

COCOMO II.2000 Post Architecture Model Driver Selection Aide

Summary Sheet

Scale Factor (SF) Section:

	X=Not Applicable	Very Low	Low	Norm	High	Very High	Extra High
PREC							
FLEX							
RESL							
TEAM							
PMAT							

EM Section:

Product Factors

	X=Not Applicable	Very Low	Low	Nominal	High	Very High	Extra high
RELY							/////
DATA		/////					/////
CPLX							
RUSE		/////					
DOCU							/////

Platform Factors

	X=Not Applicable	Very Low	Low	Nominal	High	Very High	Extra high
TIME		/////	/////				
STOR		/////	/////				
PVOL		/////					/////

Personnel Factors

	X=Not Applicable	Very Low	Low	Nominal	High	Very High	Extra high
PCON							/////
ACAP							/////
PCAP							/////
APEX							/////
PLEX							/////
LTEX							/////

Project Factors

	X=Not Applicable	Very Low	Low	Nominal	High	Very High	Extra high
TOOL							/////
SITE							
SCED							/////

SCED value for TDEV calculation

	X=Not Applicable	Very Low	Low	Nominal	High	Very High	Extra high
SCED value							/////

COCOMO II Post Architecture Model Driver Selection Aide

Scale Factors

Precedentedness (PREC)

If the product is similar to several that have been developed before then the precedentedness is high.

Features to evaluate PREC: Circle each appropriate value

.Organizational understanding of product objectives

Not Applicable	Don't Know	Very Low	Low	Norm	High	Very High	Extra High
		General	<--->	Considerable	Considerable	<--->	Thorough
		1	2	3	4	5	6

.Experience in working with related sw systems

Not Applicable	Don't Know	Very Low	Low	Norm	High	Very High	Extra High
		Moderate	<--->	Considerable	Considerable	<--->	Extensive
		1	2	3	4	5	6

.Current development of associated new hw and operational procedures

Not Applicable	Don't Know	Very Low	Low	Norm	High	Very High	Extra High
		Extensive	<--->	Moderate	Moderate	<--->	Some
		1	2	3	4	5	6

.Need for innovative data processing architecture, algorithms

Not Applicable	Don't Know	Very Low	Low	Norm	High	Very High	Extra High
		Considerable	<--->	Some	Some	<--->	Minimal
		1	2	3	4	5	6

SUM _____ (of appropriate values)

Suggested rating: (Input your rating based on the suggested rating scale. Sum --> rating)

4-7.33 -->VL 7.33-10.66 -->L 10.66-13.99 -->Norm 13.99-17.32 -->H 17.32-20.65 -->VH 20.65-24 -->EH

.Total Evaluation: Put an "X" in the applicable rating level box

Not Applicable	Don't Know	Very Low	Low	Nominal	High	Very High	Extra High
		[4..7.33]	[7.33..10.66]	[10.66..13.99]	[13.99..17.32]	[17.32..20.65]	[20.65..24]

PREC _____ (Rating Level)

COCOMO II Post Architecture Model Driver Selection Aide

Scale Factors

Development Flexibility (FLEX)

This cost driver captures the amount of constraints the product has to meet. The more flexible the requirements, schedules, interfaces, etc., the higher the rating.

Features for evaluate FLEX: Circle each appropriate value

.Need for sw conformance with pre-established requirements

Not Applicable	Don't Know	Very Low	Low	Nominal	High	Very High	Extra High
		Full	<--->	Considerable	Considerable	<--->	Basic
		1	<--->	3	4	<--->	6

.Need for sw conformance with external interface specifications

Not Applicable	Don't Know	Very Low	Low	Nominal	High	Very High	Extra High
		Full	<--->	Considerable	Considerable	<--->	Basic
		1	<--->	3	4	<--->	6

.Premium on early completion

Not Applicable	Don't Know	Very Low	Low	Nominal	High	Very High	Extra High
		High	<--->	Medium	Medium	<--->	Low
		1	<--->	3	4	<--->	6

SUM _____ (of appropriate values)

Suggested rating: (Input your rating based on the suggested rating scale. Sum --> rating)

3-4 -->VL 5-7 -->L 8-10 -->Norm 11-13 -->H 14-16 -->VH 17-18 -->EH

.Total Evaluation

Not Applicable	Don't Know	Very Low	Low	Norm	High	Very High	Extra High

FLEX _____ (Rating Level)

COCOMO II Post Architecture Model Driver Selection Aide

Scale Factors

Architecture/Risk Resolution (RESL)

This cost driver captures the thoroughness of definition and freedom from risk of the software architecture used for the product.

