Streamlined Approach to SW Estimating using COCOMO II

September 2010
Prepared by John M. Rossi
Sr. Systems Engineer
Rockwell Collins STS

Rockwell Collins

Agenda

- Problem Description
- Calibration Basics
- Collector Forms
- Initial Results
- Summary
- Lessons Learned
- Recommendations
- Questions

Why The Need For A Standardized Estimating Tool?

- Need defendable Basis of Estimates for our pursuits
  - DCMA requirement
- Requirement for bids to be generated with a consistent and repeatable process
  - No way to accurately compare bids if different estimating tools and methods are used
- Estimation models must be calibrated to STS past program actuals before use on proposal activities
  - Use of standardized tool reduces effort required for calibration
- Level of knowledge on how to use the tools varies widely

What Estimating Tools Did STS Evaluate?

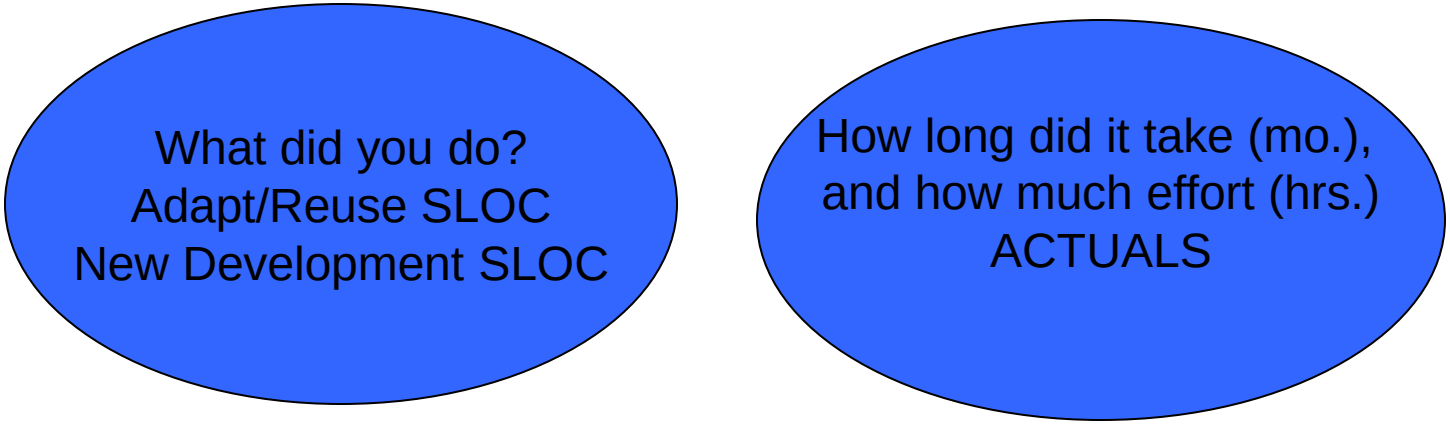
- COCOMO II 2000
- SEER-SEM
- TRUE-Planning (True-S)
- REVIC 9.2
- SLIM

