CORADMO Extensions of COCOMO II Schedule Estimation Questionnaire

1. Introduction

The Center for Software Engineering at the University of Southern California is conducting research to update the software development cost estimation model called COCOMO. The project name is COCOMO II and is led by Dr. Barry W. Boehm.

A fundamental requirement for such research is real-world software development project data. This data will be used to test hypotheses and verify the model's postulations. In return the model will be open and made available to the public. The contribution of your data will ensure the final model is useful.

The data that is contributed is important to us. We will safeguard your contribution so as not to compromise company proprietary information. Some Affiliates have an active collection program, and the data from past projects is available for the COCOMO II data collection efforts. This questionnaire can be used to extract relevant CORADMO data.

This questionnaire addresses only a project level of data granularity. The project level of granularity is data that is applicable for the whole project. This includes things like application type and development activity being reported.

This questionnaire has three sections. The first section includes general and project-level COCOMO II related questions. The remaining two section are for the extensions of COCOMO II, namely CORADMO (**CO**nstructive **R**apid **A**pplication **D**evelopment **MO**del) and its pre-processor COPSEMO (**CO**nstructive **P**hased **S**chedule and **E**ffort **Mo**del). If you have not submitted regular COCOMO-II data on this project yet, a copy of the form is available from the Points of Contact identified below.

The data collection activity for the COCOMO II research effort started in November 1994. The first calibration was published in 1997 based on 83 datapoints collected. It became popular as COCOMO II.1997 and produced estimates within 30% of the actuals 52% of the time for effort. The second calibration was published in 1998 based on 161 datapoints. It is known as COCOMO II.1998 and produces estimates within 30% of the actuals 71% of the time for effort. The aim of the COCOMO II research team is to continually update the existing COCOMO II database and to publish annual calibrations of the COCOMO II model. Hence by submitting your data to us, you play a significant role in the model calibration.

COCOMO II Points of Contact

For questions on the COCOMO II Model and its extensions, data definitions, or project data collection and management, contact:

A. Winsor Brown (Research Affiliate)	Voice: (213) 740-5703, Fax: (213) 740-4927
Cyrus Fakharzadeh (Research Assistant)	Voice: (213) 740-5703, Fax: (213) 740-4927
Barry Boehm (Project Leader)	Voice: (213) 740-5703, Fax: (213) 740-4927
Internet Electronic-Mail	cocomo-info@sunset.usc.edu

COCOMO II Data Submission Address:

COCOMO II Data Submission
Center for Software Engineering
Department of Computer Science
Henri Salvatori Room 330
University of Southern California
941 W. 37th Place
Los Angeles, CA 90089-0781
U.S.A.

[1] COnstructive RAD schedule and effort MOdel

[2] Constructive Cost Modeling (COCOMO) is defined in Software Engineering Economics by Barry W. Boehm, Prentice Hall, 1981

2. Project Level Information

As described in the Introduction section of this questionnaire, project level information is applicable for the whole project. This includes things like application type and development activity being reported.

2.A. General Information

2.A.1. Affiliate Identification Number Each separate software project contributing data will have a separate file identification number of the form XXX. XXX will be one of a random set of three-digit organization identification numbers, provided by USC Center for Software Engineering to the Affiliate.

_____

2.A.2 Project Identification Number The project identification is a three digit number assigned by the organization. Only the Affiliate knows the correspondence between YYY and the actual project. The same project identification must be used with each data submission.

_____

2.A.3 Date prepared This is the date the data elements were collected for submission.

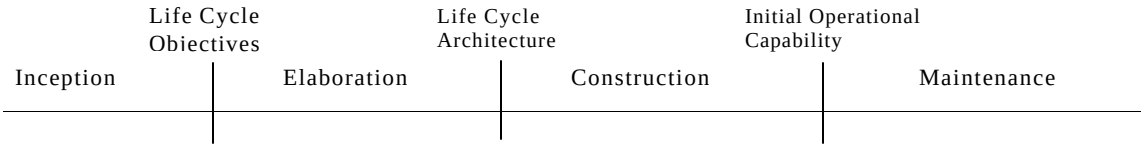
_____

2.B. Schedule Year of development. For reporting of historical data, please provide the year in which the software development was completed. For periodic reporting put the year of this submission or leave blank.

_____

2.B.1. Schedule Months. For reporting of historical data, provide the number of calendar months from the time the development began through the time it completed. For periodic reporting, provide the number of months in this development activity.