Features for evaluate RESL: Circle each appropriate value

.Risk management Plan identifies all critical risk items, establishes milestones for resolving them by PDR

Not Applicable	Don't Know	Very Low	Low	Nominal	High	Very High	Extra High
		None	Little	Some	Generally	Mostly	Fully
		1	2	3	4	5	6

.Schedule, budget, & internal milestones through PDR compatible with Risk Management Plan

Not Applicable	Don't Know	Very Low	Low	Nominal	High	Very High	Extra High
		None	Little	Some	Generally	Mostly	Fully
		1	2	3	4	5	6

.Percent of development schedule devoted to establishing architecture, given general product objectives

Not Applicable	Don't Know	Very Low	Low	Nominal	High	Very High	Extra High
		5%	10%	17%	26%	33%	40%
		1	2	3	4	5	6

.Percent of required top sw architectures available to project

Not Applicable	Don't Know	Very Low	Low	Nominal	High	Very High	Extra High
		20%	40%	60%	80%	100%	120%
		1	2	3	4	5	6

.Tool support available for resolving risk items, developing and verifying architectural specs

Not Applicable	Don't Know	Very Low	Low	Nominal	High	Very High	Extra High
		None	Little	Some	Good	Strong	Full
		1	2	3	4	5	6

.Level of uncertainty in Key architecture drivers: mission, user interface, COTS, hw, technology, performance

Not Applicable	Don't Know	Very Low	Low	Nominal	High	Very High	Extra High
		Extreme	Significant	Considerable	Some	Little	Very Little
		1	2	3	4	5	6

.Number and criticality of risk items

Not Applicable	Don't Know	Very Low	Low	Nominal	High	Very High	Extra High
		>10	5-10	2-4	1	>5	<5
		Critical	Critical	Critical	Critical	Non-critical	Non-critical
		1	2	3	4	5	6

SUM _____ (of appropriate values)

Suggested rating: (Input your rating based on the suggested rating scale. Sum --> rating)

7-12 -->VL 13-18 -->L 19-24 -->Norm 25-30 -->H 31-36 -->VH 37-42 -->EH

.Total Evaluation

Not Applicable	Don't Know	Very Low	Low	Nominal	High	Very High	Extra High
		7.07	5.65	4.24	2.83	1.41	0.00

RESL _____ (Rating Level)

COCOMO II Post Architecture Model Driver Selection Aide

Scale Factors

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.

Features for evaluate TEAM: Circle each appropriate value

.Consistency of stakeholder objectives and cultures

Not Applicable	Don't Know	Very Low	Low	Nominal	High	Very High	Extra High
		Little	Some	Basic	Considerable	Strong	Fully
		1	2	3	4	5	6

.Ability, willingness of stakeholders to accommodate other stakeholders' objectives

Not Applicable	Don't Know	Very Low	Low	Nominal	High	Very High	Extra High
		Little	Some	Basic	Considerable	Strong	Fully
		1	2	3	4	5	6

.Experience of stakeholders in operating as a team

Not Applicable	Don't Know	Very Low	Low	Nominal	High	Very High	Extra High
		None	Little	Little	Basic	Considerable	Extensive
		1	2	3	4	5	6

.Stakeholder teambuilding to achieve shared vision and commitments

Not Applicable	Don't Know	Very Low	Low	Nominal	High	Very High	Extra High
		None	Little	Little	Basic	Considerable	Extensive
		1	2	3	4	5	6

SUM _____ (of appropriate values)

Suggested rating: (Input your rating based on the suggested rating scale. Sum --> rating)

4-7.33 -->VL 7.33-10.66 -->L 10.66-13.99 -->N 13.99-17.32 -->H 17.32-20.65 -->VH 20.65-24 -->EH

.Total Evaluation

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

TEAM _____ (Rating Level)

COCOMO II Post Architecture Model Driver Selection Aide

Scale Factors

Process Maturity (PMAT)

The procedure for determining PMAT is organized around the Software Engineering Institute's Capability Maturity Model (CMM). Use the criteria and information on the "PMAT Eval" worksheet to self-evaluate the project's fulfillment of each KPA . Or, select a CMM Level Rating, below, and provide basis for selection.

