Introduction to Software Engineering

(CS350)

Lecture 03-1

Jongmoon Baik

Software Cost Estimation w/ COCOMOII

Early Design & Post-Architecture Models

$$PM_{NS} = A \times Size^{E} \times \prod_{i=1}^{n} EM_i$$

$$\text{where } E = B + 0.01 \times \sum_{j=1}^{5} SF_j$$

$$TDEV_{NS} = C \times (PM_{NS})^{F}$$

$$\text{where } F = D + 0.2 \times 0.01 \times \sum_{j=1}^{5} SF_j$$

$$= D + 0.2 \times (E - B)$$

A = 2.94 B = 0.91

C = 3.67 D = 0.28

- Early Design Model [6 EMs]:
- Post Architecture Model [16 EMs]:
  *Exclude SCED driver

EMs: Effort multipliers to reflect characteristics of particular software under development

A : Multiplicative calibration variable

E : Captures relative (Economies/Diseconomies of scale)

SF: Scale Factors

Scale Factors & Cost Drivers

- Project Level – 5 Scale Factors
  - Used for both ED & PA models
- Early Design – 7 Cost Drivers
- Post Architecture – 17 Cost Drivers
  - Product, Platform, Personnel, Project

Project Scale Factors - I

$$PM = A \times (Size)^{E} \times \prod_{i=1}^{n} EM_i$$

where A = 2.94 (for COCOMO II.2000)

- Relative economies or diseconomies of scale
  - E < 1.0 : economies of scale
    - Productivity increase as the project size increase
    - Achieved via project specific tools (e.g., simulation, testbed)
  - E = 1.0 : balance
    - Linear model : often used for cost estimation of small projects
  - E > 1.0 : diseconomies of scale
    - Main factors : growth of interpersonal communication overhead and growth of large-system overhead

Project Scale Factors - II

Scale Factors (SF_i)	Very Low	Low	Nominal	High	Very High	Extra High
PREC	thoroughly unprecedente	largely unprecedente	somewhat unprecedente	generally familiar	largely familiar	throughly familiar
	6.20	4.96	3.72	2.48	1.24	0.00
FLEX	rigorous	occasional relaxation	some relaxation	general conformity	some conformity	general goals
	5.07	4.05	3.04	2.03	1.01	0.00
RESL	little (20%)	some (40%)	often (60%)	generally(75	mostly (90%)	full (100%)
	7.07	5.65	4.24	2.83	1.41	0.00
TEAM	very difficult interactions	some difficult interactions	basically cooperative interactions	largely cooperative	highly cooperative	seamless interactions
	5.48	4.28	3.29	2.19	1.10	0.00
PMAT	SW-CMM Level 1 Lower	SW-CMM Level 1 Upper	SW-CMM Level 2	SW-CMM Level 3	SW-CMM Level 3	SW-CMM Level 5
	Or the Estimated Process Maturity Level (EPML)					
	7.80	6.24	4.68	3.12	1.56	0.00

PMAT == EPML

- EPML (Equivalent Process Maturity Level)

Key Process Areas	Almost Always (>90%)	Frequently (60-90%)	About Half (40-60%)	Occasionally (10-40%)	Rarely If Ever (<10%)	Does Not Apply	Don't Know
1 Requirements Management	☐	☐	☐	☐	☐	☐	☐
2 Software Project Planning	☐	☐	☐	☐	☐	☐	☐
3 Software Project Tracking and Oversight	☐	☐	☐	☐	☐	☐	☐
4 Software Subcontract Management	☐	☐	☐	☐	☐	☐	☐
⋮							

$$EPML = 5 \times \left(\sum_{i=1}^{n} \frac{KPA\%_i}{100} \right) \times \frac{1}{n}$$

