

CORADMO Summary

A. Winsor Brown

Abstract

*The **COCOMO RAD MODEL** (CORADMO) is currently implemented in two parts: a front end staged schedule and effort model, **COCOMO Staged Schedule and Effort MODEL** (COSSEMO), and a back end RAD model. COSSEMO's uses a different schedule estimation calculation than COCOMO II's simple one: COSSEMO's schedule estimation uses a more complex calculation for the low effort situations, those below 64 person-months. At this time there are no other COSSEMO "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 Development¹. 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 by COCOMOII-1998 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

1. Introduction	2
1.1. Another step in the evolution of COCOMO-II.....	3
1.2. COCOMO II Schedule.....	3
1.3. COCOMO-II Constructive Staged Schedule & Effort Model and Constructive RAD Schedule Estimation Model	3
2. Improving the Classic CoCoMo Model for Schedule.....	3
2.1. New Drivers.....	3
2.2. Duration Calculation.....	4
2.2.1 COCOMO II Duration Calculation	4
2.2.2 COSSEMO Duration Calculation.....	4
2.3. Process Model	7
2.4. Anchor Points, Stages and Activities.....	8
3. Model Overview.....	8
3.1. COCOMO II Constructive Staged Schedule & Effort Model (COSSEMO).....	8
3.2. Constructive RAD Schedule Estimation Model (CORADMO).....	8
3.2.1 Reuse and VHLLs (RVHL)	9
3.2.2 Development Process Reengineering and Streamlining (DPRS).....	9
3.2.3 Collaboration Efficiency (CLAB)	9
3.2.4 Architecture / Risk Resolution (RESL)	9
3.2.5 Prepositioning Assets (PPOS).....	9
4. Implementation Models.....	9
4.1. Logical COCOMO II RAD Extension.....	10
4.2. Physical COCOMO II RAD Extension	10
4.3. Stand-alone Spreadsheet Implementation	11

Table of Figures

Figure 1. Annotated RAD Opportunity Tree	4
Figure 2. A modern lifecycle model with anchor points.....	7
Figure 3. Logical Implementation Model	10
Figure 4. Physical Implementation Model.....	11
Figure 5. The COSSEMO extension and RAD Driver input portion of CORADMO.xls.....	11
Figure 6. The RAD extension calculation and display of Schedule and Effort.....	12

CORADMO Summary

1. Introduction

The evolution of CORADMO and its companion/pre-processor model COSSEMO 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

The COCOMO-II Model Manual provides the primary motive for this extension of COCOMO-II. “As COCOMO II evolves, 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

The COCOMO-II schedule, as presently implemented (COCOMO-II1998) 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-II does 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 Staged Schedule & Effort Model and Constructive RAD Schedule Estimation Model

In an effort to overcome these shortfalls, two extensions have been developed: the COCOMO-II Staged Schedule & Effort Model (COSSEMO) 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 of COCOMO-II as 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

