CORADMO Summary

A. Winsor Brown

Abstract

*The **Co**nstructive **RAD Mo**del (CORADMO) is currently implemented in two parts: a front end phase schedule and effort model,COCOMO II**P**hase **S**chedule and **E**ffort **MO**DEL (COPSEMO), and a back end RAD model. COPSEMO uses a different schedule estimation calculation than COCOMO II's simple one: a more complex calculation for the low effort situations, those below 64 person-months. At this time, there are no other COPSEMO "drivers" besides COCOMO II's calculated effort. The RAD model has its roots in the results of a 1997 CSE Focused Workshop on Rapid Application Development[1]. RAD is taken to mean application of any of a number of techniques or strategies to reduce software development cycle time. Five classes of strategies whose degree of implementation can be used to parameterize a schedule estimate given an effort estimate produced byCOCOMO II2000 were derived from the Focused Workshop's results. These strategies, which are over and above just adding people to the task, include development process re-engineering (DPRS), re-use and very high level languages (RVHL), collaboration efficiency (CLAB), architecture investment and risk Resolution (RESL), and pre-positioning of assets (PPOS).*

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

CORADMO Summary

Table of Contents

Table of Figures

Table of Tables

CORADMO Summary

1. Introduction

The evolution of CORADMO and its companion/pre-processor model COPSEMO has its roots in several activities undertaken by the Center for Software Engineering: COCOMO II, and a Rapid Application Development Focused Workshop.

1.1. Another step in the evolution of COCOMO II

TheCOCOMO IIModel Manual provides the primary motive for this extension of COCOMO II.
"AsCOCOMO IIevolves, it will have a more extensive schedule estimation model, reflecting the different classes of process model a project can use; the effects of reusable and COTS software; and the effects of applications composition capabilities."

1.2. COCOMO II Schedule

TheCOCOMO IIschedule, as presently implemented (COCOMO II.2000) reflects a waterfall process model, and not any of the currently accepted alternatives such as iterative, spiral or evolutionary. In addition, it has been observed that the COCOMO II's duration calculation seems unreasonable for small projects, those with effort under two person years. Obviously,COCOMO IIdoes not address any of the Rapid Application Development (RAD) strategies that are being employed to reduce schedule and sometimes effort as well.

1.3. COCOMO II Constructive Phase Schedule & Effort Model and Constructive RAD Schedule Estimation Model

In an effort to overcome these shortfalls, two extensions have been developed:
theCOCOMO IIPhase Schedule & Effort Model (COPSEMO) and the
Constructive RAD schedule estimation Model CoRADMo.

2. Improving the Classic COCOMO Model for Schedule

The classic COCOMO model has deficiencies in several areas: a waterfall predilection, no drivers reflecting modern schedule reduction efforts, and small-effort projects.

2.1. New Drivers

In CSE's Focussed Workshop #9 on RAD, a RAD Opportunity Tree of strategies was presented. The strategies included some techniques that were already covered by the drivers ofCOCOMO IIas well as several that were not. An analysis of these new drivers produced a set of five drivers that reflect identifiable behavioral characteristics. These were

1. Reuse and Very High-level Languages (RVHL)
2. Development Process Reengineering (DPRS)
3. Collaboration Efficiency (CLAB)
4. Architecture, Risk Resolution (RESL)
5. Prepositioning Assets (PPOS)

These new drivers are reflected in the annotated "RAD Opportunity Tree " shown in Figure 1.