SUM of PMAT Evaluation Ratings = _____ PMAT Rating: _____

0..3.5=>Very Low; 3.5..6=>Low; 6.01..13=>Nominal;
13.01..16=>High; 16.01..18=>Very High; 18=>Extra High

.Total Evaluation

Not Applicable	Don't Know	Very Low	Low	Nominal	High	Very High	Extra High
		[0..3.5]	[3.5..6]	[6.01..13]	[13.01..16]	[16.01..18]	[18]

PMAT _____ (Rating Level)

Overall Maturity Level	PMAT Eval Sum	
Very Low	<input type="checkbox"/> CMM Level 1 (lower half)	3.5
Low	<input type="checkbox"/> CMM Level 1 (upper half)	7
Nominal	<input type="checkbox"/> CMM Level 2	13
High	<input type="checkbox"/> CMM Level 3	16
Very High	<input type="checkbox"/> CMM Level 4	18
Extra High	<input type="checkbox"/> CMM Level 5	18

Basis of estimate:

- Software Process Assessment (SPA)
- Software Capability Evaluation (SCE)
- Interim Process Assessment (IPA)
- Other: _____

PMAT Eval Instructions for Key Process Area Evaluation

Enough information is provided in the following table so that you can assess the degree to which a KPA was exercised on the project.

- Almost Always (over **90%** of the time) when the goals are consistently achieved and are well established in standard operating procedures.
- Frequently (about **60 to 90%** of the time) when the goals are achieved relatively often, but sometimes are omitted under difficult circumstances.
- About Half (about **40 to 60%** of the time) when the goals are achieved about half of the time.
- Occasionally (about **10 to 40%** of the time) when the goals are sometimes achieved, but less often.
- Rarely If Ever (less than **10%** of the time) when the goals are rarely if ever achieved.
- Does Not Apply when you have the required knowledge about your project or organization and the KPA, but you feel the KPA does not apply to your circumstances (e.g. Subcontract Management).
- Don't Know when you are uncertain about how to respond for the KPA.

COCOMO II Post Architecture Model Driver Selection Aide PMAT Eval Worksheet

Key Process Area	Goals of each KPA	Almost Always	Very Often	About Half	Some Times	Rarely If Ever	Does Not Apply	Don't Know
Requirements Management: involves establishing and maintaining an agreement with the customer on the requirements for the software project.	System requirements allocated to software are controlled to establish a baseline for software engineering and management use. Software plans, products, and activities are kept consistent with the system requirements allocated to software.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Software Project Planning: establishes reasonable plans for performing the software engineering activities and for managing the software project.	Software estimates are documented for use in planning and tracking the software project. Software project activities and commitments are planned and documented. Affected groups and individuals agree to their commitments related to the software project.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Software Project Tracking and Oversight: provides adequate visibility into actual progress so that management can take corrective actions when the software project's performance deviates significantly from the software plans.	Actual results and performances are tracked against the software plans. Corrective actions are taken and managed to closure when actual results and performance deviate significantly from the software plans. Changes to software commitments are agreed to by the affected groups and individuals.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Software Subcontract Management: involves selecting a software subcontractor, establishing commitments with the subcontractor, and tracking and reviewing the subcontractor's performance and results.	The prime contractor selects qualified software subcontractors. The prime contractor and the software subcontractor agree to their commitments to each other. The prime contractor and the software subcontractor maintain ongoing communications. The prime contractor tracks the software subcontractor's actual results and performance against its commitments.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Software Quality Assurance: provides management with appropriate visibility into the process being used by the software project and of the products being built.	Software quality assurance activities are planned. Adherence of software products and activities to the applicable standards, procedures, and requirements is verified objectively. Affected groups and individuals are informed of software quality assurance activities and results. Noncompliance issues that cannot be resolved within the software project are addressed by senior management.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

