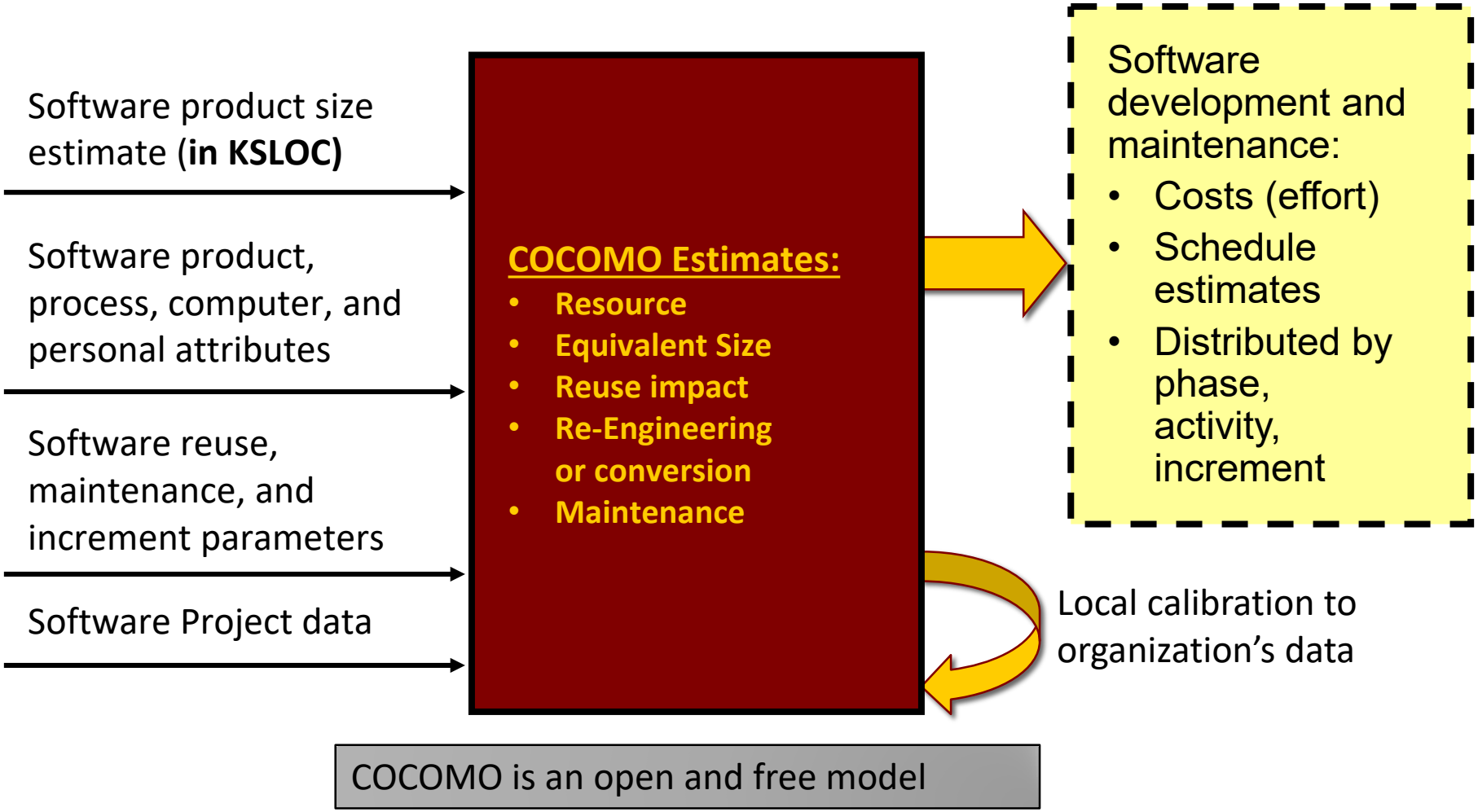
Calibrating COCOMO® II for Functional Size Metrics

ANANDI HIRA

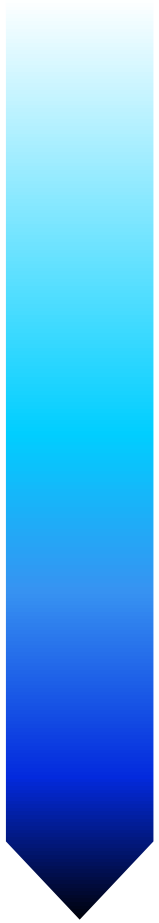
BRAD CLARK, BARRY BOEHM

COCOMO® II Model



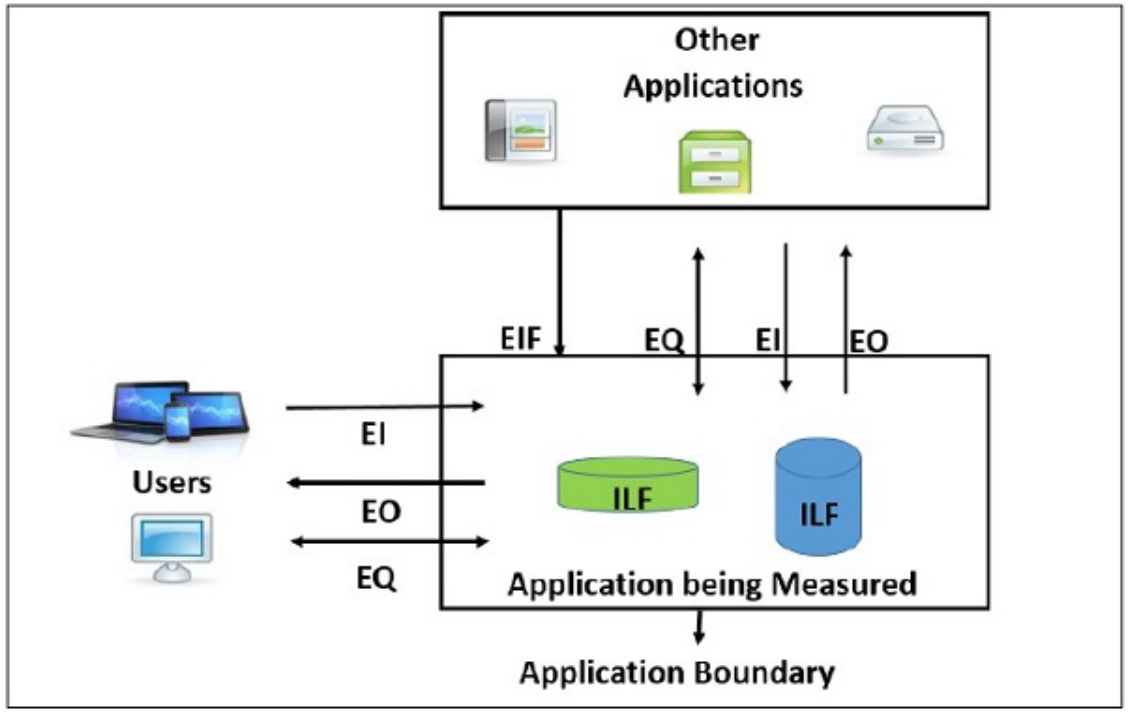
Size Metrics' Level of Abstraction

Requirement Levels	Size Metrics
Summary Goals	Story Points
User Goals	Use Cases Use Case Points (UCPs)
Sub-Functions	**IFPUG Function Points (FPs)** **COSMIC Function Points (CFPs)**
	Source Lines of Code (SLOC)

