

Beispiel: Function Points zusammenfassen

- El_1 : BibTeX-Datei importieren
- EO_1 : Abfrage über Taxonomie anzeigen lassen
- EQ_1 : Artikel über Suchmaske abfragen
- ILF_1 : Referenzen-Datenbank
- ELF_1 : externe BibTeX-Datei

Parameter	Zähler	RET	DET	Gewicht	Wert
ILF_1	1	2	7	7	7
ELF_1	1	2	7	5	5
Parameter	Zähler	FTR	DET	Gewicht	Wert
El_1	1	2	7	4	4
EO_1	1	1	7	4	4
EQ_1	1	1	7	3	3
Unadjusted Function Points (UFP)					23

Konkrete Fragen I

- Does the system require reliable backup and recovery? **3**
- Are data communications required? **2**
- Are there distributed processing functions? **0**
- Is performance critical? **1**
- Will the system run in an existing, heavily utilized operational environment? **1**
- Does the system require on-line data entry? **4**
- Does the online data entry require the input transaction to be built over multiple screens or operations? **3**

Konkrete Fragen II

- Are the master files updated on-line? **5**
- Are the inputs, outputs, files, or inquiries complex? **1**
- Is the internal processing complex? **1**
- Is the code designed to be reusable? **1**
- Are conversion and installation included in the design? **2**
- Is the system designed for multiple installations in different organizations? **2**
- Is the application designed to facilitate change and ease of use by the user? **3**

FP – Gewichtete Function-Points

- TDI (Total Degree of Influence) = Summe der Bewertungen
- VAF (Value Adjustment Factor) = $\text{TDI}/100 + 0,65$
→ Gesamteinflussfaktor: 65% - 135%
- AFP (Adjusted Function Points) = $\text{UFP} \cdot \text{VAF}$

- TDI (Total Degree of Influence) = Summe der Bewertungen
- VAF (Value Adjustment Factor) = $\text{TDI}/100 + 0,65$
→ Gesamteinflussfaktor: 65% - 135%
- AFP (Adjusted Function Points) = $\text{UFP} \cdot \text{VAF}$

Beispiel:

- $\text{TDI} = 29$
- $\text{VAF} = 29/100 + 0,65 = 0,94$
- $\text{AFP} = 23 \cdot 0,94 = 21,62$

FP – Umrechnung in Aufwand

Gesucht: Abbildung FPs \rightarrow Aufwand

- Erstellung einer neuen Erfahrungskurve
(Zählen abgeschlossener Projekte, Regressionsanalyse)

¹International Software Benchmarking Standards Group

FP – Umrechnung in Aufwand

Gesucht: Abbildung FPs \rightarrow Aufwand

- Erstellung einer neuen Erfahrungskurve
(Zählen abgeschlossener Projekte, Regressionsanalyse)
- Datenbank, z.B. ISBSG¹:
 - 3GL-Projekte: $PM = 0,971 \cdot AFP^{0,351}$
 - 4GL-Projekte: $PM = 0,622 \cdot AFP^{0,405}$
 - basierend auf 662 Projekten: $PM = 0,38 \cdot AFP^{0,37}$

¹International Software Benchmarking Standards Group

FP – Umrechnung in Aufwand

Gesucht: Abbildung FPs \rightarrow Aufwand

- Erstellung einer neuen Erfahrungskurve
(Zählen abgeschlossener Projekte, Regressionsanalyse)
- Datenbank, z.B. ISBSG¹:
 - 3GL-Projekte: $PM = 0,971 \cdot AFP^{0,351}$
 - 4GL-Projekte: $PM = 0,622 \cdot AFP^{0,405}$
 - basierend auf 662 Projekten: $PM = 0,38 \cdot AFP^{0,37}$
- grobe Schätzung mit Faustregeln Jones (1996):
 - Entwicklungsdauer (Monate) = $AFP^{0.4}$
 - Personen = $AFP / 150$ (aufgerundet)
 - Aufwand (Personenmonate) = Personen \cdot Entwicklungsdauer

¹International Software Benchmarking Standards Group

FP – Umrechnung in Aufwand

Gesucht: Abbildung FPs \rightarrow Aufwand

- Erstellung einer neuen Erfahrungskurve
(Zählen abgeschlossener Projekte, Regressionsanalyse)
- Datenbank, z.B. ISBSG¹:
 - 3GL-Projekte: $PM = 0,971 \cdot AFP^{0,351}$
 - 4GL-Projekte: $PM = 0,622 \cdot AFP^{0,405}$
 - basierend auf 662 Projekten: $PM = 0,38 \cdot AFP^{0,37}$
- grobe Schätzung mit Faustregeln Jones (1996):
 - Entwicklungsdauer (Monate) = $AFP^{0.4}$
 - Personen = $AFP / 150$ (aufgerundet)
 - Aufwand (Personenmonate) = Personen \cdot Entwicklungsdauer