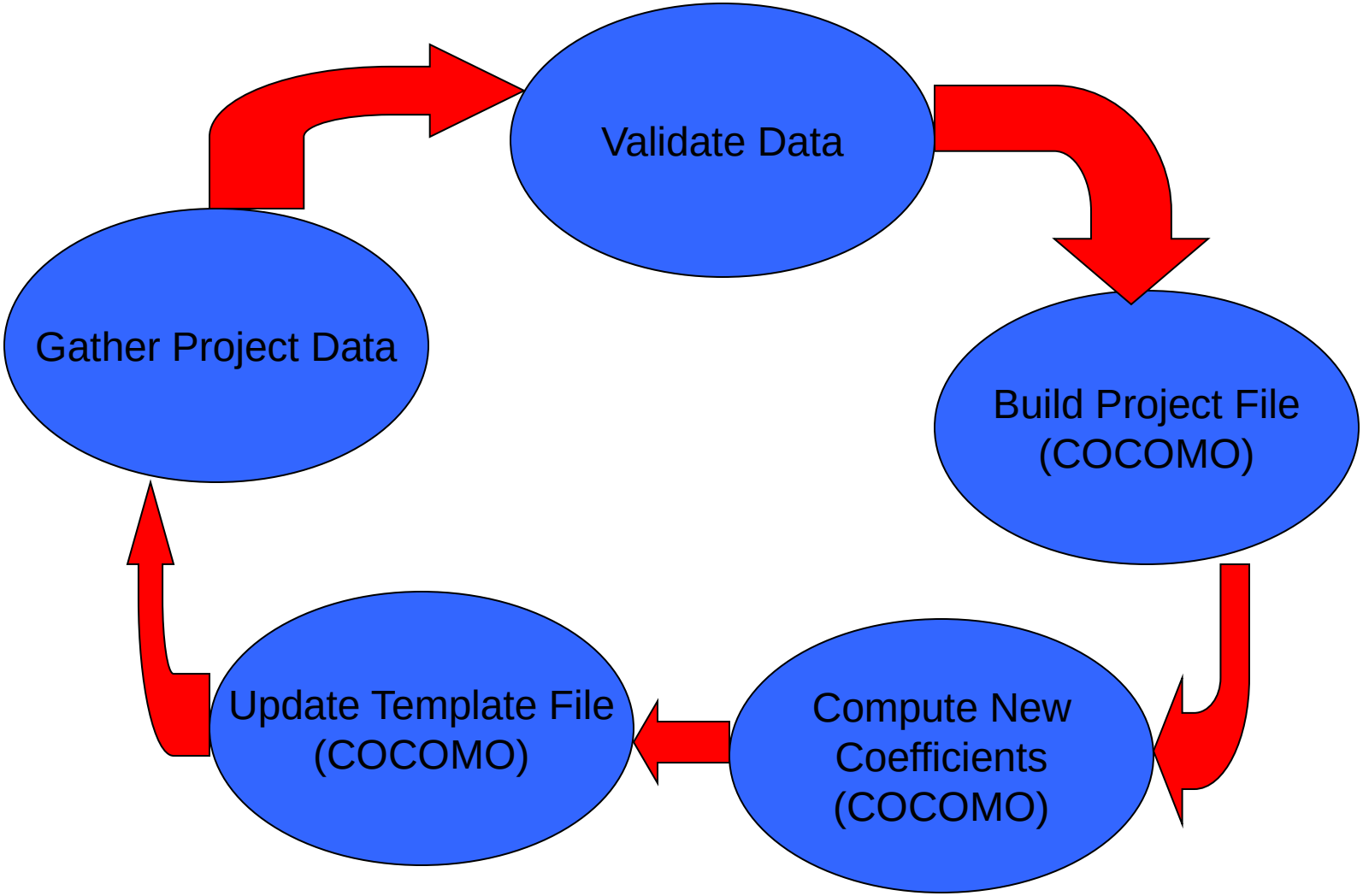
COCOMO II 2000 Was Selected

- Pros
  - Latest revision of the COCOMO II model
  - Extensive documentation to the model
  - Model factors are changeable to the project
  - Can be calibrated to STS Program Actuals
  - Output uses standard Excel spreadsheets

- Cons
  - Need a training course on setting up and running the program
  - Requires accurate Source Lines Of Code (SLOC) counts for each functional area to be estimated
  - Many complex factors to learn which affect output
  - Embeds non-engineering functions in effort estimates (Program Office, CM, QA, Manuals)

Calibration Data Basics





COCOMO Calibration Goals

- By following the process…
- Calibrated values from COCOMO approach Program Actuals over many STS programs
- Calibrated COCOMO model becomes the standard Software Estimating tool for creating bids for proposals and ECPs for STS
- Using the Calibrated COCOMO Template establishes a defendable Basis of Estimate (BOE) for written narratives

We Selected COCOMO II, Now What?

- Need Data…And lots of it!!
- Need an easy, intuitive way to capture the SLOC from Programs
- Need to capture Program Actuals to build a calibration file with the SLOC

What Is SLOC?

- SLOC = “Source Lines Of Code”
- One SLOC is one logical line of code (an If-Then-Else is one line of code)
  - In COCOMO 81, known as Delivered Source Instructions (DSI). One If-Then-Else = X lines of code
- Only Source lines that are DELIVERED as part of the product are included -- test drivers and other support software is excluded
- SOURCE lines are created by the project staff -- code created by applications generators is excluded
- Declarations are counted as SLOC
- Comments are not counted as SLOC

Key Terminology

- New Code: Completely new file development
  - Files developed from scratch for the program
  - Not ported from another program

- Reused Code: Code that is 100% reused (NO modifications)
  - Files taken from previous program baseline w/o change
  - Files taken from other programs w/o change

- Adapted Code: Existing code that will be modified (adapted) to work for the objectives of the program.
  - Existing files that will be modified for program objectives
  - Existing files that will delete code for program objectives
  - Existing files that will add code from other sources
  - Need the SLOC size **before** any modifications are made (baseline)

Tools To Capture New / Adapted Code Count

- SLOCCount: Standard STS tool for computing SLOC
- Count LOC: Alternate tool for computing SLOC
- Beyond Compare: Compare files for differences (New, Deleted, Modified lines of code)
- Perl Script: Find_Mods
  - In-house tool to diff files or directories.
  - Outputs Metrics needed for collector

Collector Forms

- Rockwell Collins STS developed the SLOC Collector to capture data from programs
  - SLOC Collector
  - Project Info Collector
  - Engineers will complete form

- Both forms used to collect project SLOC Actuals and COCOMO factors for completed projects.
  - Forms are embedded into the Software Development Folders (SDF)

153												
154	**Surface Search**	**Total Subsystem Size**	B52OSS/IUS/TGP	**27180**	c							
155												
156		**Header File(s)**	TGP_Numeric_Entry_Page.h		c	57						
157			TGP_Search_Page.h		c	81						
158			TGP_Search_Table_Page.h		c	59						
159			TGP_Search_Utilities_Page.h		c	114						
160			TGP_SPI_Loading_Page.h		c		132	1		10	4	0.6
161			AGWCP_Output.h		c		109	12		10	4	0.6
162			TGP_Shared.h		c		118	5		10	4	0.6
163												

New Code Example. Only column F needs filled out.

Grayed out cells **SHOULD NOT** be filled in.