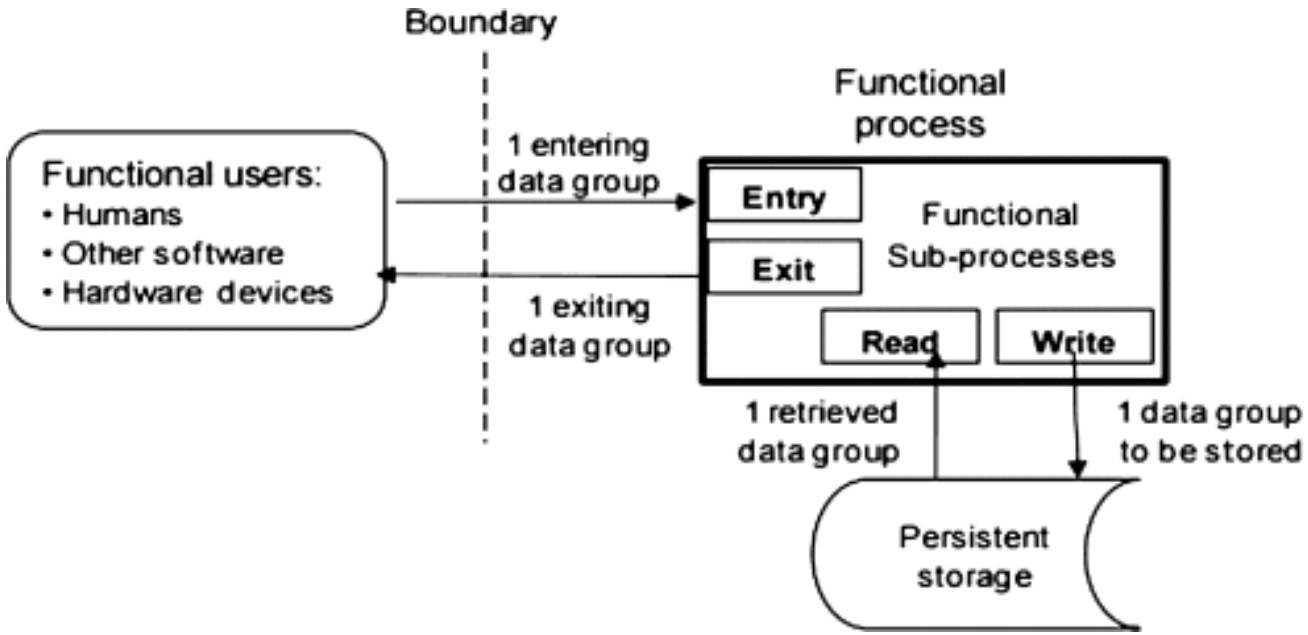


2 Prominent Functional Size Methods

IFPUG SOFTWARE MODEL



COSMIC SOFTWARE MODEL



COCOMO® II Effort Model Format

$$PM = A \times Size^{(B + 0.1 \times (\sum SF))} \times \prod EM$$

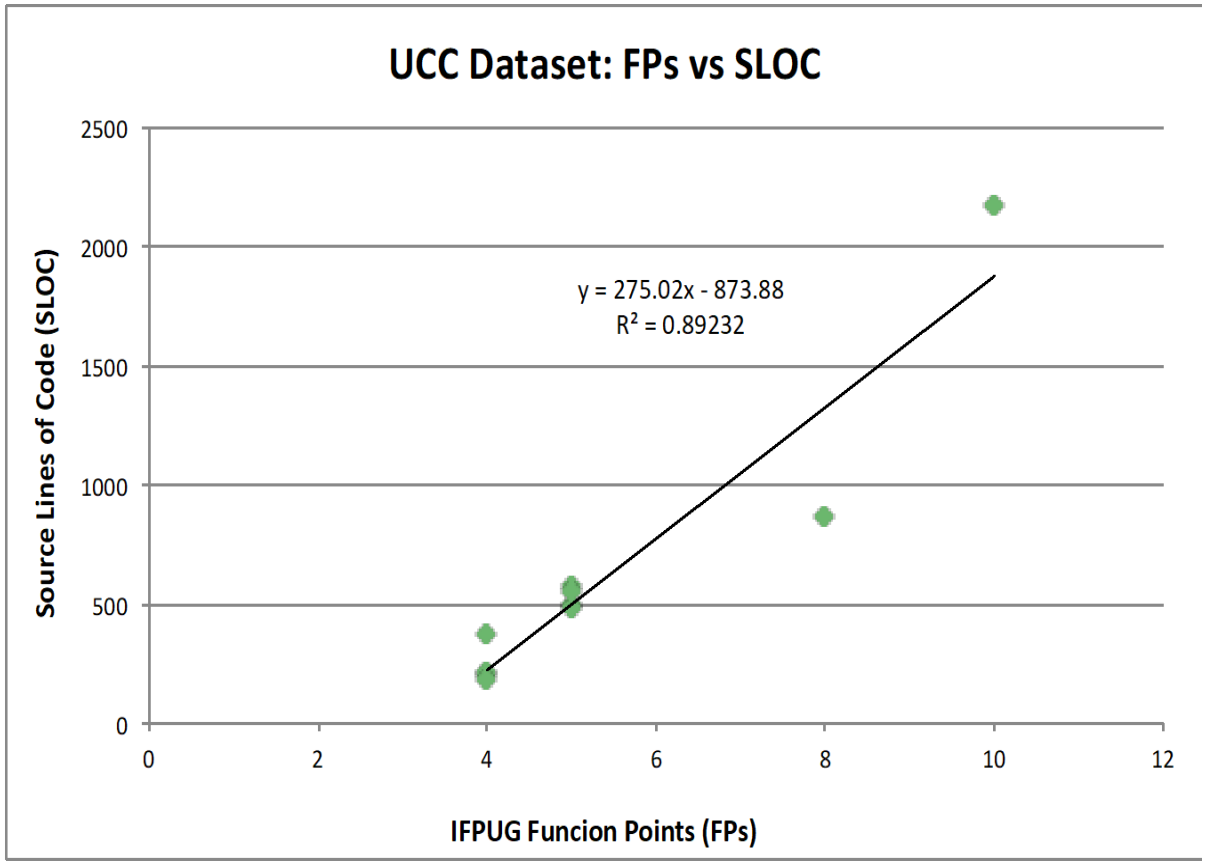
Exponent ranges from 0.9 to 1.2, with 1.0991 as default

