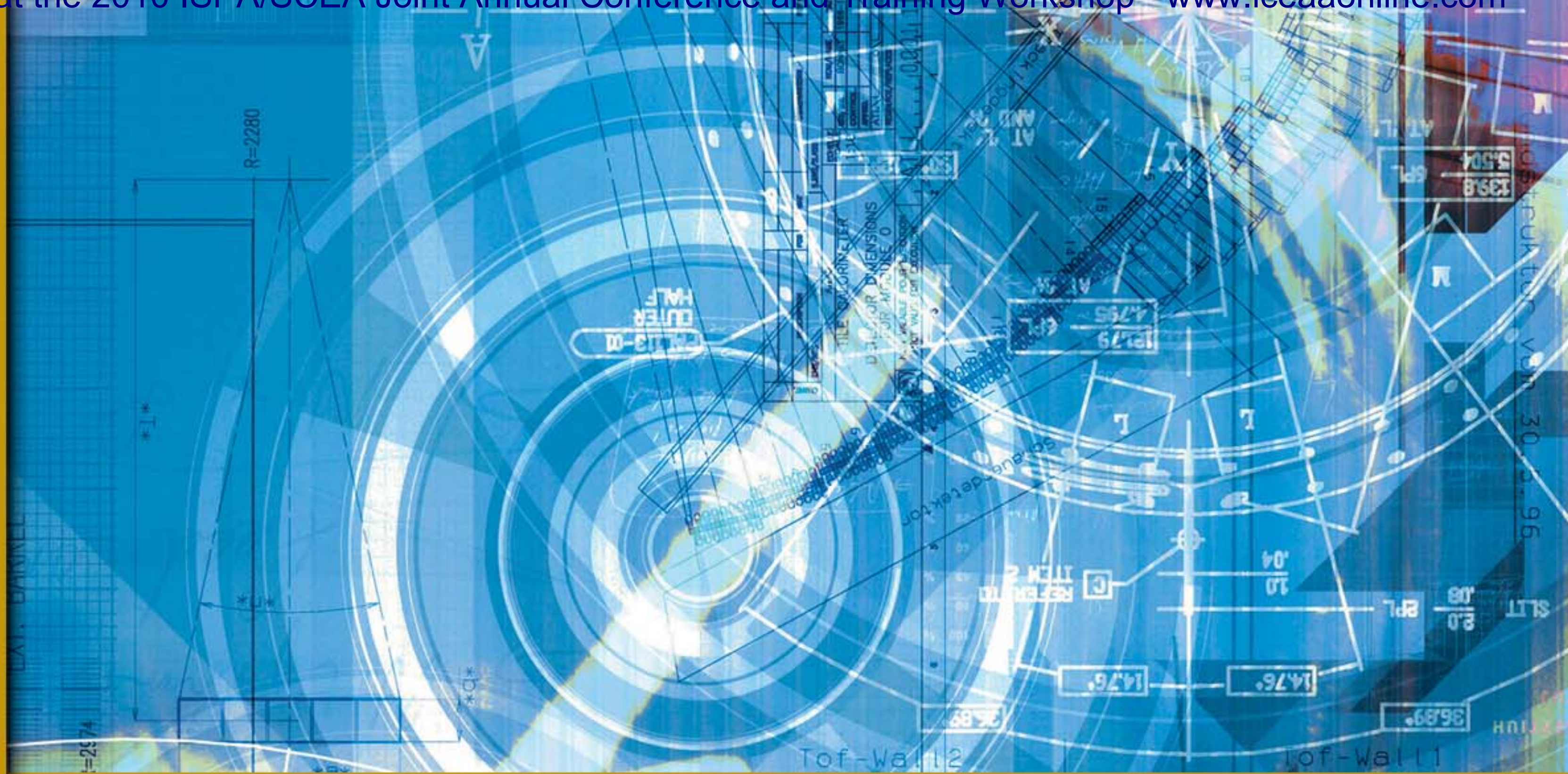
Presented at the 2010 ISPA/SCEA Joint Annual Conference and Training Workshop - www.iceaaonline.com



Software Schedule Estimating Linked to Cost Estimates
Presented to the SCEA Conference in San Diego, CA
June 10, 2010

Sarah Springer, MCR sspringer@mcri.com
Dan Feldman, MCR dfeldman@mcri.com
Greg Hogan, BAH hogan_gregory@bah.com

MCR, LLC
390 N. Sepulveda Blvd., Suite 1050
El Segundo, CA 90245

Critical Thinking. Solutions Delivered.