	A	B	C	D	E	F	G	H	I	J	K	L	M
1					Primary		Initial	Total					
2	Subsystem	File Type	File Location/Name	Total	Language	New	Adapted	Modified	Reused	IM	SU	AA	UNFM
3				SLOC		SLOC	SLOC	SLOC	SLOC	(%)	(%)	(%)	
4													
5	IOS	Total Subsystem Size	/smt/avio/	27606	C++								
6		Header File(s)	/b52off_MalfActive.h		C++		25	1			30	0	0.2
7			/b52off_malf_data.cxx		C++		4147	600			30	0	0.2
8			/WTXOffense.cxx		C++		1880	5			30	0	0.2
9			/b52off_malfs.h		C++		7200	760			30	0	0.2
10			/WeaponCommonUtils.h		C++		403	50			20	2	0.2
11			/WeaponLoadIC.h		C++		114	10			20	2	0.2
12			/WeaponStatus.h		C++		82	1			20	0	0.2
13			/WpnConfig.h		C++		95	5			20	2	0.2
14			/mgstargetdata.h		C++		166	12			30	2	0.2
15			/malfs.h		C++		850	39			30	2	0.2
16			/IOS_Shared.h		C++		684	4			30	0	0.2
17			/DtcUtils.h		C++		496	1			30	0	0.2
18			/weapon_defs.h		C++		191	1			30	0	0.2
19													
20			/MalfActive.cpp		C++		694	19			30	2	0.2
21			/MapDisplay.cpp		C++		3603	71			30	8	0.2
22			/MgsLib.cpp		C++		677	4			30	6	0.2
23			/mgsTargetData.cpp		C++		553	49			20	6	0.2
24			/WeaponCommonUtils.cpp		C++		1707	537			20	8	0.2
25			/WeaponLoadIC.cpp		C++		898	180			30	8	0.2
26			/WeaponStatus.cpp		C++		381	7			30	4	0.2
27			/WpnConfig.cpp		C++		702	6			30	4	0.2
28			/WeaponLoadOff.cpp		C++		936	1			30	2	0.2
29			/Update_Wpn_IOS_data.c		C++		1122	2			30	2	0.2

Need Total Subsystem size BEFORE modifications!

Adapted Code Example. Fill in all non-grayed out cells.

Grayed out cells **SHOULD NOT** be filled in.

Project Info Collector

- Defines the Program Effort Adjustment Factors (EAFs)
  - Can be defined for whole project, OR tailored per subsystem
- Program Attributes are program wide
- Hover help available for all entries

Project EAFs

	A	B	C	D	E	F	G	H	I	J
1									Don't	
2		Effort Multiplier Attributes	Very Low	Low	Nom	Hi	Very Hi	Xtra Hi	Know	Justification
3										
4	Product	Required SW Reliability (RELY)								
5		Database Size (DATA)								
6		Documentation (DOCU)								
7		Product Complexity (CPLX)								
8		Develop for Reuse (RUSE)								
9										
10	Platform	Execution Time Constraint (TIME)								
11		Main Storage Constraint (STOR)								
12		Platform Volatility (PVOL)								
13										
14	Personnel	Analyst Capability (ACAP)								
15		Programmer Capability (PCAP)								
16		Personnel Continuity (PCON)								
17		Applications Experience (APEX)								
18		Platform Experience (PLEX)								
19		Language/Tool Experience (LTEX)								
20										
21	Project	Use of SW Tools (TOOL)								
22		Multi-site Development (SITE)								
23										
24		Required Development Schedule (SCED)								

The effect of software failures is low, easily recoverable losses.

