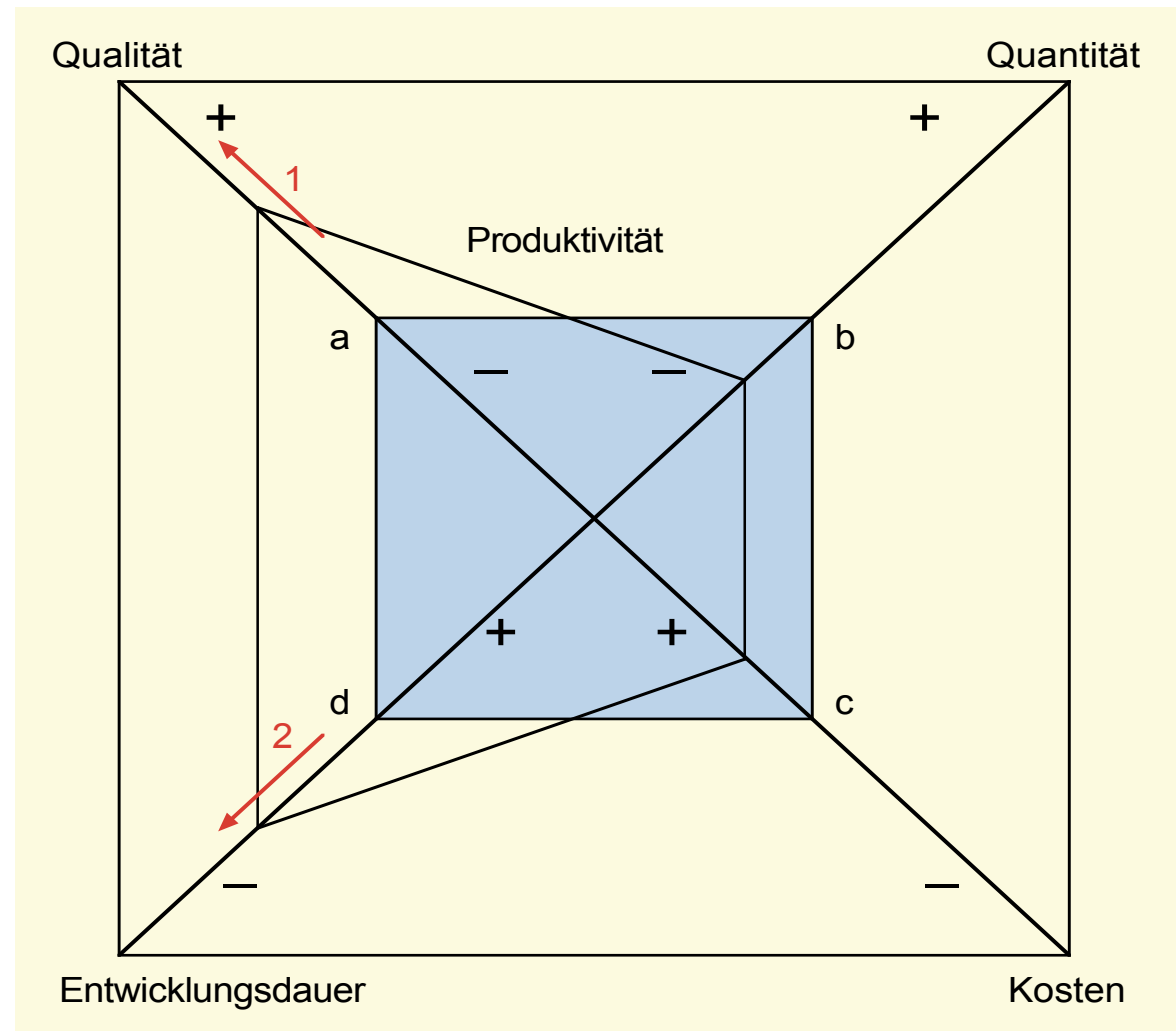
- Die meisten Modelle basieren auf dem geschätzten Umfang des zu erstellenden Software-Produktes in „**Anzahl der Programmzeilen**“ bzw. in ***Lines of Code (LOC)***.
  - Bei höheren Sprachen werden z.B. alle Vereinbarungs- und Anweisungszeilen geschätzt.
  - Der geschätzte Umfang wird durch einen Erfahrungswert für die Programmierproduktivität (in LOC) eines Mitarbeiters pro Jahr oder Monat geteilt.
  - Ergebnis: geschätzter Aufwand in **Personenjahren (PJ, auch MJ)** oder **Personenmonaten (PM, auch MM)**
  - 1 PJ = 9 PM oder 10 PM (Urlaub, Krankheit, Schulung, ...)
  - Der so ermittelte Aufwand wird durch die nach der Terminvorgabe zur Verfügung stehende Entwicklungszeit geteilt.
  - Ergebnis: Anzahl der einzusetzenden, parallel arbeitenden Mitarbeiter.

# Einflussfaktoren der Aufwandsschätzung (1)

- Quantität
- Qualität
- Entwicklungsdauer
- Kosten

bedingen einander

➔ **Teufelsquadrat**



# Einflussfaktoren der Aufwandsschätzung (2)

## ■ Quantität

### ■ Größe des Programmtextes

in Planungsphase unbekannt

- | Maß „Anzahl Programmzeilen“ (LOC)
- | lineare oder überproportionale Beziehung zwischen LOC und dem Aufwand

### ■ Funktions- und Datenumfang

- | Maß unabhängig von einer Programmiersprache

früh bekannt

### ■ evtl. zusätzliche Gewichtung mit **Komplexität**

- | qualitative Maße, z.B. „leicht“, „mittel“ und „schwer“
- | Abbildung auf Zahlenreihe. Beispiel: Noten zwischen 1 und 6.

