Cost estimation with COCOMO II

Article · November 2002

CITATIONS
15

READS
6,931

1 author:

Barry Boehm
University of Southern California
373 PUBLICATIONS **21,026** CITATIONS

SEE PROFILE

Cost Estimation with COCOMO II

Barry Boehm

CS 577a, Fall 2002

October 14, 2002

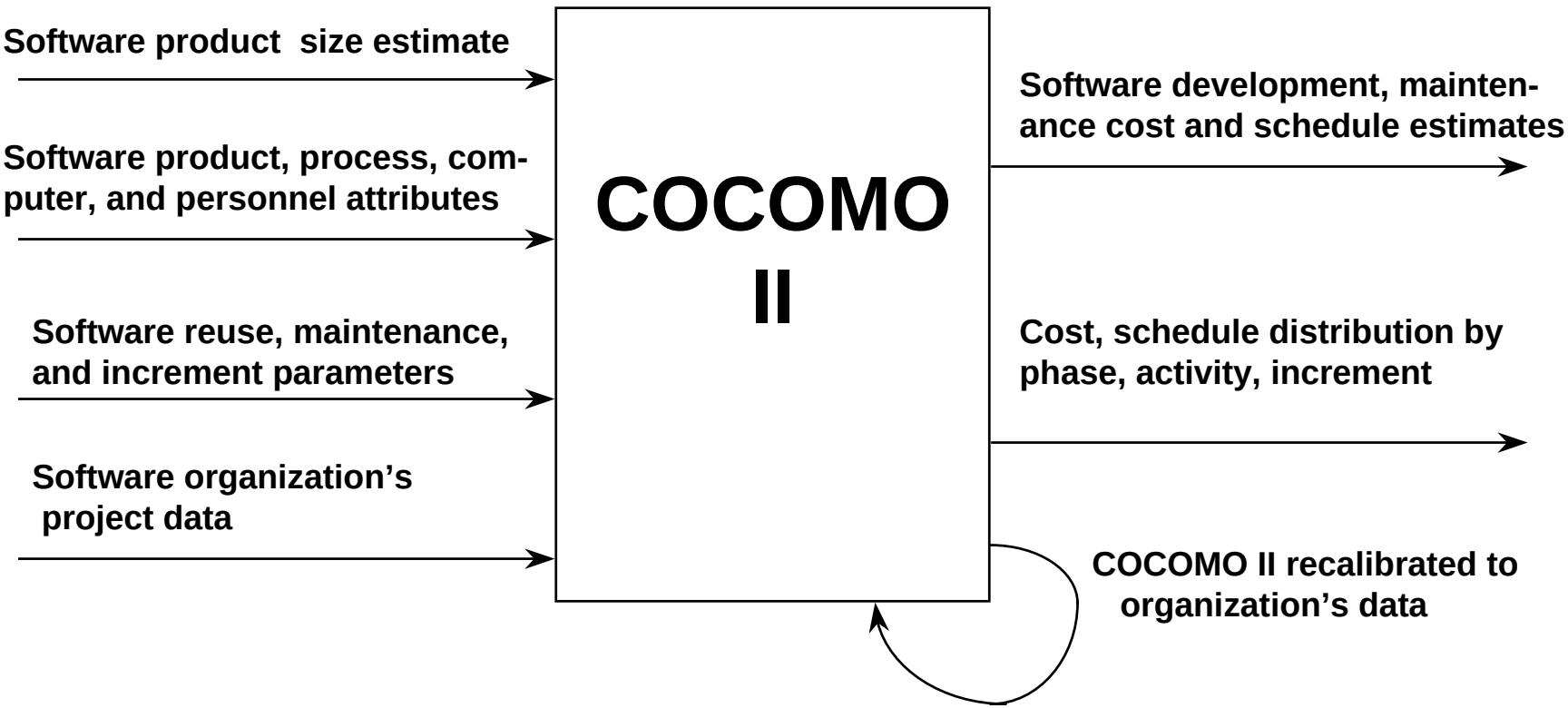
(boehm@sunset.usc.edu)

(http://sunset.usc.edu)