CORADMO Summary

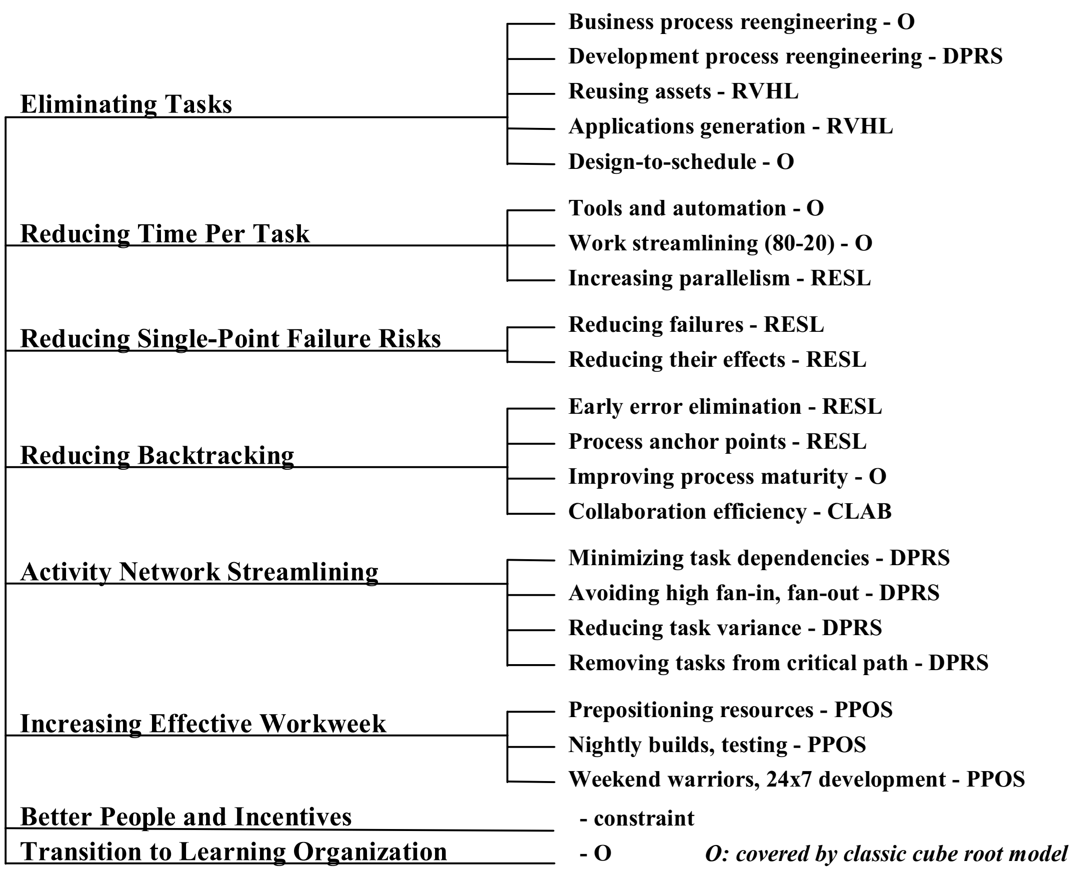

Figure 1. Annotated RAD Opportunity Tree

2.2. Duration Calculation

TheCOCOMO IIschedule, as presently implemented (in COCOMO II.2000) reflects a waterfall process model and its duration calculation seems unreasonable for small projects, those with effort under two person years.

2.2.1 COCOMO II Duration Calculation

TheCOCOMO IIduration calculation is based on an equation that has demonstrated historical accuracy, at least for large projects.

$$\text{Months} \sim 3\sqrt[3]{\text{Person-Months}}$$

This model component completely breaks down at very low efforts (16 person-months of effort) and is very questionable below a few person-years of effort.

2.2.2 COPSEMO Duration Calculation

COCOMO's effort and schedule estimates are focused on Elaboration and Construction (the Phases between LCO and IOC. Inception corresponds to the COCOMO's "Requirements" activity, which is actually an additional (fixed percentage) effort, beyond the effort calculated by COCOMO.

CORADMO Summary

Another important difference of COPSEMO's schedule estimation from COCOMO II's simple schedule estimation is the use of a more complex calculation for the low effort situations. The initialCOCOMO IIbaseline schedule equation is

$$TDEV= (3.67 * PMbar^{(0.28 + 0.2 * (B-0.91))} * SCED\%/100$$

where *TDEV* is the calendar time in months from the determination of a product's requirements baseline to the completion of an acceptance activity certifying that the product satisfies its requirements. *PMbar* is the estimated person-months excluding the SCED effort multiplier, *B* is the sum of project scale factors (discussed in the next chapter) and SCED% is the compression / expansion percentage in the SCED effort multiplier.

The TDEV calculations mean that the calculated schedule is related, approximately, to three times the cube root of the effort. For low-effort situations, especially below twenty-seven (27) person months, this yields a very pessimistic and unlikely duration of nine (9) months applying three (3) FSP people. Therefore, a new baseline schedule equation for efforts below 16 months has been chosen which is based on the square root of the effort, yielding equal FSPs and schedule months. A linear interpolation is used between the high-end applicability of 64 person months (which corresponds to a schedule of 14.4 months for a 100KSLOC EHART using 1998 average driver values), and the low end point of 16 person months.