COCOMO II Post Architecture Model Driver Selection Aide PMAT Eval Worksheet

Key Process Area	Goals of each KPA	Almost Always	Very Often	About Half	Some Times	Rarely If Ever	Does Not Apply	Don't Know
Software Configuration Management: establishes and maintains the integrity of the products of the software project throughout the project's software life cycle.	Software configuration management activities are planned. Selected software work products are identified, controlled, and available. Changes to identified software work products are controlled. Affected groups and individuals are informed of the status and content of software baselines.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Organization Process Focus: establishes the organizational responsibility for software process activities that improve the organization's overall software process capability.	Software process development and improvement activities are coordinated across the organization. The strengths and weaknesses of the software processes used are identified relative to a process standard. Organization-level process development and improvement activities are planned.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Organization Process Definition: develops and maintains a usable set of software process assets that improve process performance across the projects and provides a basis for cumulative, long- term benefits to the organization.	A standard software process for the organization is developed and maintained. Information related to the use of the organization's standard software process by the software projects is collected, reviewed, and made available.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Training Program: develops the skills and knowledge of individuals so they can perform their roles effectively and efficiently.	Training activities are planned. Training for developing the skills and knowledge needed to perform software management and technical roles is provided. Individuals in the software engineering group and software-related groups receive the training necessary to perform their roles.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Integrated Software Management: integrates the software engineering and management activities into a coherent, defined software process that is tailored from the organization's standard software process and related process assets.	The project's defined software process is a tailored version of the organization's standard software process. The project is planned and managed according to the project's defined software process.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Software Product Engineering: integrates all the software engineering activities to produce and support correct, consistent software products effectively and efficiently.	The software engineering tasks are defined, integrated, and consistently performed to produce the software. Software work products are kept consistent with each other.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Intergroup Coordination: establishes a means for the software engineering group to participate actively with the other engineering groups so the project is better able to satisfy the customer's needs effectively and efficiently.	The customer's requirements are agreed to by all affected groups. The commitments between the engineering groups are agreed to by the affected groups. The engineering groups identify, track, and resolve intergroup issues.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

COCOMO II Post Architecture Model Driver Selection Aide PMAT Eval Worksheet

Key Process Area	Goals of each KPA	Almost Always	Very Often	About Half	Some Times	Rarely If Ever	Does Not Apply	Don't Know
Peer Review: removes defects from the software work products early and efficiently.	Peer review activities are planned. Defects in the software work products are identified and removed.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Quantitative Process Management: controls the process performance of the software project quantitatively.	The quantitative process management activities are planned. The process performance of the project's defined software process is controlled quantitatively. The process capability of the organization's standard software process is known in quantitative terms.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Software Quality Management: involves defining quality goals for the software products, establishing plans to achieve these goals, and monitoring and adjusting the software plans, software work products, activities, and quality goals to satisfy the needs and desires of the customer and end user.	The project's software quality management activities are planned. Measurable goals for software product quality and their priorities are defined. Actual progress toward achieving the quality goals for the software products is quantified and managed.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Defect Prevention: analyzes defects that were encountered in the past and takes specific actions to prevent the occurrence of those types of defects in the future.	Defect prevention activities are planned. Common causes of defects are sought out and identified. Common causes of defects are prioritized and systematically eliminated.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Technology Change Management: involves identifying, selecting, and evaluating new technologies, and incorporating effective technologies into the organization.	Incorporation of technology changes are planned. New technologies are evaluated to determine their effect on quality and productivity. Appropriate new technologies are transferred into normal practice across the organization.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Process Change Management: involves defining process improvement goals and, with senior management sponsorship, proactively and systematically identifying, evaluating, and implementing improvements to the organization's standard software process and the projects' defined software processes on a continuous basis.	Continuous process improvement is planned. Participation in the organization's software process improvement activities is organization wide. The organization's standard software process and the projects' defined software processes are improved continuously.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

COCOMO II Post Architecture Model Driver Selection Aide

Product Factors

Required Software Reliability (RELY)

This is the measure of the extent to which the software must perform its intended function over a period of time.

Not Applicable	Don't Know	Very Low	Low	Nominal	High	Very High	Extra High
		slight inconvenience	low, easily recoverable losses	moderate, easily recoverable losses	High financial losses	risk to human life	/////
							/////

RELY _____ (Rating Level)

Data Size (DATA)

This measure attempts to capture the affect large data requirements have on product development e.g. testing. The rating is determined by calculating D/P, where D is the number of bytes of data in the testing data base at the end of system test and P is the number of SLOCS.

Not Applicable	Don't Know	Very Low	Low	Nominal	High	Very High	Extra High
			DB bytes/Pgm SLOC < 10	$10 \leq D/P < 100$	$100 \leq D/P < 1000$	$D/P \geq 1000$	/////
		<---> /////			1000		/////
							/////

DATA _____ (Rating Level)

Required Usability (RUSE)

This cost driver accounts for the additional effort needed to construct components intended for reuse on the current or future projects.

Not Applicable	Don't Know	Very Low	Low	Nominal	High	Very High	Extra High
		/////	None	across project	across program	across product line	across multiple product line
		/////					

RUSE _____ (Rating Level)

Documentation match to life cycle needs (DOCU)

This captures the suitability of the project's documentation to its life-cycle needs.