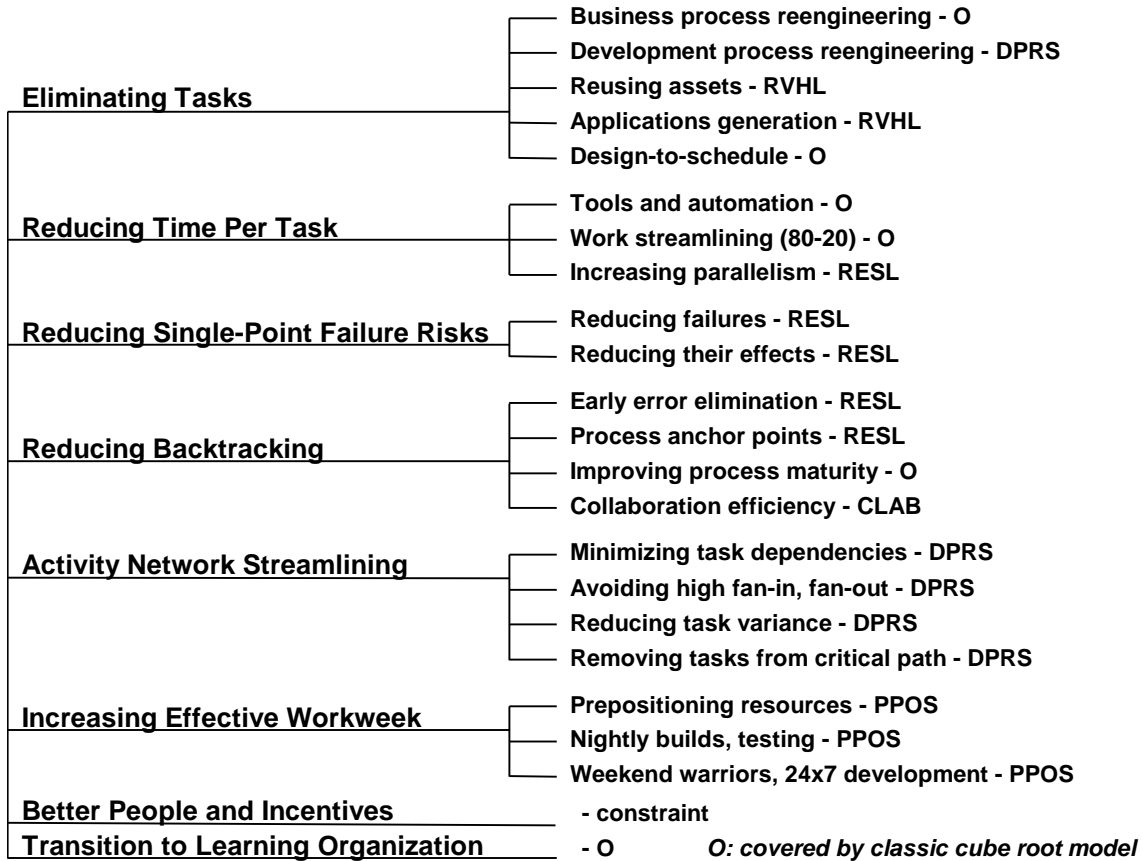


Figure 1. Annotated RAD Opportunity Tree

2.2. Duration Calculation

The COCOMO-II schedule, as presently implemented (in COCOMO-II1998) 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

The COCOMO-II duration 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 COSSEMO Duration Calculation

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, which is actually an additional (fixed percentage) effort, above and beyond the effort calculated by COCOMO.

CORADMO Summary

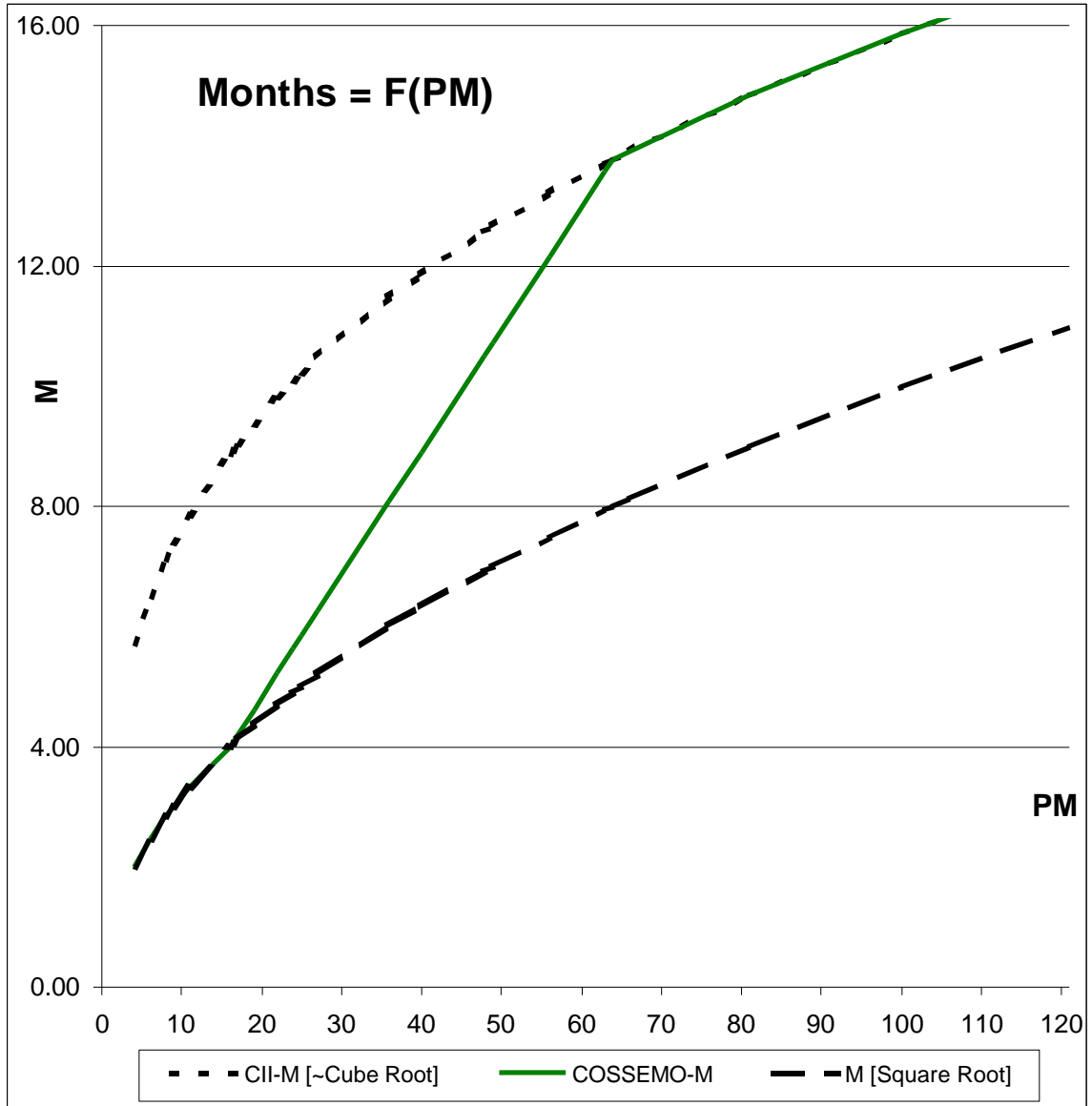
Another important difference of COSSEMO's schedule estimation from COCOMO II's simple schedule estimation is the use of a more complex calculation for the low effort situations. The initial COCOMO II baseline schedule equation is