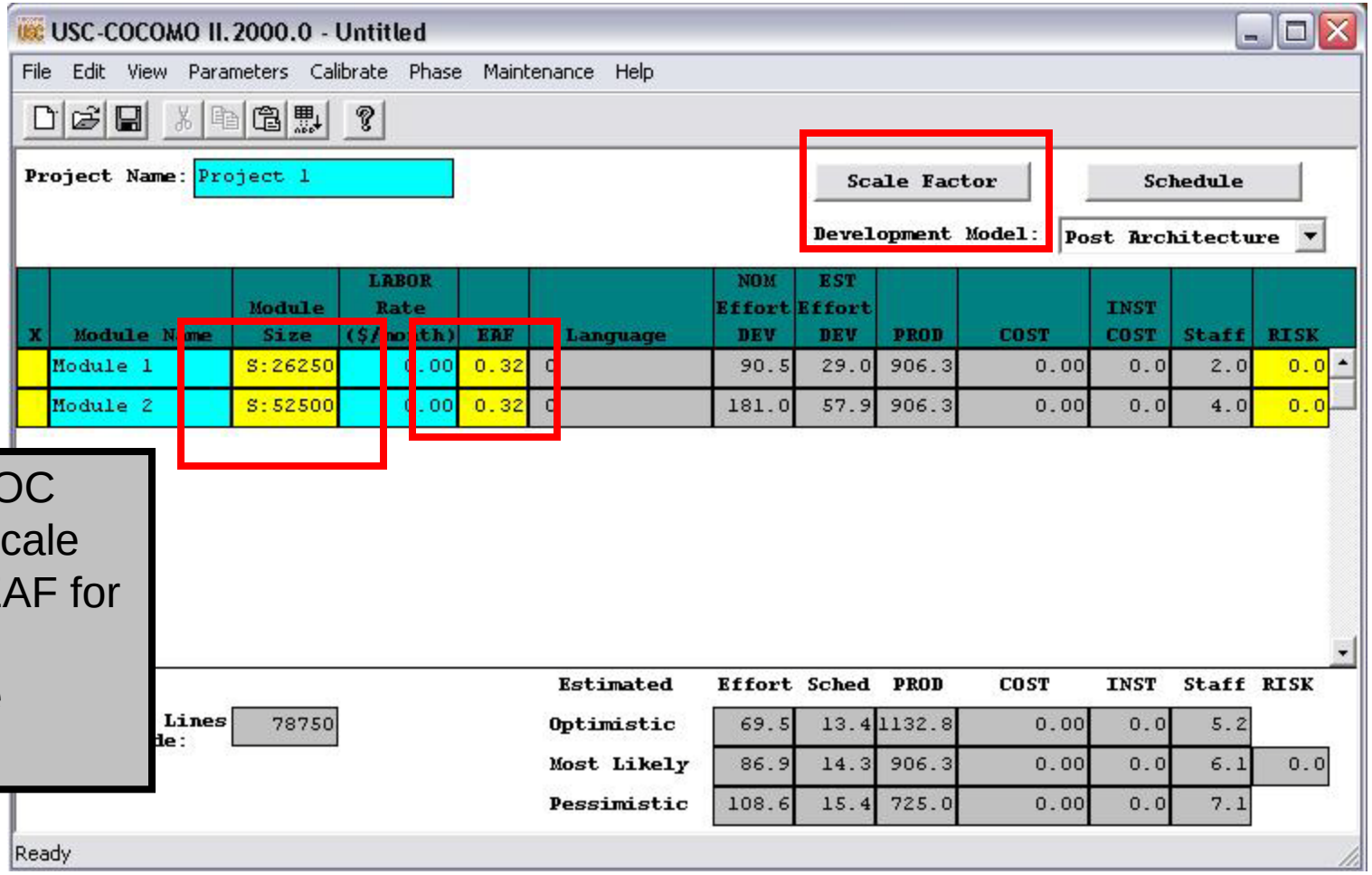
Project Attributes

	A	B	C	D	E	F	G	H	I
1								Don't	
2	Project Scale Factors	Very Low	Low	Nom	Hi	Very Hi	Xtra Hi	Know	Justification
3									
4	Precedentness (PREC)								
5	Devel Flexibility (FLEX)								
6	Architecture/Risk Resolution (RESL)								
7	Team Cohesion (TEAM)								
8	Process Maturity (PMAT)								
9									
10									

If the product is similar to previously developed projects, PREC is high. Use the table in the Instruction guide to develop the PREC rating for your program.

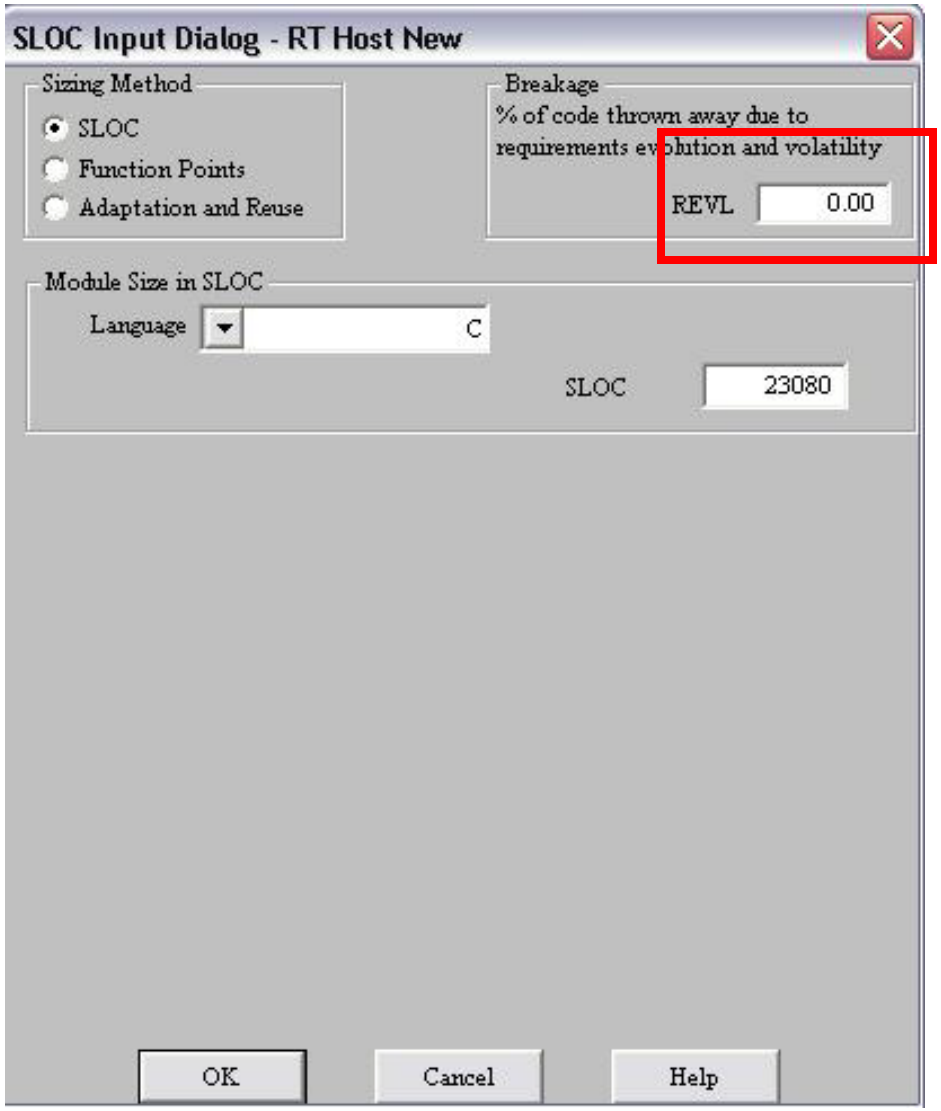
Creating Calibration File

- COCOMO II has a built in Calibration method
- A calibration file can be created by entering actual data (SLOC and Effort) for each project
- The project can be characterized by any number of modules representing the subsystems
- The data entered should reflect the actual SLOC produced (New, Adapted, Reused), the EAF's associated with those subsystems, and the Scale Factors for the Project as a whole



Enter the ESLOC data, project Scale Factors, and EAF for all modules comprising the project.

