Lessons Learned in Estimating the Software Cost of a Ground Station with COTS Integration

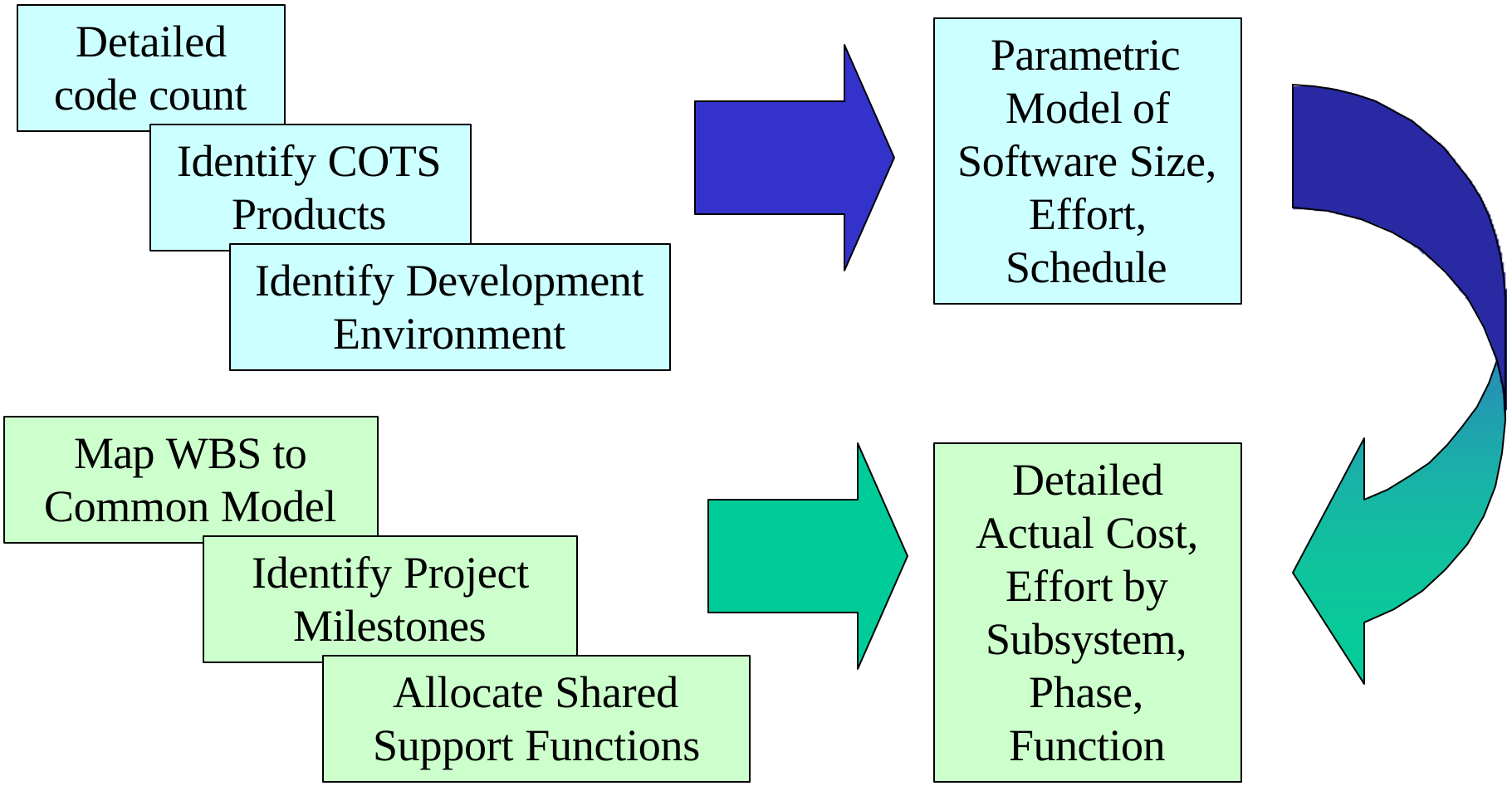
Kathy Bradford

22 February 2001

Short History of an Integrated COTS Procurement

- RFP requested a mostly COTS ground station
- Original proposal effort based on IR&D experience:
  - IR&D experience not well quantified
  - COTS integration not separately identified or estimated
  - Script, 4GL code was assumed to be 4x as productive, so estimated SLOC counts were divided by 4
- Project was a success - completion was on schedule
  - But actuals greater than bid
  - Adjusted actuals - enhancements still greater than bid
- How do we do better next time?