Beispiel (mit Jones-Schätzung):

- Entwicklungsdauer (Monate) = $21.62^{0.4} = 3.42$
- Personen = $21.62/150 \rightarrow 1$
- Aufwand (Personenmonate) = $1 \cdot 3.42 = 3.42$

¹International Software Benchmarking Standards Group

FP – Umrechnung in LOC

Mittlere Anzahl Codezeilen pro FP (Jones 1995):

Sprache	ØLOC
Assembler	320
C	128
FORTRAN	107
COBOL (ANSI 85)	91
Pascal	91
C++	53
Java	53
Ada 95	49
Smalltalk	21
SQL	12

Bewertung der Function-Point-Methode

- + Wird als beste verfügbare Methode zur Schätzung kommerzieller Anwendungssysteme angesehen (Balzert 1997)
- + Sinnvoll einsetzbar, wenn Erfahrungswerte von vergleichbaren Projekten vorliegen (Kemerer 1987)
 - Bewertung der Systemmerkmale subjektiv (Symons 1988)
- + Studie: mittlere FP-Abweichung zwischen 2 „Zählern“ nur 12% (Kemerer und Porter 1992)
 - Zählen der FPs relativ aufwendig

- Für 4GLs (Query Languages, Report Writers, ...)
- Haben nicht unbedingt mit OOP-Objekten zu tun
- Gewichtete Schätzung von
 - Anzahl verschiedener „Screens“
 - Anzahl erstellter „Reports“
 - Anzahl zu entwickelnder 3GL-Module
- Vorteil: Einfacher und weniger zu schätzen:
vergleichbare Präzision wie Function-Point-Schätzung (Banker u. a. 1991)
47% des Aufwands für Function-Point-Schätzung (Kauffman und Kumar 1993)

Object Points

Screens

# views contained	# data tables		
	< 4	< 8	8+
< 3	1	1	2
3-7	1	2	3
> 8	2	3	3

Reports

# sections contained	# data tables		
	< 4	< 8	8+
0-1	2	2	5
2-3	2	5	8
4+	5	8	8

Views: Menge logisch zusammengehöriger Daten (z.B. Kundenstammdaten)

Data Tables = # Server data tables + # Client data tables (Tabellen, die abgefragt werden müssen, um Daten zu bestimmen)

Jede 3GL-Komponente: 10 object points

COCOMO = Constructive Cost Model (Boehm 1981)

- Basiert auf Auswertung sehr vieler Projekte
- Eingaben: Projektkomplexität (3 Stufen), Systemgröße
- Ausgaben: Realisierungsaufwand, Entwicklungszeit
- Drei Genauigkeitsstufen (steigender Aufwand):
 - *Basic*: Aufwand = $a \cdot \text{KLOC}^b$, Dauer = $c \cdot \text{Aufwand}^d$
 - *Intermediate*: Dekomposition, 15 Einflussfaktoren (Kategorien: Produkt, Projekt, Computer, Personal)
 - *Advanced*: Einflussfaktoren pro Phase

a, b konstant, abhängig von Projektkomplexität:

- *Organic*: $PM = 2,4 \cdot KDSI^{1,05} \cdot M$
 - wohl verstandene Anwendungsdomäne mit kleinen Teams
- *Semidetached*: $PM = 3,0 \cdot KDSI^{1,12} \cdot M$
 - komplexere Projekte, bei dem Teams nur begrenzte Erfahrungen haben
- *Embedded*: $PM = 3,6 \cdot KDSI^{1,20} \cdot M$
 - Projekte, eingebettet in komplexe Systeme aus Hardware, Software, Vorschriften und betriebliche Abläufe

KDSI = Kilo Delivered Source Instructions

M ergibt sich aus Einflussfaktoren

Damals:

- Wasserfallprozess
- nur Neuentwicklung
- Mainframes

Heute:

- Inkrementelle Entwicklung
- Wiederverwendung, COTS-Komponenten
- PCs
- Reengineering
- Code-Generierung

→ COCOMO II (Boehm u. a. 1995)

Unterscheidung nach Phasen (Boehm u. a. 2000):

- Frühe Prototypenstufe
- Frühe Entwurfsstufe
- Stufe nach Architekturentwurf

Spätere Schätzung → höhere Genauigkeit

COCOMO II – Early prototyping level

Eingaben:

- Object Points (OP)
- Produktivität (PROD):

Erfahrung/Fähigkeiten der Entwickler	-	-	-	o	+	++
Reife/Fähigkeiten der CASE-Tools	-	-	-	o	+	++
PROD (NOP/Monat)	4	7	13	25	50	

- Wiederverwendungsanteil *%reuse* in Prozent

COCOMO II – Early prototyping level

Eingaben:

- Object Points (OP)
- Produktivität (PROD):