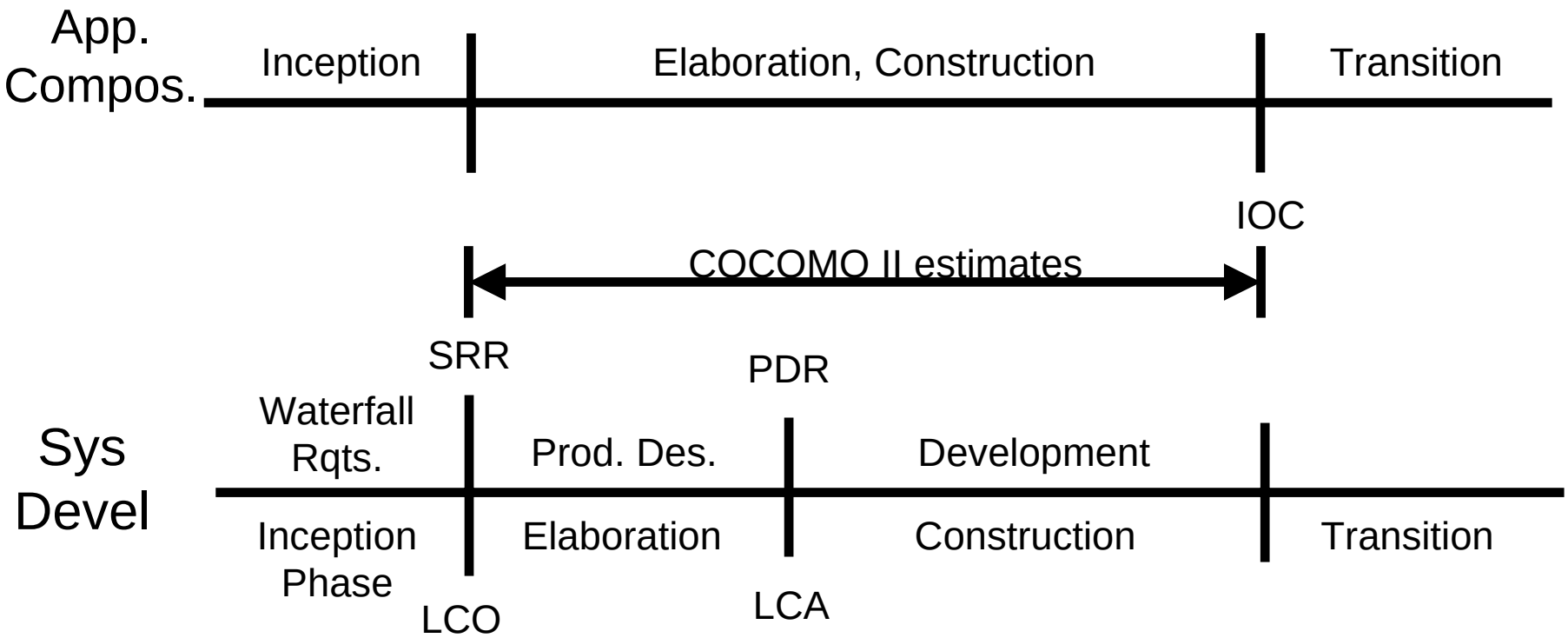
Outline

- Model Overview
  - Sizing, Reuse, and Scale Factors
  - Effort Multipliers
  - Other Features
- Example of Use: Demo
- Model Reinterpretation for CS 577

COCOMO II Overview



Relations to MBASE*/Rational Anchor Point Milestones



*MBASE: Model-Based (System) Architecting and Software Engineering

Early Design and Post-Arch Model

- **Effort:**

$$PM_{estimated} = A \times (Size)^{(SF)} \times \left(\prod_i EM_i \right)$$

- **Size**
  - **KSLOC (Thousands of Source Lines of Code)**
  - **UFP (Unadjusted Function Points) * KSLOC/UFP**
    - **KSLOC/UFP factor varies by language**
  - **EKSLOC (Equivalent KSLOC) used for adaptation**
- **SF: Scale Factors (5)**
- **EM: Effort Multipliers (7 for ED, 17 for PA)**

Scaling Exponent Approach

- Nominal person-months = A*(size)**B
- B = 0.91 + 0.01 $\sum$(exponent driver ratings)
  - B ranges from 0.91 to 1.23
  - 5 drivers; 6 rating levels each
- Exponent drivers:
  - Precedentedness
  - Development flexibility
  - Architecture/ risk resolution
  - Team cohesion
  - Process maturity (derived from SEI CMM)