Standard TRW S&ITG Process for Auditing Software Productivity



Setting Up the Project Baseline

- Established 1 page outline of project
  - Major Milestones
    - ATP and DD250
    - Technical reviews
    - Incremental delivery schedule
    - Formal test events
  - IPT (product & functional) organization
    - Contractors, subcontractors
  - Major elements of cost (Product/Subsystems, Sites, Deliveries)
    - Major COTS elements by subsystem

Analysis of Actuals

- Map project work breakdown structure (WBS) to standard model
  - Standard WBS used to archive, compare completed projects
  - Has subsets by function for differentiating between products or subsystems
- Used map to resummarize effort, cost
  - Determine cost of each Product (subsystem)
  - Split development into H/W and S/W efforts
  - Provides framework for quantifying and analyzing effort by development phase
- Used time-phased actuals, program profile
  - Determine headcount by phase, product, function

Detailed Code Count: Will the real code stand up?

- Completed products captured in the project Configuration Management (CM) data base
- COTS Integration systems have LOTS of file types
- Differentiated contents by file extension:
  - Code include 3GL, 4GL source and scripts
  - Tables are info for COCOMO sizing (database size)

File Extensions by Type	No.
3GL (C, C++ awk)	6
4GL (screen builders, code builders)	4
Scripts (make, shell, vendor specific)	13
Tables (vendor specific)	12
Intermediates (bin, dat, o, out, etc.)	23
Total file types:	58

Code Counting Terminology, Methods

- Source files include 3GL, 4GL language; scripts such as make files, sh/csh/ksh, COTS-specific scripting languages
- LOC = Lines of code, defined by line terminator, but not a comment or a blank line
- SLOC = Source LOC, executable statement as defined in context of the language

			Lines of Code in 1000s (K)	
Product	CM Files	Source Files	LOC	SLOC
A	2,770	333	118.6	46.8
B	264	196	24.9	22.2
C	1,118	172	44.0	18.4
D	810	172	28.8	18.9
E	2,616	643	176.5	78.4
F	866	128	55.0	16.7
G	49	44	12.7	12.4
H	3,602	293	60.7	38.8
Total	12,095	1,981	521.2	252.7

Further identification required - differentiating New from Improved

- More terminology:
  - Reuse - source code created else where, and used with some reengineering and retest effort.
  - Modified - reuse code with some changes or additions to the code, as well as reengineering and retest effort.
  - Derivative code - multiple source files derived from one original with some % modification
  - Non-product code - tests or stubs, not part of final delivery
- Summarized, counted code inspected by IPT leads:
  - Identified reuse, derivative code products and original source
  - Non-product code identified and eliminated from counts
  - Verified derivative parameters (%redesign, %recode, %retest)

Interviews result in better definition of the real work performed

	Lines of Code in 1000s (K)				
Product	SLOC	New	Reuse	Derivative	ESLOC
A	46.8	12.8	34.0		16.6
B	22.2	22.2			22.2
C	18.4	16.1		2.3	16.3
D	18.9	18.9			18.9
E	78.4	69.6	8.8		69.8
F	16.7	16.7			16.7
G	12.4	4.9		7.5	5.9
H	38.8	33.0		5.8	33.8
Total	252.7	194.3	42.8	15.6	200.2

COTS products count, too.