Booz | Allen | Hamilton

MCR CRITICAL THINKING. SOLUTIONS DELIVERED.

Outline

- Acknowledgments
- Problem Statement
- Purpose
- Data Used in Analysis
- Process
- Schedule-/Cost-Estimating Relationship (SER/CER) results
- Developing Comprehensive Cost Model
- Crosschecks
- Strengths and Weaknesses of SERs/CERs
- Example
- Conclusions
- Way Forward
- References
- Acronyms

Acknowledgments

- Dr. Stephen Book, MCR
- Ray Covert, MCR
- Dr. Neal Hulkower, MCR
- Rick Garcia, MCR

MCR CRITICAL THINKING. SOLUTIONS DELIVERED.

Problem Statement

- **Software development has become an increasingly important part of acquisition for the DoD**
- **"Weapon Systems Acquisition Reform Act" of 2009 (WSARA) identifies the need for consideration of trade-offs and impact of cost, schedule and performance in our estimates**

- **Need for comprehensive software cost and schedule models and visibility into those models**

MCR CRITICAL THINKING. SOLUTIONS DELIVERED.

Hypothesis

- Equivalent Source Lines of Code (ESLOC) is widely considered good quantitative measure of software size and effort
- ESLOC attempts to capture the relative effort of each software build, and is an obvious cost driver
- Resource loaded software schedules are also function of effort (i.e., ESLOC), and other factors such as development environment and constraints
- This implies relationships exist between ESLOC, cost, and duration
- In this study we developed a comprehensive joint cost and schedule model for software intensive Ground Programs

Why?

- Currently, academic and commercial software exist that estimates cost and schedule
  - Constructive Cost Model (COCOMO II)
  - Software Evaluation and Estimation of Resources- Software Estimation Model (SEER-SEM)
  - There are several others
- Why develop another software model that estimates cost and schedule?
  - Advantages of having insight into the data behind the model
  - Allows cost and schedule to be intrinsically “linked” without making any additional assumptions about the correlation between cost and schedule
  - Produces a joint probability curve

Data Used in Analysis and Crosschecks

- Aerospace Corporation data (circa 2003-2004)
  - NOTE: Data provided in published Aerospace Corp. TOR 2004 (8311)-1; not shown in this briefing
  - Data Records Contain Data Categorized by
    - Operating Environment
    - Application Type
    - Software Language
    - ESLOC, Developer Months, and Schedule Duration in Months
  - Data contain effort for software development phases of product design, code and Computer Software Configuration Item (CSCI) testing
    - Software requirements and system-level integration & test not included
- Results Crosschecked with the following
  - COCOMO II
  - Metrics developed from other data
  - Ground Program-level Schedule-Estimating Relationship (SER)

MCR CRITICAL THINKING. SOLUTIONS DELIVERED.

Process, Chart 1 of 2

- **Normalize Data**
  - Computer Languages normalized using "rule of thumb" calibration factors
    - Conversion for Ada to equivalent lines of C++: 0.96 (0.96 * Ada SLOC = C++ equiv SLOC)
    - Converted Java to equivalent lines of C++: 1.22
  - Stratify by operating environment, then by application type
- **Analyze schedule duration by operating environment**
  - Ground-only AND Ground plus Military Mobile data analyzed
  - Addition of Military Mobile data added degrees of freedom and results were found to be consistent with Ground data
  - 86 data points analyzed
  - Eight different application types

MCR Critical Thinking. Solutions Delivered.