## ■ Qualität

- | Je höher die Qualitätsanforderungen, desto größer ist der Aufwand.
- | Es gibt nicht **die** Qualität, sondern es gibt verschiedene Qualitätsmerkmale.
- | Jedem Qualitätsmerkmal lassen sich Kennzahlen zuordnen.

# Methoden zur Kosten- und Termschätzung

## ■ Beispiel:

- Es soll ein Software-Produkt mit geschätzten 21.000 LOC realisiert werden
- Durchschnittliche Produktivität pro Mitarbeiter: 3.500 LOC/Jahr [HP, Grady 92]
- ➔ 6 Mitarbeiterjahre werden benötigt
- ➔ Arbeiten 3 Mitarbeiter im Team zusammen, so werden 2 Jahre bis zur Fertigstellung benötigt.

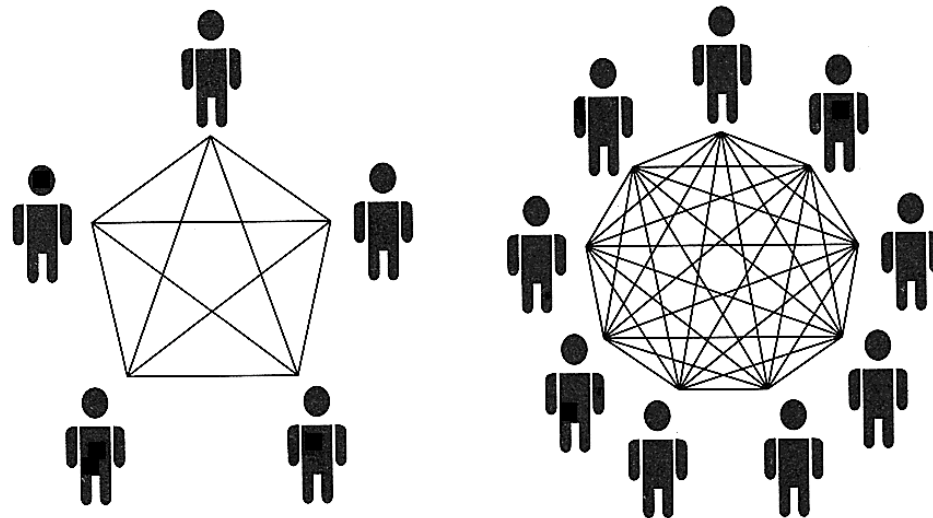
## ■ **Faustregeln**

- Eine durchschnittliche Software-Entwicklung liefert ungefähr 350 Quellcodezeilen (ohne Kommentare) pro Personenmonat.
- Dabei umfasst die Personen-Zeit alle Phasen von der Definition bis zur Implementierung.

# Einflussfaktoren der Aufwandsschätzung (3)

## ■ Zusätzlicher Faktor: **Produktivität**

- Wird von vielen verschiedenen Faktoren beeinflusst
- Die **Lernfähigkeit** und **Motivation** der Mitarbeiter ist entscheidend.

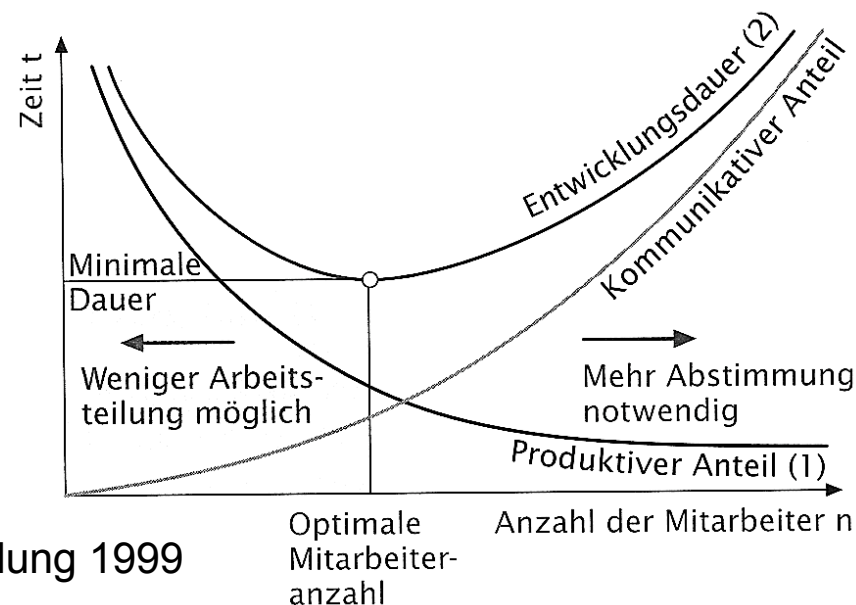


■ **Brooksches Gesetz:** „*Adding manpower to a late software project makes it later*“

# Einflussfaktoren der Aufwandsschätzung (4)

## ■ Entwicklungsdauer

- Soll die Zeit verkürzt werden, dann werden mehr Mitarbeiter benötigt.
- Mehr Mitarbeiter erhöhen den Kommunikationsaufwand im Entwicklungsteam.
- Der höhere Kommunikationsanteil (z.B. durch Unterweisung in der Einarbeitungszeit) reduziert die Produktivität.
- Kann die Entwicklungsdauer verlängert werden, so werden weniger Mitarbeiter benötigt.



Graphik: Steinweg, C.  
Projektkompass Softwareentwicklung 1999

# Aufwandsschätzung: Analogiemethode <sup>(1)</sup>

---

- Bei hoher Übereinstimmung kann der bekannte Aufwand unverändert übernommen werden
- **Faustregel:**
  - Software-Entwicklung mit weitgehender **Übernahme von existierender Software** benötigen ungefähr **1/4** der Zeit von Neuentwicklungen
- **Nachteile der Analogiemethode**
  - intuitive, sehr globale Schätzung auf der Basis individueller Erfahrungen
  - keine Nachvollziehbarkeit der Schätzergebnisse

# Analogiemethode (2)

---

## ■ Beispiel: Analogiemethode

- abgeschlossenes Produkt: Pascal-Compiler: 20 MM
- zu entwickelndes Produkt: Modula-2-Compiler?
  - 20% neue Konstrukte
  - 50% des Codes wiederverwendbar
  - 50% müssen überarbeitet werden
- Schätzung
  - 50% leicht modifiziert:  $1/4$  von 10 MM = 2,5 MM
  - 50% völlige Überarbeitung: 10 MM
  - 20% zusätzliche Neuentwicklung hoher Komplexität:  
 $4 \text{ MM} * 1,5$  (Komplexitätszuschlag) = 6 MM
  - Schätzung Modula-2: 18,5 MM.