$$TDEV = (3.0 * PMbar^{(0.33 + 0.2 * (B-1.01))} * SCED\%/100$$

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. As a result, 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.

CORADMO Summary



CORADMO Summary

2.3. Process Model

The COSSEMO model is based on the lifecycle anchoring concepts discussed by Boehm². 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 Corporation³, Figure 1 shows the stages around the anchor points.

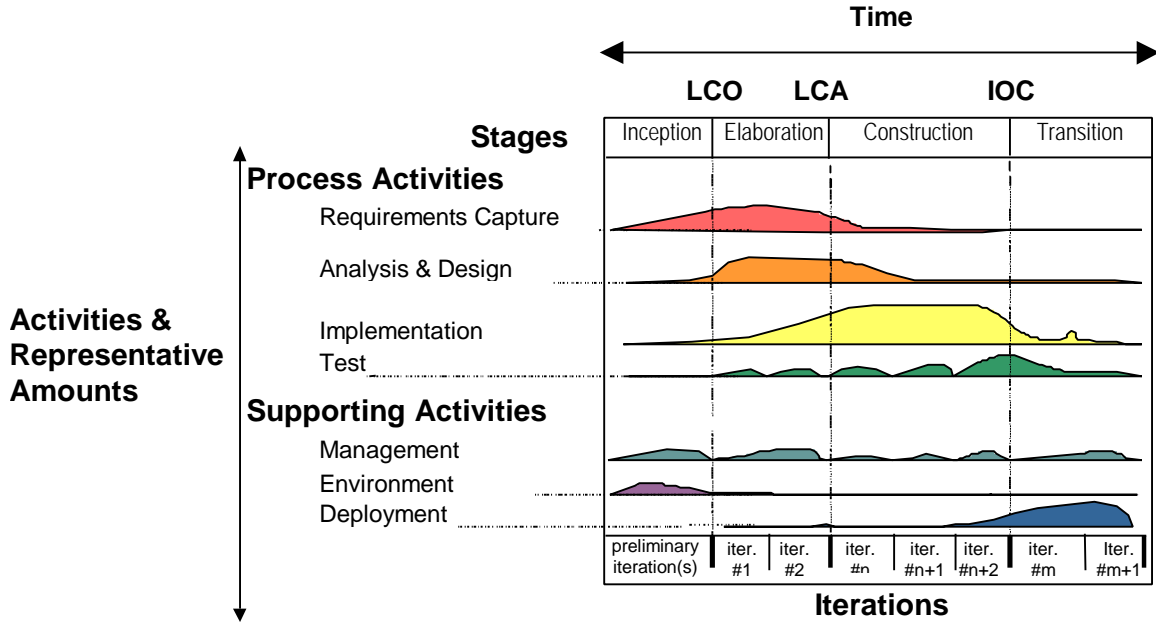


Figure 2. A modern lifecycle model with anchor points

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/>.