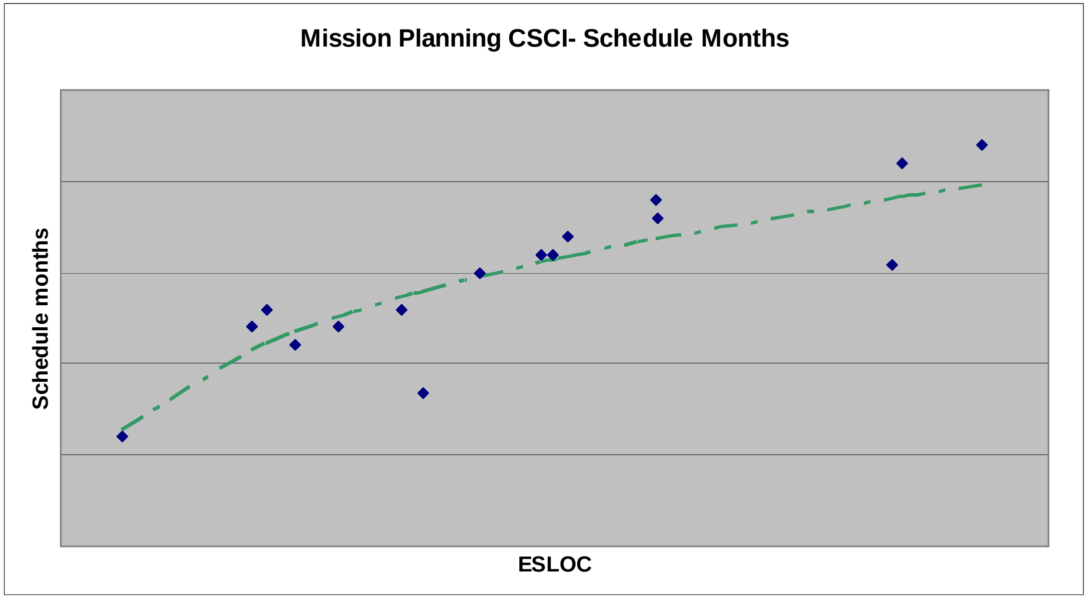
Process, Chart 2 of 2

- Regress data to derive relationships between ESLOC and both schedule (months) and effort (staff months)
  - Minimize standard percent error while constraining percent bias to zero (ZMPE)
  - Test variety of functional forms for every application type
  - Choose CER/SER with smallest standard error
- Crosscheck resulting CERs and SERs
- Develop joint cost and schedule model for Software development-intensive ground programs

MCR CRITICAL THINKING. SOLUTIONS DELIVERED.

Example: Mission-Planning SER

- Operating Environment: Ground and Military mobile
- Application Type: Mission-Planning
- Input is ESLOC; Output is schedule (months)



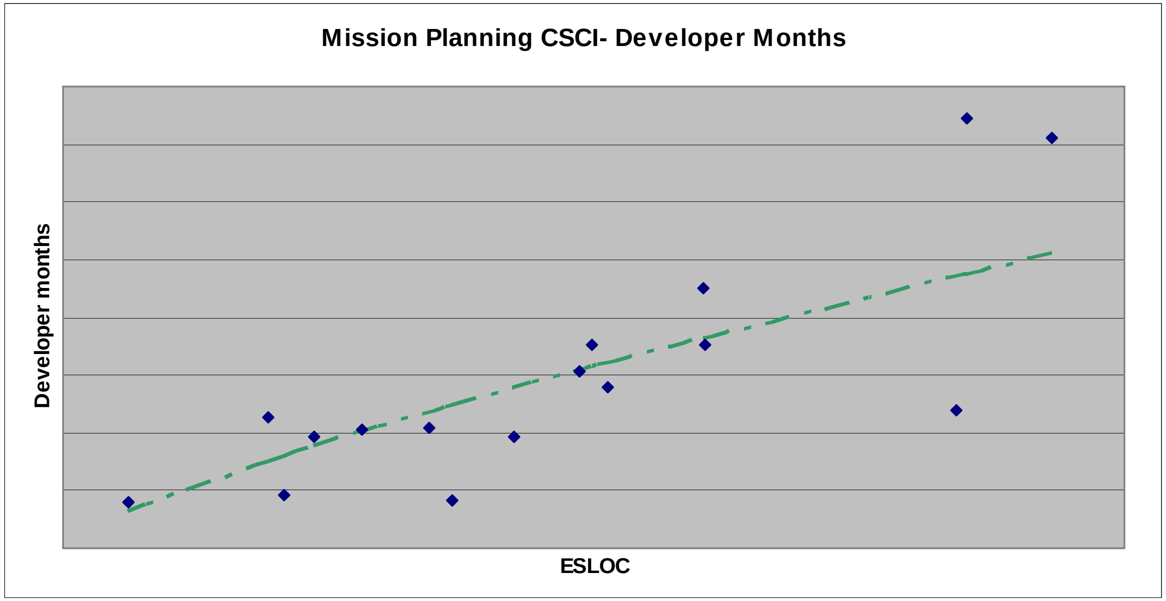
Schedule Months = $(ESLOC)^{0.2862}$

SER Statistics	
Number of points	16
Degrees of freedom	15
SER form	$y = x^b$
%SEE	10%
R^2	77%

MCR CRITICAL THINKING. SOLUTIONS DELIVERED.

