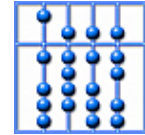


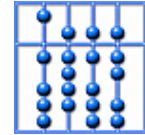
Vorlesung Projektmanagement und Teamorganisation

Dr. Bernhard Schätz
Leopold-Franzens Universität Innsbruck
Sommersemester 2003



Übersicht

1. Übersicht
2. Projektmanagement und Software-Engineering
3. Projektstrukturen und Personalaktivitäten
4. Projektvorbereitung und Projektplanung
 1. Projektinitiierung
 2. Schätzverfahren
 3. Planung (Zeit-, Personal- und Ressourcenplanung)
5. Projektkontrolle und Projektsteuerung
6. Projektabschluss und Prozessverbesserung
7. Ausblick: Der Faktor Mensch im Projektmanagement

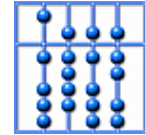


4.2 Schätzverfahren

„Was man nicht misst, das kann man nicht steuern.“

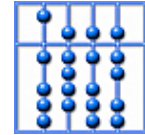
[Tom de Marco, *Controlling Software Projects*, 1982.]

- Messung von Softwaresystemen:
 - Zu Projektbeginn: Projektplanung (Schätzung)
 - Bei Durchführung: Projektstatus/Projektsteuerung

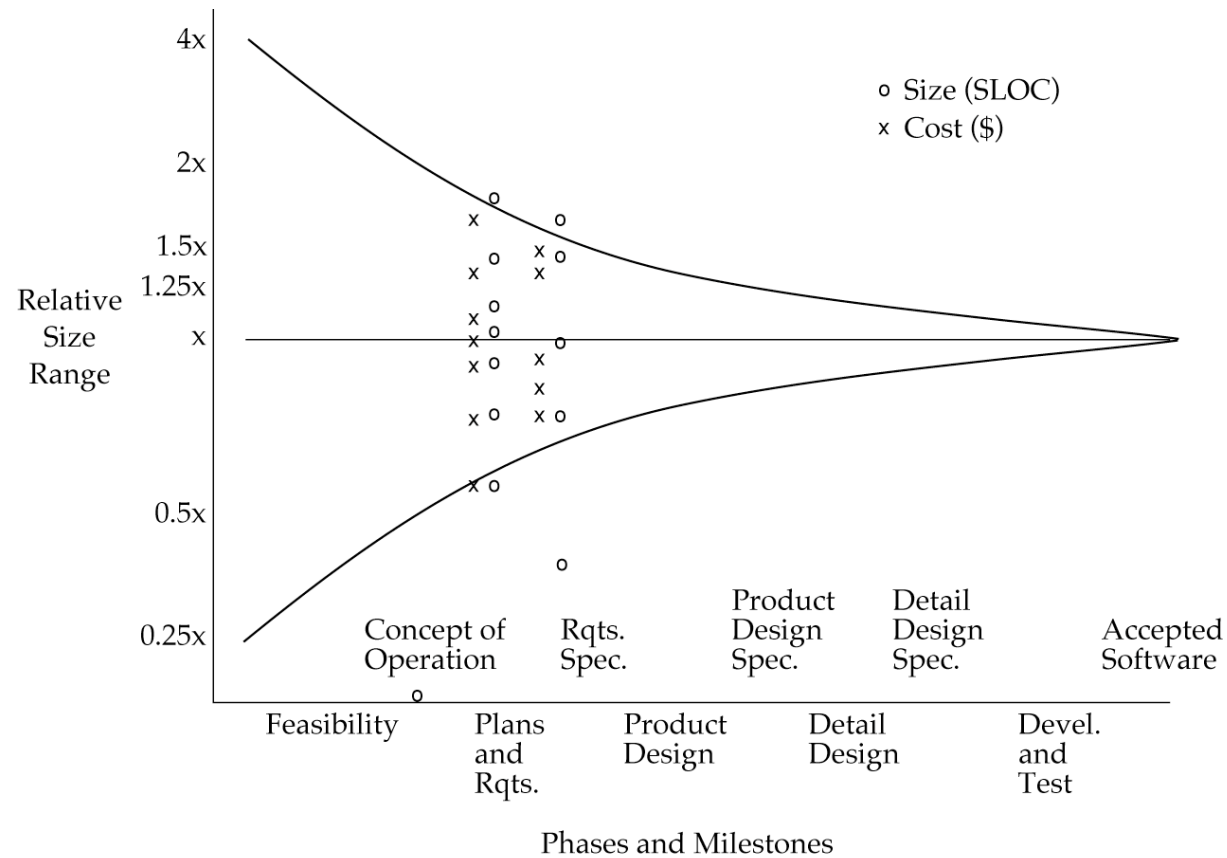


Schätzen von Entwicklungsaufwänden

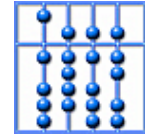
- Ziel des Schätzens:
 - Bestimmung Entwicklungsaufwand
 - Realisierungsaufwand
 - Realisierungszeit
 - Abhängig von
 - Systemkomplexität
 - Produktivität
 - Möglichst vor Systemrealisierung
- Prinzipielle Strategie: per Analogie
- Schätzen:
 - Black-Box/White-Box
 - Mischformen
 - Modellbasiert