PA Model – Product EMs

Effort Multiplier	Very Low	Low	Nominal	High	Very High	Extra High
RELY	slight inconven-ience	low, easily recoverable losses	moderate, easily recoverable losses	high financial loss	risk to human life	
	0.82	0.92	1.00	1.10	1.26	n/a
DATA		DB bytes/Pgm SLOC < 10	10 <= D/P < 100	100 <= D/P < 1000	D/P>=1000	
	n/a	0.90	1.00	1.14	1.28	n/a
RUSE		none	across project	across program	across product line	across multiple product lines
	n/a	0.95	1.00	1.07	1.15	1.24
DOCU	Many life-cycle needs uncovered	Some life-cycle needs uncovered.	Right-sized to life-cycle needs	Excessive for life-cycle needs	Very excessive for life-cycle needs	
	0.81	0.91	1.00	1.11	1.23	n/a
CPLX	See CPLX table					
	0.73	0.87	1.00	1.17	1.34	1.74

PA Model - CPLX

Effort Multiplier	Control Operations	Computational Operations	Device-dependent Operations	Data Management Operations	User Interface Management Operations
Very Low	Straight-line code with a few non-nested structured programming operators: DOs, CASEs, IF-THEN-ELSEs. Simple module composition via procedure calls or simple scripts.	Evaluation of simple expressions: e.g., A=B+C*(D-E)	Simple read, write statements with simple formats.	Simple arrays in main memory. Simple COTS-DB queries, updates.	Simple input forms, report generators.
Low	...	...	...	...	...
Nominal	Mostly simple nesting. Some intermodule control. Decision tables. Simple callbacks or message passing, including middleware-supported distributed processing	Use of standard math and statistical routines. Basic matrix/vector operations.	I/O processing includes device selection, status checking and error processing.	Multi-file input and single file output. Simple structural changes, simple edits. Complex COTS-DB queries, updates.	Simple use of widget set.
High	...	...	...	...	...
Very High	...	...	...	...	...
Extra High	Multiple resource scheduling with dynamically changing priorities. Microcode-level control. Distributed hard real-time control.	Difficult and unstructured numerical analysis: highly accurate analysis of noisy, stochastic data. Complex parallelization.	Device timing-dependent coding, micro-programmed operations. Performance-critical embedded systems.	Highly coupled, dynamic relational and object structures. Natural language data management.	Complex multimedia, virtual reality, natural language interface.

PA Model – Platform EMs

Effort Multiplier	Very Low	Low	Nominal	High	Very High	Extra High
TIME			≤ 50% use of available execution time	70% use of available execution time	85% use of available execution time	95% use of available execution time
	n/a	n/a	1.00	1.11	1.29	1.63
STOR			≤ 50% use of available storage	70% use of available storage	85% use of available storage	95% use of available storage
	n/a	n/a	1.00	1.05	1.17	1.46
PVOL		Major change every 12 mo.;	Major: 6 mo.; Minor: 2 wk.	Major: 2 mo.;Minor: 1	Major: 2 wk.;Minor: 2	
	n/a	0.87	1.00	1.15	1.30	n/a

PA Model – Personnel EMs

Effort Multiplier	Very Low	Low	Nominal	High	Very High	Extra High
ACAP	15th percentile	35th percentile	55th percentile	75th percentile	90th percentile	
	1.42	1.19	1.00	0.85	0.71	n/a
PCAP	15th percentile	35th percentile	55th percentile	75th percentile	90th percentile	
	1.34	1.15	1.00	0.88	0.76	n/a
PCON	48% / year	24% / year	12% / year	6% / year	3% / year	
	1.29	1.12	1.00	0.90	0.81	n/a
APEX	<= 2 months	6 months	1 year	3 years	6 years	
	1.22	1.10	1.00	0.88	0.81	n/a
LTEX	<= 2 months	6 months	1 year	3 years	6 year	
	1.20	1.09	1.00	0.91	0.84	n/a
PLEX	<= 2 months	6 months	1 year	3 years	6 year	
	1.19	1.09	1.00	0.91	0.85	n/a