Where

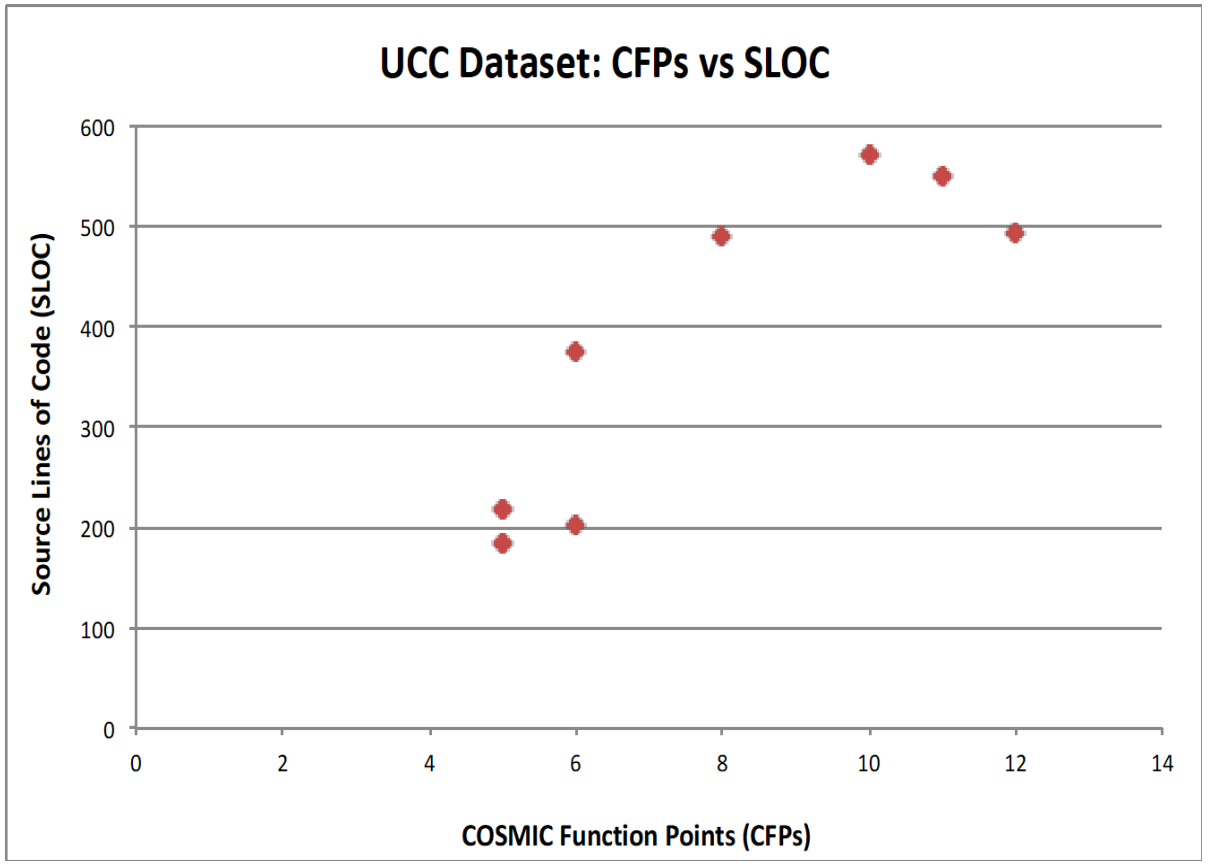
PM	=	Software development effort (in Person-months)
Size	=	Size in Thousand Equivalent Source Lines of Code (KESLOC)
A	=	Calibrated Productivity constant (ESLOC/PM)
B	=	Calibrated Exponent constant
SF	=	Scale Factors – have exponential effect
EM	=	Effort Multipliers – have multiplicative effect

Example FP and CFP vs SLOC (UCC Dataset)

IFPUG FUNCTION POINTS (FPS)



COSMIC FUNCTION POINTS (CFPS)



Objective/Goal

Adjust COCOMO® II parameters:

- Scale Factors – how quickly effort grows with respect to size
  - Precedentedness, Development Flexibility, Team Cohesion, Risk and Architecture Resolution, and Process Maturity
- Effort Multipliers – if necessary
  - Perhaps Product drivers, such as Product Complexity (CPLX)?

2 Steps:

- **Opinions of improved parameter values**
- Bayesian Analysis to combine opinion and regression

Participation Requirements

- Familiar with software development at project level, either as project lead, estimator, or engineer.
- Experience with either or both IFPUG/COSMIC Function Points
  - Or other types of functional size metric
- Experience estimating software development cost is very helpful
- Experience with COCOMO® II or other software estimation models is helpful.

Delphi Workshop

Participant Information (Name and email for internal use only, will not be shared.)

Name: _____ Email: _____

Years of experience in:

Software Development/Engineering: _____ Software Estimating: _____

For the following 2 questions, use this range:

1: Little or none

2: Some

3: A moderate amount

4: An extensive amount

5: An extensive amount, plus experience teaching COCOMO/FSM

Expertise with using:

COCOMO® II model: _____ IFPUG FPs/COSMIC FPs: _____

Participant Information

Small P… …170 Function Points

Scale Factor and brief descr. For Very Low and Extra High

Provide labor hours required for IFPUG & COSMIC Function Points

	Very Low (VL) # Hours	Nominal (N) # Hours	Extra High (XH) # Hours
F… Very Lo… …unpreced… Nominal (N): Somewh… …precede… Extra High (XH): Thoroughly famili…	FP:	5,013	FP:
			CFP:
Development Flexibility Very Low (VL): Rigorous Nominal (N): Some Relaxation Extra High (XH): General Goals		5,013	FP:
			CFP:
Architecture & Risk Resolution Very Low (VL): Little (20%) Nominal (N): Often (60%) Extra High (XH): Fully (100%)		5,013	FP:
			CFP:
Team Cohesion Very Low (VL): Very Difficult Interaction Nominal (N): Basically Cooperative Interactions Extra High (XH): Seamless Interactions	FP:	5,013	FP:
	CFP:		CFP:
Process Maturity Very Low (VL): SW-CMM/CMMI Level 1 Lower Nominal (N): SW-CMM/CMMI Level 2 Extra High (XH): SW-CMM/CMMI Level 5	FP:	5,013	FP:
	CFP:		CFP:

Voting Form – Small Project

170 Function Points
~ 9,010 SLOC Java code
COCOMO II estimate
5,013 hours, 11.2 mo.

Small P… …170 Function Points

Scale Factor and brief descr. For Very Low and Extra High

Provide labor hours required for IFPUG & COSMIC Function Points

	Very Low (VL) # Hours	Nominal (N) # Hours	Extra High (XH) # Hours
F… Very L… …nprecede… Nominal (N): Somewh… …precede… Extra High (XH): Thoroughly famili…	FP:	5,013	FP:
			CFP:
Development Flexibility Very Low (VL): Rigorous Nominal (N): Some Relaxation Extra High (XH): General Goals		5,013	FP:
			CFP:
Architecture & Risk Resolution Very Low (VL): Little (20%) Nominal (N): Often (60%) Extra High (XH): Fully (100%)		5,013	FP:
			CFP:
Team Cohesion Very Low (VL): Very Difficult Interaction Nominal (N): Basically Cooperative Interactions Extra High (XH): Seamless Interactions	FP:	5,013	FP:
	CFP:		CFP:
Process Maturity Very Low (VL): SW-CMM/CMMI Level 1 Lower Nominal (N): SW-CMM/CMMI Level 2 Extra High (XH): SW-CMM/CMMI Level 5	FP:	5,013	FP:
	CFP:		CFP:

Voting Form – Large Project

1,000 Function Points
~ 53,000 SLOC Java code
COCOMO II estimate
35,187 hours, 20.6 mo.

Scale Factors

FACTORS WITH EXPONENTIAL EFFECT

Characteristic	Very Low	Nominal / High	Extra High
Organizational understanding of product objectives	General	Considerable	Thorough
Experience in working with related software systems	Moderate	Considerable	Extensive
Concurrent development of associated new hardware and operational procedures	Extensive	Moderate	Some
Need for innovative data processing architectures, algorithms	Considerable	Some	Minimal

Precedentedness (PREC)

If a product is similar to several previously developed projects, then the precedentedness is high.

Development Flexibility (FLEX)

The PREC and FLEX scale factors are largely intrinsic to a project and uncontrollable. The next three factors identify management controllables by which projects can reduce diseconomies of scale by reducing sources of project turbulence, entropy, and rework.

Feature/Parameter Rating	**Very Low**	**Nominal**	**Extra High**
Need for software conformance with pre-established requirements	Full	Considerable	Basic
Need for software conformance with external interface specifications	Full	Considerable	Basic
Combination of inflexibilities above with premium on early completion	High	Medium	Low

Architecture/Risk Resolution (RESL) (1/2)

This factor combines two of the scale drivers in Ada COCOMO, “Design Thoroughness by Product Design Review (PDR)” and “Risk Elimination by PDR” [Boehm and Royce 1989; Figures 4 and 5]. The below table consolidates the Ada COCOMO ratings to form a more comprehensive definition for the COCOMO II RESL rating levels. It also relates the rating level to the MBASE/RUP Life Cycle Architecture (LCA) milestone as well as to the waterfall PDR milestone. The RESL rating is the subjective weighted average of the listed characteristics.

Architecture/Risk Resolution (RESL) (2/2)

Characteristic/Rating	Very Low	Nominal	Extra High
Risk management plan identifies all critical risk items and establishes milestones for resolving them	None	Some	Fully
Schedule, budget, and internal milestones compatible with risk management plan	None	Some	Fully
% of development schedule devoted to establishing architecture	5	17	40
% of required top software architects	20	60	120
Tool support available for resolving risk items and verifying architectural specs	None	Some	Full
Level of uncertainty in key architecture drivers	Extreme	Considerable	Very Little
Number and criticality of risk items	> 10 Critical	2-4 Critical	<5 Non-Critical

Team Cohesion (TEAM) (1/2)

Characteristic/Rating	Very Low	Nominal	Extra High
Consistency of stakeholder objectives and cultures	Little	Basic	Full
Ability, willingness of stakeholders to accommodate other stakeholders' objectives	Little	Basic	Full
Experience of stakeholders in operating as a team	None	Little	Extensive
Stakeholder teambuilding to achieve shared vision and commitments	None	Little	Extensive