Example: Mission-Planning CER

- Operating Environment: Ground and Military mobile
- Application Type: Mission-Planning
- Input is ESLOC; Output is effort (staff months)



Developer Months = 0.087 * (ESLOC) $^{0.7735}$

CER Statistics	
Number of points	16
Degrees of freedom	15
SER form	$y = ax^b$
%SEE	35%
R^2	64%

MCR CRITICAL THINKING. SOLUTIONS DELIVERED.

Scope of SERs/CERs, Chart 1 of 2

- Data excludes effort and schedule for Software Systems Engineering / Integration & Test (SEIT)
- Leveraged SEER / COCOMO II / DoD Data to account for all phases of software development
  - SEIT Effort factor: 15-55%, Schedule: 19-60%
    - Low end of range: portion of requirements and design accomplished in Phase A; prototype demonstrated
    - High end of range: new technology being developed
  - Model SEIT factor as range based on knowledge of program

System Requirements Design	Software Requirements Analysis	Preliminary Design	Detailed Design	Code and Unit Test	Component Integration and Test	SW/HW I&T	System I&T
		Activites covered by Aerospace data					

- Schedule dependencies captured in schedule build-up
  - Interface with outside programs or space vehicle
  - Task dependencies and schedule margins

- Program CSCIs need to be mapped into given Aerospace CSCI classifications
- Risk
  - Establish ESLOC Range: Low, Most Likely, and High Values for each CSCI
  - Uncertainty around SEIT add-on factor
  - To model risks, use %SEE (Standard Error of the Estimate) of the relationship
  - Other Technical Risks
    - Database is robust and contains variety of programmatic risks
    - Other technical risks should be added only when a valid reason exists as to why they are not covered in historical database
- Schedule and Cost both driven by ESLOC assumptions

MCR CRITICAL THINKING. SOLUTIONS DELIVERED.

Crosschecks

- **CERs / SERs crosschecked against the COCOMO II model**
- **Using planned Software build schedule to account for overlap / dependencies, CSCI level SERs combined to forecast total program duration**
  - **Resulting schedule estimate crosschecked with program-level SER developed from analogous ground programs**
  - **Found to be within 3% of build-up schedule for our program**
- **Crosschecks suggest the comprehensive schedule model developed from using SERs is consistent with historical data outside the Aerospace database**

MCR CRITICAL THINKING. SOLUTIONS DELIVERED.

Strengths and Weaknesses of SERs/CERs

- **Strengths**
  - The statistics are "good"
    - Under 30% SEE considered "great quality" CER/SER and between 30% and 60% SEE still has usefulness
    - SEEs for all SERs developed here are under 34% and all CERs are under 45%
  - Mathematically modeled links between cost and schedule
  - Not "black box": we have visibility into relationships
- **Weaknesses**
  - ESLOC is an imperfect measure of Software
  - Does not account for the effects of schedule "compression"
  - Schedules that are driven heavily by hardware development or other outside factors are most likely not an appropriate application for these SERs

Fictitious Program Example: ESLOC

- **Fictitious Example**
  - **4 CSCIs in 3 builds**
  - **Most Likely Values shown**
  - **Risk assessment discussed but specific bounds not shown**
- **Inputs/Risk**
  - **Most Likely ESLOC for each CSCI based on Cost Analysis Requirements Description (CARD) (program office) or independent assessment (oversight group)**
  - **Low and High values for ESLOC based on risk assessment**
  - **Triangular is typically the best distribution to use at this level**

CSCI Name	Classification	ESLOC Build 1	ESLOC Build 2	ESLOC Build 3
CSCI 1	C&C	40,000	40,000	40,000
CSCI 2	Msn Planning	35,000	35,000	35,000
CSCI 3	Signal Processing	60,000	40,000	60,000
CSCI 4	Support	30,000	30,000	30,000

MCR CRITICAL THINKING. SOLUTIONS DELIVERED.

