

# 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

# Effort Multiplier RCPX: Product Reliability and Complexity

RELY Required reliability  
 DOCU Documentation match to life-cycle needs  
 CPLX Product Complexity  
 DATA Data base size

Grad: Punkte:	very low 1	low 2	nominal 3	high 4	very high 5	extra high 6	
<b>RCPX</b>							
RELY	very little	little	some	basic	strong		
DOCU	very little	little	some	basic	strong		
CPLX	very little	little	some	basic	strong		
DATA		small	moderate	large	very large		
$\sum$ Punkte:	5,6	7,8	9–11	12	13–15	16–18	19–21
$EM_{RCPX}$	0.49	0.60	0.83	1.00	1.33	1.91	2.72

# Effort Multiplier PDIF: Platform Difficulty

TIME Execution time constraints

STOR Main storage constraints

PVOL Platform volatility

Grad: Punkte:	low 2	nominal 3		high 4	very high 5	extra high 6
<b>PDIF</b>						
TIME		≤50%		≤65%	≤80%	≤90%
STORE		≤50%		≤65%	≤80%	≤90%
PVOL	very stable	stable	somewhat	volatile		
$\sum$ Punkte:	8	9	10–12	13–15	16,17	
$EM_{PDIF}$	0.87	1.00	1.29	1.81	2.61	

# Effort Multiplier PERS: Personnel Capability

ACAP Analyst capability (gemessen als Perzentil)  
PCAP Programmier capability (gemessen als Perzentil)  
PCON Personnel continuity (gemessen durch Personalfluktuation)

Grad: Punkte:	very low 1	low 2	nominal 3	high 4	very high 5		
<b>PERS</b>							
ACAP	15%	35%	55%	75%	90%		
PCAP	15%	35%	55%	75%	90%		
PCON	48%	24%	12%	6%	3%		
$\sum$ Punkte:	3,4	5,6	7,8	9	10,11	12,13	14,15
$EM_{PERS}$	2.12	1.62	1.26	1.00	0.83	0.63	0.50

# Effort Multiplier PREX: Personnel Experience

AEXP Applications experience  
 PLEX Platform experience  
 LTEX Language/tool experience

Grad:	very low	low	nominal	high	very high		
Punkte:	1	2	3	4	5		
<b>PREX</b>							
AEXP	≤2 Mo.	6 Mo.	1 J.	3 J.	6 J.		
PLEX	≤2 Mo.	6 Mo.	1 J.	3 J.	6 J.		
LTEX	≤2 Mo.	6 Mo.	1 J.	3 J.	6 J.		
$\sum$ Punkte:	3,4	5,6	7,8	9	10,11	12,13	14,15
$EM_{PREX}$	1.59	1.33	1.22	1.00	0.87	0.74	0.62

# Effort Multiplier FCIL: Facilities

TOOL Use of Software Tools  
 SITE Multisite Development

Grad:	very low	low	nominal	high	very high	
Punkte:	1	2	3	4	5	6

**FCIL**

TOOL	(1)	(2)	(3)	(4)	(5)		→ nächste Folie
SITE	(1)	(2)	(3)	(4)	(5)	(6)	→ nächste Folie

$\sum$ Punkte:	2	3	4,5	6	7,8	9,10	11
$EM_{FCIL}$	1.43	1.30	1.10	1.00	0.87	0.73	0.62



## TOOL:

- ① Editor, Compiler, Debugger
- ② einfaches CASE-Werkzeug, schlechte Integration
- ③ Basis-Life-Cycle-Tools moderat integriert
- ④ weitergehende, reife Life-Cycle-Tools moderat integriert
- ⑤ weitergehende, proaktive Life-Cycle-Tools gut integriert mit Prozessen, Methoden und Wiederverwendung

# TOOL und SITE

## TOOL:

- ① Editor, Compiler, Debugger
- ② einfaches CASE-Werkzeug, schlechte Integration
- ③ Basis-Life-Cycle-Tools moderat integriert
- ④ weitergehende, reife Life-Cycle-Tools moderat integriert
- ⑤ weitergehende, proaktive Life-Cycle-Tools gut integriert mit Prozessen, Methoden und Wiederverwendung

## SITE:

- ① Telefon prinzipiell vorhanden und Post
- ② individuelles Telefon und Fax
- ③ E-Mail (niedrige Bandbreite)
- ④ elektronische Kommunikation mit großer Bandbreite
- ⑤ elektronische Kommunikation mit großer Bandbreite, gelegentliche Videokonferenzen
- ⑥ Interaktive Multimedia

# Effort Multiplier SCED: Schedule

Es besteht Notwendigkeit, den Zeitplan zu straffen bzw. es wird mehr Zeit als notwendig eingeräumt.