PA Model – Project EMs

Effort Multiplier	Very Low	Low	Nominal	High	Very High	Extra High
TOOL	edit, code, debug	simple, frontend, backend CASE, little integration	basic life-cycle tools, moderately integrated	strong, mature life-cycle tools, moderately integrated	strong, mature, proactive life-cycle tools, well integrated with processes, methods, reuse	
	1.17	1.09	1.00	0.90	0.78	n/a
SITE	Inter-national	Multi-city and Multi-company	Multi-city or Multi-company	Same city or metro. area	Same building or complex	Fully collocated
	Some phone, mail	Individual phone, FAX	Narrow band email	Wideband electronic communication.	Wideband elect. comm., occasional video conf.	Interactive multimedia
	1.22	1.09	1.00	0.93	0.86	0.80
SCED	75% of nominal	85% of nominal	100% of nominal	130% of nominal	160% of nominal	
	1.43	1.14	1.00	1.00	1.00	n/a

Calibration & Prediction Accuracy

COCOMO Calibration

	COCOMO 81	COCOMO II.1997	COCOMO II.2000
Project Data Points	63	83	161
Calibration		10% Data, 90% Expert	Bayesian

MRE: PRED (.30) Values

	COCOMO 81	COCOMO II.1997	COCOMO II.2000
Effort - By Organization	81%	52% 64%	75% 80%
Schedule - By Organization	65%	61% 62%	72% 81%

COCOMO II Family

Model	No. of Drivers		Sizing
	Environment	Process	
Application Composition	**2**	**0**	**Application Points**
Early Design	**7**	**5**	**Function Points or SLOC**
Post Architecture	**17**	**5**	**Function Points or SLOC**
COCOMO81	**15**	**1**	**SLOC (FP Extension)**

COCOMO Model Comparison

	COCOMO	Ada COCOMO	COCOMO II: Application Composition	COCOMO II: Early Design	COCOMO II: Post-Architecture
Size	Delivered Source Instructions (DSI) or Source Lines of Code (SLOC)	DSI or SLOC	Application Points	Function Points (FP) and Language or SLOC	FP and Language or SLOC
Reuse	Equivalent SLOC = Linear *f*(DM,CM,IM)	Equivalent SLOC = Linear *f*(DM,CM,IM)	Implicit in Model	Equivalent SLOC = nonlinear *f*(AA, SU,UNFM,DM,CM,IM)	Equivalent SLOC = nonlinear *f*(AA, SU,UNFM,DM,CM,IM)
Rqts. Change	Requirements Volatility rating: (RVOL)	RVOL rating	Implicit in Model	Change % : RQEV	RQEV
Maintenance	Annual Change Traffic (ACT) = %added + %modified	ACT	Object Point ACT	*f*(ACT,SU,UNFM)	*f*(ACT,SU,UNFM)
Scale (b) in $MM_{NOM}=a(Size)^b$	Organic: 1.05 Semidetached: 1.12 Embedded: 1.20	Embedded: 1.04 -1.24 depending on degree of: • early risk elimination • solid architecture • stable requirements • Ada process maturity	1.0	.91-1.23 depending on the degree of: • precedentedness • conformity • early architecture, risk resolution • team cohesion • process maturity (SEI)	.91-1.23 depending on the degree of: • precedentedness • conformity • early architecture, risk resolution • team cohesion • process maturity (SEI)
Product Cost Drivers	RELY, DATA, CPLX	RELY*, DATA, CPLX*, RUSE	None	RCPX*▪, RUSE*▪	RELY, DATA, DOCU*▪, CPLX▪, RUSE*▪
Platform Cost Drivers	TIME, STOR, VIRT, TURN	TIME, STOR, VMVH, VMVT, TURN	None	Platform difficulty: PDIF*▪	TIME, STOR, PVOL(=VIRT)
Personnel Cost Drivers	ACAP, AEXP, PCAP, VEXP, LEXP	ACAP*, AEXP, PCAP*, VEXP, LEXP*	None	Personnel capability and experience: PERS*▪, PREX*▪	ACAP*, AEXP▪, PCAP*, PEXP*▪, LTEX*▪, PCON*▪
Project Cost Drivers	MODP, TOOL, SCED	MODP*, TOOL*, SCED, SECU	None	SCED, FCIL*▪	TOOL*▪, SCED, SITE*▪

* Different Multipliers
▪ Different Rating Scale

USC-COCOMO II.2000 Demo.

http://csse.usc.edu/csse/research/COCOMOII/cocomo_downloads.htm

Q & A