Circle the life-cycle stages that the schedule covers:

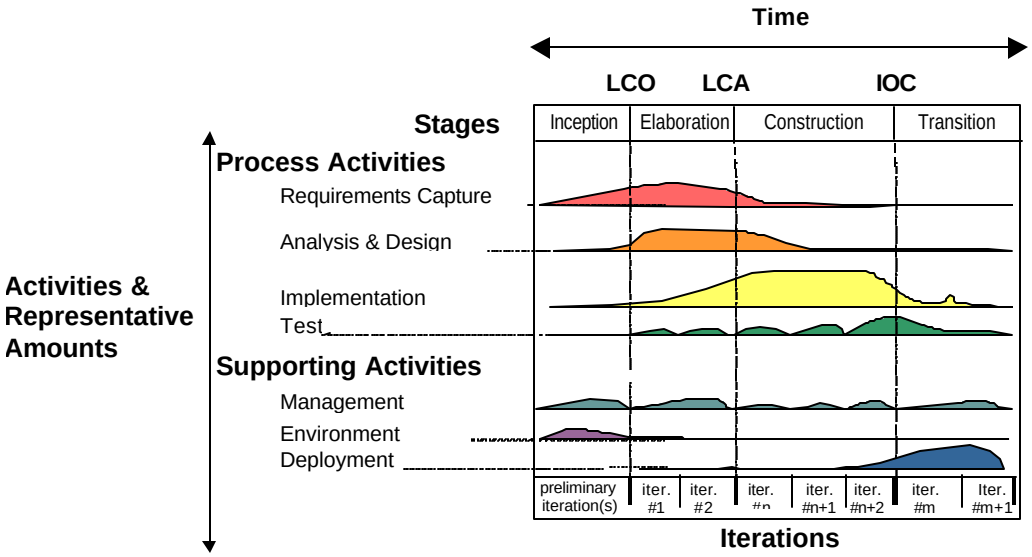


See the Appendix A for definitions of the LCO, LCA, and IOC milestones. The COCOMO II model covers the effort required from the completion of the LCO to IOC. If you are using a waterfall model, the corresponding milestones are the Software Requirements Review, Preliminary Design Review, and Software Acceptance Test.

Schedule in months: _____

3. Constructive Phased Schedule and Effort Model (COPSEMO)

COPSEMO is based on the lifecycle anchoring concepts discussed by Boehm[3]. The anchor points are defined as Life Cycle Objectives (LCO), Life Cycle Architecture (LCA), and Initial Operational Capability (IOC). An illustration from Rational Corporation[4] showing the stages around the anchor points is shown below.



The correspondence between Rational's and COPSEMO's & CORADMO's "Stages" and the life cycle anchor points is shown in the following table along with an indication of the relative amounts of the different activities.

COCOMO II Submodel Usage	**Early Design** ——— **Post-Architecture**			**Maintenance**
	LCO	**LCA**	**IOC**	
Activities \ Stage	**Inception**	**Elaboration**	**Construction**	**Transition**
Requirements Capture	Some usually	Most, peaks here	Minor	None
Analysis & Design	A little	Majority, mostly constant effort	Some	Some, for repair during ODT&E
Implementation	Practically none	Some, usually for risk reduction	Bulk; mostly constant effort	Some, for repair during ODT&E
Test	None	Some, for prototypes	Most for unit, integration and qualification test.	Some, for repaired code.

COPSEMO & CORADMO use the word "stage" so it is not confused with the classic waterfall phases: Requirements, Analysis, Design, Code, Test and Maintenance.

[3] Barry W. Boehm, "Anchoring the Software Process," *IEEE Software,* 13, 4, July 1996, pp. 73-82

[4] Rational Corp., "Rational Objectory Process 4.1 – Your UML Process", available at http://www.rational.com/support/techpapers/toratobjprcs/.

CORADMO Extensions of COCOMO II Schedule Estimation Questionnaire

COCOMO's effort and schedule estimates are focused on Elaboration and Construction (the stages between LCO and IOC. Inception corresponds to the COCOMO's "Requirements" activity in a waterfall process model, which is actually an additional (fixed percentage) effort, above and beyond the effort calculated by COCOMO. The table also indicates the areas in which the COCOMO II Early Design and Post-Architecture submodels are normally used.

Planned Allocations

3.A.1. Percentage Effort per Phase. Allocate the effort (person months) used in each of the phases as a percentage of the total effort during Elaboration and Construction. The sum of the percentages of Elaboration and Construction should be 100%. The effort during Inception (as a percentage of total Elaboration and Construction) is added to get the Total IE&C, which should be greater than 100%.

		LCO	LCA	IOC		
Phase	**Inception**	**Elaboration**	**Construction**	**Total E & C**	**Total I E & C**
%Effort				**100%**	

3.A.2. Percentage Schedule per Phase. Allocate the schedule (calendar months) for each of the phases as a percentage of the total schedule during Elaboration and Construction. The sum of Elaboration and Construction should be 100%. The schedule during Inception (as a percentage of total Elaboration and Construction) is added to get the Total IE&C, which should be greater than 100%.