Not Applicable	Don't Know	Very Low	Low	Nominal	High	Very High	Extra High
		Many life-cycle needs not covered	Some life-cycle needs not covered	Right-sized to life-cycle needs	Excessive for life-cycle needs	Very excessive life-cycle needs	/////
							/////

DOCU _____ (Rating Level)

COCOMO II Post Architecture Model Driver Selection Aide

Product Factors

Product Complexity (CPLX)

Complexity is divided into five areas: control operations, computational operations, device-dependent operations, data management operations, and user interface management operations. Select the area or combination of areas that characterize the product or a sub-system of the product. The complexity rating is the subjective weighted average of these areas.

Features for evaluate CPLX: Circle each appropriate value

Control Operation	Very Low	Low	Nominal	High	Very High	Extra High
	Straight-line code with a few non-nested structured programming operators: Dos, CASEs, IF-THEN-ELSEs Simple module composition via procedure calls or simple scripts.	Straightforward nesting of structured programming operators. Mostly simple predicates.	Mostly simple nesting. Some intermodule control. Decision tables. Simple callbacks or message passing, including middleware-supported distributed.	Highly nested structured programming, structured programming operators with many compound predicates. Queue and stack control. Homogenous distributed processing. Single processor soft read-time control.	Reentrant and recursive coding. Fixed priority interrupt handling. Task synchronization, complex callbacks, heterogeneous distributed processing. Single-processor hard real-time control.	Multiple resource scheduling with dynamically changing priorities. Microcode-level control. Distributed hard real-time control.
	1	2	3	4	5	6

Computational Operations	Evaluation of simple expressions: e.g., $A=B+C*(D-E)$	Evaluation of moderate-level expressions: e.g., $D=\text{SQRT}(B**2-4.*A*C)$	Use of standard math and statistical routines. Basic matrix/vector operations.	Basic numerical analysis; analysis: multivariate interpolation, ordinary differential equations. Basic truncation, roundoff concerns.	Difficult but structured near-singular matrix numerical analysis: near-singular matrix equations, partial differential equations. Simple parallelization	Difficult & unstructured numerical analysis: highly accurate analysis of noisy, stochastic data. Complex parallelization.
	1	2	3	4	5	6

Device-dependent Operations	Simple read, write statements with simple formats.	No cognization needed of particular processor or I/O device characteristics. I/O done at GET/PUT level.	I/O processing includes device selection, status checking and error processing.	Operations at Physical I/O level (Physical storage address translations; seeks, reads, etc.) Optimized I/O overlap.	Routines for interrupt diagnosis, servicing, masking. Communication line handling. Performance-intensive embedded systems.	Device timing-dependent coding, micro-programmed operations. Performance-critical embedded systems.
	1	2	3	4	5	6

COCOMO II Post Architecture Model Driver Selection Aide

Product Factors

	Very Low	Low	Nominal	High	Very High	Extra High
Data Management Operations	Simple Arrays in main memory. Simple COTS-DB queries, updates.	Single file subsetting with no data structure changes, no edits, no intermediate files. Moderately complex COTS-DB queries, updates.	Multie-file input and single file output. Simple structural changes, simple edits. Complex COTS-DB queries, updates.	Simple triggers activated by data stream contents. Complex data restructuring	Distributed database coordination. Complex triggers. Search optimization.	Highly coupled, dynamic relational & object structures. Natural language data management.
	1	2	3	4	5	6
User Interface Management Operations	Simple input forms, report generators.	Use of simple graphic user interface (GUI) builders.	Simple use of widget set.	Widget set development and extension. Simple voice I/O, multimedia.	Moderately complex 2D/3D, dynamic graphics, multimedia.	Complex multimedia, virtual reality.
	1	2	3	4	5	6

SUM : _____ (of appropriate values)

Suggested rating: (Input your rating based on the suggested rating scale. Sum --> rating)

5-8 -->VL 9-12 -->L 13-17 -->N 18-22 -->H 23-26 -->VH 27-30 -->EH

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

CPLX _____ (Rating Level)

COCOMO II Post Architecture Model Driver Selection Aide

Personnel Factors

Platform Factor

Execution Time Constraint (TIME)

This is a measure of the execution time constraint imposed upon a software system.

Not Applicable	Don't Know	Very Low	Low	Nominal	High	Very High	Extra High
		/////	/////	<=50% use of available execution time	70%	85%	95%
		/////	/////				

TIME _____ (Rating Level)

Main Storage Constraint (STOR)

This rating represents the degree of main storage constraint imposed on a software system or subsystem.