SCED = Verkürzung bzw. Verlängerung des nominalen Zeitplans.

	75%	85%	100%	130%	160%
$EM_{SCED}$	1.43	1.14	1.00	1.00	1.00

## Nominaler Aufwand

Personenmonate  $PM_{NS} = A \cdot KLOC^E \cdot EM$  mit  $EM_{SCED} = 1,0$

## Nominaler Aufwand

Personenmonate  $PM_{NS} = A \cdot KLOC^E \cdot EM$  mit  $EM_{SCED} = 1,0$

mit  $A = 2,94$  und  $E = B + \sum w_i/100$  mit  $B = 1,01$ .

## Nominaler Aufwand

Personenmonate  $PM_{NS} = A \cdot KLOC^E \cdot EM$  mit  $EM_{SCED} = 1,0$

mit  $A = 2,94$  und  $E = B + \sum w_i/100$  mit  $B = 1,01$ .

Annahme: es herrschen einfache Verhältnisse:

→  $\forall i : w_i = 0 \Rightarrow E = 1,01$  (bester Fall)

→ nominale Effort-Multiplier = 1,00 (Normalfall)  $\Rightarrow EM = 1,00$

## Nominaler Aufwand

Personenmonate  $PM_{NS} = A \cdot KLOC^E \cdot EM$  mit  $EM_{SCED} = 1,0$

mit  $A = 2,94$  und  $E = B + \sum w_i/100$  mit  $B = 1,01$ .

Annahme: es herrschen einfache Verhältnisse:

→  $\forall i : w_i = 0 \Rightarrow E = 1,01$  (bester Fall)

→ nominale Effort-Multiplier = 1,00 (Normalfall)  $\Rightarrow EM = 1,00$

Geschätzte Lines-of-Code = 100.000

→  $PM_{NS} = 2,94 \times 100^{1,01} \times 1,0 = 307,86$  Monate  $\approx 25\frac{1}{2}$  Jahre

Nominale Entwicklungsdauer (Kalenderzeit in Monaten)

$$TDEV_{NS} = C \times PM_{NS}^{D+0,2 \times (E-B)}$$

mit  $C = 3,67$  und  $D = 0,28$ .



## Nominale Entwicklungsdauer (Kalenderzeit in Monaten)

$$TDEV_{NS} = C \times PM_{NS}^{D+0,2 \times (E-B)}$$

mit  $C = 3,67$  und  $D = 0,28$ .

Beispiel:  $TDEV_{NS} = 3,67 \times 307,86^{0,28+0,2 \times 0} = 18,26$

# Entwicklungsdauer

Nominale Entwicklungsdauer (Kalenderzeit in Monaten)

$$TDEV_{NS} = C \times PM_{NS}^{D+0,2 \times (E-B)}$$

mit  $C = 3,67$  und  $D = 0,28$ .

Beispiel:  $TDEV_{NS} = 3,67 \times 307,86^{0,28+0,2 \times 0} = 18,26$

Anzahl Entwickler

$$N = PM_{NS} / TDEV_{NS}$$

# Entwicklungsdauer

Nominale Entwicklungsdauer (Kalenderzeit in Monaten)

$$TDEV_{NS} = C \times PM_{NS}^{D+0,2 \times (E-B)}$$

mit  $C = 3,67$  und  $D = 0,28$ .

Beispiel:  $TDEV_{NS} = 3,67 \times 307,86^{0,28+0,2 \times 0} = 18,26$

Anzahl Entwickler

$$N = PM_{NS} / TDEV_{NS}$$

Beispiel:  $N = 307,86 / 18,26 \approx 17$

# Verkürzte Entwicklungsdauer

Chef: „Wieso 18 Monate? Geht das nicht schneller?“

# Verkürzte Entwicklungsdauer

Chef: „Wieso 18 Monate? Geht das nicht schneller?“

$PM_{NS}$  geht von  $SCED = 1,0$  aus.