# Aufwandsschätzung: Relationsmethode (1)

- Das zu schätzende Produkt wird direkt mit ähnlichen Entwicklungen verglichen.
- Aufwandsanpassung erfolgt im Rahmen einer formalisierten Vorgehensweise.
- Für die Aufwandsanpassung stehen Faktorenlisten und Richtlinien zur Verfügung.
- Beispiel:

## Programmiersprache

PL/1 = 100

COBOL = 120

ASSEMBLER = 140

## Programmiererfahrung

5 Jahre = 80

3 Jahre = 100

1 Jahr = 140

## Dateiorganisation

sequentiell = 80

indexsequentiell = 120

- Werte geben an, in welcher Richtung und wie stark die einzelnen Faktoren den Aufwand beeinflussen.

# Relationismethode (2)

---

## ■ Beispiel (Fortsetzung):

### ■ Ein neues Produkt soll in PL/1 realisiert werden

- | Das Entwicklungsteam hat im Durchschnitt 3 Jahre Programmiererfahrung
- | Es ist eine indexsequentielle Dateiorganisation zu verwenden.

### ■ Zum Vergleich: Entwicklung...

- | die im Assembler programmiert wurde
- | eine sequentielle Dateiorganisation verwendete
- | von einem Team mit 5 Jahren Programmiererfahrung erstellt wurde

### ■ Geht man davon aus, dass alle 3 Faktoren den Aufwand gleichgewichtig beeinflussen, dann ergibt sich folgende Kalkulation:

- | Assembler zu PL/1:  $140$  zu  $100 = 40$  Punkte Einsparung
- | 5 Jahre zu 3 Jahre:  $80$  zu  $100 = 20$  Punkte Mehraufwand
- | sequentiell zu indexsequentiell:  $80$  zu  $120 = 40$  Punkte Mehraufwand

### ■ Es ergibt sich ein Mehraufwand von 20 Punkten.

# Aufwandsschätzung: Multiplikatormethode (1)

---

- Das zu entwickelnde System wird soweit in Teilprodukte zerlegt, bis jedem Teilprodukt ein bereits feststehender Aufwand zugeordnet werden kann (z.B. in LOC).
- Der Aufwand pro Teilprodukt wird meist durch Analyse vorhandener Produkte ermittelt.
- Oft werden auch die Teilprodukte bestimmten Kategorien zugeordnet wie
  - Steuerprogramme
  - E/A-Programme
  - Datenverwaltungsroutinen
  - Algorithmen usw.
- Die Anzahl der Teilprodukte, die einer Kategorie zugeordnet sind, wird mit dem Aufwand dieser Kategorie multipliziert.
- Die erhaltenen Werte für eine Kategorie werden dann addiert, um den Gesamtaufwand zu erhalten.
- Auch „Aufwand-pro-Einheit-Methode“ genannt.

# Aufwandsschätzung: Multiplikatormethode (4)

- Beispiel:
- Die Aufteilung eines zu schätzenden Produkts in Teilprodukte hat folgendes ergeben:

■ <b>Kategorie</b>	<b>Teil- produkte</b>	<b>Summe LOC</b>	<b>Aufwands- faktor</b>	<b>LOC bewertet</b>
■ Steuerprogramm	1*500 LOC	500	1,8	900
■ E/A-Programme	1*700+2*500	1700	1,5	2550
■ Datenverwaltung	1*800+2*250	1300	1,0	1300
■ Algorithmen	1*300+5*100	800	2,0	1600
■ <b>Summe</b>				<b>6350</b>

# Aufwandsschätzung: Multiplikatormethode (5)

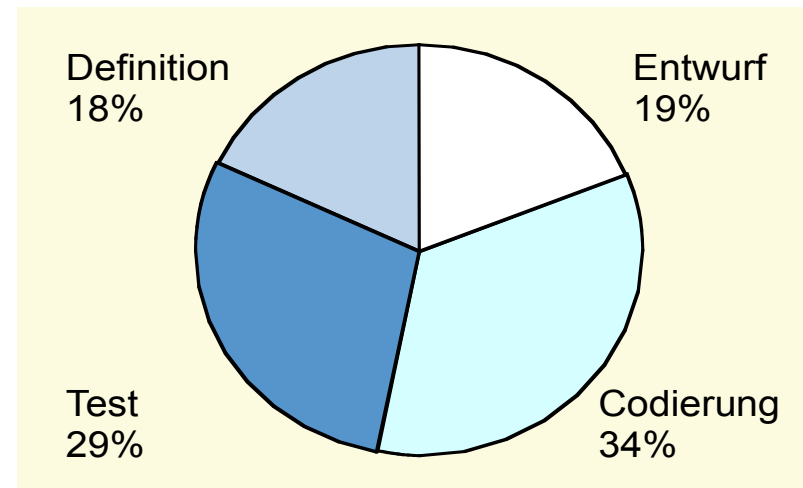
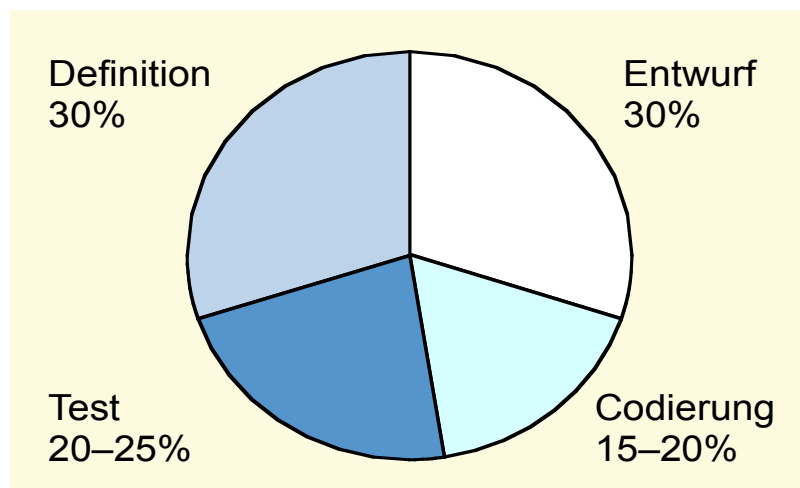
---