Enter all data for each subsystem as Equivalent SLOC (ESLOC), as calculated from the collector form. (REVL=0 for completed projects)

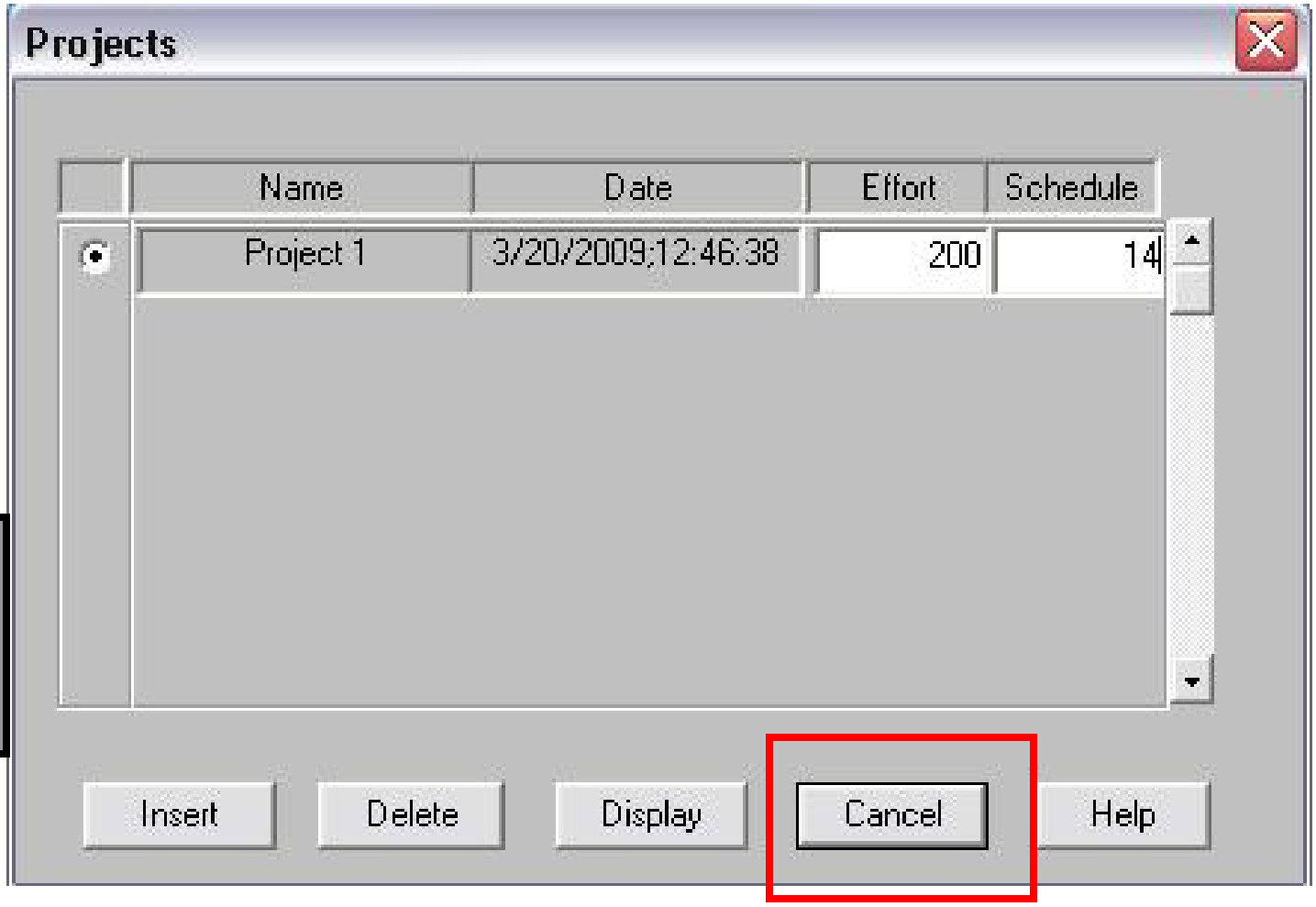


Project Effort Actuals

- Once a project has been defined, the effort and schedule actuals for that project need to be entered
- This is done by selecting the Project selection from the Calibrate pull-down
- The inputs are Effort and Schedule
  - Effort is defined in Person-Months (PM)
  - Schedule is input in Months duration