CORADMO Summary

2.4. Anchor Points, Stages 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 \ 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 test, integration test and qualification test.	Some, for repaired code.

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

3. Model Overview

There are two parts of the current model, COSSEMO and CORADMO. They both assume that data is available from a COCOMO II model.

3.1. COCOMO II Constructive Staged Schedule & Effort Model (COSSEMO)

The COSSEMO 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 stages: Inception, Elaboration and Construction. The predicted effort and schedule from a COCOMO II run correspond to the sum of the Elaboration and Construction stages' effort and schedule, respectively. The percentages of effort and schedule Elaboration and Construction stages thus total 100% and are used to distribute the sum accordingly. The percentages of effort and schedule for the Inception stage are also applied to the COCOMO II run's effort and schedule, respectively. Thus, the sum of the effort or schedule for three stages can actually total more than 100% of the COCOMO II run'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 stage 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 and provided in full detail in Part 3 of this report.

CORADMO Summary

3.2.1 Reuse and VHLLs (RVHL)

The impact of re-use of 3GL production code is handled directly in the COCOMO II model 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 stages. Higher rating levels reflect the potential schedule compression impacts in Inception and Elaboration stages due to faster prototyping, 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 of COCOMO II Post-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)

The COCOMO II Architecture / Risk Resolution driver (RESL) enables parallel construction activities without the COCOMO II assumed effect of increased integration and testing costs. There is not any impact on the effort or schedule in the Inception and Elaboration stages. 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 four implementations of the CORADMO/COSSEMO 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 spreadsheets, both the standalone spreadsheet extension and the multiple parallel version that is part of the Technology Impact Analyzer. The first three of these models are shown below. The fourth implementation model is described in detail in the Volume 1 of the KBSA Report.

CORADMO Summary

4.1. Logical COCOMO II RAD Extension

Figure 3 shows a conceptual logical block diagram for implementation of the RAD Model. It assumes that the regular COCOMO II implementation is extended with stage distributions which are potential driven by language level (e.g., 3GL or 4GL), experience, etc. The output of COCOMO II is used as a baseline for effort and schedule by the RAD Extension. The stage distributions extension allocates the baseline effort and schedule by stage. The RAD extension itself is controlled by the five drivers (discussed in section 3), resulting in the RAD effort and schedule by stage.