- Terminology again:
  - COTS - a product purchased or supplied, which is used as delivered. Most products supply some well-defined customization interfaces; but most of capability is in proprietary, execute only code.
- IPT Leads verify COTS products integrated into product:
- COTS size in FPs; sources include:
    - Vendor (rarely)
    - Comparison to well-known product lines, knowledge bases
    - Entries in reference manual index, or function list
  - Best to compare results of methods 2 and 3.
- Product Functionality required (in % of total functions)
    - Search on well-known function names
    - Inspection of index for known elements
  - Product quality, support, knowledge characteristics
    - Maturity vs. volatility
    - Vendor responsiveness
    - Ease of use, familiarity

IPT leads also define the characteristics of the development environment, staff

- Questionnaires derived from popular models:
  - SEER-SEM, COCOMO II, COCOTS, Price-S
  - Staff experience levels
  - Host/Target complexity, stability/volatility
  - Requirements completeness, volatility

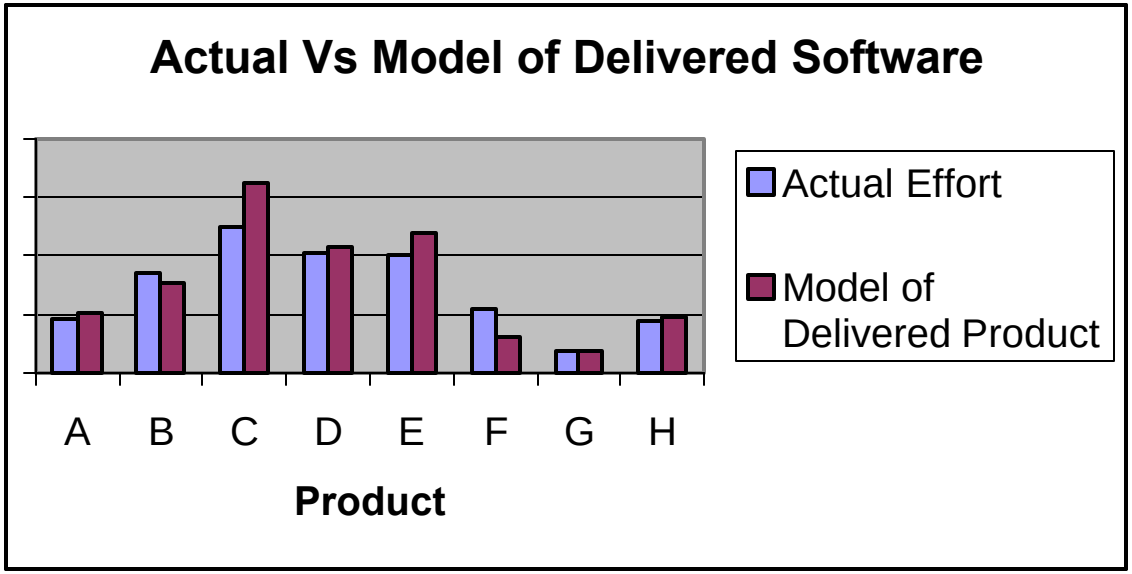
- Parametric model of delivered software product created:
  - Use new, reuse, derivative code sizes
  - Development environment parameters as supplied by IPT leads
  - Constraints as identified in program plan, IPT interviews
- Compare modeled effort and schedule to actuals

Actual vs Model of Delivered Product

A calibrated model provides:

- Validation of provided data
- Basis of estimate for future enhancements

Product	% Diff
A	12%
B	-9%
C	31%
D	4%
E	20%
F	-43%
G	0%
H	6%
Total	7%

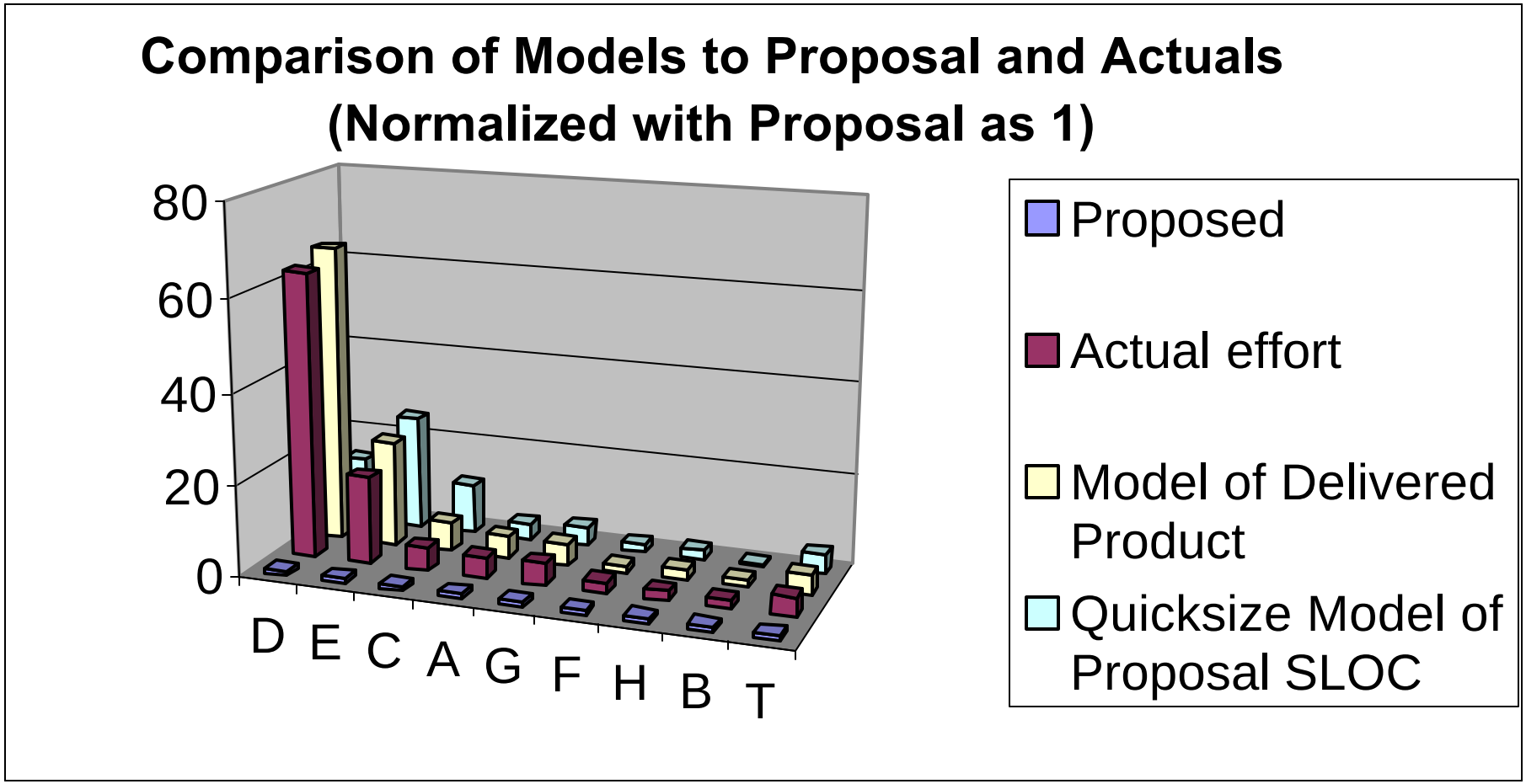


- Product F was forced to restart with alternate COTS product 28% into the schedule.
- Product C was severely constrained by staffing problems.
- Difference <2% with these corrections