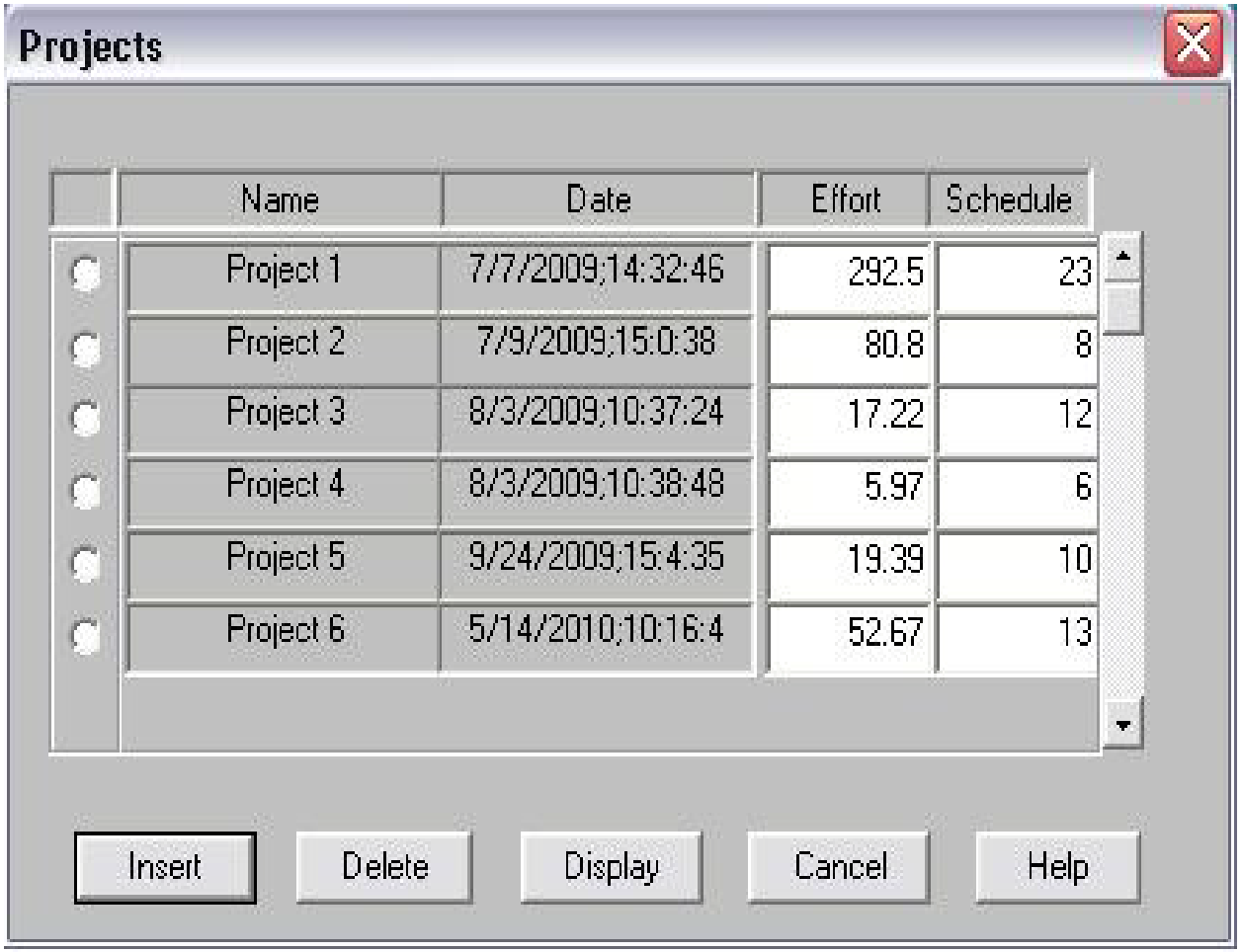
Enter the actual Project Effort in Person-Months, and Schedule in Months.

When completed with Project 1, select the Cancel button.



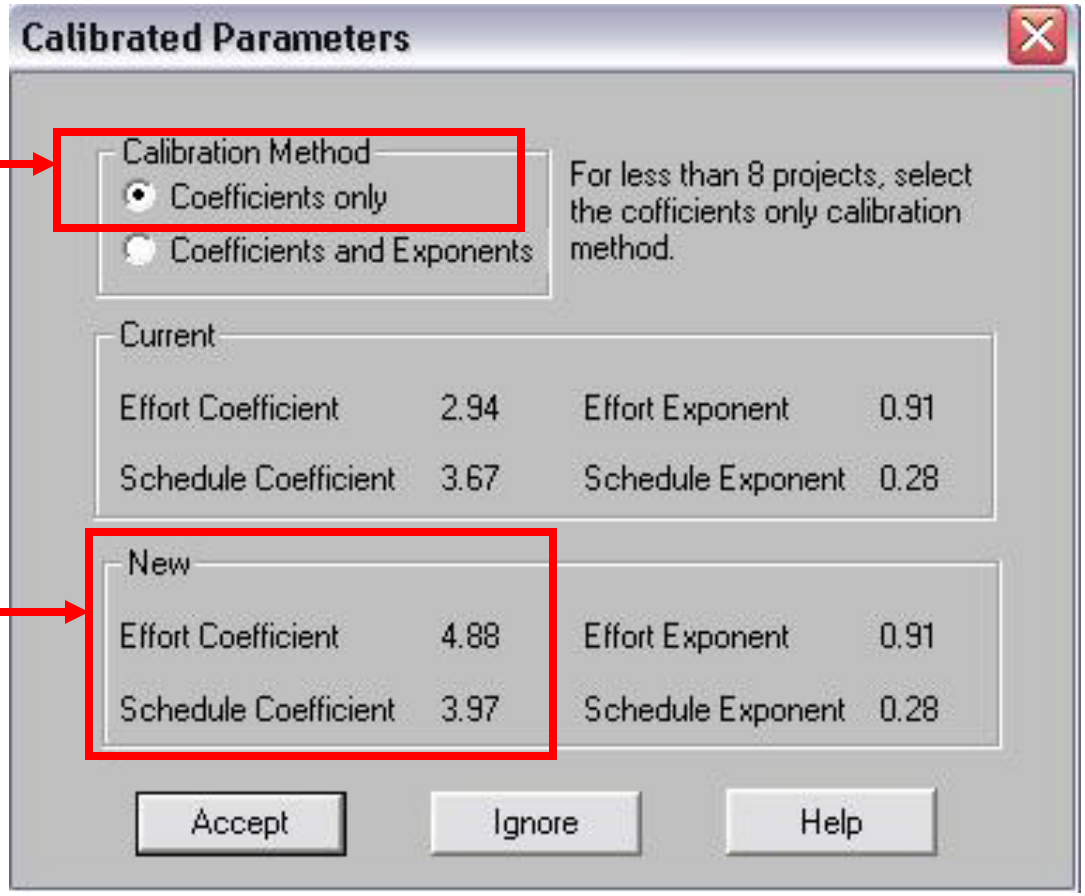
Repeat those same steps for each project.