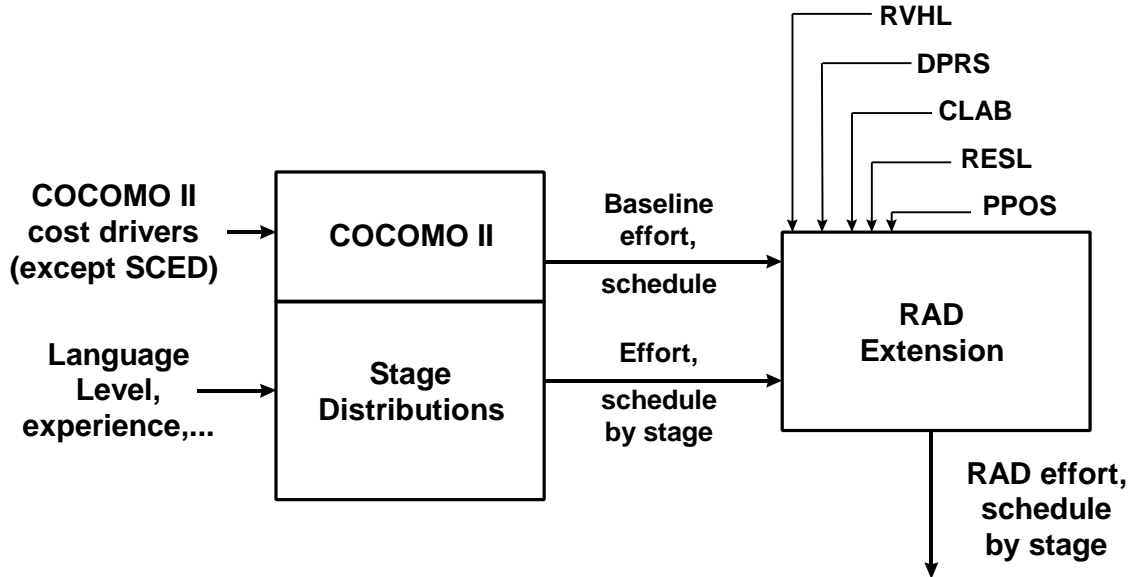


Figure 3. Logical Implementation Model

4.2. Physical COCOMO II RAD Extension

Figure 4 shows the current implementation strategy for the COCOMO II RAD extension. The upper right box represents the COCOMO II.98 model as implemented by COCOMO.exe, self-identified as “COCOMO II.1998.0” in its “About USC-COCOMOII” dialog box. Also part of the COCOMO II implementation suite is a spreadsheet called COCOMO.xls which is designed to import two CSV files that can be exported from COCOMO.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 COSSEMO Extension by links to the COCOMO.xls spreadsheet. The COSSEMO 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 COCOMO.xls spreadsheet via links, although that value can be over-ridden by the user.

CORADMO Summary

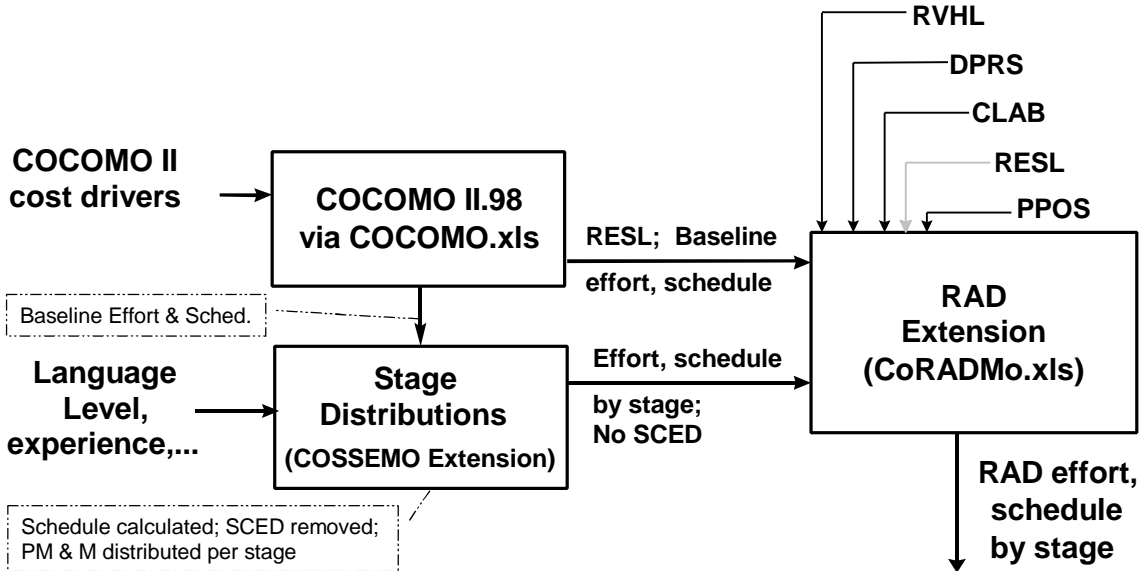


Figure 4. Physical Implementation Model

4.3. Stand-alone Spreadsheet Implementation

Figure 5 and Figure 6 contain a stand-alone implementation of the COSSEMO and CORADMO extensions.