Ficticious Program Example: Application of CERs/SERs

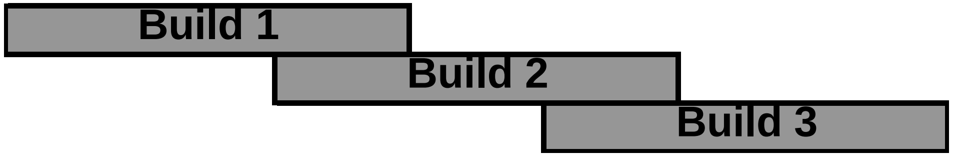
- **Classification of each CSCI determines which CER/SER to use**
  - **CERs and SERs are both applied to the ESLOC assumptions**
  - **Both schedule and software development cost are driven by ESLOC in model**
  - **Uncertainty around each CER/SER determined by standard error**

CSCI Name	Classification	ESLOC Build 1	ESLOC Build 2	ESLOC Build 3
CSCI 1	C&C	40,000	40,000	40,000
CSCI 2	Msn Planning	35,000	35,000	35,000
CSCI 3	Signal Processing	60,000	40,000	60,000
CSCI 4	Support	30,000	30,000	30,000

CRITICAL THINKING. SOLUTIONS DELIVERED.

Fictitious Program Example: Schedule

- **Build schedule**
  - 3 builds correspond with builds in previous chart
  - Each build's start date dependent on previous build
  - Linked in model to reflect program's build structure
  - Builds will often overlap
- **Schedule for each build determined by the "long pole"**
  - "Long pole" depends on size and complexity of each CSCI



MCR CRITICAL THINKING. SOLUTIONS DELIVERED.

Build Schedule

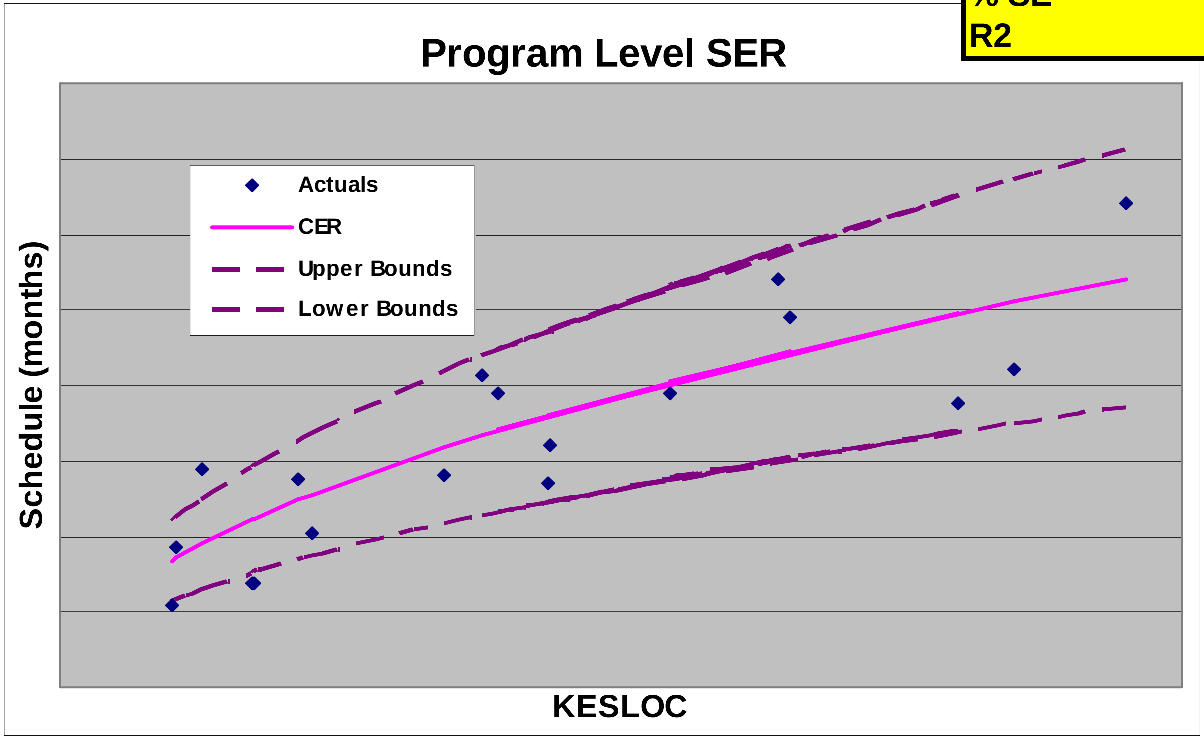
- **Important to crosscheck proposed build schedule**
  - **Often the schedule for a build is determined by the “long pole” CSCI (critical path)**
  - **With the integration process required with most S/W items the bulk of code in that build could also drive the schedule**
- **First build schedule from bottom up and then crosscheck it with a top-down estimate**
- **In this example the build schedule was crosschecked with a program-level SER developed from 18 analogous programs**

MCR CRITICAL THINKING. SOLUTIONS DELIVERED.