Güte der Schätzung

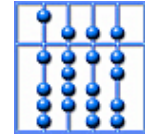


- Genauigkeit Schätzung □ Genauigkeit Produkt/Prozessbeschreibung



Messen Systemkomplexität

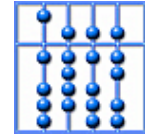
- Komplexitätsfaktoren
 - Größe: Anzahl der Bausteine
 - Diversität: Unterschiedlichkeit der Bausteine
 - Vernetzung: Abhängigkeit der Bausteine
- Grundlagen der Softwareschätzung
 - Messung/Schätzung von Systemkomplexität
 - Komplexität, Aufwand, Laufzeit
 - Modellparameter: Projekt vs. Prozess



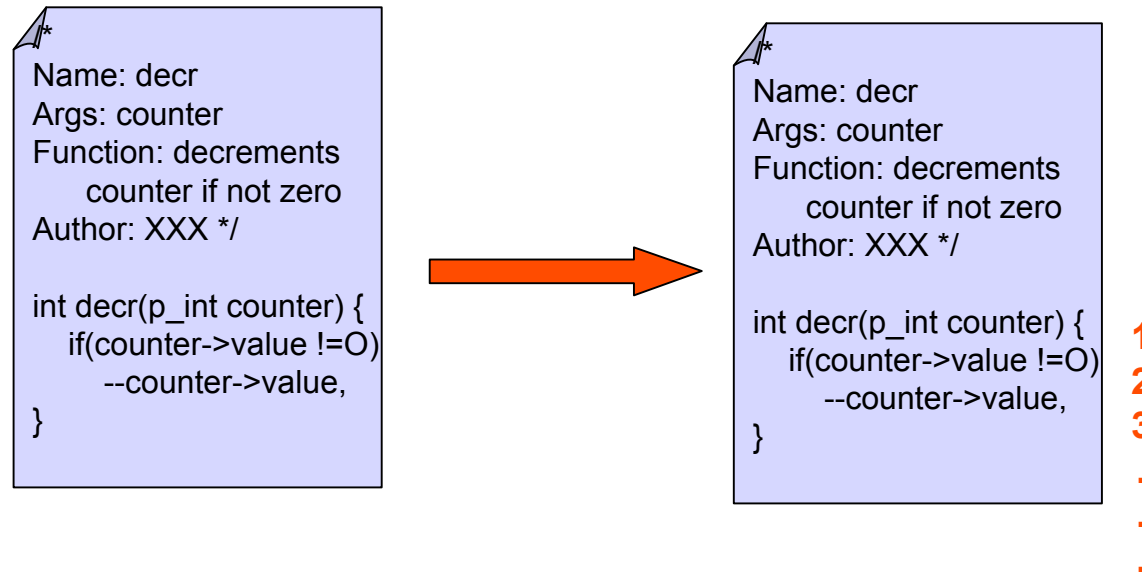
Softwarekomplexität

```
/*  
Name: decr  
Args: counter  
Function: decrements  
         counter if not zero  
Author: XXX */  
  
int decr(p_int counter) {  
    if(counter->value != 0)  
        --counter->value,  
}  
}
```

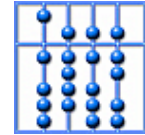
- Softwaremetrik Code
 - Ansatz: Software = Programmcode
 - Einheit: Lines of Code
 - Meßverfahren: Softwareumfang = Anzahl Programmzeilen
- Vorzug: Einfache Messung



Code als Komplexitätsmaß

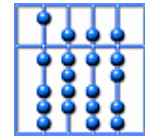


- Verbesserung: Standardisierte Code-Zeilen
 - Größe: Zählregeln
 - Diversität: Sprachebene (Assembler, Prozedural, Objektorientiert)
 - Vernetzung: Nicht berücksichtigt
- Problem:
 - Implementierungsmaßzahl
 - Konsequenz: Erst nach Codierung anwendbar

