3. December 2002

seminar cost estimation W 2002/2003

COCOMO

Constructive cost model

Department of Information Technology
University of Zurich

Nancy Merlo-Schett

"To help people reason about

the cost and schedule implications

of their software decisions.“

B. Boehm 2000

OVERVIEW

overview
COCOMO I
reengineering
COCOMO II
equations
tools
conclusion

overview
COCOMO I
reengineering
COCOMO II
equations
tools
conclusion

The tree levels of the COCOMO I

The **basic** model ..	• is single-valued and static
The intermediate model ..	• computes software development effort as a function of program size and a set of **fifteen "cost drivers"** that include subjective assessments of product, hardware, personnel, and project attributes.
The advanced or detailed model ..	• incorporates all characteristics of the intermediate version with an assessment of the cost driver's impact on **each step** (analysis, design, etc.) of the software engineering process.

overview
COCOMO I
reengineering
COCOMO II
equations
tools
conclusion

Three development modes COCOMO I

Development Mode	Project Characteristics			
	Size	Innovation	Deadline/ constraints	Dev. Environment
Organic	Small	Little	Not tight	Stable
Semi-detached	Medium	Medium	Medium	Medium
Embedded	Large	Greater	Tight	Complex hardware/customer interfaces

overview
COCOMO I
reengineering
COCOMO II
equations
tools
conclusion

Two main equations COCOMO I

development effort (MM)

$MM = a * KDSI^{b}$

based on MM (**152** hours per MM) = one month of effort by one person.

effort and development time (TDEV)

$TDEV = 2.5 * MM^{c}$

The coefficents a, b and c depend on the *mode of the development*

Intermediate COCOMO I

overview
COCOMO I
reengineering
COCOMO II
equations
tools
conclusion

Intermediate COCOMO

$MM = a * KDSI^{b}$

$TDEV = 2.5 * MM^{c}$

Intermediate COCOMO	a	b	c
Organic	**3.2**	1.05	0.38
Semi-detached	**3.0**	1.12	0.35
Embedded	**2.8**	1.20	0.32

Example COCOMO I

overview
COCOMO I
reengineering
COCOMO II
equations
tools
conclusion

TDEV=2.5 * MM c

Intermediate COCOMO

MM = a * KDSI b

TDEV=2.5 * MM c

Required: database system for an office automation project.
Project = **organic** (a=3.2, b = 1.05, c=0.38),
4 modules to implement:

data entry	0.6 KDSI
data update	0.6 KDSI
query	0.8 KDSI
report generator	1.0 KDSI
System SIZE	**3.0** KDSI

Efforts are rated as follows
(all others nominal, 1.0):

complexity	**high (1.15)**
storage	**high (1.06)**
experience	**low (1.13)**
prog capabilities	**low (1.17)**

$MM_{Korr} = (1.15*1.06*1.13*1.17) * 3.2 * 3.0^{1.05}$
$MM_{Korr} = 1.61 * 3.2 * 3.17$
$MM_{Korr} = 16.33$

$TDEV = 2.5 * 16.33^{0.38}$
TDEV = 7.23 (>7months to complete)

How many people should be hired?
MM_{Korr} / TDEV = team members
16.33 / 7.23 = 2.26 (>2 team members)

overview
COCOMO I
reengineering
COCOMO II
equations
tools
conclusion

Advantages / Drawbacks COCOMO I

Advantages

transparent

Drawbacks

Information for estimating KDSI not always available

mis-classification of the development mode

historical data not always available

overview
COCOMO I
reengineering
COCOMO II
equations
tools
conclusion

Reengineering COCOMO

ReEngineering COCOMO I needs

- New software processes
- New phenomenas: size, reuse
- Need for decision making based on **incomplete information**

Focused issues are

- Non-sequential and rapid-development process models
- Reuse-driven approaches involving commercial-off-the-shelf (COTS)
- Reengineering (reuse, translated code)
- Applications composition
- Application generation capabilities
- Software process maturity effects
- Process-driven quality estimation

overview
COCOMO I
reengineering
COCOMO II
equations
tools
conclusion

Differences between COCOMO I & COCOMO II

COCOMO

COCOMO I	COCOMO II
KDSI	KSLOC
waterfall model	three phases
Point estimate	range estimate
Three development modes	five scale factors
Fifteen cost drivers	seven / seventeen cost drivers
63 data points	161 data points
reengineering	software reuse and requirement volatility

Nominal Schedule Estimation Equations COCOMO II

overview
COCOMO I
reengineering
COCOMO II
equations
tools
conclusion

$$PM_{NS} = AxSize^{E} x \prod_{i=1}^{n} EM_i$$

$$E = B + 0.01x \sum_{j=1}^{5} SF_j$$

$$TDEV_{NS} = Cx(PM_{NS})^{F}$$

$$F = D + 0.2x0.01x \sum_{j=1}^{5} SF_j$$

where
A = 2.94, B = 0.91, C = 3.67, D = 0.28

Nominal Schedule Estimation Equations COCOMO II

overview
COCOMO I
reengineering
COCOMO II
equations
tools
conclusion

$$\text{Size} = \left(1 + \frac{\text{REVL}}{100}\right) \times (\text{New KSLOC} + \text{Equivalent KSLOC})$$