Program Level SER

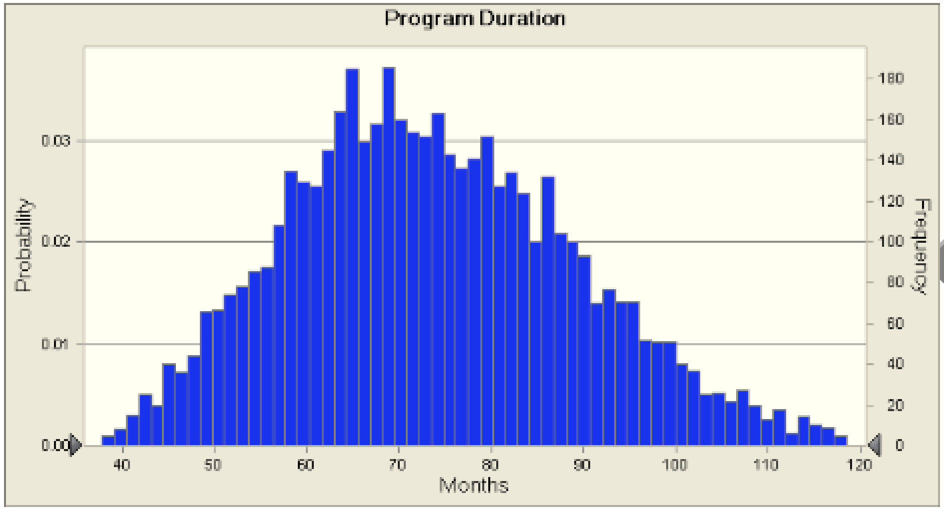
- Program-Level SER developed from 18 programs
- Data points included from a variety of contractors
- Include Space/Ground programs

Duration=2.1197*(KESLOC)^0.5210	
% Bias	0.00%
% SE	31.55%
R2	73.99%



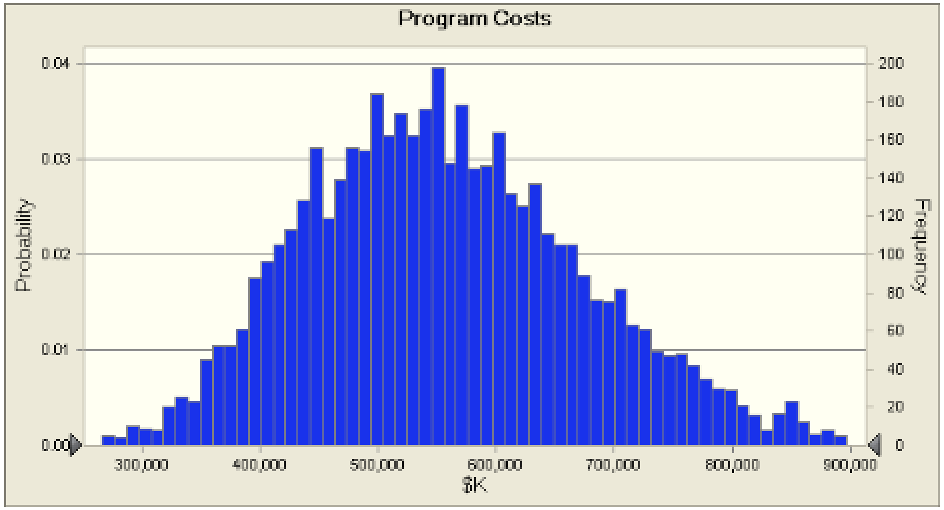
MCR CRITICAL THINKING. SOLUTIONS DELIVERED.

Results- Cost and Schedule Density Curves



Statistics:	Forecast values
Trials	5,000
Mean	74
Median	73
Mode	---
Standard Deviation	16
Variance	253
Skewness	0.3942
Kurtosis	2.92
Coeff. of Variability	0.2150

Statistics:	Forecast values
Trials	5,000
Mean	559,139
Median	549,457
Mode	---
Standard Deviation	120,385
Skewness	0.4752
Kurtosis	3.22
Coeff. of Variability	0.2153