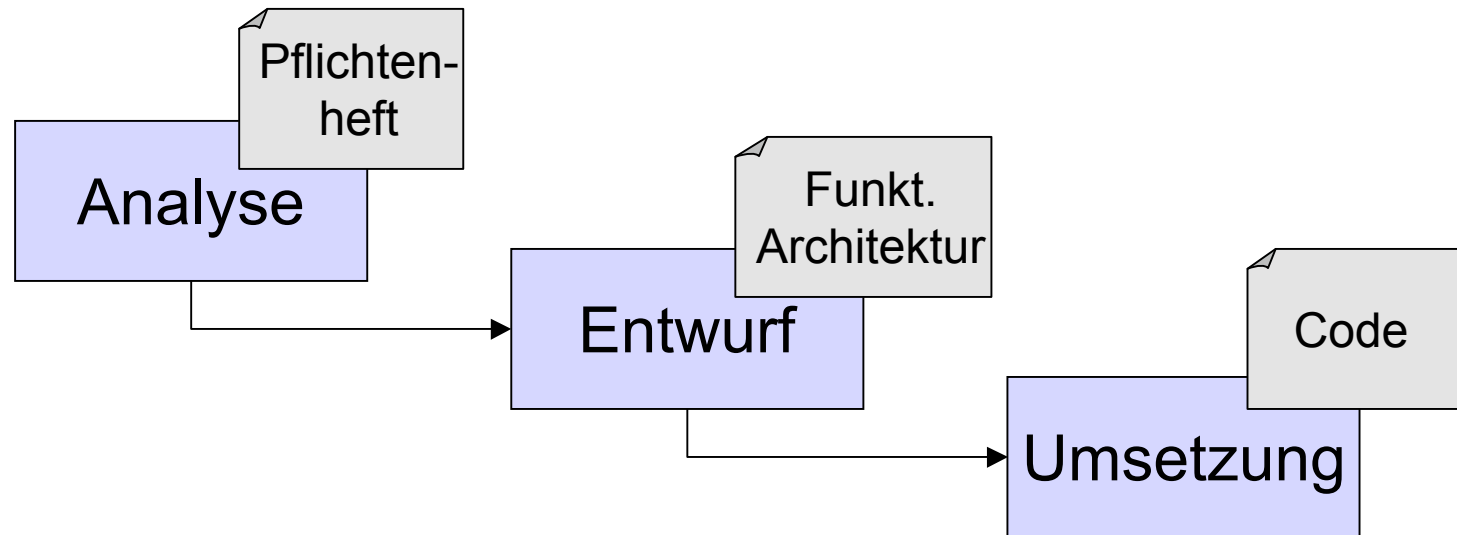


Weitere Metriken

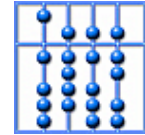
- Komponentenmetriken:
 - I.A. Metriken „algorithmischer“ Komplexität
 - Halstead: Anzahl (verwendeter) Operatoren/Operanden
 - McCabe: Anzahl Knoten/Kanten Kontrollflussgraph
 - OO-Metriken: Anzahl Vererbungsstufen, Attribute, Methoden
- Vorteile:
 - I.A. einfache bzw. automatisierbare Zählverfahren
 - Operieren auf Code/abstrahiertem Code
- Einschränkung:
 - Erst auf Implementierungsebene einsetzbar
 - Fokus auf Implementierungsgüte



Komplexität und Prozess



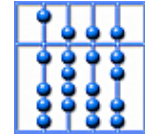
- Verschiedene Maße in verschiedenen Phasen
 - Implementierung: Codezeilen
 - Post-Architektur (Feindesign): Abstrakter Code
 - Prä-Architektur: Funktionspunkte



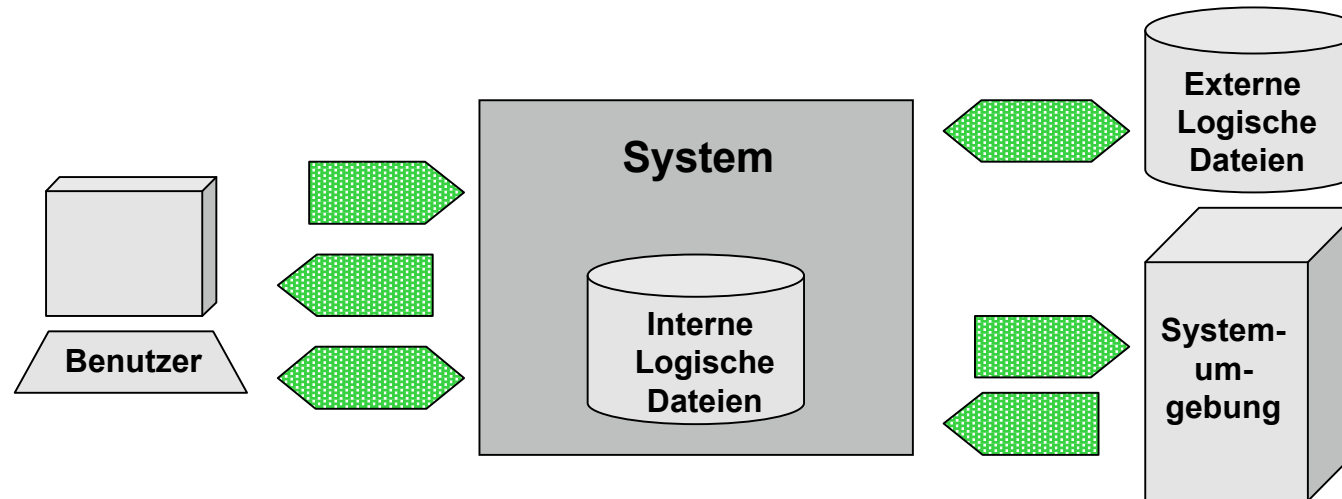
Funktionspunktanalyse

- Ziel: Messung Datenflusskomplexität
 - Größe: Anzahl der Datenflussschnittstellen
 - Diversität: Datenflussarten
 - Vernetzung: Gemeinsame Datenquellen/-senken

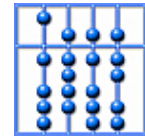
- Verfahren:
 - Analyse aus Sicht der Systemumgebung
 - Messung über funktionale Schnittstellen
 - Externe Eingabe
 - Externe Ausgabe
 - Externe Anfrage
 - Interne Logische Dateien
 - Externe Logische Dateien