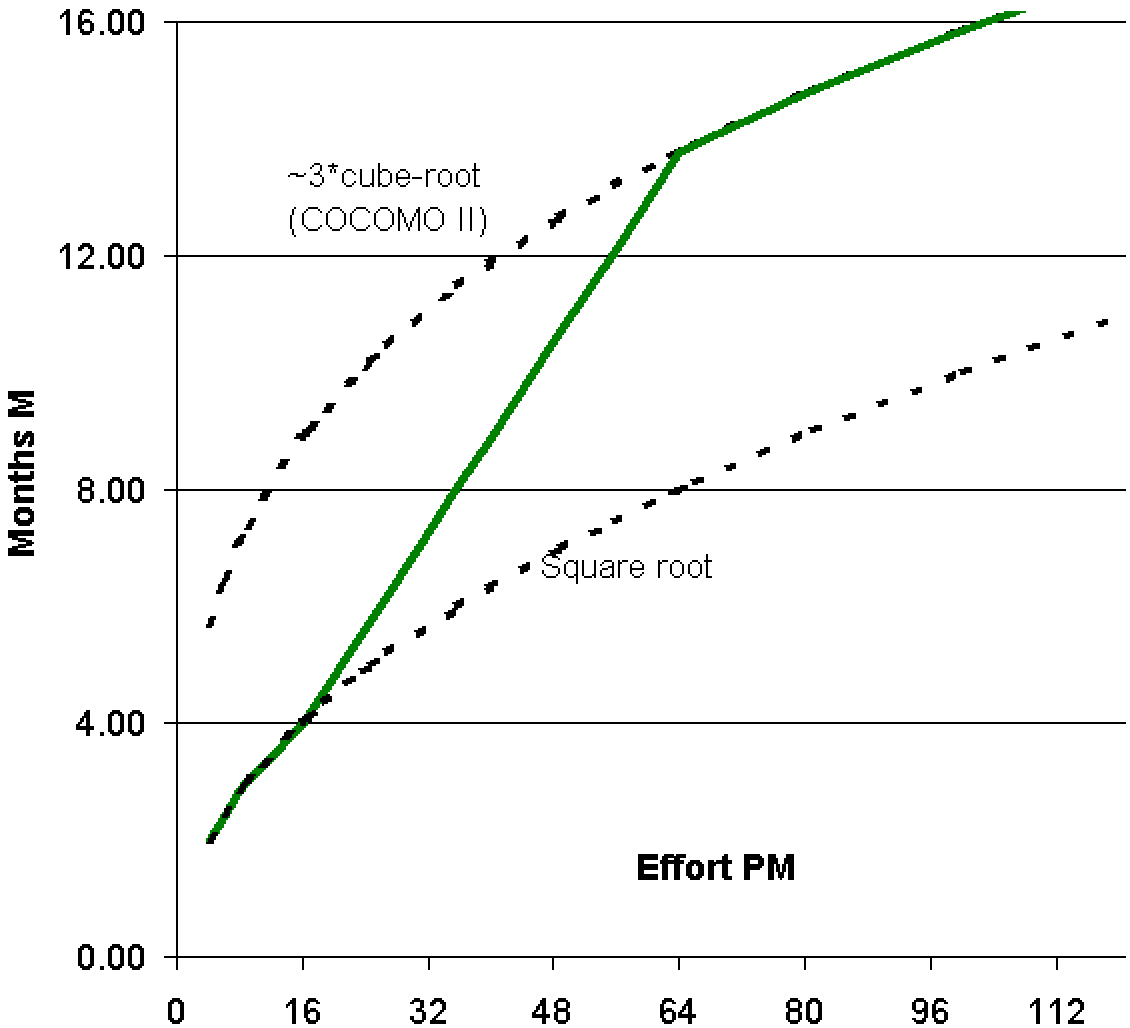

Figure 2.COCOMO IISchedule Estimate vs. COPSEMO Schedule Estimate

2.3. Process Model

The COPSEMO model is based on the lifecycle anchoring concepts discussed by Boehm[2]. The anchor points are defined as Life Cycle Objectives (LCO), Life Cycle Architecture (LCA), and Initial Operational Capability (IOC). An augmented illustration based on one from the Rational Corporation[3], Figure 3, shows the phases around the anchor points.