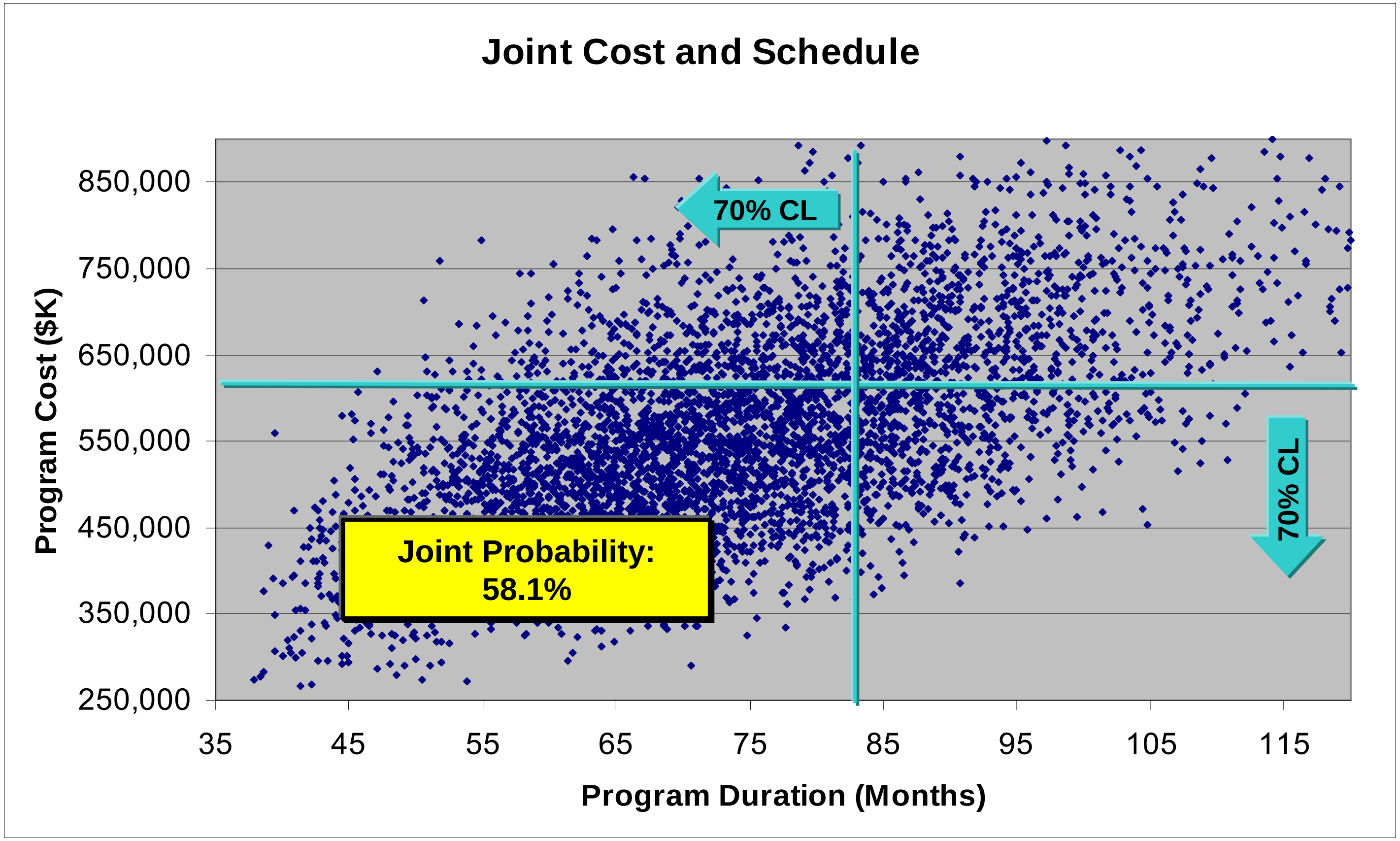
Crosschecks for Schedule in Example

- Schedule
  - Mean duration of all builds within range given by COCOMO II output
    - Post Architecture Model used
    - Nominal settings for all scale drivers and effort multipliers
  - Program Level SER
    - Resulting mean duration for example – 74 mos
    - Mean Program Level SER duration for example- 70 mos
    - Within the percent standard error for the SER
  - Other Program metrics

MCR CRITICAL THINKING. SOLUTIONS DELIVERED.

Results- Joint Cost and Schedule Estimate

- **Calculated cost to schedule correlation = 0.63**



70% cost and 70% schedule ≠ 70% Program!

CRITICAL THINKING. SOLUTIONS DELIVERED.