Erfahrung/Fähigkeiten der Entwickler	-	-	-	o	+	++
Reife/Fähigkeiten der CASE-Tools	-	-	-	o	+	++
PROD (NOP/Monat)	4	7	13	25	50	

- Wiederverwendungsanteil *%reuse* in Prozent

Abgeleitete Größen:

- New Object Points (NOP): berücksichtigen Wiederverwendung
$$NOP = OP \cdot (100 - \%reuse)/100$$
- Aufwand in Personenmonaten $PM = NOP/PROD$

Unterstützt Prototypen, Wiederverwendung

COCOMO II – Early design level

- Schätzung basiert auf Function Points (LOCs werden daraus abgeleitet)
- Personenmonate $PM_{NS} = A \cdot KLOC^E \cdot EM + PM_m$ bei nominalem Zeitplan

COCOMO II – Early design level

- Schätzung basiert auf Function Points (LOCs werden daraus abgeleitet)
- Personenmonate $PM_{NS} = A \cdot KLOC^E \cdot EM + PM_m$ bei nominalem Zeitplan
- $A = 2,94$ in initialer Kalibrierung
- Exponent E :
 - 5 Faktoren w_i für Exponent E (5 = sehr klein, 0 = sehr groß):
Erfahrung mit Domäne, Flexibilität des Entwicklungsprozesses, Risikomanagement, Teamzusammenhalt, Prozessreife
 - $E = B + \sum w_i/100$ mit $B = 1.01$

COCOMO II – Early design level

- Schätzung basiert auf Function Points (LOCs werden daraus abgeleitet)
- Personenmonate $PM_{NS} = A \cdot KLOC^E \cdot EM + PM_m$ bei nominalem Zeitplan
- $A = 2,94$ in initialer Kalibrierung
- Exponent E :
 - 5 Faktoren w_i für Exponent E (5 = sehr klein, 0 = sehr groß): Erfahrung mit Domäne, Flexibilität des Entwicklungsprozesses, Risikomanagement, Teamzusammenhalt, Prozessreife
 - $E = B + \sum w_i/100$ mit $B = 1.01$
- Effort Multiplier EM :
 - 7 lineare Einflussfaktoren (6 Stufen, Standard: 1.00, in Tabelle nachschlagen): Produktgüte und -komplexität, Plattformkomplexität, Fähigkeiten des Personals, Erfahrung des Personals, Zeitplan, Infrastruktur
 - $EM = \prod Effort-Multiplier_i$

COCOMO II – Early design level

- Schätzung basiert auf Function Points (LOCs werden daraus abgeleitet)
- Personenmonate $PM_{NS} = A \cdot KLOC^E \cdot EM + PM_m$ bei nominalem Zeitplan
- $A = 2,94$ in initialer Kalibrierung
- Exponent E :
 - 5 Faktoren w_i für Exponent E (5 = sehr klein, 0 = sehr groß): Erfahrung mit Domäne, Flexibilität des Entwicklungsprozesses, Risikomanagement, Teamzusammenhalt, Prozessreife
 - $E = B + \sum w_i/100$ mit $B = 1.01$
- Effort Multiplier EM :
 - 7 lineare Einflussfaktoren (6 Stufen, Standard: 1.00, in Tabelle nachschlagen): Produktgüte und -komplexität, Plattformkomplexität, Fähigkeiten des Personals, Erfahrung des Personals, Zeitplan, Infrastruktur
 - $EM = \prod Effort-Multiplier_i$
- Korrekturfaktor PM_m bei viel generiertem Code (höhere Produktivität; nicht weiter diskutiert hier)

Faktoren für Exponent E I

- Erfahrung mit Anwendungsbereich (PREC)
 - Erfahrung mit vorliegendem Projekttyp
 - 5 keine Erfahrung
 - 0 vollständige Vertrautheit
- Entwicklungsflexibilität (FLEX)
 - Grad der Flexibilität im Entwicklungsprozess
 - 5 Prozess vom Kunden fest vorgegeben
 - 0 Kunde legt nur Ziele fest
- Risikomanagement (RESL)
 - Umfang der durchgeführten Risikoanalyse
 - 5 keine Risikoanalyse
 - 0 vollständige und genaue Risikoanalyse

Faktoren für Exponent E II

- Teamzusammenhalt (TEAM)
 - Vertrautheit und Eingespieltheit des Entwicklungsteams
 - 5 schwierige Interaktionen
 - 0 integriertes und effektives Team ohne Kommunikationsprobleme
- Prozessreife (EPML)
 - Reife des Entwicklungsprozesses (z.B. CMM);
 - beabsichtigt: gewichteter Anteil der mit “ja” beantworteten Fragen im CMM-Fragebogen
 - pragmatisch: CMM-Level
 - 5 niedrigster CMM-Level
 - 0 höchster CMM-Level