End result will be a calibration file composed of many STS projects. This will produce more accurate effort estimates based on STS actuals.

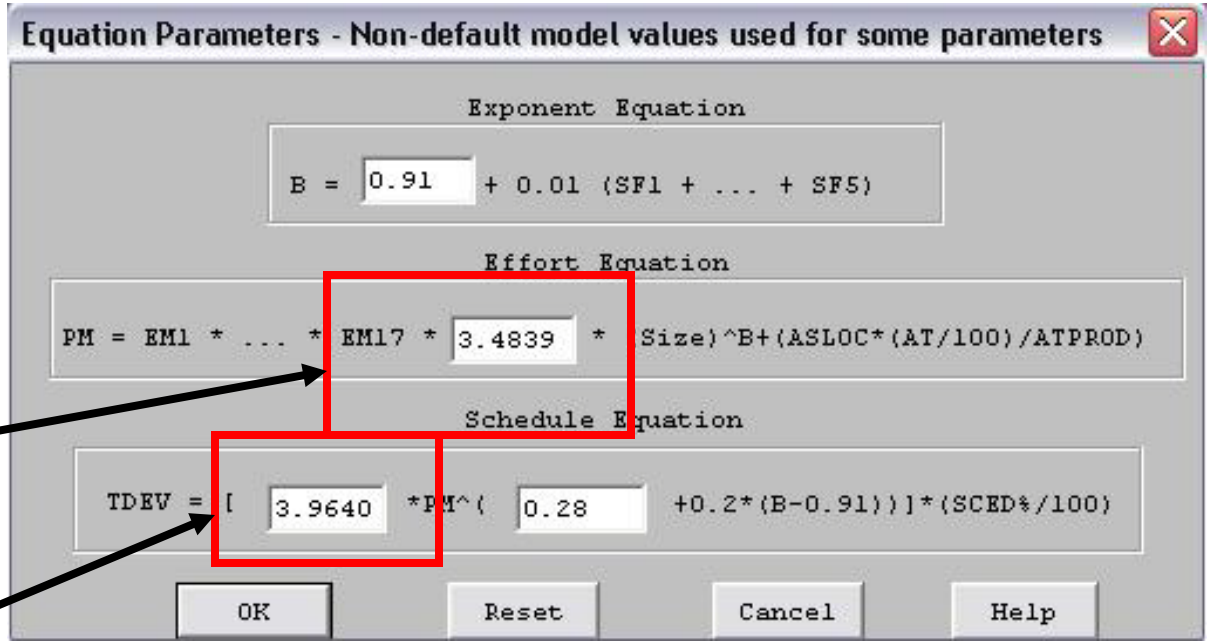


After selecting Compute, select Coefficients Only.

The New computed Coefficient(s) for calibration are shown.



After Accepting, New Coefficients are now part of the estimation equation.