## ■ Bewertung

- Es ist eine umfangreiche, empirische Datensammlung und -auswertung erforderlich, um die zu berücksichtigenden Faktoren unternehmensspezifisch zu bewerten.
- Die Koeffizienten müssen permanent überprüft werden, um den technischen Fortschritt zu berücksichtigen.

# Aufwandsschätzung: Prozentsatzmethode

- Aus abgeschlossenen Entwicklungen wird ermittelt, wie der Aufwand sich auf die einzelnen Entwicklungsphasen verteilt hat.
- Bei neuen Entwicklungen schließt man entweder eine Phase zunächst vollständig ab und ermittelt aus dem Ist-Aufwand dann anhand der Aufwandsverteilung den Soll-Aufwand für die restlichen Phasen,
- oder man führt eine detaillierte Schätzung einer Phase durch und schließt hieraus dann auf den Gesamtaufwand.
- Kann bereits frühzeitig eingesetzt werden, wenn der Aufwand für mindestens eine Phase durch den Einsatz einer anderen Methode bestimmt wurde.



# Bewertung

---

- Keine der aufgeführten Basismethoden allein ist ausreichend.
- Je nach Zeitpunkt und Kenntnis von aufwandsrelevanten Daten sollte die eine oder andere Methode eingesetzt werden.
- Für frühzeitige, grobe Schätzungen müssen die
  - Analogie-
  - Relations- und
  - Prozentsatzmethode eingesetzt werden.
- Sind die Einflussfaktoren während der Entwicklung dann genauer bekannt, dann sollten die genaueren Methoden, wie die
  - Multiplikatormethode Verwendung finden.

# Aufwandsschätzung: COCOMO-Methode (1)

---

- COnstructive COst MOdel von B. Boehm entwickelte Sammlung von Schätzmodellen
- Modell 1 (Basisversion):
- Modell 2 (Zwischenversion):
  - zusätzliche Berücksichtigung von kostenbeeinflussenden Faktoren
- Modell 3 (Detailversion):
  - zusätzliche Berücksichtigung der unterschiedlichen Projektphasen, Verteilung des Aufwandes auf die Projektphasen

# Aufwandsschätzung: COCOMO-Methode (2)

---

- Unterscheidung von 3 Projektkategorien zur Anpassung der Kennzahlen:
  - Einfache Projekte
    - | (vertraute Systemumgebung, große Erfahrung, wenige Schnittstellen)
  - Mittelschwere Projekte
  - Komplexe Projekte
    - | (enge Verzahnung mit Systemumgebung, zahlreiche Schnittstellen, geringe Erfahrung)

# COCOMO-Methode: Basisversion (1)

---

- Berechnung des Aufwandes (LOC -> MM):

$$A = C * KLOC^B$$

mit

A: Entwicklungsaufwand in MM

B,C: Konstanten gemäß Tabelle

Projekt	C	B
Einfach	2,4	1,05
Mittel	3,0	1,12
Komplex	3,6	1,20

# COCOMO-Methode: Basisversion (2)

---

- Berechnung der optimalen Entwicklungszeit:

$$T = D * A^E$$

mit

T: Entwicklungszeit

D,E : Konstanten gemäß Tabelle

Projekt	D	E
Einfach	2,5	0,38
Mittel	2,5	0,35
Komplex	2,5	0,32

# Aufwandsabschätzung - Aufgabe

## Aufgabe:

Es soll eine einfache Online-Anwendung erstellt werden.  
Für die abgeschlossene Definitions- und Entwurfsphase wurden  
15 MM benötigt.

<b>Erfahrungswerte:</b>	Definitionsphase	18%
	Entwurfsphase	19%
	Realisierung	53%
	Einführungsphase	10%

- **Fragen:**
- Aufwand der gesamten Entwicklung in MM?
  -
- Restaufwand /-dauer?
  -
- Benötigte Teamgröße ?

# Aufwandsabschätzung - Lösung

## Aufgabe:

Es soll eine einfache Online-Anwendung erstellt werden.

Für die abgeschlossene Definitions- und Entwurfsphase wurden 15 MM benötigt.

<b>Erfahrungswerte:</b>	Definitionsphase	18%
	Entwurfsphase	19%
	Realisierung	53%
	Einführungsphase	10%

## ■ Fragen:

- Aufwand der gesamten Entwicklung in MM?

- $(15 \text{ MM} * 100) / 37 = 40,54 \text{ MM}$

- Restaufwand /-dauer?

- Gesamtaufwand – Aufwand für Planung und Entwurf  
 $(40,54 \text{ MM} - 15 \text{ MM}) = 25,54 \text{ MM}$   
opt. Dauer =  $2,5 * 25,54^{0,35}$  gleich ca. 8 Monate

- Benötigte Teamgröße ?