$$\text{Equivalent KSLOC} = \text{Adapted KSLOC} \times \left(1 - \frac{\text{AT}}{100}\right) \times \text{AAM}$$

$$\text{where AAM} = \begin{cases} \dfrac{\text{AA} + \text{AAF} \times (1 + [0.02 \times \text{SU} \times \text{UNFM}])}{100}, & \text{for AAF} \leq 50 \\ \dfrac{\text{AA} + \text{AAF} + (\text{SU} \times \text{UNFM})}{100}, & \text{for AAF} > 50 \end{cases}$$

$$\text{AAF} = (0.4 \times \text{DM}) + (0.3 \times \text{CM}) + (0.3 \times \text{IM})$$

overview
COCOMO I
reengineering
COCOMO II
equations
tools
conclusion

Scale factors COCOMO II

Scale Factors (W_i)		annotation
PREC	If a product is similar to several previously developed project, then the precedentedness is high	replaces Development Mode, largely intrinsic to a project and uncontrollable
FLEX	Conformance needs with requirements / external interface specifications, ...	
RESL	Combines Design Thoroughness and Risk Elimination (two scale factors in Ada).	Identify management controllables by which projects can reduce diseconomies of scale by reducing sources of project turbulence, entropy and rework.
TEAM	accounts for the sources of project turbulence and entropy because of difficulties in synchronizing the project's stakeholders.	
PMAT	ratingways: 1. by the results of an organized evaluation based on the SEI CMM, 2. 18 Key Process Areas in the SEI CMM.	

overview
COCOMO I
reengineering
COCOMO II
equations
tools
conclusion

Cost Drivers COCOMO II

	Early Design cost drivers	Post-Architecture cost drivers
Product reliability and complexity	RCPX	RELY, DATA, CPLX, DOCU
Required reuse	RUSE	RUSE
Platform difficulty	PDIF	TIME, STOR, PVOL
Personnel capability	PERS	ACAP, PCAP, PCON
Personnel experience	PREX	AEXP, PEXP, LTEX
Facilities	FCIL	TOOL, SITE
Required Development Schedule	SCED	SCED

overview
COCOMO I
reengineering
COCOMO II
equations
tools
conclusion

Code Category COCOMO II

New code:	new code, from scratch
Reused code:	pre-existing code, *black-box,* applied as it is
Adapted code:	pre-existing code, *white-box*, modified
COTS component:	leased, licensed, source code is not available
Automatically translated code:	pre-existing code, translated by automated tools (keyword: reengineering/conversion)

overview
COCOMO I
reengineering
COCOMO II
equations
tools
conclusion

Comparison COCOMO II

	Early Design model	Post-Architecture model
deployment	used to make rough estimates of a project's cost and duration before its entire architecture is determined.	used after project's overall architecture is developed.
Information available	not enough	fine-grain cost estimation can be supported
cost drivers	set of **seven** cost drivers (Product Reuse, Platform, Personnel, Facilities, Schedule)	set of **seventeen** multiplicative cost drivers grouped into four categories (Product factors, Platform factors, Personnel factors, Project factors).
involves ...	exploration of alternative software/system architecture and concepts of operation	actual development and maintenance of a software product

Extensions COCOMO II

overview
COCOMO I
reengineering
COCOMO II
equations
tools
conclusion

- estimating the cost of software COTS integration (COCOTS)
- Application Composition Model
- phase distributions of schedule and effort (COPSEMO)
- rapid application development effort and schedule adjustments (CORADMO)
- quality in terms of delivered defect density (COQUALMO)
- effects of applying software productivity strategies / improvement (COPROMO)
- System Engineering (COSYSMO)

overview
COCOMO I
reengineering
COCOMO II
equations
tools
conclusion

Extensions; status quo COCOMO II

Extension	Status quo	still experimental
Application Composition Model		yes
COCOTS	not fully formulated and validated	yes
COPSEMO	about to complete Delphi round 1	yes
CORADMO	Delphi round 2 completed	yes
COQUALMO	Bayesian Analysis	

overview
COCOMO I
reengineering
COCOMO II
equations
tools
conclusion

Reengineering COCOMO

ReEngineering COCOMO I needs

- New software processes
- New phenomenas: size, reuse
- Need for decision making based on **incomplete information**

Focused issues are

- Non-sequential and rapid-development process models
- Reuse-driven approaches involving commercial-off-the-shelf (COTS)
- Reengineering (reuse, translated code)
- Applications composition
- Application generation capabilities
- Software process maturity effects
- Process-driven quality estimation

Advantages COCOMO II

overview
COCOMO I
reengineering
COCOMO II
equations
tools
conclusion

Advantages

COCOMO II is an industry standard

very profound information is easy available

clear and effective calibration process by combining delphi with algorithmic cost estimation techniques (Bayesian method)

various extension for almost every purpose are available

Tool support (also for the various extensions)

Drawbacks COCOMO II

overview
COCOMO I
reengineering
COCOMO II
equations
tools
conclusion

Drawbacks

waterfall predilection

extensions are still experimental

duration calculation for small projects is unreasonable

TOOLS COCOMO II

overview
COCOMO I
reengineering
COCOMO II
equations
tools
conclusion

OUTLOOK / CONCLUSION COCOMO II

overview
COCOMO I
reengineering
COCOMO II
equations
tools
conclusion

release a new calibration annually

calibrate the different extensions

To keep track with the future software engineering trends