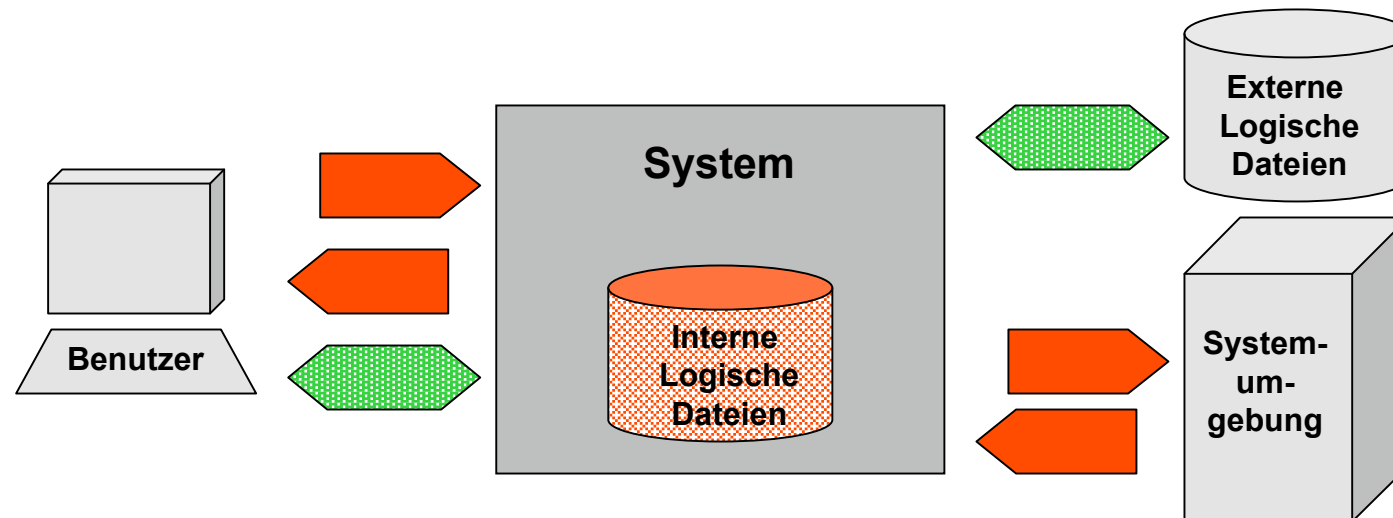
Funktionspunktanalyse



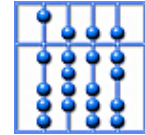
- Analyse Systemkomplexität über Funktionalität:
 - Analyse aus Sicht der Systemumgebung
 - Messung über funktionale Schnittstellen:
 - Externe Eingabe
 - Externe Ausgabe/Externe Anfrage
 - Interne Logische Daten/Externe Logische Daten
- Vorteile:
 - Auf hohem Niveau (Funktionspunkte)
 - Frühzeitig (Systemspezifikation)



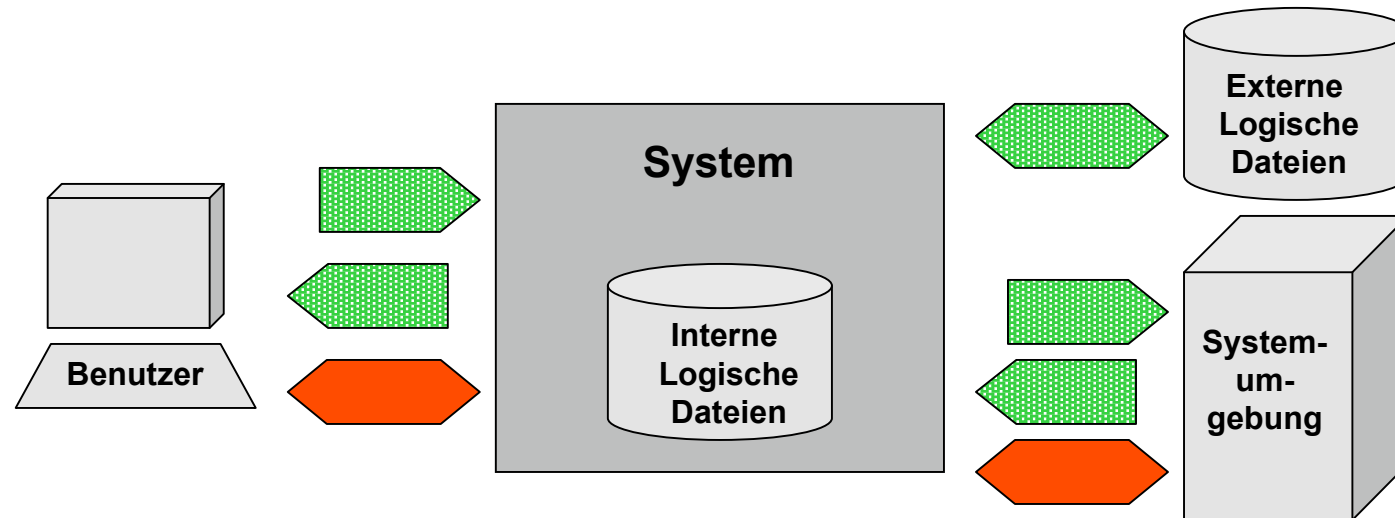
FPA: Externe Ein-/Ausgabe



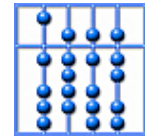
- Eingabe bzw. Ausgabe von Umgebung an System
 - Logisch zusammenhängend
 - Benutzer- oder Kontrolleingabe bzw. -ausgabe
 - Eingabe: Verändert /erweitert internen Datenzustand
 - Ausgabe: Erhält Datenzustand, ist unabhängig von Benutzereingaben