		LCO	LCA	IOC		
Phase	**Inception**	**Elaboration**	**Construction**	**Total E & C**	**Total I E & C**
%Schedule				**100%**	

3.A.3. Person-Power per Phase. Indicate the average number of people actually working during this period of each of the phases. If the loading was not approximately constant during the period except for typical, limited ramp-ups, please indicate the degree of variation by providing the Persons-Max and Persons-Min, and the number of months with that number of people (max and min, respectively). NOTE: summing persons across stages is illogical and incorrect.

			LCO	LCA	IOC			
Phase	**Inception**		**Elaboration**		**Construction**		**Total E & C**	**Total I E & C**
Persons-Ave.							**X**	**X**
	Heads	Mon.	Heads	Mon.	Heads	Mon.	**X**	**X**
Persons-Max							**X**	**X**
Persons-Min							**X**	**X**

4. Constructive Rapid Application Development Model (CORADMO)

The Constructive Rapid Application Development Model has its roots in the results of a 1997 CSE Focused Workshop on Rapid Application Development[5]. RAD is taken to mean an application of any of a number of techniques or strategies to reduce software development cycle time. A "RAD Opportunity Tree" presented at the workshop identified five classes of strategies whose degree of implementation can be used to parameterize a schedule estimate given an effort estimate produced by COCOMO II-2000. These strategies are preferable to just adding people to the task. The six classes are: development process re-engineering (DPRS), rapid prototyping (RPRO), collaboration support (CLAB), architecture and risk resolution (RESL), pre-positioning of assets (PPOS) and RAD capability of personnel (RCAP). RESL corresponds to the COCOMO II scale driver; the other five are new. All have their effects reflected as multipliers on effort (person months, PM), schedule (months, M) and/or number of personnel (P). Person months of effort can actually be increased because certain pro-active strategies, like pre-positioning of assets, are only possible with extra effort.

CORADMO utilizes the COCOMO extension that allocates effort and schedule to the phases, which are anchored at the LCO/LCA/IOC points in a development life cycle. A phased schedule and effort distribution is needed because the effects of the RAD strategies identified above are different for the different stages. Also, a new mathematical function is used to calculate (predict) the calendar months for a given amount of effort: the function is only radically different in low (under 16) person-month's efforts where it seems more normal have an equal number of people and months to accomplish the task. At the higher (greater than 120) person-month's efforts, the traditional COCOMO II-2000 function is used which is based on the traditional cube-root-like function of effort. A smooth curve is fit within these ranges.

The intent of CORADMO is to calculate/predict the schedule (months, M), personnel (P), and adjusted effort (person-months, PM) based on the distribution of effort and schedule to the various phases, and impacts of the selected schedule driver ratings on the M, P, and PM of each phases.

4.A.1. Rapid Prototyping (RPRO). The first driver for CORADMO is RPRO (Rapid Prototyping) that expresses the degree to which the personnel have experience in Rapid Prototyping. This driver reflects schedule compression in Inception and Elaboration stages due to faster prototyping or option exploration. For this driver, the effort compression is hypothesized to be the same as the schedule compression; that is, the team size would stay the same over a shorter period. The rating for this driver depends on the amount of Rapid Prototyping Experience the development team has had in the domain of the project being evaluated. Since the rating applies to the team, it must include the experience of the managers and team leaders and their experience takes precedence over the average of the rest of the team working in the Inception and Elaboration phases. Below is the rating scale for this driver.

Very Low	Low	Nominal	High	Very High	Don't Know	N/A
None	On average, personnel have experience on less than one recent project using Rapid Prototyping	Most personnel have worked on more than one project using Rapid Prototyping	On average, personnel have worked on more than two projects using Rapid Prototyping	All personnel have worked on at least three projects using Rapid Prototyping		

[5] B. Boehm, S. Chulani, and A. Eyed, "Knowledge Summary: USC-CSE Focused Workshop on Rapid Application Development", USC-CSE Technical Report, June 1997.

4.A.2. Development Process Reengineering and Streamlining (DPRS). The degree to which the project and organization allow and encourage streamlined or re-engineered development processes: the current level of bureaucracy is a clear indicator. The schedule compression or expansion, because of this driver, doesn't alter staff level (P). The following table can be used to make a subjective average to determine the level of bureaucracy.