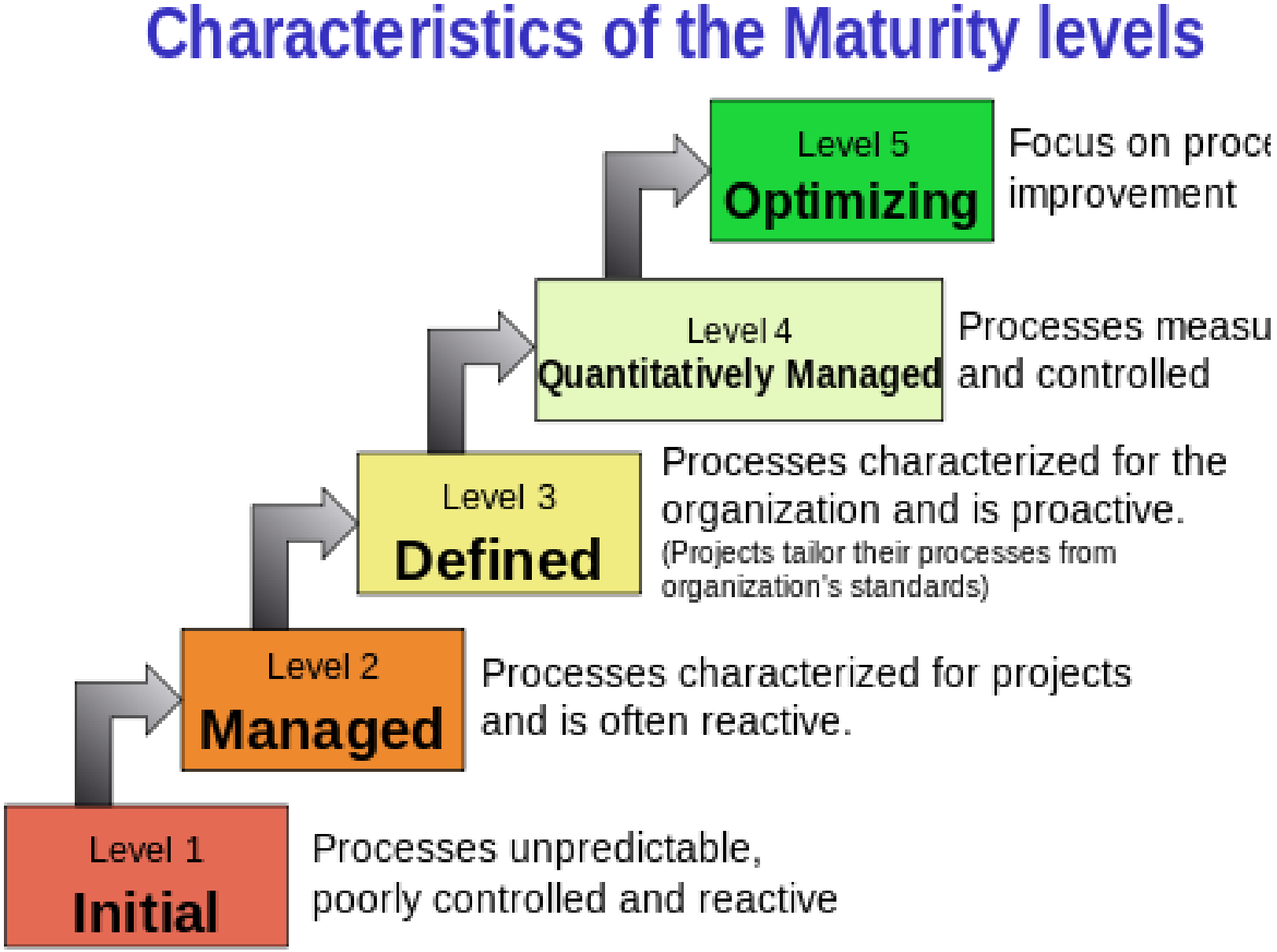
Team Cohesion (TEAM) (2/2)

Very Low	Low	Nominal	High	Very High	Extra High
Very Difficult Interactions	Some Difficult Interactions	Basically Cooperative Interactions	Largely Cooperative	Highly Cooperative	Seamless Interactions
0.0548	0.0438	0.0329	0.0219	0.011	0

New Value?

Process Maturity (PMAT)

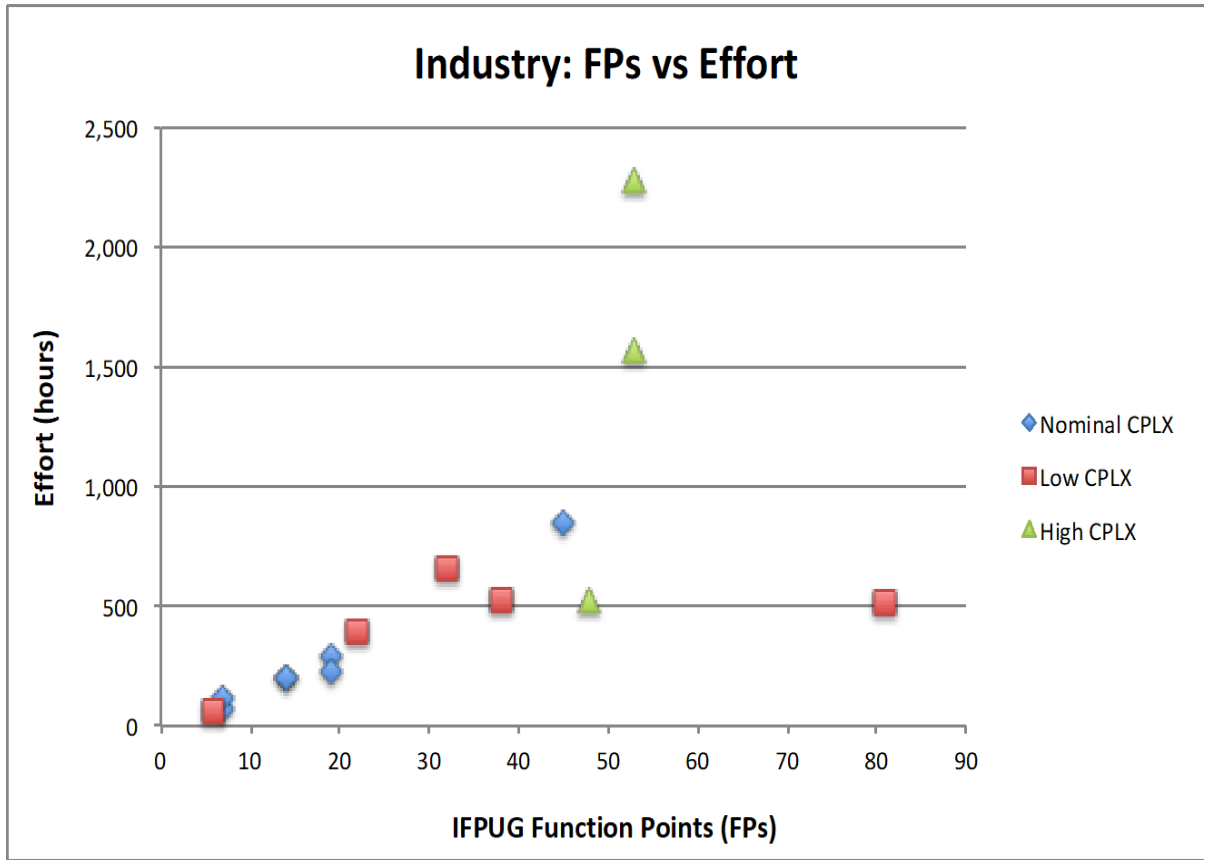
The procedure for determining PMAT is organized around the Software Engineering Institute's Capability Maturity Model (CMM). The time period for rating Process Maturity is the time the project starts. There are two ways of rating Process Maturity. The first captures the result of an organized evaluation based on the CMM.