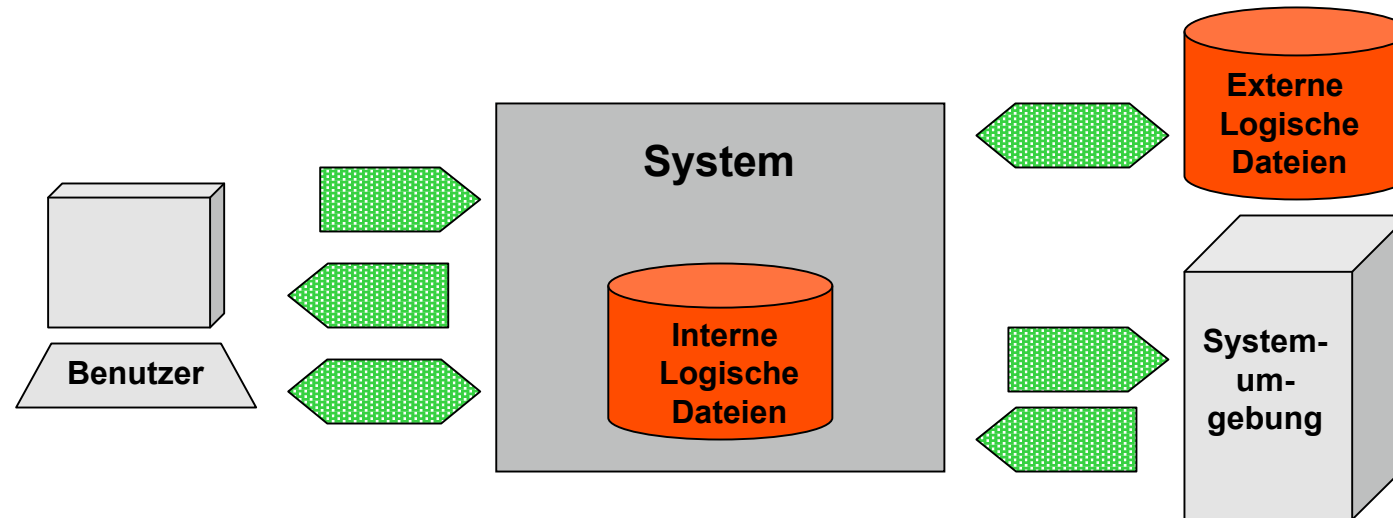
FPA: Externe Anfrage



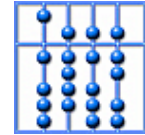
- Externe Anfrage
 - Ein-/Ausgabekombination
 - Logisch zusammenhängend
 - Von Benutzer oder Systemumgebung
 - Eingabe löst sofortige Reaktion aus
 - Ändert Interne Logische Daten nicht



FPA: Interne Logische Dateien



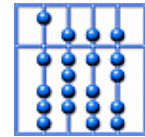
- Gruppe von intern bzw. extern verarbeiteter Daten
 - Logisch zusammenhängend
 - Benutzer- oder Kontrolldaten
 - Interne: Erzeugt, verwaltet oder bearbeitet
 - Externe: Von Außen zur Verfügung gestellt



Berechnung Funktionspunkte

For ILF and EIF				For EO and EQ				For EI			
Record Elements	Data Elements			File Types	Data Elements			File Types	Data Elements		
	1 - 19	20 - 50	51+		1 - 5	6 - 19	20+		1 - 4	5 - 15	16+
1	Low	Low	Avg	0 or 1	Low	Low	Avg	0 or 1	Low	Low	Avg
2 - 5	Low	Avg	High	2 - 3	Low	Avg	High	2 - 3	Low	Avg	High
6+	Avg	High	High	4+	Avg	High	High	3+	Avg	High	High

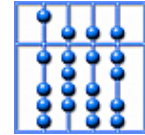
- Berechnung:
 - Ermittlung Liste aller Funktionsarten nach Typ
 - Beispiel: EI: Eingabe Kundendaten mit 5 Feldern
 - Bestimmung Komplexität pro Funktionsart
 - Beispiel: 1 ILF, 5 Elemente = LOW
 - Berechnung der Funktionspunkte pro Funktionsart



Berechnung Funktionspunkte (2)

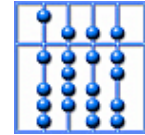
Function Type	Complexity-Weight		
	Low	Average	High
Internal Logical Files	7	10	15
External Interfaces Files	5	7	10
External Inputs	3	4	6
External Outputs	4	5	7
External Inquiries	3	4	6

- Berechnung:
 - Bestimmung Anzahl/Art Funktionspunkte:
 - Z.B. 3 EI Low + 2 EI Average = $3 \cdot 3 + 2 \cdot 4 = 17$
 - Bestimmung Unangepasste Funktionspunkte:
 - Summe aller gewichteten Funktionstypen
 - Umrechnung in KLOC:
 - 1 FP = ca. 320 Ass.-LOC, 130 C-LOC, 90 Pascal-LOC



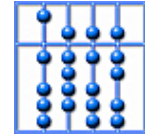
CoCoMo

- Verfahren: Constructive Cost Modeling
- Entwickler: Barry Boehm
- Anwendungsgebiete:
 - Informationssysteme (Honeywell, IBM, US Army)
 - Eingebettete System (AT&T, Boeing, Motorola)
- Ansatz:
 - Eingabe:
 - Messung/Abschätzung der Systemkomplexität
 - Abschätzung von Prozessparametern
 - Ausgabe:
 - Abschätzung des Realisierungsaufwands
 - Abschätzung der minimalen Realisierungszeit



Modellbasiertes Schätzen

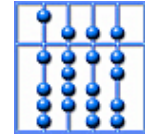
- Schematisiertes Analogieverfahren
- Ansatz: Aus Projektdaten math. Modell
 - Definition geeigneter Modellparameter
 - Projekt (Größe, Komplexität, Randbedingungen)
 - Projektumfeld (Team, Infrastruktur)
 - Bestimmung von Modellgleichungen
 - Berechnung Projektaufwand aus Parameter
 - Berechnung Projektlaufzeit aus Parameter
 - Überprüfung Modell
 - Anwendung
 - Bestimmung Parameter
 - Berechnung Aufwand/Laufzeit