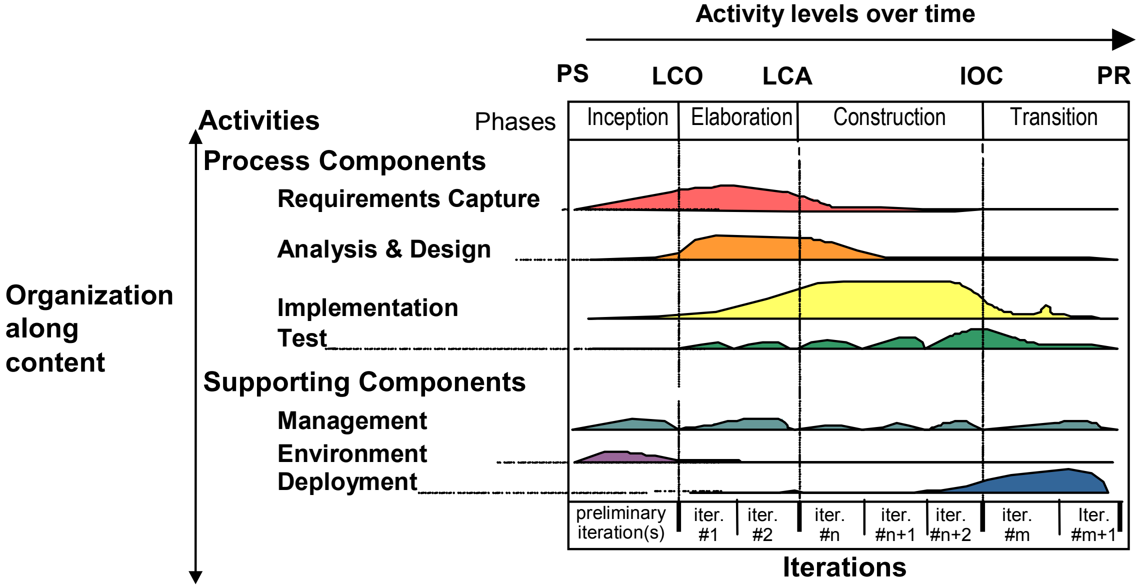

Figure 3. A modern lifecycle model with anchor points

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

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

2.4. Anchor Points, Phases and Activities

The diagram shows various activities, and implies iterations and the relative effort and duration of typical cycles within an iteration. The following table provides some more detail on the relative proportion of the activities, and some details.

COCOMO II Submodel Usage	Early Design		Post-Architecture	Maintenance
	LCO	LCA		IOC
Activities / Phase	**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 test, integration test and qualification test.	Some, for repaired code.

Table 1. Phases, Anchor Points, and relative amount and kind of Activities

3. Model Overview

There are two parts of the current model, COPSEMO and CORADMO. They both assume that data is available from aCOCOMO IImodel.

3.1. COCOMO II Constructive Phase Schedule & Effort Model (COPSEMO)

The COPSEMO part of the model currently has no drivers, per se. The model does allow for the specification of the percentages of effort and schedule to be applied to the different phases: Inception, Elaboration and Construction. The predicted effort and schedule from aCOCOMO IIrun correspond to the sum of the Elaboration and Construction phases' effort and schedule, respectively. The percentages of effort and schedule Elaboration and Construction phases thus total 100% and are used to distribute the sum accordingly. The percentages of effort and schedule for the Inception phase are also applied to theCOCOMO IIrun's effort and schedule, respectively. Thus, the sum of the effort or schedule for three phases can actually total more than 100% of theCOCOMO IIrun's effort and schedule.

3.2. Constructive RAD Schedule Estimation Model (CORADMO)

The CORADMO model has five drivers. Each driver has both rating levels, which are selected by a user based on the characteristics of the software project, its development organization, and its milieu. There are numeric schedule and effort multiplier values per phase for each rating level. The rating levels are described in detail in Part 2 of this report, which corresponds to a subset of the information gathering worksheet for users of the model and its tools. The rating levels and their corresponding numerical values are summarized below.

3.2.1 Reuse and VHLLs (RVHL)

The impact of re-use of 3GL production code is handled directly in theCOCOMO IImodel via the re-use sub-model and its effect on size. This CORADMO driver reflects the impact of re-use of code (other than production code) and/or the use of very high level languages, especially during the Inception and Elaboration phases. Higher rating levels reflect the potential schedule compression impacts in Inception and Elaboration phases due to faster prototyping, and option exploration. Clearly, this impact will be dependent on the level of capability and experience in doing this, such as Rapid Prototyping experience. The values of the multipliers corresponding to the rating levels are the same for both effort and schedule; this implies that the staff level (number of full time software personnel) is held constant.

3.2.2 Development Process Reengineering and Streamlining (DPRS)

The schedule impact of this driver reflects the inverse of the level of bureaucracy in which the developers must operate. More succinctly stated, this driver captures the degree to which the project and organization allow and encourage streamlined or re-engineered development processes. A detailed rating level scale is provided for this driver (see Part 3 of this report). The values of the multipliers corresponding to the rating levels are the same for both effort and schedule; this implies that the staff level (number of full time software personnel) is held constant.

3.2.3 Collaboration Efficiency (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). Rather than invent a new behavioral characteristic, this driver's rating level is primarily determined by an appropriate combination ofCOCOMO IIPost-Architecture SITE and TEAM driver ratings and the PREX Early Design driver ratings. The SITE rating needs to be augmented by the team's collaboration tool maturity and experience. The effects of collaboration tools are expected to help in domain analysis, option analysis, and negotiation. A detailed rating level process and scale is provided for this driver (see Part 3 of this report). The values of the multipliers corresponding to the rating levels are the same for both effort and schedule; this implies that the staff level (number of full time software personnel) is held constant.