- $25,54 / 8 = 3,2$  gleich ca. 3 Mitarbeiter

# Aufwandsschätzung: Function Point-Methode (1)

---

- Allen J. Albrecht (Ausarbeitung nach Balzert, 00)
- Ausgangspunkte:
  - Aufwand hängt vom Umfang und vom Schwierigkeitsgrad des neuen Produktes ab
  - Umfang wird nicht in Lines of Code (LOC) ausgedrückt, sondern direkt aus den Anforderungen abgeleitet
- Vorgehen
  - Zuordnung jeder Produktanforderung in eine von fünf Kategorien

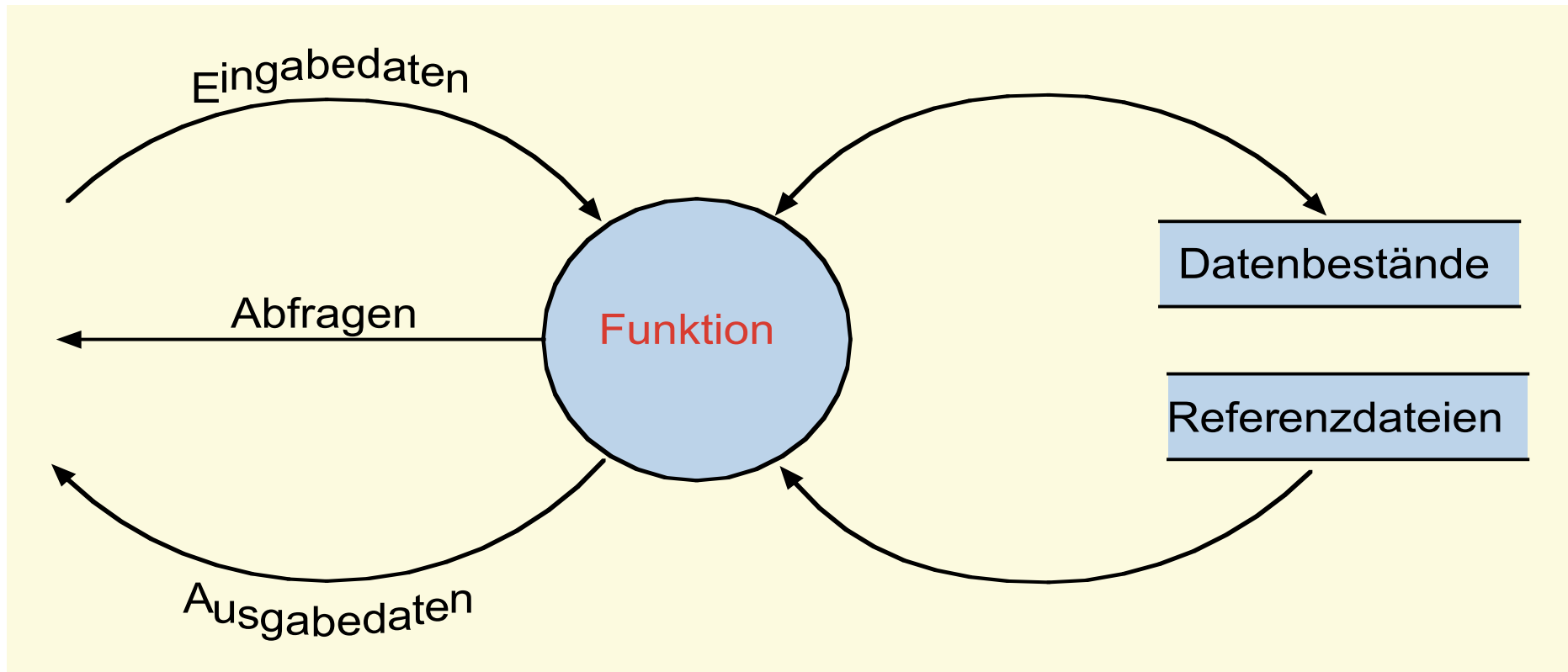
# Die 7 Schritte der *Function-Point* Methode

---

1. Kategorisierung jeder Produkthanforderung  
Eingabedaten, Abfragen, Ausgabedaten, Datenbestände, Referenzdaten
2. Klassifizierung jeder Produkthanforderung  
einfach, mittel, komplex
3. Eintrag in Berechnungsformular
4. Bewertung der Einflussfaktoren
5. Berechnung der bewerteten ***Function Points*** (FP)
6. Ermitteln des Personalaufwands aus einer **FP-PM**  
(Personenmonaten)-Kurve oder Tabelle
7. Aktualisierung der empirischen Daten als Schätzgrundlage  
für Folgeprojekt

# 1. Kategorisierung der Produktanforderungen (1)

## ■ Kategorien



- Ermittelt pro Funktion im Lastenheft
- Erfordert Identifikation und Bewertung der Einzelfunktionen

# 1. Kategorisierung der Produktanforderungen (2)

---

- Beispiel:

- **/LF 20/:**

„Benachrichtigung der Kunden  
(Anmeldebestätigung, Abmeldebestätigung,  
Änderungsmitteilungen, Rechnung, Werbung)“

- Diese Anforderung ist der Kategorie „Ausgabedaten“ zuzuordnen
- Da es sich um 5 verschiedene Ausgaben handelt, wird im folgenden von 5 Ausgaben ausgegangen.

## 2. Klassifizierung der Produktanforderungen

- Einordnung der Anforderungen in eine der Klassen „einfach“, „mittel“ oder „komplex“

Hauptschwierigkeit!