Einfaches CoCoMo-Modell

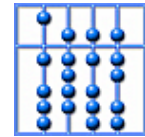
- Eingaben:
 - Ermittlung Systemgröße (KLOC)
 - Systemklasse:
 - Organic: Vertrautes Problem, erfahrenes Team
 - Semi-Detached: Zwischenform
 - Embedded: Schwere Randbedingungen, neues Problem, wenig Erfahrung

- Ausgaben:
 - Realisierungsaufwand (PM)
 - Entwicklungszeit (TDEV)



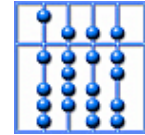
CoCoMo-Gleichungen

- Aufwand:
 - OM: $PM = 2,4 * (KLOC)^{1,05}$
 - SDM: $PM = 3,0 * (KLOC)^{1,12}$
 - EM: $PM = 3,6 * (KLOC)^{1,2}$
- Entwicklungszeit:
 - OM: $TDEV = 2,5 * (PM)^{0,32}$
 - SDM: $TDEV = 2,5 * (PM)^{0,35}$
 - EM: $TDEV = 2,5 * (PM)^{0,38}$
- Rechenbeispiel: 10.000 LOC
 - Semi Detached Model
 - $PM = 3,0 * 10^{1,12} = 39,54$ PM
 - $TDEV = 2,5 * 39,54^{0,35} = 9,05$ CM
 - Embedded Model
 - $PM = 3,5 * 10^{1,2} = 55,44$ PM
 - $TDEV = 2,5 * 55,44^{0,38} = 11,47$ CM



Modellparameter

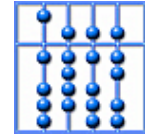
- Gleichung:
 - Aufwand: $PM = A * (KLOC)^B$
 - Entwicklungszeit: $TDEV = C * (PM)^D$
- Parameterarten: $F(P) = L * P^E$
 - **Linear**: Unkritisch (“Balance of Scales”)
 - **Polynomial**: Polynomial: Unkritisch (Eingabeparameter)
 - **Exponential**: Exponentiell: (“Diseconomy of Scales”)
- Rechenbeispiel: Semi Detached Model :
 - 10.000 LOC
 - $PM = 3,0 * 10^{1,12} = 39,54$ PM
 - $TDEV = 2,5 * 39,54^{0,35} = 9,05$ CM
 - 20.000 LOC
 - $PM = 3,0 * 10^{1,12} = 85,95$ PM
 - $TDEV = 2,5 * 85,95^{0,35} = 11,87$ CM



CoCoMo II

- Beobachtung:
 - Schätzung stark abhängig von
 - Eigenschaften zu erstellendes System
 - Eigenschaften erstellendes Unternehmen

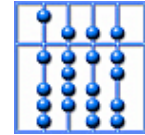
 - Unterschiedliche Parameter:
 - Linear (Frühes Design): Projektorientiert
 - Mensch: Qualifikation Personal
 - Technik: Produktgüte, Legacy, Plattform, Entwicklungsumgebung, Zeitplan
 - Exponentiell: Prozessorientiert
 - Technik: Randbedingungen, Risikomanagement, Prozessreife
 - Mensch: Erfahrung, Kohäsion