3.2.4 Architecture / Risk Resolution (RESL)

TheCOCOMO IIArchitecture / Risk Resolution driver (RESL) enables parallel construction activities without theCOCOMO IIassumed effect of increased integration and testing costs. There is not any impact on the effort or schedule in the Inception and Elaboration phases. There is no change in effort because of RESL, only potential for schedule compression at higher RESL ratings. For this driver to be effective, it is assumed that a higher level of staffing is available and used during construction. Thus, the multipliers corresponding to the rating levels are not the same for both effort and schedule.

3.2.5 Prepositioning Assets (PPOS)

This driver reflects the degree to which assets are pre-tailored to a project or physically pre-positioned and furnished to the project for use on demand. The assets include skilled or particularly knowledgeable, people's skill-level increases, and pro-active team building. The assets that are being pre-positioned also 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! PPOS multipliers reflect the increased effort associated with the pre-positioning activities as well as the corresponding decrease in schedule and increased personnel required.

4. Implementation Models

There are three implementations of the CORADMO/COPSEMO model at this time. The logical implementation model shows how the various drivers and models interrelate. The physical implementation model shows how the logical implementation model has been realized in spreadsheet models. The three models are shown below.

4.1. LogicalCOCOMO IIRAD Extension

Figure 4 shows a conceptual logical block diagram for implementation of the RAD Model. It assumes that the regularCOCOMO IIimplementation is extended with phase distributions which are potential driven by language level (e.g., 3GL or 4GL), experience, etc. The output ofCOCOMO IIis used as a baseline for effort and schedule by the RAD Extension. The phase distributions extension allocates the baseline effort and schedule by phase. The

RAD extension itself is controlled by the five drivers (discussed in section 3), resulting in the RAD effort and schedule by phase.

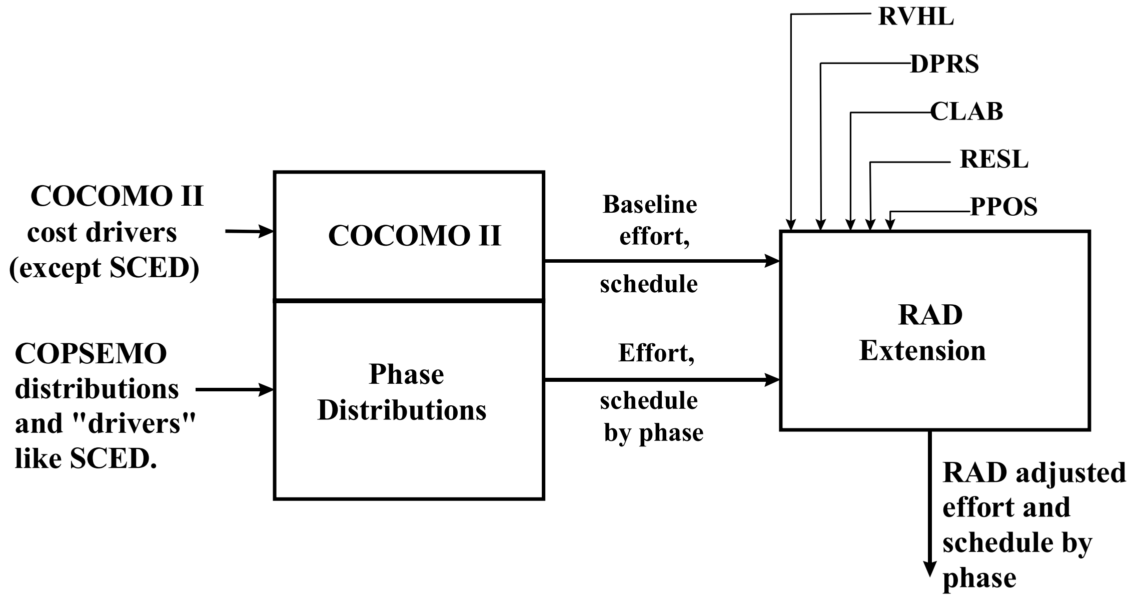

Figure 4. Logical Implementation Model

4.2. PhysicalCOCOMO IIRAD Extension

Figure 5 shows the shows the current implementation strategy for theCOCOMO IIRAD extension. The upper left box represents the COCOMO II.2000 model as implemented byCOCOMOII.exe, self-identified as "COCOMO II.2000.0" in its "About USC-COCOMO II" dialog box. Also part of theCOCOMO IIimplementation suite is a spreadsheet called COCOMOII_charts.xls which is designed to import two CSV files that can be exported fromCOCOMOII.exe and make their information available in spread sheet form (it also generates many useful charts and graphs of the data). The baseline effort and schedule as well as the values for all the drivers are acquired the COPSEMO Extension by links to the COCOMOII_charts.xls spreadsheet. The COPSEMO Extension, which is actually implemented as part of the RAD extension (CoRADMo.xls) distributes the effort (with no SCED impact) and schedule for subsequent operation by the RAD extension proper. Only the five new RAD drivers need to be input into the RAD extension: RESL is actually acquired from the COCOMOII_charts.xls spreadsheet via links, although that value can be over-ridden by the user.