Abweichung von der nominalen Entwicklungsdauer

$$TDEV = TDEV_{NS} \times SCED / 100$$

# Verkürzte Entwicklungsdauer

Chef: „Wieso 18 Monate? Geht das nicht schneller?“

$PM_{NS}$  geht von  $SCED = 1,0$  aus.

Abweichung von der nominalen Entwicklungsdauer

$$TDEV = TDEV_{NS} \times SCED / 100$$

Wir verkürzen auf 75%:

$$TDEV = 18,26 \times 75 / 100 = 10,3 \text{ Monate}$$

# Verkürzte Entwicklungsdauer

Chef: „Wieso 18 Monate? Geht das nicht schneller?“

$PM_{NS}$  geht von  $SCED = 1,0$  aus.

Abweichung von der nominalen Entwicklungsdauer

$$TDEV = TDEV_{NS} \times SCED / 100$$

Wir verkürzen auf 75%:

$$TDEV = 18,26 \times 75 / 100 = 10,3 \text{ Monate}$$

Chef: „Super!“

# Verkürzte Entwicklungsdauer

Chef: „Wieso 18 Monate? Geht das nicht schneller?“

$PM_{NS}$  geht von  $SCED = 1,0$  aus.

Abweichung von der nominalen Entwicklungsdauer

$$TDEV = TDEV_{NS} \times SCED / 100$$

Wir verkürzen auf 75%:

$$TDEV = 18,26 \times 75 / 100 = 10,3 \text{ Monate}$$

Chef: „Super!“

Wir setzen  $SCED = 75$  in PM-Formel ein.

$$PM = 2,94 \times 100^{1,01} \times 1,0 \times 1,43 = 440,23$$

$$\text{Erhöhung des Aufwands: } 440,23 / 307,86 = 1,43$$



# Verkürzte Entwicklungsdauer

Chef: „Wieso 18 Monate? Geht das nicht schneller?“

$PM_{NS}$  geht von  $SCED = 1,0$  aus.

Abweichung von der nominalen Entwicklungsdauer

$$TDEV = TDEV_{NS} \times SCED / 100$$

Wir verkürzen auf 75%:

$$TDEV = 18,26 \times 75 / 100 = 10,3 \text{ Monate}$$

Chef: „Super!“

Wir setzen  $SCED = 75$  in PM-Formel ein.

$$PM = 2,94 \times 100^{1,01} \times 1,0 \times 1,43 = 440,23$$

$$\text{Erhöhung des Aufwands: } 440,23 / 307,86 = 1,43$$

Chef: „43% mehr Kosten? Seid Ihr wahnsinnig?“

Berücksichtigt:

- Auswirkungen erwarteter Änderungen von Anforderungen
- Ausmaß/Aufwand der möglichen Wiederverwendung
  - Aufwand für Entscheidung, ob Wiederverwendung
  - Aufwand für das Verstehen existierenden Codes
  - Aufwand für Anpassungen
- 17 verfeinerte lineare Einflussfaktoren
- Schätzung basiert auf LOC

# COCOMO II – Post-architecture level

- Produktgüte/-kompl → Verlässlichkeit, Datenbasisgröße, Komplexität, Dokumentation
- Plattformkomplexität → Laufzeit-, Speicherbeschränkungen, Plattformdynamik
- Fähigkeiten Personal → Fähigkeiten der Analysten/Entwickler, Kontinuität des Personals
- Erfahrung Personal → Domänenenerfahrung der Analysten/Entwickler, Erfahrung mit Sprache und Werkzeugen
- Infrastruktur → Tools, verteilte Entwicklung+Kommunikation

# Einflussfaktoren (Cost Drivers) für Cocomo-2

	--	-	o	+	++	+++
<b>Product Attributes</b>						
RELY – Required reliability	0.82	0.92	1.00	1.10	1.26	
DATA – Data base size		0.90	1.00	1.14	1.28	
CPLX – Product Complexity	0.73	0.87	1.00	1.17	1.34	1.74
RUSE – Required Reusability		0.95	1.00	1.07	1.15	1.24
DOCU – Doc. match to life-cycle needs	0.81	0.91	1.00	1.11	1.23	
<b>Computer Attributes</b>						
TIME – Execution time constr.			1.00	1.11	1.29	1.63
STOR – Main storage constr.			1.00	1.05	1.17	1.46
PVOL – Platform volatility		0.87	1.00	1.15	1.30	