	A	B	C	E	G	H	J	L	M	O	Q	R	T	V	W	Y	AA
1	CORADMO																
2	Step																
3	1.0 Currently implemented only for projects (not modules)																
4	2.0 Get COCOMO-II.98 data																
5	2.0	BOLD: required		ITALIC: optional		Project: 32K Norm+SCED=1.07			Total Size: 32000			PM_C= 142.2		M_C= 14.77			
6	(automatically from COCOMO.xls)																
7	Including Scale Factor Ratings																
13																	
14	3.0	Start with M_BS (i.e. After applying COSSEMO's M=fr(PM)) to COCOMO-II.98 Effort (PM_C) and P_BS calculated from PM_CM_BS.															
20	M_BS= 16.0 P_BS= 8.8796																
21																	
22	4.0	Get Eff% & Sched % per stage															
23	% Effort	14.0		% Sched	40.0		% Effort	28.0		% Sched	40.0		% Effort	72.0	% Sched	60.0	
24	Inception			Inception			Elaboration			Elaboration			Construction		Construction		
25																	
26	5.0	Distribute PM(PM_C) and M(M_BS) according to the Effort% & Schedule% per stage; and calculate P for the stage from its PM/M.															
36																	
37	Effort %	Inception			Elaboration			Construction			Total ESC		Total				
38	Schedule %	14.0			28.0			72.0			100.0		114.0				
39	PM/Ave(P)	0.35			0.70			1.20			1.00		Does not apply				
40	PM/M=P	PM	M	P	PM	M	P	PM	M	P	PM	M	P	PM	M	P-ave	
41	BS	32000	19.91	8.41	3.11	39.82	8.41	8.22	102.09	9.81	10.86	142.2	16.0	8.9	162.1	22.4	7.2
42																	
43	6.0 Get the Schedule Multipliers values.																
44																	
45																	
46																	
47																	
48	N	RVHL	0.980	0.980	1.000	0.99	0.99	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
49	N	DPRS	0.900	0.900	1.000	0.950	0.950	1.000	0.950	0.950	1.000	0.950	0.950	1.000	1.000	1.000	1.000
50	N	CLAB	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
51	N	RESL	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
52	N	PPOS	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
53	II 0.882 0.882 1.000 0.941 0.941 1.000 0.950 0.950 1.000																
54																	

Figure 5. The COSSEMO extension and RAD Driver input portion of CORADMO.xls

CORADMO Summary

COConstructive RAD schedule estimation MOdel (CoRADMO) Development

Sheet: EDSP0-Dev

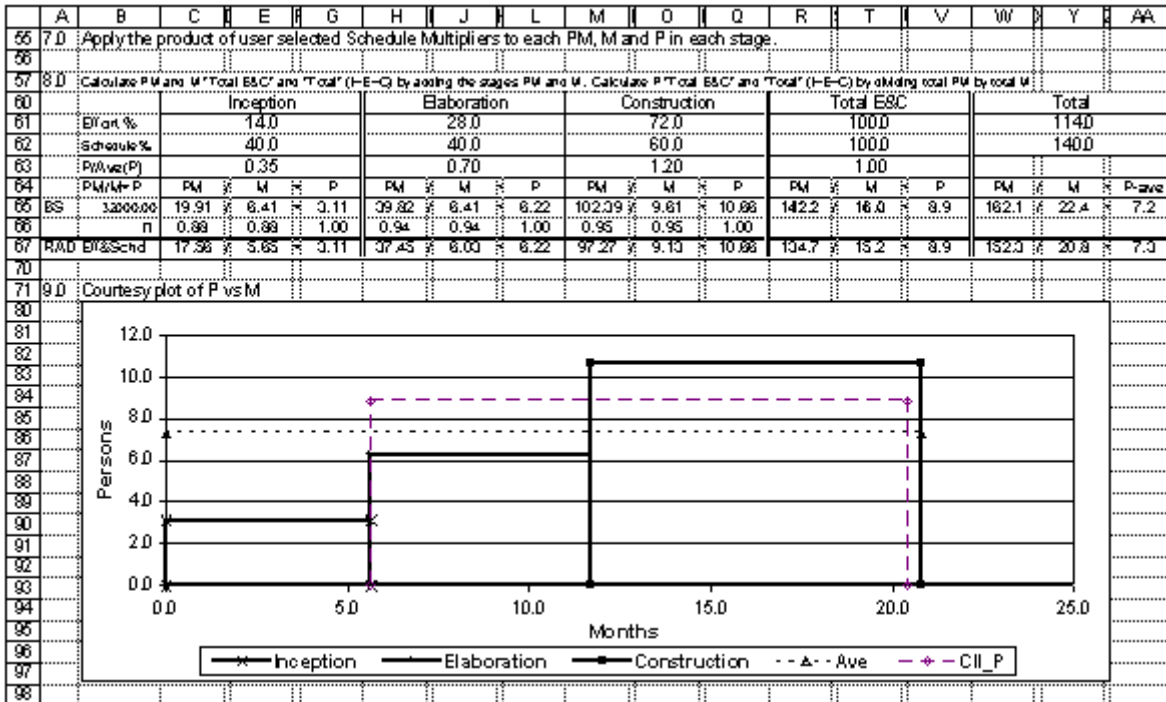


Figure 6. The RAD extension calculation and display of Schedule and Effort