CORADMO Summary

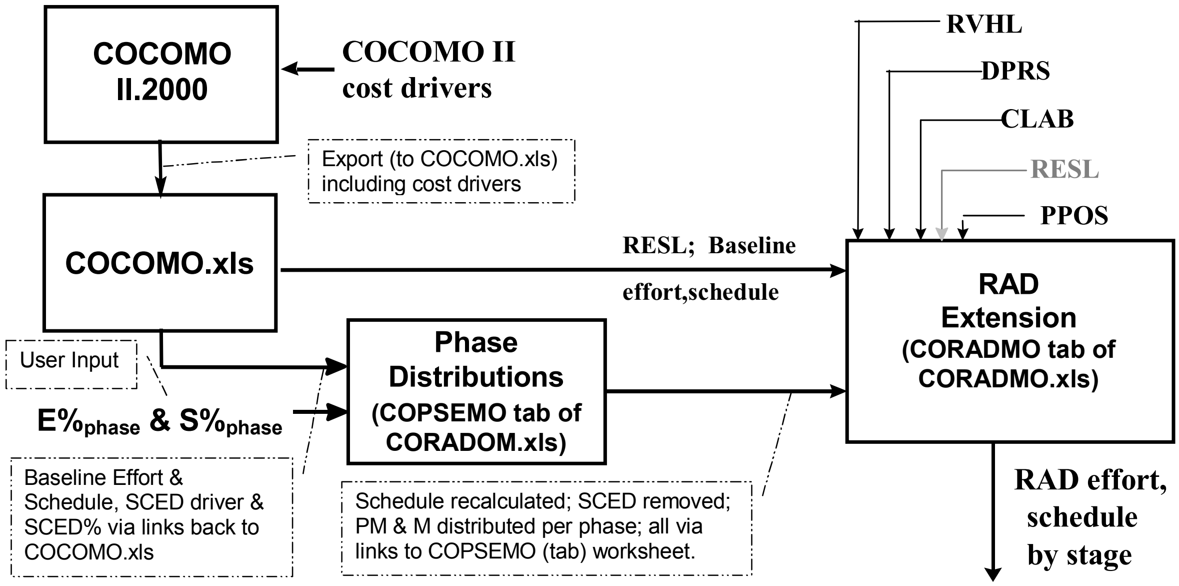

Figure 5. Physical Implementation Model

4.3. Stand-alone Spreadsheet Implementation

Figure 6 and Figure 7 contain a stand-alone implementation of the COPSEMO and CORADMO extensions.

COPSEMO

Step **NOTE: BOLD** implies a required value. ***Italic*** implies an optional value

1.0 ....F Currently implemented only for projects (not modules)

Get COCOMO II.2000 data and adjustments

2.0 ***Project: 5KNomNomNom*** **TotalSize= 5,000** **PM_C= 17.3** **M_C= 9.08**

(automatically from COCOMO_charts.xls) **SCED_R= N** **SCEDinc: 0%** **SCEDV= 1.00** P_C= 1.9013

Work hours per months (non-overhead or "billable"; COCOMO-II.2000 default= 152)= **152**

3.0 Calculate Baseline values: PM_BS by adjusting work hours per months; M_BS by applying COPSEMO's M=f(PM) to PM_BS; and P_BS from PM_BS/M_BS.

Calculate Months_Base-Schedule (M_BS) as function of COCOMO II.2000 calculated effort (PM_CnoSCED)

If PM_C< 16 then M_BS = Sqrt(PM_CnoSCED); and therefore P=M

If 16<=PM_C<64 then M_BS=(Mof64-4)/48*PMnoSCED)+(4-16*(Mof64-4)/48)

If PM=>64 then M_BS =3.67*(PM_C^(0.28+0.2*(B_C-0.91)))*SCEDph/100

M_BS= =IF(PMnoSCED<16.001,SQRT(PMnoSCED),IF(PMnoSCED<64,((Mof64-4)/48*PMnoSCED)+(4-16*(Mof64-4)/48),3.67*POWER(PMnoSCED,(0.28+0.2*(B-0.91)))*SCEDph/100))