# Einflussfaktoren (Cost Drivers) für Cocomo-2

	--	-	o	+	++	+++
<b>Personell attributes</b>						
ACAP – Analyst capability	1.42	1.19	1.00	0.85	0.71	
PCAP – Programmer capability	1.34	1.15	1.00	0.88	0.76	
AEXP – Applications experience	1.22	1.10	1.00	0.88	0.81	
PLEX – Platform experience	1.19	1.09	1.00	0.91	0.85	
LTEX – Language/tool exp.	1.20	1.09	1.00	0.91	0.84	
<b>Project attributes</b>						
TOOL – Use of software tools	1.17	1.09	1.00	0.90	0.78	
SITE – Multisite development	1.22	1.09	1.00	0.93	0.86	0.80
SCED – Required dev. schedule	1.43	1.14	1.00	1.00	1.00	

- alle Schätzungen basieren auf Erfahrung
- kontinuierlich schätzen
- verschiedene Techniken anwenden

## 2 Entwicklungsprozesse

- Lernziele
- Wasserfallmodell
- V-Modell
- Testgetriebene Entwicklung
- Inkrementelle Entwicklung
- Spiralmodell
- Rational Unified Process
- Cleanroom Development
- Extreme Programming (XP)
- Capability Maturity Model
- Persönlicher Softwareprozess

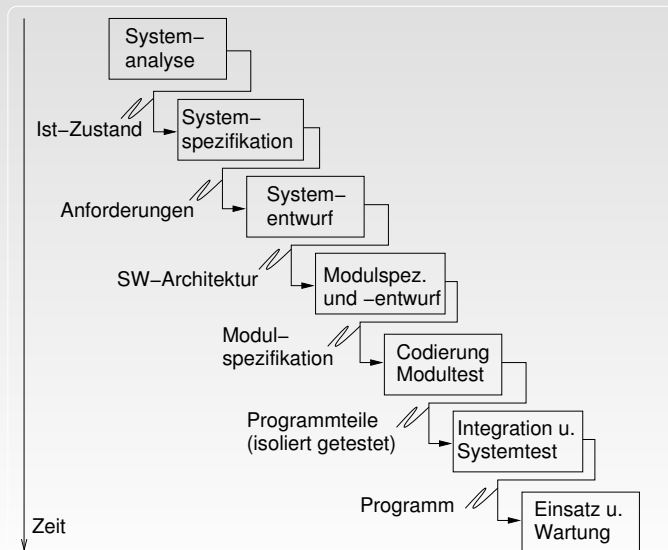
- verschiedene Software-Entwicklungsprozessmodelle kennen
- Vor- und Nachteile/Anwendbarkeit abwegen können
- die Besonderheit von Prozessen der Software-Entwicklung kennen



## Grundlagen für guten Prozess:

- Wohldefiniiertheit
  - sonst Falsch-, Nicht-, Mehrarbeit
  - sonst Information nicht verfügbar oder unverständlich
- Quantitatives Verstehen
  - objektive Grundlage für Verbesserungen
- Kontinuierliche Änderung
  - sonst Prozess schnell überholt
- Angemessenheit
  - Prozess muss dem zu lösenden Problem angemessen sein
  - d.h. effektiv und effizient sein

# Striktes Wasserfallmodell nach Royce (1970)



# Striktes Wasserfallmodell Royce (1970)

Eigenschaften dieses Modells:

- dokumenten-getrieben: jede Aktivität erzeugt Dokument
- streng sequenzielle Aktivitäten

# Striktes Wasserfallmodell Royce (1970)

Eigenschaften dieses Modells:

- dokumenten-getrieben: jede Aktivität erzeugt Dokument
- streng sequenzielle Aktivitäten
- + klar organisierter Ablauf
- + relativ einfache Führung
- + hohe Qualität der Dokumentation

# Striktes Wasserfallmodell Royce (1970)

Eigenschaften dieses Modells:

- dokumenten-getrieben: jede Aktivität erzeugt Dokument
- streng sequenzielle Aktivitäten
- + klar organisierter Ablauf
- + relativ einfache Führung
- + hohe Qualität der Dokumentation
  - Entwicklung bildet langen Tunnel
  - 90%-fertig-Syndrom
  - Spezifikationsmängel lassen sich kaum frühzeitig erkennen
  - Entwicklung beim Kunden wird ignoriert

# V-Modell von Boehm (1979)