Creating Modified Model of Original Proposal

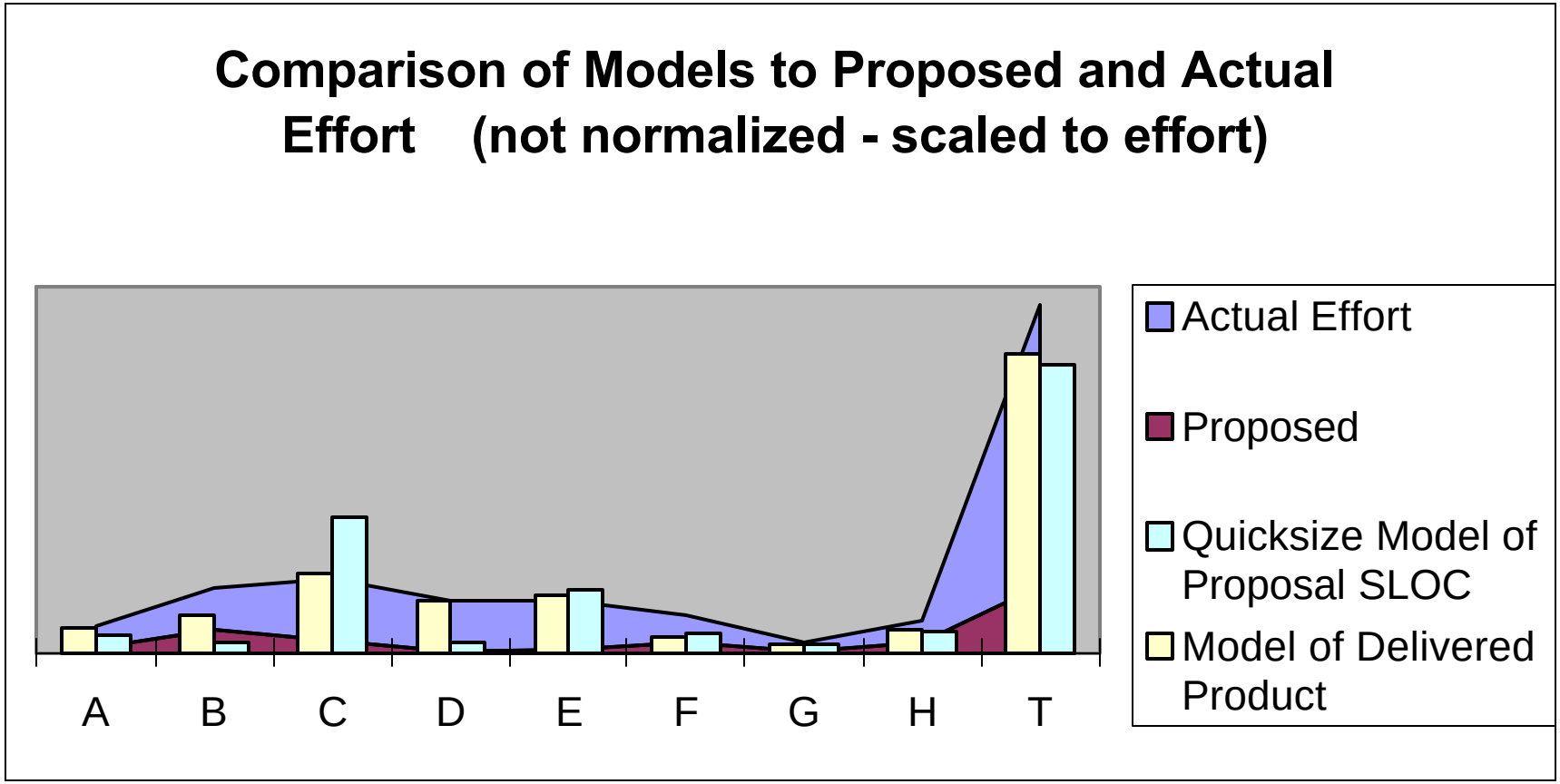
- Needed to verify the “improved” process
- Created another parametric model using:
  - Original 3GL, 4GL and script code from the proposal
    - In proposal, script and 4GL assumed to be 25% as hard, so were counted 4:1 in models
    - Corrected script, 4GL lines counts back to original estimate
  - COTS function counts, parameters from IPT interviews
    - Used SEER-SEM “quicksize” of COTS products to estimate impact of COTS integration
- Compared the proposal to the Quicksize model, and to the actuals and model of the delivered product

Comparison of Proposal to Other Methods



- 7 of 8 products are far closer to reality using the modified modeling methods.

Quicksize using COTS FP arrives at Effort and Schedule within 2% of Actuals



What Next?

- Using COTS FPs and experience parameters, created a much better model of this project
- More COTS integration programs are nearing delivery now; verify the process, productivity profiles on other efforts
- Use the results to help current program management:
  - Use results to help quantify effects of mid-term COTS fall-out
  - Use the history and models to better determine intermediate milestones
- Use the results to establish better cost-risk profiles for COTS integration programs