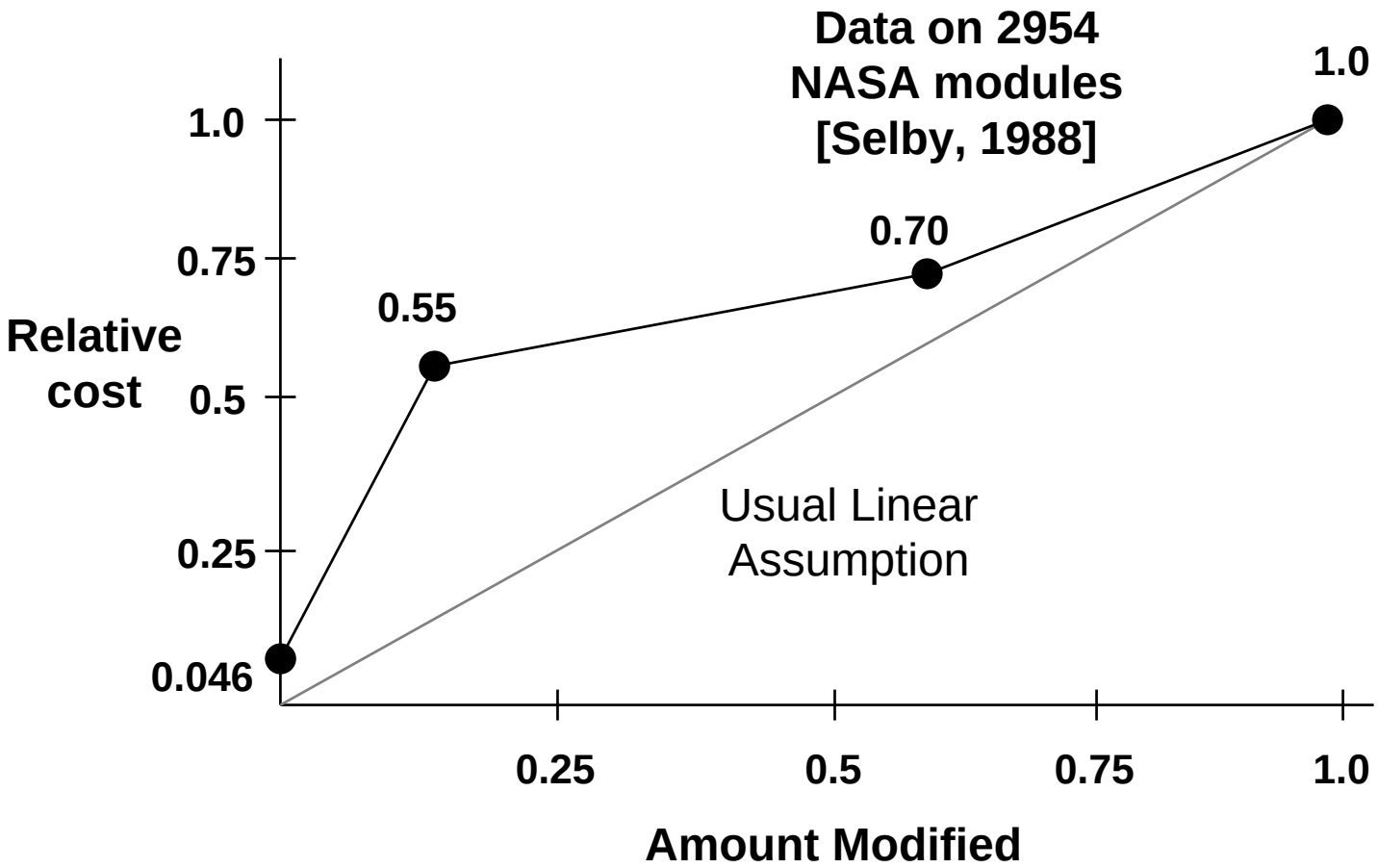
Project Scale Factors

$$PM_{estimated} = 3.67 \times (Size)^{(SF)} \times \left(\prod_i EM_i \right)$$

$$SF = 0.91 + 0.01 \times \sum w_i$$

Scale Factors (*Wi*)	Very Low	Low	Nominal	High	Very High	Extra High
PREC	thoroughly unprecedented	largely unprecedented	somewhat unprecedented	generally familiar	largely familiar	throughly familiar
FLEX	rigorous	occasional relaxation	some relaxation	general conformity	some conformity	general goals
RESL	little (20%)	some (40%)	often (60%)	generally (75%)	mostly (90%)	full (100%)
TEAM	very difficult interactions	some difficult interactions	basically cooperative interactions	largely cooperative	highly cooperative	seamless interactions
PMAT	weighted sum of 18 KPA achievement levels					