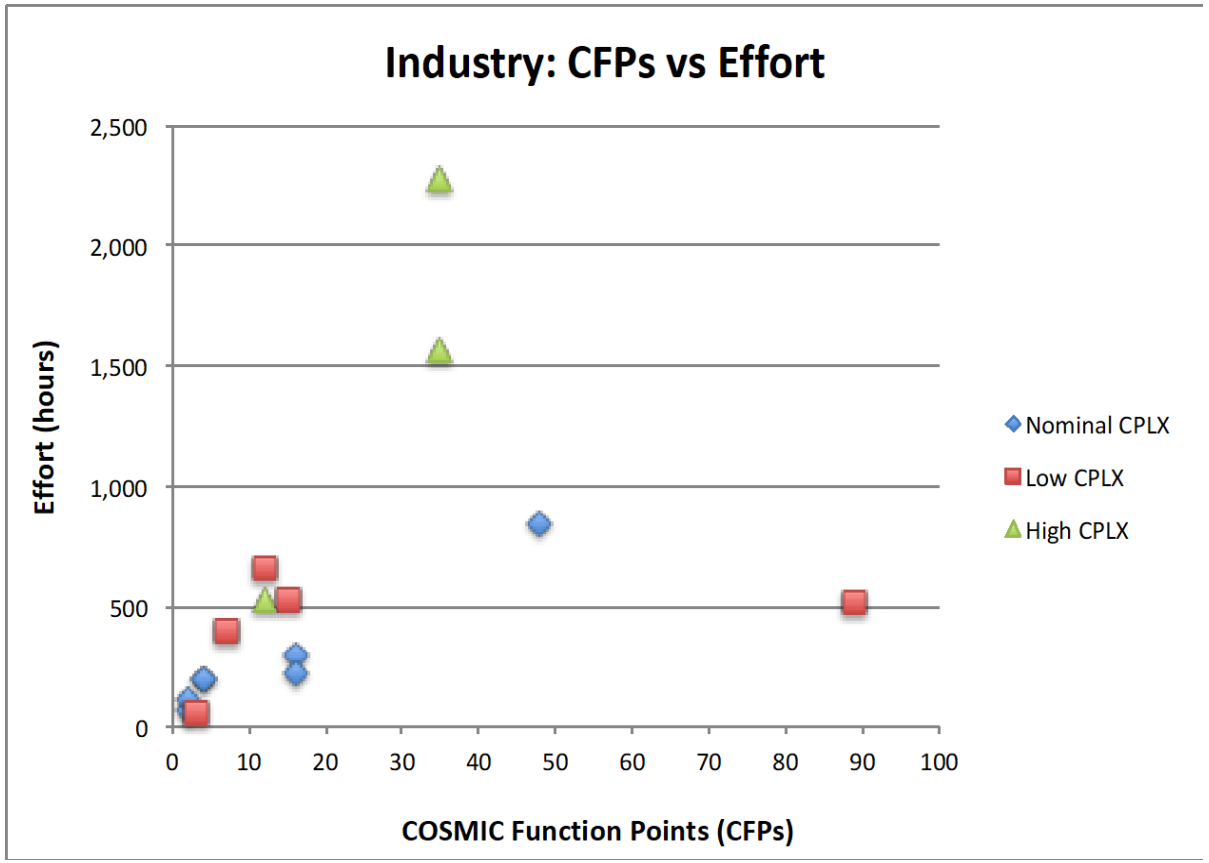
Adjust any COCOMO® II Effort Mutipliers?

Example Complexity Levels (Industry Dataset)

IFPUG FUNCTION POINTS (FPS)

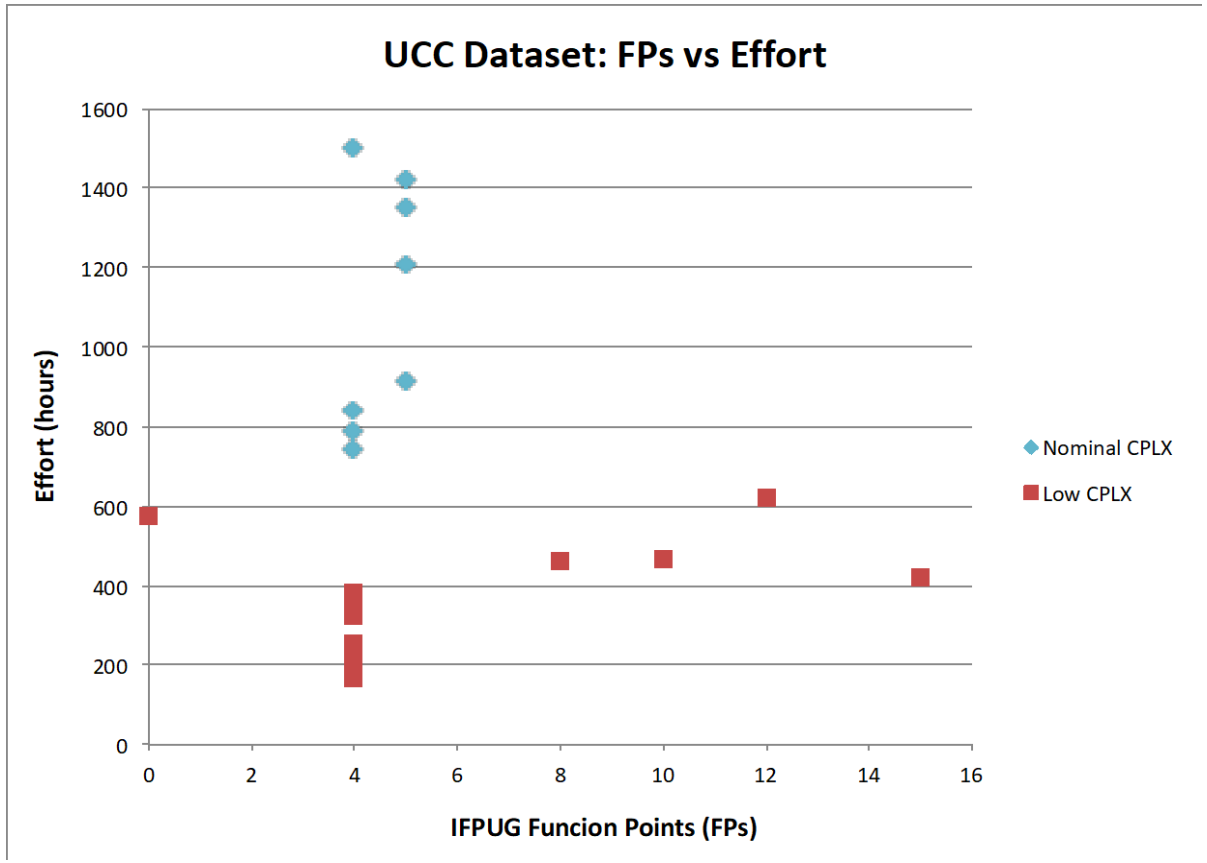


COSMIC FUNCTION POINTS (CFPS)