- Beispiel: Klassifizierung der Datenbestände einer Funktion

■ Kriterium	einfach	mittel	komplex
Anzahl unterschiedl. Datenelemente	1-5	6-10	>10
Eingabeprüfung	formal logisch	formal logisch	formal DB-Zugriff
Ansprüche an die Bedienung	gering	normal	hoch

# 3. Eintrag in Berechnungsformular

Kategorie	Anzahl	Klassifizierung	Gewichtung	Zeilensumme
Eingabedaten		einfach	x 3	=
		mittel	x 4	=
		komplex	x 6	=
Abfragen		einfach	x 3	=
		mittel	x 4	=
		komplex	x 6	=
Ausgaben		einfach	x 4	=
		mittel	x 5	=
		komplex	x 7	=
Datenbestände		einfach	x 7	=
		mittel	x 10	=
		komplex	x 15	=
Referenzdaten		einfach	x 5	=
		mittel	x 7	=
		komplex	x 10	=
Summe			E1	=
Einflußfaktoren (ändern den Function Point-Wert um ± 30%)	1 Verflechtung mit anderen Anwendungssystemen (0–5)			=
	2 Dezentrale Daten, dezentrale Verarbeitung (0–5)			=
	3 Transaktionsrate (0–5)			=
	4 Verarbeitungslogik			=
	a Rechenoperationen (0–10)			=
	b Kontrollverfahren (0–5)			=
	c Ausnahmeregelungen (0–10)			=
d Logik (0–5)			=	
5 Wiederverwendbarkeit (0–5)			=	
6 Datenbestands- Konvertierungen (0–5)			=	
7 Anpaßbarkeit (0–5) =			=	
Summe der 7 Einflüsse		E2		=
Faktor Einflußbewertung = E2 / 100 + 0,7	E3	=		
Bewertete Function Points: E1 * E3				=

IBM-  
Werte

3. Schritt

4. Schritt  
Berechnung  
der  
Gewichtung

$$E3 = E2 / 100 + 0.7$$

# 4. Bewertung der Einflussfaktoren

- Die Einflussfaktoren beziehen sich auf die Anwendung als Ganzes und nicht auf einzelne Funktionen oder Funktionspunkte.
- Es wird nicht nur das bloße Vorliegen eines Zustandes bewertet, sondern sein Einfluss auf die Entwicklung.

## ■ Alternative Ansätze für Bewertung der Einflussfaktoren:

Ansatz	Anzahl der Faktoren	Bewertungs- spanne	Faktor Einflussbewertung
■ nach Albrecht	14 Faktoren	(0 ... 5)	(0 ... 70) / 1,64
■ nach IBM	7 Faktoren	(0 ... 5 / 0 ... 10)	(0 ... 60) / 100 + 0,7
■ nach IFPUG	14 Faktoren	(0 ... 5)	(0 ... 70) / 100 + 0,65
■ nach Hürten	7 Faktoren	(-2,5% ... 2,5%) (-5% ... 5%)	-30% ... 30% (*)

(\*) Auswirkung der Einflussfaktoren in Prozent der *Function Points*

## 5. Berechnung der bewerteten *Function Points*

---

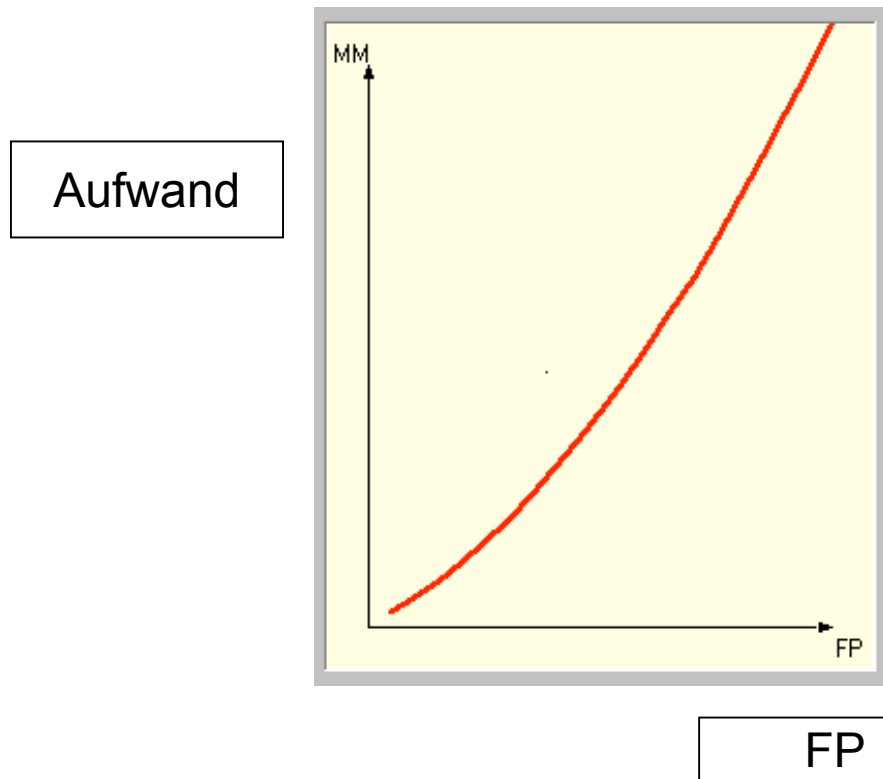
- E1 = Summe der Kategorien
- E2 = Summe der 7 Einflüsse

$$\text{Bewertete } \textit{Function Points} = E1 * ( E2 / 100 + 0,7 )$$

- Die Summe der Einflussfaktoren (ein Wert zwischen 0 und 60) ändert den *Function Point*-Wert um +/- 30 %