Not Applicable	Don't Know	Very Low	Low	Nominal	High	Very High	Extra High
		/////	/////	<=50% use of available storage	70%	85%	95%
		/////	/////				

STOR _____ (Rating Level)

Platform Volatility (PVOL)

Platform is used here to mean the complex of hardware and software (OS, DBMS, etc.) the software product calls on to perform its tasks.

Not Applicable	Don't Know	Very Low	Low	Nominal	High	Very High	Extra High
		/////	major change every 12 mo.;	major: 6mo.	major : 2mo.	major : 2 wk	/////
		/////	minor change every 1 mo.	minor: 2wk.	minor : 1 wk	minor : 2 days	
		/////					/////

PVOL _____ (Rating Level)

COCOMO II Post Architecture Model Driver Selection Aide

Personnel Factors

Personnel Continuity (PCON)

The rating scale for PCON is in terms of the project's annual personnel turnover.

Not Applicable	Don't Know	Very Low	Low	Nominal	High	Very High	Extra High
		48%/year	24%/year	12%/year	6%/year	3%/year	/////
							/////

PCON _____ (Rating Level)

Analyst Capability (ACAP)

Analysts are personnel that work on requirements, high level design and detailed design.

Not Applicable	Don't Know	Very Low	Low	Nominal	High	Very High	Extra High
		15th percentile	35th percentile	55th percentile	75th percentile	90th percentile	/////
							/////

ACAP _____ (Rating Level)

Programmer Capability (PCAP)

Evaluation should be based on the capability of the programmers as a team rather than as individuals. Major factors that should be considered in the rating are ability, efficiency and thoroughness, and the ability to communicate and cooperate.

Not Applicable	Don't Know	Very Low	Low	Nominal	High	Very High	Extra High
		15th percentile	35th percentile	55th percentile	75th percentile	90th percentile	/////
							/////

PCAP _____ (Rating Level)

Applications Experience (APEX)

This rating is dependent on the level of applications experience of the project team developing the software system or subsystem. The ratings are defined in terms of the project team's equivalent level of experience with this type of application.

Not Applicable	Don't Know	Very Low	Low	Nominal	High	Very High	Extra High
		<=2 months	6 months	1 year	3 years	6 years	/////
							/////

APEX _____ (Rating Level)

Platform Experience (PLEX)

The Post-Architecture model broadens the productivity influence of PLEX, recognizing the importance of understanding the use of more powerful platforms, including more graphic user interface, database, networking, and distributed middleware capabilities.

Not Applicable	Don't Know	Very Low	Low	Nominal	High	Very High	Extra High
		<=2 months	6 months	1 year	3 years	6 years	<--->
							/////

PLEX _____ (Rating Level)

Language and Tool Experience (LTEX)

This is a measure of the level of programming language and software tool experience of the project team developing the software system or subsystem.

Not Applicable	Don't Know	Very Low	Low	Nominal	High	Very High	Extra High
		<=2 months	6 months	1 year	3 years	6 years	<--->
							/////

LTEX _____ (Rating Level)

COCOMO II Post Architecture Model Driver Selection Aide

Project Factors

Use of Software Tools (TOOL)

The tool rating ranges from simple edit and code, very low, to integrated lifecycle management tools, very high.

Not Applicable	Don't Know	Very Low	Low	Nominal	High	Very High	Extra High
		edit, code, debug	simple, front-end, backend CASE, little integration	basic life-cycle tools, moderately integrated	strong, mature life-cycle tools, moderately integrated	strong, mature, proactive life-cycle tools, well integrated with process, methods, reuse	/////
							/////

TOOL _____ (Rating Level)

Multisite Development (SITE)

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).

Not Applicable	Don't Know	Very Low	Low	Nominal	High	Very High	Extra High
		some phone, mail	individual phone, FAX	Narrowband email	Wideband electronic communication	Wideband elect. comm., occasional video conference	Interactive Multimedia

SITE _____ (Rating Level)

Required development Schedule (SCED)

This rating measures the schedule constraint imposed on the project team developing the software. The ratings are defined in terms of the percentage of schedule stretch-out or acceleration with respect to a nominal schedule for a project requiring a given amount of effort.

SCED Rating:

Not Applicable	Don't Know	Very Low	Low	Nominal	High	Very High	Extra High
% schedule compression:		75%	85%	100%	130%	160%	/////
							/////

SCED _____ (Rating Level) **SCED Percentage:** _____