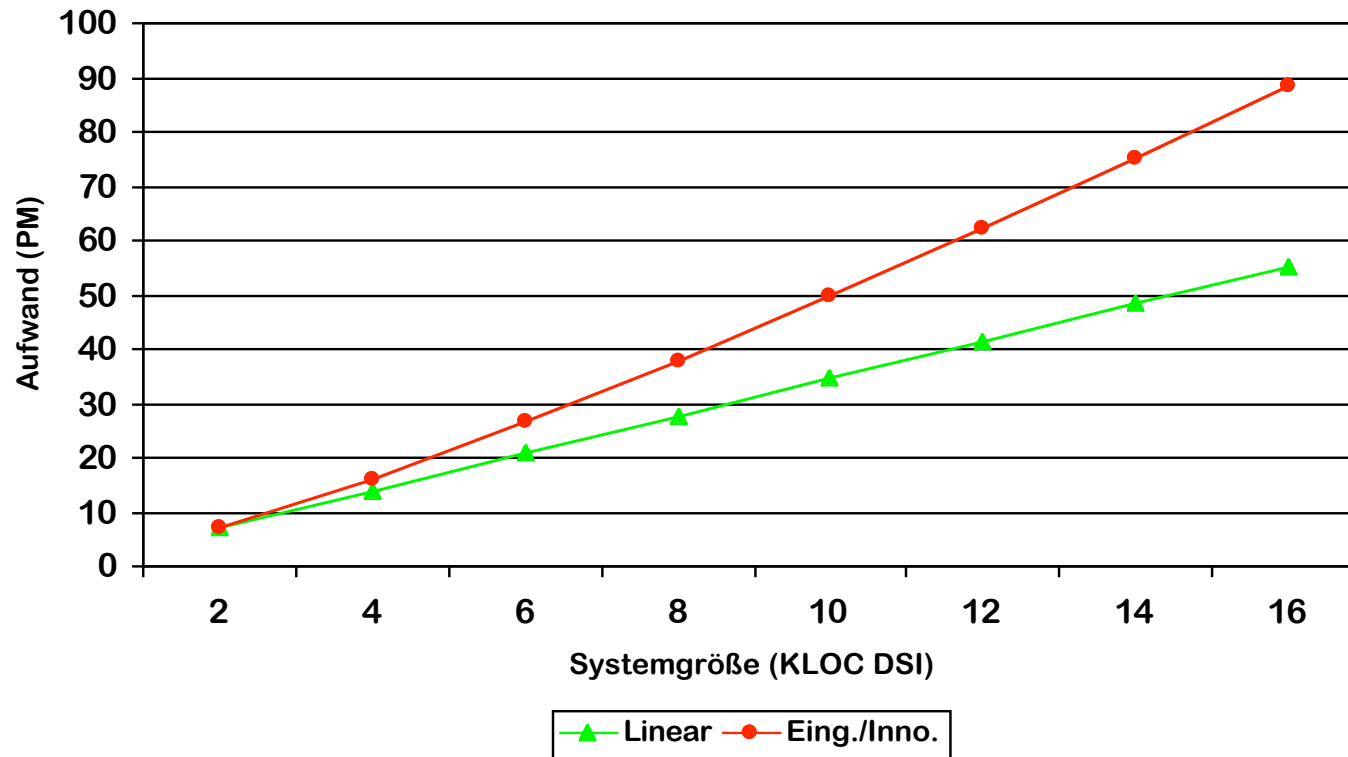
CoCoMo II Modellgleichungen

- Parameter:
 - Aufwandsfaktor EM aus Einzelfaktoren EM_i
 - Skalierfaktor B aus Einzelgewichten W_i
 - Grundfaktor A
 - Zeitplanfaktor SCED%

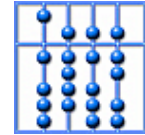
- Gleichungen
 - Skalierfaktor: $B = 0,91 + 0,01 * \text{Summe}(W_i)$
 - Grundaufwand: $PM_{\text{Grund}} = A * \text{Größe}(FPA) ^ B$
 - Aufwandsfaktor: $EM = \text{Produkt}(EM_i)$
 - Gesamtaufwand: $PM_{\text{Gesamt}} = EM * PM_{\text{Grund}}$
 - Laufzeit: $TDEV = [3,67 * PM_{\text{Gesamt}} ^{(0,28+0,2*(B-1,01))}] * \text{SCED\%/100}$



CoCoMo II - Aufwand

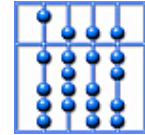


- Generell:
 - Auswirkung des exponentiellen Faktors Prozess
 - Faktor > 1
 - “Diseconomy of Scales”

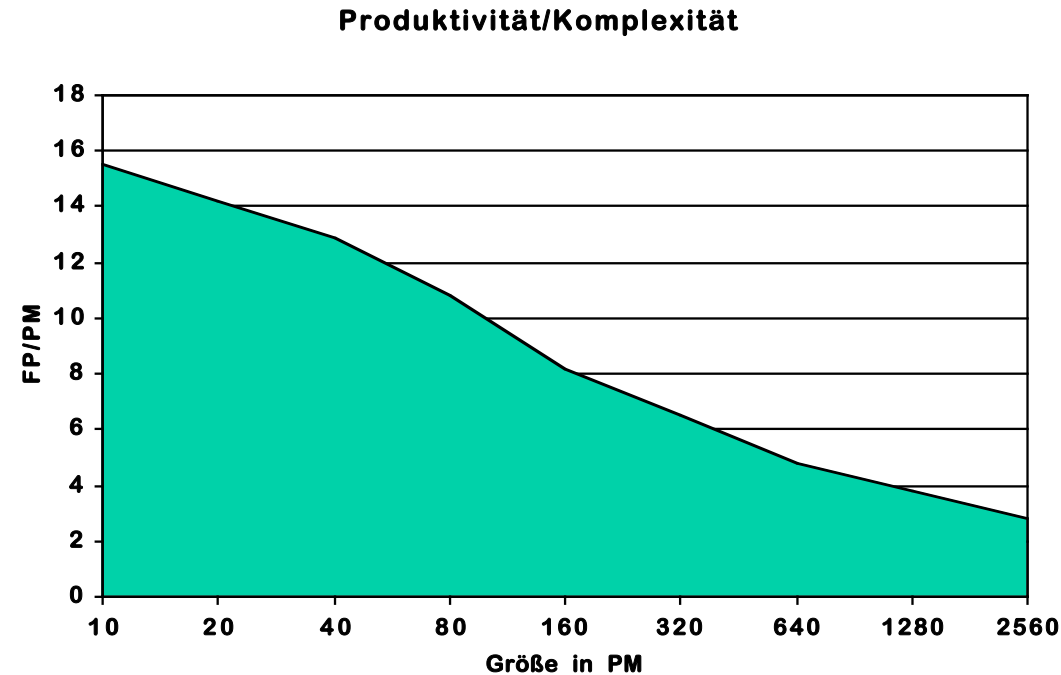


Komplexität und Produktivität

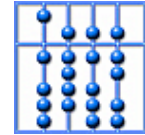
- Ziel Schätzung:
 - Bestimmung Realisierungsaufwand
 - Bestimmung Realisierungszeit
- Produktivität:
 - Bestimmung Mitarbeiterzeit/Systemeinheit
 - Berechnung: „Aufwand = Produktivität * Größe“
 - Beobachtung:
 - Durchschnittl. Produktivität: 5 FP/PM
 - IS-Produktivität: 8 FP/PM
 - ES-Produktivität: 4 FP/PM



Komplexität und Produktivität

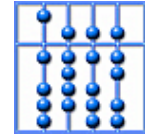


- Produktivität beeinflusst:
 - Durch administrativen Aufwand
 - Durch personellen Aufwand
 - Durch Abstimmungsaufwand