PM_BS= 17.3 M_BS= 4.26 P_BS= 4.055

4.0 Get Effort % & Schedule % per stage

Default values: *Italic*

	Inception	Elaboration	Construction	Transition	Total E&C	Total
Effort %	*6.0* **6.0**	*24.0* **24.0**	*76.0* **76.0**	*12* **12.0**	100.0	118.0
Schedule %	*12.5* **12.5**	*37.5* **37.5**	*62.5* **62.5**	*12.5* **12.5**	100.0	125.0
P/Ave(P)	0.48	0.64	1.22	0.96	1.00	Does not apply
	PM / M : P	PM / M : P	PM / M : P	PM / M : P	PM / M : P	PM / M : P-ave
PSE Distributed	1.04 / 0.53 : 1.95	4.14 / 1.60 : 2.60	13.12 / 2.66 : 4.93	2.07 / 0.53 : 3.89	**17.3** / **4.26** : 4.05	20.4 / 5.32 : 3.83

Ave(P) refers to the average number of persons on the project; it is the same as PM_BS/M_BS for the entire project, and each stage's P/Ave(P) is the same as stage's Effort%/Schedule%.

5.0 Plot of P vs. M. "Input" values: **BOLD**

	Inception	Elaboration	Construction	Transition	Total E&C	Total
Effort %	**6.0**	**24.0**	**76.0**	12.0	100.0	106.0
Schedule %	**12.5**	**37.5**	**62.5**	12.5	100.0	112.5
P/Ave(P)	0.48	0.64	1.22	1.00	1.00	
	PM / M : P	PM / M : P	PM / M : P	PM / M : P	PM / M : P	PM / M : P-ave
PSE Distributed	1.04 / 0.53 : 1.95	4.14 / 1.60 : 2.60	13.12 / 2.66 : 4.93	2.07 / 0.53 : 3.89	**17.3** / **4.3** : 4.1	20.4 / 5.32 : 3.83

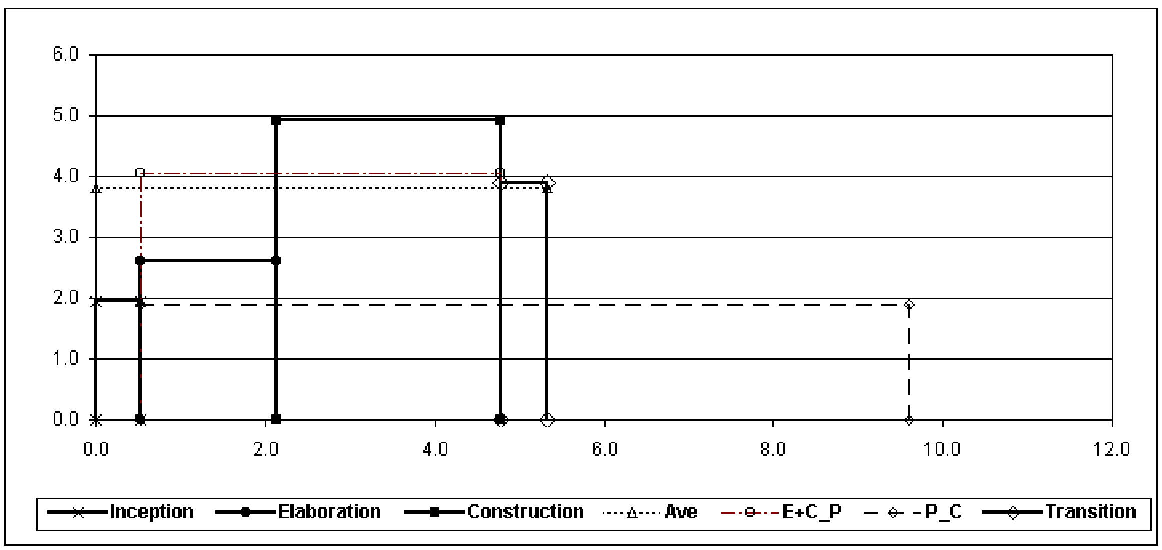

Figure 6. The COPSEMO extension portion of CORADMO.xls

CORADMO Summary

CORADMO . Currently implemented only for projects (not modules)

Step **BOLD: required values carried forward from COPSEMO;** ***Italic: optional values carried forward from COPSEMO***

1.0 Get COCOMO II.2000 data and adjustments from COPSEMO

Project:* *5KNomNomNom **TotalSize= 5,000** **PM_C= 17.3** **M_C= 9.08** P_C= 1.901

Including Schedule parameters **SCED_R= N** **SCEDinc: 0%** **SCEDV= 1.00**

Including Scale Factor Ratings **PREC_R: N** **FLEX_R: N** **RESL_R: N** **TEAM_R: N** **PMAT_R: N**

2.0 Get COPSEMO Distribution information: Values specified or Calculated in COPSEMO. Baseline/Input vales: **BOLD**

Eff% & Sched % per stage (per CoPSEMo)