## 6. Ablesen des Aufwands in MM

- Produktivität nimmt bei großen Projekten ab → nicht-linearer Zusammenhang



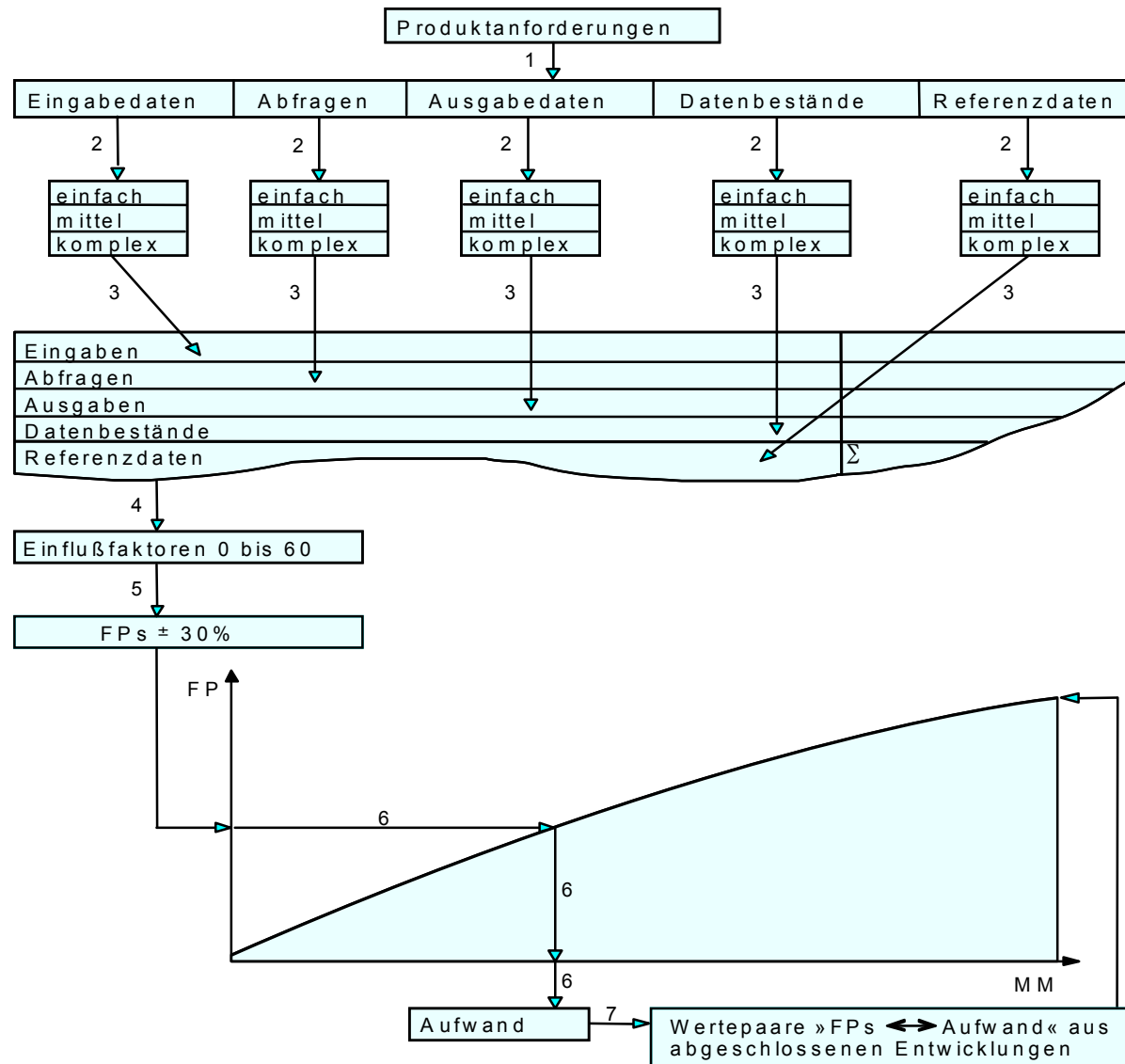
Wertepaare (FP, PM) aus abgeschlossenen Entwicklungen des eigenen Unternehmens

## 7. Aktualisierung der empirischen Daten

---

- Nach der Beendigung einer mit der *Function Point*-Methode geschätzten Entwicklung ist das neue Wertepaar **FP** → tatsächliche **PM** zu verwenden, um die bestehende Kurve zu aktualisieren.

# Zusammenfassung



1. Schritt:  
Kategorisierung für  
jede Anforderung

2. Schritt:  
Klassifizierung jeder  
Anforderung

3. Schritt: Eintrag  
der jeweiligen  
Anzahl in das  
Berechnungs-  
formular und  
Ermittlung der  
unbewerteten FPs

4. Schritt:  
Bewertung der  
Einflußfaktoren

5. Schritt:  
Berechnung der  
bewerteten FPs

6. Schritt: Ablesen  
des Aufwandes

7. Schritt:  
Aktualisierung der  
Wertepaare, Neu-  
berechnung der  
Aufwandskurve

# Die *Function Point*-Methode: Voraussetzungen

---

- Eine Bewertung kann erst dann durchgeführt werden, wenn die Projektanforderungen bekannt sind.
- Eine Bewertung sollte von Mitarbeitern durchgeführt werden, die ein ausreichendes Wissen über die Anforderungen haben.
- Bei der Bewertung muss die gesamte Anwendung / Projektanforderung betrachtet werden.
- Die Projektanforderungen sollten nicht besonders minutiös zergliedert werden, sondern die gesamte Anwendung muss im Blickfeld stehen.
- Beim Einsatz dieser Methode ist sehr streng darauf zu achten, dass das Anwendungsprojekt aus Sicht des Benutzers betrachtet wird.
- Unternehmensspezifische Schulung und Vorgabe von Beispielen, damit die Wirkung individueller Schätzungen (Klassifizierung und Bewertung der Einflussfaktoren) minimiert werden.
- Der Ist-Aufwand muss für die Nachkalkulation ermittelbar sein.