A Effort Coefficient

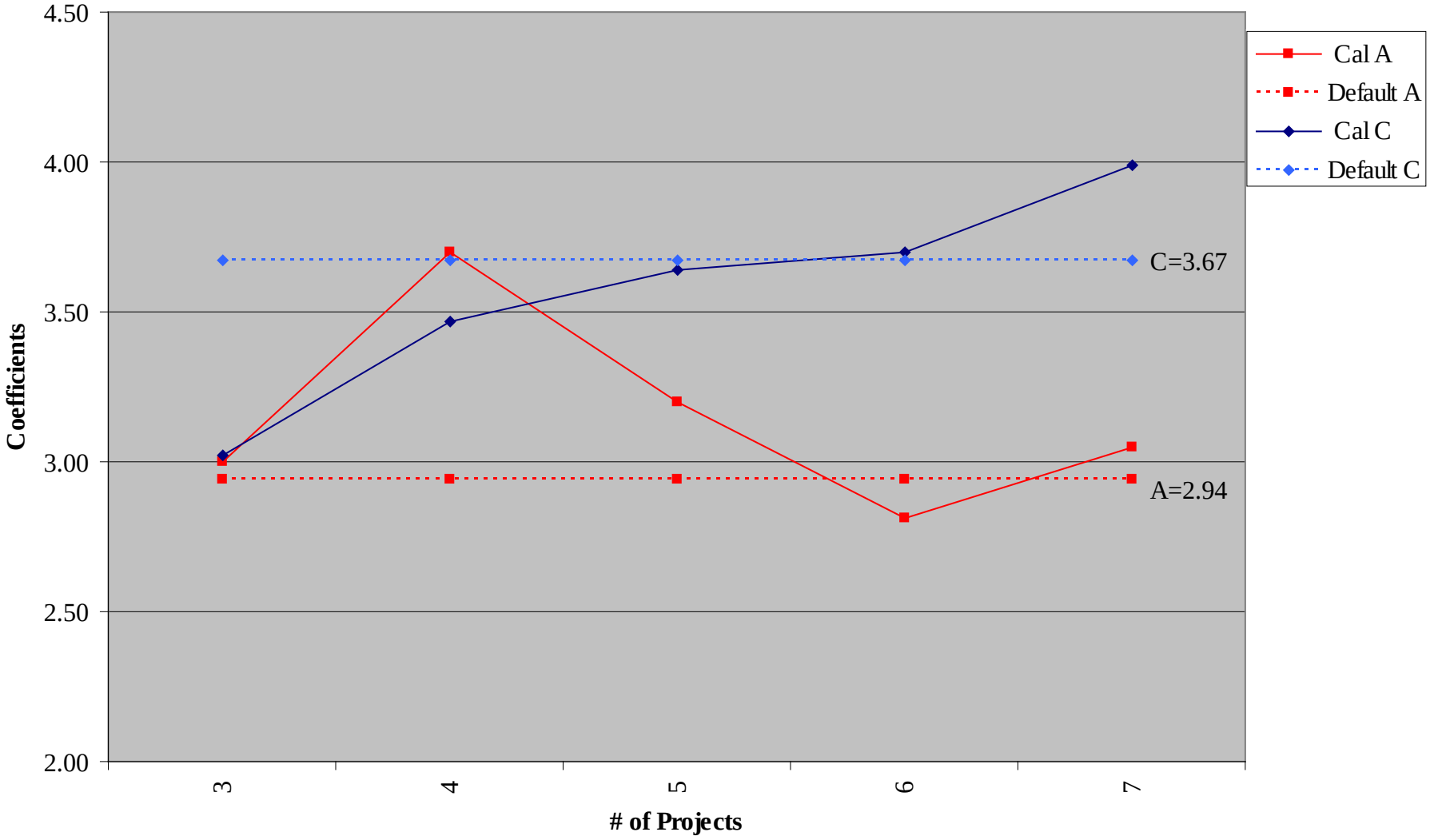
C Schedule Coefficient

Calibration Results

- Now we have several projects for calibration, what does that tell us?

	A	B	C	D	E	F	G	H	I	J	K
1			Actual	Cocomo	Cocomo 2000	Cocomo 2000	Cocomo 2000	Calibrated	Calibrated	Calibrated	Calibrated
2		Project	PM	2000	Plans + Req	Variance	Variance	Cocomo	Plans + Req	Variance	Variance
3				PM	PM	PM	(%)	PM	PM	PM	%
4											
5	1	Project 1	292.50	295.153	315.814	**23.31**	**7.97%**	282.091	301.837	**9.34**	**3.19%**
6	2	Project 2	80.80	80.895	86.558	**5.76**	**7.13%**	77.315	82.727	**1.93**	**2.38%**
7	3	Project 3	5.97	5.549	5.937	**-0.03**	**-0.55%**	5.304	5.675	**-0.29**	**-4.94%**
8	4	Project 4	17.22	7.273	7.782	**-9.44**	**-54.81%**	6.951	7.438	**-9.78**	**-56.81%**
9	5	Project 5	19.39	31.888	34.120	**14.73**	**75.97%**	30.477	32.610	**13.22**	**68.18%**
10	6	Project 6	52.67	105.944	113.360	**60.69**	**115.23%**	101.255	108.343	**55.67**	**105.70%**
11	7	Project 7									
12	8										
13		**Average**				**18.99**	**43.61%**			**15.04**	**40.20%**
14											
15			**Note: PM = 152 Hrs.**								
16											
17											

Calibration Coefficients



Variance Explanations

- Data Collection on 1st two programs done imprecisely
- Initial collector forms not as rigorous, or intuitive as current forms (Updated and more user friendly)
- More knowledge garnered from USC on COCOMO factors and definitions (flowed into Collectors)
  - 1st several projects lacked knowledge
- Project 4 mainly a HW project (SW only effort extracted)
- Only 6 projects (Really need >8 quality data projects)
  - Several more projects currently in work

STS COCOMO Templates

- 2 Templates are available:
- STS COCOMO Template Calibrated
  - STS Calibrated coefficients are part of this file
  - Effort Factors tailored to typical STS project
  - Examples of types of code estimates shown

- STS COCOMO Template Uncalibrated
  - Only Effort Factors tailored to typical STS project
  - Uses out-of-the-box calibration coefficients
  - Examples of types of code estimates shown

Summary

- STS SW Estimating getting better
  - More programs needed for calibration
  - End Users more knowledgeable in completing forms
- Collector Forms still evolving
  - Still fairly complex even with additional work
  - Becoming easier to populate
- Data flowed into database to auto-generate COCOMO project file
  - Generate estimates faster

Lessons Learned

- COCOMO model and factors very complex
  - Definitions of factors hard to understand
  - End user knowledge varies across programs
- Engineers/PMs perceive non-value added work
  - Collecting SLOC still an uphill battle with Projects
    - Automation only minimally helps
    - “What’s the charge #?”
  - See no immediate benefit to them
- WBS structure not aligned with model to accurately capture actuals
  - Added analysis time to align w/model
  - May lead to errors in effort
- Collectors help, but are not perfect.
  - Collectors improved with feedback from end users
  - Data collection more thorough now

Recommendations

- Standardized Tool Needed
  - Pick one and stick to it
  - Become educated on complexity of tool
- Single focal point for collecting SLOC and Actuals
  - Keep out of engineers hands
  - Speeds process of creating calibration projects
- Get Buy-in from all functional areas on benefits
  - Why does it benefit the corporation? -> More accurate estimates.
- Capture costs in way that aligns with SW Estimating model
  - Align WBS with model to accurately capture actuals