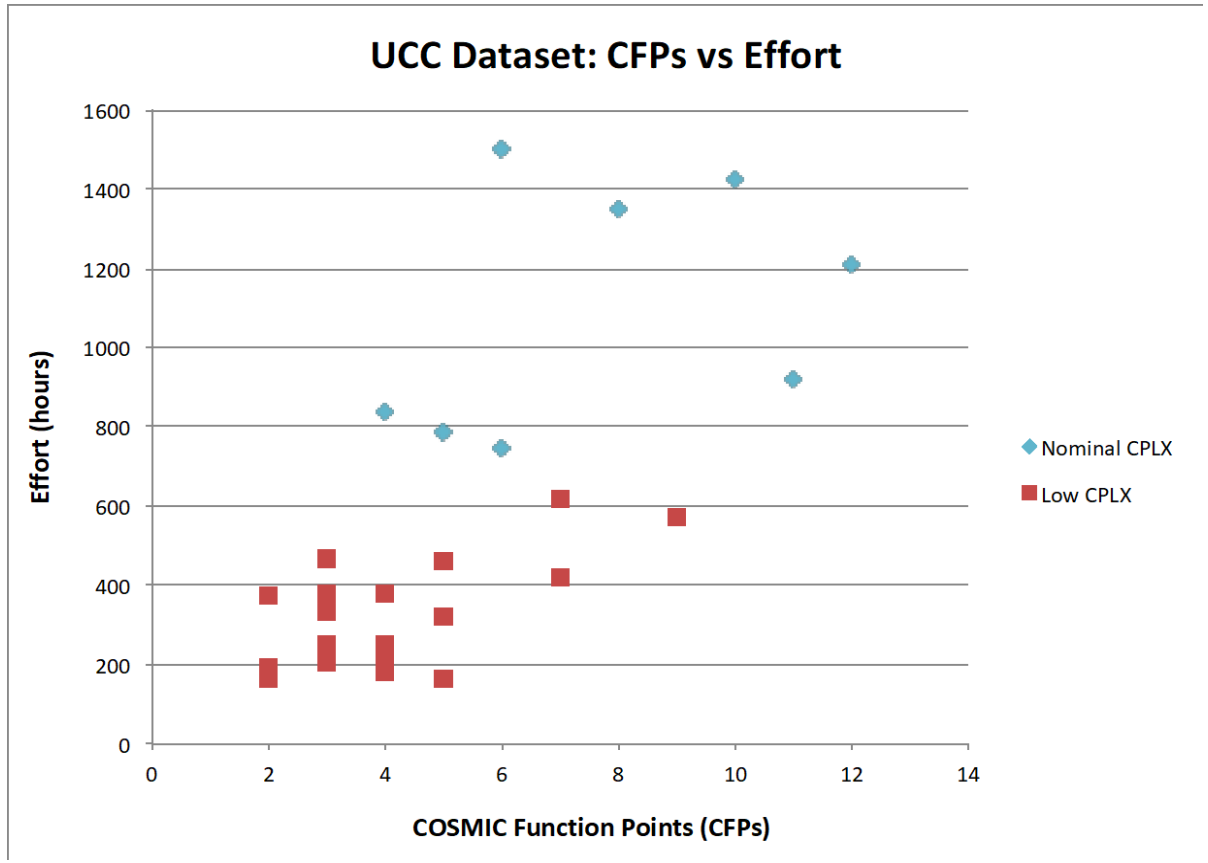


Example Complexity Levels (UCC Dataset)

IFPUG FUNCTION POINTS (FPS)



COSMIC FUNCTION POINTS (CFPS)



Product Complexity (CPLX) (1/3)

	Control Operations	Computation Operations	Device Dependent Operations	Data Management Operations	User Interface Operations
Very Low	Straight-line code with few non-nested structured programming operations	Evaluation of simple expressions: e.g., A = B + C*(D-E)	Simple read, write statements with simple formats.	Simple arrays in main memory.	Simple input forms, report generators.
Low	Straightfoward nesting of structured programming operators. Mostly simple predicates.	Evaluation of moderate-level expressions: e.g., D = SQRT(B*2-4*A*C)	No congnizance needed of particular processor or I/O device characteristics.	Single file subsetting with no data structure changes, no edits, no intermediate files.	Use of simple GUI builders.
Nominal	Mostly simple nesting. Some inter-module control. Decision tables, simple callbacks or message passing.	Use of standard math and statistical routines. Basic matrix/vector operations.	I/O Processing includes device selection, status checking and error processing.	Multi-file input and single file output. Simple structural changes, simple edits.	Simple use of widgets.

Product Complexity (CPLX) (2/3)

	Control Operations	Computation Operations	Device Dependent Operations	Data Management Operations	User Interface Operations
High	Highly nested structured programming operators with many compound predicates. Queue and stack control.	Basic numerical analysis.	Operations at physical I/O level. Optimized I/O overlap.	Simple triggers activated by data stream contents.	Widget development and extension. Simple voice I/O, multimedia.
Very High	Reentrant and recursive coding. Fixed-priority interrupt handling. Task sync, complex callbacks.	Difficult but structured numerical analysis.	Routines for interrupt diagnosis, servicing, masking. Communication line handling.	Distributed database coordination. Complex triggers. Search optimization.	Moderately complex 2D/3D, dynamic graphics, multimedia.
Extra High	Multiple resource scheduling.	Difficult and unstructured numerical analysis	Device timing-dependent coding, micro=programmed operations.	Highly coupled, dynamic relational and object structures.	Complex multimedia, virtual reality, natural language interface.

Product Complexity (CPLX) (3/3)

Very Low	Low	Nominal	High	Very High	Extra High	Productivity Range (Max/Min)
Straight—line, simple math, read/write, arrays	Simple predicates, I/O device not known, simple GUI builders	Simple nesting, decision-tables, math & stats, device selection, widgets	Highly nested, compound predicates, numerical analysis, operations at I/O level, triggers, Multimedia	Recursive coding, complex callbacks, routines, distr DB, complex triggers, 2D/3D graphics	Resource scheduling, device timing, coupled and dynamic structures, natural language	
0.73	0.87	1	1.17	1.34	1.74	**2.38**

Effort Multipliers

PRODUCT AND PLATFORM FACTORS

Required Software Reliability (RELY)

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

Very Low	Low	Nominal	High	Very High	Productivity Range (Max/Min)
Slight Inconvenience	Low, easily recoverable losses	Moderate, easily recoverable losses	High Financial loss	Risk to human life	
0.82	0.0624	0.0468	0.0312	1.26	**1.54**

Database Size (DATA)

Capture the effect large test data requirements have on product development. Rating is determined by calculating the ratio of bytes in the testing database to SLOC in the program.

Consider effort required to generate the test data that will be used to test program.