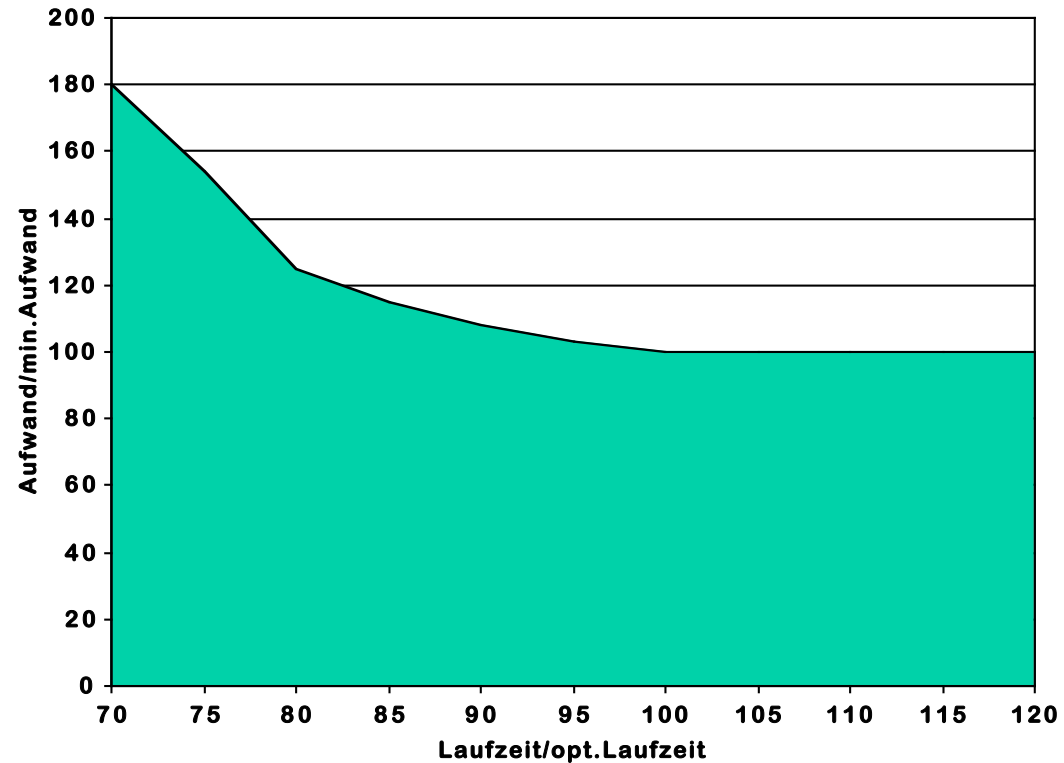


Realisierungsaufwand und Laufzeit

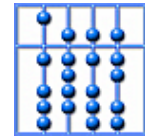
- Ziel Schätzung:
 - Bestimmung Realisierungsaufwand
 - Bestimmung Realisierungszeit
- Intensität:
 - Bestimmung „Projektlaufzeit/Aufwand“
 - Berechnung: „Laufzeit = Intensität * Aufwand“
 - Beobachtung:
 - Laufzeit polynomial in Aufwand
 - Laufzeitvorgaben beeinflussen Aufwand
 - Maximale Laufzeitverkürzungen: 70-75%



Realisierungsaufwand und Laufzeit

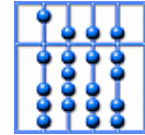


- Laufzeit beeinflusst Aufwand:
 - Durchschnittliche Laufzeit minimiert Aufwand
 - Zusätzlicher Aufwand erlaubt Kürzung (70%)
 - Erhöhte Laufzeit verringert Aufwand nicht



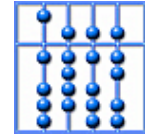
Aufwandsfaktoren

- Projektparameter
 - Parameter wirken linear auf Aufwand
 - Arten von Parameter:
 - Produktfaktoren, z.B.:
 - Zuverlässigkeit
 - Anforderungen an Dokumentation
 - Vorbereitung Wiederverwendung
 - Personalfaktoren, z.B.:
 - Fluktuation
 - Anwendungserfahrung
 - Projektfaktoren:
 - Werkzeugeinsatz
 - Zeitplan
 - Beeinflussung:
 - Technisch, projekttechnisch
 - Konsequenz: Beeinflussung durch Projektmanagement



Skalierfaktoren

- Prozessparameter
 - Faktoren wirken exponentiell auf Aufwand/Laufzeit
 - Arten von Faktoren:
 - Portfolio: Projektdomäne vs. Unternehmensknowhow
 - Prozess: Vorgaben, Reife, Risikomanagement
 - Personal: Teamkohäsion
 - Beeinflussung Prozessparameter:
 - Marktstrategisch: Projektauswahl
 - Unternehmensstrategisch: Entwicklungskultur
 - Personalstrategisch: Personalkultur
 - Konsequenz: Beeinflussung durch Unternehmensmanagement



Zusammenfassung

- Drei Merksätze zur Schätzung
 - Funktionelle Abhängigkeit kann Code als Komplexitätsmaß ersetzen.
 - Entwicklungsaufwand wird von Projektfaktoren linear, von Prozessfaktoren exponentiell beeinflusst.
 - Laufzeit kann bei Mehraufwand maximal auf 70-75% gekürzt werden.
- Zwei Warnungen zur Schätzung:
 - Detaillierung bestimmt Genauigkeit (20-50%)
 - Modelle benötigen Kalibrierung