## Die *Function Point*-Methode: Vorteile <sup>(1)</sup>

---

- Produktanforderungen, nicht LOC als Ausgangspunkt
- Anpassbarkeit an verschiedene Anwendungsbereiche (Änderung der Kategorien)
- Anpassbarkeit an neue Techniken (Änderung der Einflussfaktoren und der Einflussbewertung)
- Anpassbarkeit an unternehmensspezifische Verhältnisse (Änderung der Einflussfaktoren, der Einflussbewertung und der Klassenfaktoren pro Klasse)

## Die *Function Point*-Methode: Vorteile (2)

---

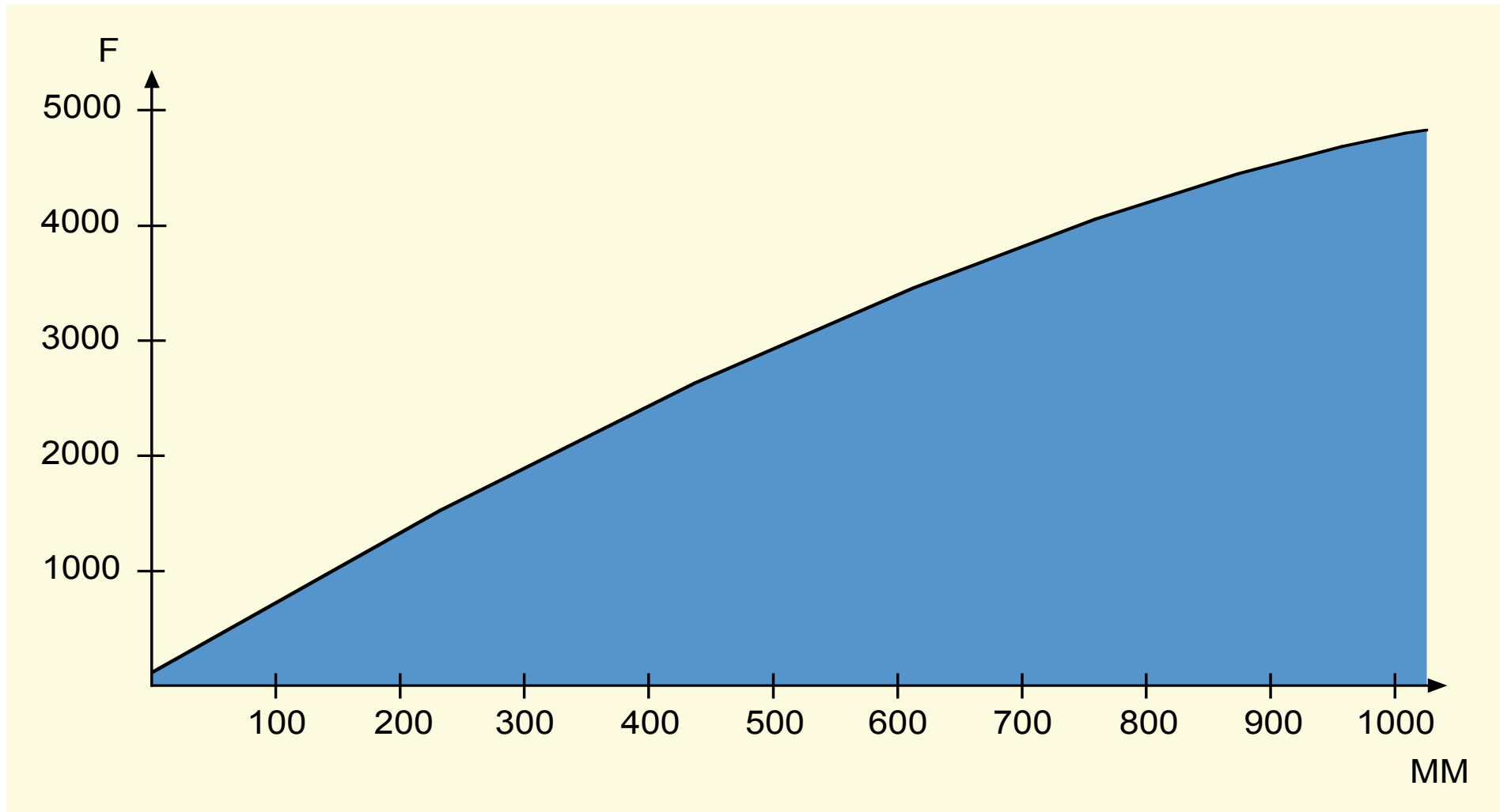
- Verfeinerung der Schätzung entsprechend dem Entwicklungsfortschritt (iterative Methode)
  - Beispiel
    1. Schätzung auf der Grundlage des Lastenheftes
    2. Schätzung auf der Grundlage des Pflichtenheftes
    3. Schätzung nach Erstellung des formalen Modells
- Erste Schätzung bereits zu einem sehr frühen Zeitpunkt möglich (Planungsphase)
- Festgelegte methodische Schritte
- Leicht erlernbar
- Benötigt nur einen geringen Zeitaufwand
- Gute Transparenz
- Gute Schätzgenauigkeit
- Werkzeugunterstützungen verfügbar

## Die *Function Point*-Methode: Nachteile

---

- Es kann nur der Gesamtaufwand geschätzt werden. Eine Umrechnung auf einzelne Phasen muss mit der Prozentsatzmethode erfolgen.
- In der Originalform von Albrecht personalintensiv und nicht automatisierbar
- Zu stark funktionsbezogen
- Qualitätsanforderungen werden nicht berücksichtigt
- Mischung von Projekt- und Produkteigenschaften bei den Einflussfaktoren.

# Anhang: Die IBM-Kurve: FP -> MM

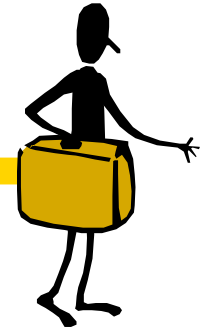


# Kernphasen der Software-Entwicklung

---

- Problemanalyse und Planung
- Systemspezifikation
- Entwurf
- Implementierung
- Integration und Test
- Wartung

# Zusammenfassung, Kernpunkte



- Techniken zur Aufwandsabschätzung
- Einfache Methoden
- Kombination von einfachen Methoden
  - COCOMO
  - Function Point-Methode

# Was kommt beim nächsten Mal?



- Fortsetzung der Diskussion der Phasen
- Systemspezifikation