	Inception	Elaboration	Construction	Total E&C	Total
Effort %	**6.0**	**24.0**	**76.0**	100.0	118.0
Schedule %	**12.5**	**37.5**	**62.5**	100.0	125.0
P/Ave(P)	0.48	0.64	1.22	1.00	Does not apply
	PM / M = P	PM / M = P	PM / M = P	PM / M = P	PM / M = P-ave
PSE Distributed	1.04 / 0.53 = 1.95	4.14 / 1.60 = 2.60	13.12 / 2.66 = 4.93	**17.3** / **4.26** = 4.05	20.4 / 5.32 = 3.83

3.0 Get the Schedule Multipliers values.

RVHL N | DPRS N | CLAB N | RESL N | PPOS N

		Inception			Elaboration			Construction		
		PM	/ M	= P	PM	/ M	= P	PM	/ M	= P
N	RVHL	0.980	0.980	1.000	0.990	0.990	1.000	1.000	1.000	1.000
N	DPRS	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
N	CLAB	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
N	RESL	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
N	PPOS	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
	Π	0.980	0.980	1.000	0.990	0.990	1.000	1.000	1.000	1.000

4.0 Apply the product of user selected Schedule and Effort Multipliers to each PM, M and P in each stage.

Input vales: **BOLD**

	Inception	Elaboration	Construction	Total E&C	Total
	PM / M = P	PM / M = P	PM / M = P	PM / M = P	PM / M = P-ave
PSE Distributed	1.04 / 0.53 = 1.95	4.14 / 1.60 = 2.60	13.12 / 2.66 = 4.93	**17.3** / **4.3** = 4.1	18.3 / 4.8 = 3.8
Π	0.98 0.98 1.00	0.99 0.99 1.00	1.00 1.00 1.00		
RAD Eff&Schd	1.01 / 0.52 = 1.95	4.10 / 1.58 = 2.60	13.12 / 2.66 = 4.93	17.2 / 4.2 = 4.1	18.2 / 4.8 = 3.8

Ave(P) refers to the average number of persons on the project: in the absence of Schedule Multiplier effects.
it is the same as PM_BS/M_BS for the entire project, and each stage's P/Ave(P) is the same as stage's Effort%/Schedule%.

5.0 Plot of P vs M. Input values in **BOLD**

RVHL= **N** DPRS= **N** CLAB= **N** RESL= **N** PPOS= **N**

	Inception	Elaboration	Construction	Total E&C	Total
Effort %	**6.0**	**24.0**	**76.0**	100.0	106.0
Schedule	**12.5**	**37.5**	**62.5**	100.0	112.5
P/Ave(P)	0.48	0.64	1.22	1.00	
	PM / M = P	PM / M = P	PM / M = P	PM / M = P	PM / M = P-ave
PSE Distributed	1.04 / 0.53 = 1.95	4.14 / 1.60 = 2.60	13.12 / 2.66 = 4.93	**17.3** / **4.3** = 4.1	18.3 / 4.8 = 3.8
Π	0.98 0.98 1.00	0.99 0.99 1.00	1.00 1.00 1.00		
RAD Eff&Schd	1.01 / 0.52 = 1.95	4.10 / 1.58 = 2.60	13.12 / 2.66 = 4.93	17.2 / 4.2 = 4.1	18.2 / 4.8 = 3.8

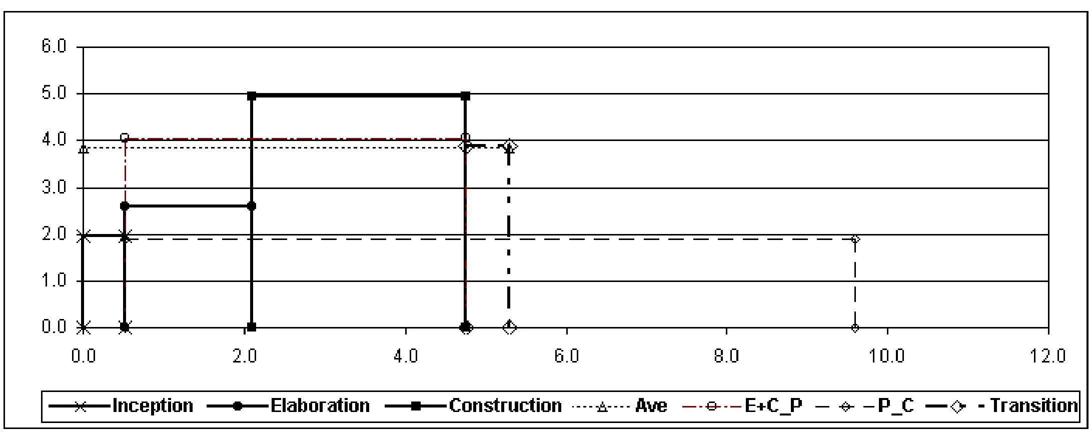

Figure 7. The RAD extension calculation and display of Schedule and Effort