	VL	L	N	H	VH
Number of approvals required per task	Excessive	Occasionally Reduced	Mature	Actively Reduced	Actively Minimized
Time taken per approval	Excessive	Occasionally Reduced	Mature	Actively Reduced	Actively Minimized
Reduced task dependencies, critical path tasks	None	Little	Mature Tech. Adopted	Advanced Tech. Adopted	Pioneering
Follow-up to expedite task completion	None	Little	Encouraged	Emphasized	Strongly Emphasized
Process measurement & streamlining	None	Little	Mature Tech. Adopted	Advanced Tech. Adopted	Pioneering
	Heavily Bureaucratic	Bureaucratic	Basic good business practices	Partly streamlined	Fully streamlined

Very Low	Low	Nominal	High	Very High	Don't Know	N/A
Heavily Bureaucratic	Bureaucratic	Basic good business practices	Partly streamlined	Fully streamlined		

4.A.3. Collaboration Support and Experience (CLAB). Teams and team members who can collaborate effectively can reduce both effort and schedule; those that don't collaborate effectively have increased schedule and effort (due to wasted time). With this multiplier, staff level does not change based on collaboration support.

4.A.3.1. Collaboration Tools. Collaboration support is impacted by TOOL, but only for tools that support or enable collaboration. However, the tool technology impact is lessened in the case of a co-located team with high experience ratings (PREX, the combination of application, platform, language and tool experience taken from the early design ratings).

Collaboration Tools	Very Low	Low	Nominal	High	Very High
	None special	Common-access project files	Collaborative tool exercise: NetMeeting	Basic special collaboration support, e.g. GroupSystems.com	Advanced collaboration support: domain aids, agents

4.A.3.2. Multisite Development (SITE). Collaboration support is clearly impacted by SITE ratings. Given the increasing frequency of multisite developments, and indications that multisite development effects are significant, the SITE cost driver has been added in COCOMO II. Determining its cost driver rating involves the assessment and averaging of two factors: site collocation (from fully collocated to international distribution) and communication support (from surface mail and some phone access to full interactive multimedia). See the User's Manual.

	Very Low	Low	Nominal	High	Very High	Extra High
SITE: Collocation	Inter-national	Multi-city and Multi-company	Multi-city or Multi-company	Same city or metro area	Same building or complex	Fully collocated
SITE: Communications	Some phone, mail	Individual phone, FAX	Narrowband email	Wideband electronic communication	Wideband electronic communication, occasional video conferencing	Interactive multimedia
Average of above two ratings						

The rating scale for this driver is the fuzzy average of the Collaboration Tools and Multisite Development.

Very Low	Low	Nominal	High	Very High	Extra High	Don't Know

4.A.4. Architecture & Risk Resolution (RESL). This rating is exactly the same as the COCOMO II RESL rating. The architecture portion enables parallel construction, thus reducing schedule during the construction phase assuming that staff level increases during construction while applying the same effort. Good risk resolution in a schedule driven development effort applying RAD strategies increases the probability of the strategies' success.

Very Low	Low	Nominal	High	Very High	Extra High	Don't Know	N/A
<== Use COCOMO II's RESL Rating Level ==>							

4.A.5. Prepositioning Assets (PPOS). This driver assesses the degree to which assets are pre-tailored to a project and furnished to the project for use on demand. This clearly has impacts from people skills and team building. The assets that are being pre-positioned include processes and tools, and architecture and componentry.

In order to take advantage of PPOS, the organization must either be taking a product-line approach or have made a 3, 6 or 10% pre-Inception effort investment!

Nominal	High	Very High	Extra High	Don't Know	N/A
Basic project legacy, no tailoring	Some prepositioning & tailoring	Key items prepositioned & tailored	All items prepositioned & tailored		

4.A.6 RAD Capability of Personnel (RCAP). This accounts for the effects of RAD personnel capability and experience in Rapid Application Development projects. This driver also includes the TEAM effect.

CORADMO Extensions of COCOMO II Schedule Estimation Questionnaire

4.A.6.1 Team Cohesion (TEAM). The Team Cohesion cost driver accounts for the sources of project turbulence and extra effort due to difficulties in synchronizing the project's stakeholders: users, customers, developers, maintainers, interfacers, others. See the Model Definition Manual for more details.

Very Low	Low	Nominal	High	Very High	Extra High	Don't Know	N/A
Very difficult interactions	Some difficult interactions	Basically cooperative interactions	Largely cooperative	Highly cooperative	Seamless interactions		

4.A.6.2 PERS-R and PREX-R. PERS-R is the Early Design Capability rating, adjusted to reflect the performers' capability to rapidly assimilate new concepts and material, and to rapidly adapt to change. PREX-R is the Early Design Personnel Experience rating, adjusted to reflect the performers' experience with RAD languages, tools, components, and COTS integration. Below is the rating scale for these ratings.

FACTOR	Very Low	Low	Nominal	High	Very High	Extra High
PERS-R	25th percentile	40th percentile	55th percentile	70th percentile	85th percentile	95th percentile
PREX-R	≤2 months	4 months	6 months	1 year	3 years	6 years

The rating scale for this driver is the fuzzy average of TEAM and the rating scale above.

Very Low	Low	Nominal	High	Very High	Extra High	Don't Know