Low	Nominal	High	Very High	Productivity Range (Max/Min)
DB Bytes/ Project SLOC < 10	10 ≤ DB Bytes/ Project SLOC < 100	100 ≤ DB Bytes/ Project SLOC < 1000	DB Bytes/ Project SLOC > 1000	
0.9	1	1.14	1.28	**1.42**

Developed for Reusability (RUSE)

Accounts for additional effort needed to construct components intended for reuse on current or future projects. This effort is consumed with creating more generic design of software, more elaborate documentation, and more extensive testing to ensure components are ready for use in other applications.

Low	Nominal	High	Very High	Extra High	Productivity Range (Max/Min)
None	Across Project	Across Program	Across Product Line	Across Multiple Product Lines	
0.95	1	1.07	1.15	1.24	**1.31**

Documentation Math to Life Cycle Needs (DOCU)

Cost driver is evaluated in terms of the suitability of the project's documentation to its life cycle needs.

Very Low	Low	Nominal	High	Very High	Productivity Range (Max/Min)
Many life cycle needs uncovered	Some life cycle needs uncovered	Right-sized to life cycle needs	Excessive for life cycle needs	Very excessive for life cycle needs	
0.81	0.91	1	1.11	1.23	**1.52**

Execution Time Constraint (TIME)

Measure of the execution time constraint imposed upon a software system. Expressed in terms of the percentage of available execution time expected to be used by the system or subsystem consuming the execution time resource.

Nominal	High	Very High	Extra High	Productivity Range (Max/Min)
≤ 50% use of available execution time	70% use of available execution time	85% use of available execution time	95% use of available execution time	
1	1.11	1.29	1.63	**1.63**

Main Storage Constraint (STOR)

Represents the degree of main storage constraint imposed on a software system or subsystem. Many application expand to consume whatever resources are available. The rating is in terms of the percentage of available main storage expected to be used by the system or subsystem.

Nominal	High	Very High	Extra High	Productivity Range (Max/Min)
≤ 50% use of available storage	70% use of available storage	85% use of available storage	95% use of available storage	
1	1.05	1.17	1.46	**1.46**

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. This driver represents the amount of effort required to maintain the software product to be compatible with changes to platform.

Low	Nominal	High	Very High	Productivity Range (Max/Min)
Major change every year; Minor change every month	Major change every 6 mo., Minor change every 2 weeks	Major change every 2 mo., Minor change every week	Major change every 2 weeks, Minor change every 2 days	
0.87	1	1.15	1.3	**1.49**

Effort Multipliers

PERSONNEL AND PROJECT FACTORS

Analyst Capability (ACAP)

Analysts are personnel who work on requirements, high-level design, and detailed design. The major attributes that should be considered in this rating are analysis and design ability, efficiency and thoroughness, and the ability to communicate and cooperate.

The rating should not consider the level of experience. Percentile is with respect to all developers and analysts across the US.

Very Low	Low	Nominal	High	Very High	Productivity Range (Max/Min)
15th percentile	35th percentile	55th percentile	75th percentile	90th percentile	
1.42	1.19	1	0.85	0.71	**2.0**

Programmer Capability (PCAP)

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

The experience of the programmer should not be considered here.

Very Low	Low	Nominal	High	Very High	Productivity Range (Max/Min)
15th percentile	35th percentile	55th percentile	75th percentile	90th percentile	
1.34	1.15	1	0.88	0.76	**1.76**

Personnel Continuity (PCON)

Representation of how consistent the team remains over the duration of the project.

With personnel turnover, knowledge is lost with team members. Existing team members must spend time to train new members, and new members require learning effort.

Very Low	Low	Nominal	High	Very High	Productivity Range (Max/Min)
52%/year stayed	64%/year stayed	88%/year stayed	94%/year stayed	97%/year	
1.29	1.12	1	0.9	0.81	**1.51**

Applications Experience (APEX)

The rating for this cost driver depends 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.

Very Low	Low	Nominal	High	Very High	Productivity Range (Max/Min)
< 2 years	6 months	1 year	3 years	6 years	
1.22	1.1	1	0.88	0.81	**1.51**

Platform Experience (PLEX)

This cost driver recognizes the importance of understanding the use of more powerful platforms, including graphic user interface, database, networking, and distributed middleware capabilities, as required by the software product.

Very Low	Low	Nominal	High	Very High	Productivity Range (Max/Min)
< 2 years	6 months	1 year	3 years	6 years	
1.2	1.09	1	0.91	0.84	**1.4**

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.

Includes use of tools that perform requirements and design representation and analysis, configuration management, document extraction, library management, program style and formatting, consistency checking, planning and control, etc.

Very Low	Low	Nominal	High	Very High	Productivity Range (Max/Min)
< 2 years	6 months	1 year	3 years	6 years	
1.19	1.09	1	0.91	0.85	**1.43**

Use of Software Tools (TOOL)

Very Low	Low	Nominal	High	Very High	
Edit, code, debug	Simple, frontend, backed 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 processes, methods, reuse	**Productivity Range (Max/Min)**
1.17	1.09	1	0.9	0.78	**1.5**

Multisite Development (SITE) (1/2)

Determining this cost driver's rating involves assessing and judgement-based averaging of 2 factors: site collocation (from fully collocated to international distribution) and communication support (from surface mail and some phone access to full interactive multimedia).

Note: if a team is fully collocated, it doesn't need interactive multimedia to achieve an Extra High rating. Email would usually be sufficient.

* Did not include communication support in hand-out to avoid confusion.

Multisite Development (SITE) (2/2)

Very Low	Low	Nominal	High	Very High	Extra High	Productivity Range (Max/Min)
Team spread out inter-nationally	Multi-city and multi-company	Multi-city or multi-company	Same city or metro area	Same building or complex	Fully collocated	
Some phone, email	Individual phone	Email	Wide-band electronic communication	Occasional video conference	Interactive multi-media	
1.22	1.09	1	0.93	0.86	0.8	**1.53**

Required Development Schedule (SCED) (1/2)

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.

Accelerated schedules tend to produce more effort in the earlier phases to eliminate risks and refine the architecture and more effort in the later phases to accomplish more testing and documentation in parallel.

Stretch-outs do not add or decrease effort. They lead to savings from smaller team sizes and are generally balanced by the need to carry project administrative functions over a longer period of time.

Required Development Schedule (SCED) (2/2)

Very Low	Low	Nominal	High	Very High	Productivity Range (Max/Min)
75% of Nominal	85% of Nominal	100% of Nominal	130% of Nominal	160% of Nominal	
1.43	1.14	1	1	1	**1.43**

Review Results