Nonlinear Reuse Effects



Reuse and Reengineering Effects

- Add Assessment & Assimilation increment (AA)
  - Similar to conversion planning increment
- Add software understanding increment (SU)
  - To cover nonlinear software understanding effects
  - Coupled with software unfamiliarity level (UNFM)
  - Apply only if reused software is modified
- Results in revised Equivalent Source Lines of Code (ESLOC)
  - AAF = 0.4(DM) + 0.3 (CM) + 0.3 (IM)
  - ESLOC = ASLOC[AA+AAF(1+0.02(SU)(UNFM))], AAF $\leq$ 0.5
  - ESLOC = ASLOC[AA+AAF(SU)(UNFM))], AAF > 0.5

Software Understanding Rating / Increment

	Very Low	Low	Nom	High	Very High
Structure	Very low cohesion, high coupling, spaghetti code.	Moderately low cohesion, high coupling.	Reasonably well - structured; some weak areas.	High cohesion, low coupling.	Strong modularity, information hiding in data/control structures.
Application Clarity	No match between program and application world views.	Some correlation between program and application .	Moderate correlation between program and application .	Good correlation between program and application .	Clear match between program and application world views.
Self - Descriptiveness	Obscure code; documentation missing, obscure or obsolete.	Some code commentary and headers; some useful documentation.	Moderate level of code commentary, headers, documentation.	Good code commentary and headers; useful documentation; some weak areas.	Self - descriptive code; documentation up-to-date, well-organized, with design rationale.
SU Increment to ESLOC	50	40	30	20	10

Other Major COCOMO II Changes

- Range versus point estimates
- Requirements Volatility (Evolution) included in Size
- Multiplicative cost driver changes
  - Product CD's
  - Platform CD's
  - Personnel CD's
  - Project CD's
- Maintenance model includes SU, UNFM factors from reuse model
  - Applied to subset of legacy code undergoing change

Post-Architecture EMs - Product:

	Very Low	Low	Nominal	High	Very High	Extra High
Required Reliability (RELY)	slight incon - venience EM = .82	low, easily recoverable losses .92	moderate, easily recoverable losses 1.0	high financial loss 1.10	risk to human life 1.26	
Database Size (DATA)		DB bytes/ Pgm SLOC< 10 .90	10 < D/P < 100 1.0	100 < D/P < 1000 1.14	D/P >1000 1.28	
Complexity (CPLX)	see Complexity Table					
Required Reuse (RUSE)		none	across project	across program	across product line	across multiple product lines
Documentation Match to Lifecycle (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	

Post-Architecture Complexity:

	Control Operations	Computational Operations	Device-dependent Operations	Data Management Operations	User Interface Management Operations
Very Low	•••	•••	•••	•••	•••
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	•••	•••	•••	•••	•••

Post-Architecture EMs - Platform:

	Very Low	Low	Nominal	High	Very High	Extra High
Execution Time Constraint (TIME)			≤ 50% use of available execution time	70%	85%	95%
Main Storage Constraint (STOR)			≤ 50% use of available storage	70%	85%	95%
Platform Volatility (PVOL)		major change every 12 mo.; minor change every 1 mo.	major: 6 mo.; minor: 2 wk.	major: 2 mo.; minor: 1 wk.	major: 2 wk.; minor: 2 days	

Post-Architecture Ems- Personnel:

	Very Low	Low	Nominal	High	Very High	Extra High
Analyst Capability (ACAP)	15th percentile	35th percentile	55th percentile	75th percentile	90th percentile	
Programmer Capability (PCAP)	15th percentile	35th percentile	55th percentile	75th percentile	90th percentile	
Personnel Continuity (PCON)	48%/year	24%/year	12%/year	6%/year	3%/year	
Application Experience (AEXP)	< 2 months	6 months	1 year	3 years	6 years	
Platform Experience (PEXP)	< 2 months	6 months	1 year	3 years	6 years	
Language and Tool Experience (LTEX)	< 2 months	6 months	1 year	3 years	6 years	

Post-Architecture EMs - Project:

	Very Low	Low	Nominal	High	Very High	Extra High
Use of Software Tools (TOOL)	edit, code, debug	simple, frontend, backend CASE, little integration	basic lifecycle tools, moderately integrated	strong, mature lifecycle tools, moderately integrated	strong, mature, proactive lifecycle tools, well integrated with processes, methods, reuse	
Multisite Development: Collocation (SITE)	International	Multi-city and Multi - company	Multi-city or Multi - company	Same city or metro. area	Same building or complex	Fully collocated
Multisite Development: Communicati ons (SITE)	Some phone, mail	Individual phone, FAX	Narrowband email	Wideband electronic communica - tion	Wideband elect. comm, occasional video conf.	Interactive multimedia
Required Development Schedule (SCED)	75% of nominal	85%	100%	130%	160%	

Early Design vs. Post-Arch EMs:

Early Design Cost Driver	Counterpart Combined Post - Architecture Cost Drivers
Product Reliability and Complexity	RELY, DATA, CPLX, DOCU
Required Reuse	RUSE
Platform Difficulty	TIME, STOR, PVOL
Personnel Capability	ACAP, PCAP, PCON
Personnel Experience	AEXP, PEXP, LTEX
Facilities	TOOL, SITE
Schedule	SCED

Other Model Refinements

- Initial Schedule Estimation

$$TDEV = \left[3.67 \times \left(\overline{PM} \right)^{(0.28+0.2\times(B-\ 0.91))} \right] \times \frac{SCED\ \%}{100}$$

where $\overline{PM}$ = estimated person months excluding Schedule multiplier effects

- Output Ranges

Stage	Optimistic Estimate	Pessimistic Estimate
Application Composition	0.50 E	2.0 E
Early Design	0.67 E	1.5 E
Post-Architecture	0.80 E	1.25 E

  - 80% confidence limits: 10% of time each below Optimistic, above Pessimistic
  - Reflect sources of uncertainty in model inputs

Outline

- Model Overview
  - Sizing, Reuse, and Scale Factors
  - Effort Multipliers
  - Other Features
- Example of Use: Demo
- Model Reinterpretation for CS 577

Example of Use: Demo

- Estimate effort and schedule to build USC COCOMO II
- Show sensitivity analysis capabilities
- Use as example of Fast Function Point sizing
  - Best sizing method for CS 577 projects

Fast Function Point Sizing

- Count number of files of different types
  - File: grouping of data elements handled similarly by software
  - External Input EI: files entering software system
  - External Output EO: files exiting software system
  - Internal Logical IL: internal files used by software system
  - External Interface EIF: files passed/shared between software systems
  - External Query EQ: input and immediate output response
- Use Average complexity weights for all files
  - FP = 4 * EI + 5 * EO + 10 * IL + 7 * EIF + 4 * EQ
- USC COCOMO II FP = 4(12) + 5(7) + 10(7) + 0 + 0 = 153
  - Java, C++ SLOC = 153(50) = 7650 SLOC
  - HTML, Power Builder = 153(20) = 3060 SLOC
  - Can use averages for mixes of languages

Outline

- Model Overview
  - Sizing, Reuse, and Scale Factors
  - Effort Multipliers
  - Other Features
- Example of Use: Demo
- Model Reinterpretation for CS 577

Using COCOMO II in CS 577

- Begin with COCOMO II estimate
  - Using Fast Function Point sizing
  - Using adjustments to CS 577 below
  - Focus on 577b Construction phase
- Cross-check with bottom-up team estimate
  - Source lines of code (SLOC)
  - Effort by activity, rough 577b milestone plan
- Adjust, try to reconcile both estimates

COCOMO II Estimates for 577b

- Disregard COCOMO II schedule estimates
- Use COCOMO II effort estimates to determine how large a team needed for 12-week fixed schedule
  - Assuming 12 hours/week of dedicated effort per person
  - Assuming 10 of the 12 weeks fill COCOMO II Construction phase (72% of total effort estimate)
  - Assuming 100 hours/person-month for COCOMO estimates
- For 577b Construction phase, these are equivalent:
  - 1 577b team member effort = (10 weeks)(12 hours/week) = 120 hours
  - 1.67*[estimated COCOMO II person month] = (1.67)(100 hours)(0.72) = 120 hours
- So, COCOMO II estimated 577b team size = 1.67*[estimated COCOMO II person months]
- Ideal COCOMO II estimate = (5 team members)(1.67) = 8.33PM