Conclusions

- Software-intensive programs that have a detailed schedule at the CSCI level might be able to apply these parametric modeling techniques
  - Primary estimating methodology
    - Modeling techniques useful as a primary methodology for programs in earlier stages
  - Crosscheck
    - Resource- loaded schedules should be crosschecked with a parametric model to ensure that the detailed schedule is consistent with historical data
- SERs not intended as a Planning Schedule
  - More aggressive schedule (such as an Integrated Master Schedule) is typically used for planning
  - But should have risk plans (margin) in place to cover schedule slips consistent with parametric estimate

Way Forward

- Analysis of additional operating environments
- Additional data collection and analysis of the data
- Examination of additional parameters (where possible)
- Incorporation of "Knee in the Curve" analysis for schedule compression
- Analysis of the impact of possible incremental development productivity decline
- SEIT Add-on factor specific to operating environment, application domain, and phase of program

MCR CRITICAL THINKING. SOLUTIONS DELIVERED.

References

- Gayek, J., et al., "Software Cost and Productivity Model," Aerospace Corporation, February 2004
- Boehm, B., et al., "Software cost estimation with COCOMO II" Englewood Cliffs, NJ: Prentice-Hall, 2000 (www.sunset.usc.edu/csse/research/COCOMOII/cocomo_main.html)
- SEER-SEM Manual, version 6.0.28, December 2002
- Covert, R. "Joint Cost, Schedule and Performance Risk Modeling Revisited" presented to Space Systems Cost Analysis Group, 2009

CRITICAL THINKING. SOLUTIONS DELIVERED.

Acronyms

- CARD= Cost Analysis Requirements Description
- CER = Cost Estimating Relationship
- COCOMO = Constructive Cost Model
- CSCI= Computer Software Configuration Item
- ESLOC= Equivalent Source Lines of Code
- SEE = Standard Error of the Estimate
- SEER-SEM = Software Evaluation and Estimation of Resources- Software Estimation Model
- SER = Schedule Estimating Relationship
- SRDR= Software Resources Data Report
- S/W = Software
- WSARA = Weapon Systems Acquisition Reform Act
- ZMPE= Zero Percent Bias, Minimum Percent Error

CRITICAL THINKING. SOLUTIONS DELIVERED.

Backup

MCR CRITICAL THINKING. SOLUTIONS DELIVERED.

SER Results

- **Operating Environment of interest: Ground programs**
- **Input is ESLOC; Output is schedule (months)**
- **Percent standard error ranges from 20-34%**
- **Data suggest economies of scale for software development schedules**
  - **Similar to COCOMO's nominal equation**

	Cmd Cntl	Database	Msn Planning	Signal Proc	OS Exec	Support	Test
# Points	19	7	16	8	7	18	11
DF	17	6	15	6	5	17	9
SER Form	$y = a+bx$	$y = x^b$	$y = x^b$	$y = a+bx$	$y = a+bx$	$y = x^b$	$y = ax^b$
%SEE	2.4*	11.1%	10.4%	27.2%	21.7%	33.9%	20.3%
R^2	95%	52%	77%	80%	53%	29%	64%
a	9.1903			11.5509	19.7798		0.1410
b	0.0003	0.2815	0.2862	0.0002	0.0002	0.2689	0.4879

**Command & Control is an additive error SER and is expressed as Standard Error, not %SEE*

NOTE: Data in published Aerospace Corp. TOR 2004 (8311)-1; not shown in this briefing

MCR CRITICAL THINKING. SOLUTIONS DELIVERED.

CER Results

- **Operating Environment of interest- Ground programs**
- **Input is ESLOC; Output is effort (staff months)**
- **Percent standard error ranges from 20-45%**
- **Data suggests diseconomies of scale for software development cost**
  - **Similar to COCOMO's nominal equation**

	Cmd Cntl	Database*	Msn Planning	Signal Proc*	OS Exec	Support	Test	Combination of Data**
# Points	19	7	16	8	7	18	11	86
DF	17	6	15	6	5	17	9	84
CER Form	$y = ax^b$	$y = a+bx$	$y = ax^b$	$y = a+bx$	$y = a+bx$	$y = a+bx$	$y = ax^b$	$y = ax^b$
%SEE	20%	69.5	35%	131.2	41%	45%	43%	45%
R^2	96%	55%	64%	66%	92%	83%	67%	68%
a	0.0012	28.7875	0.0870	65.9943	11.5000	34.6947	0.0004	0.0163
b	1.1633	0.0081	0.7735	0.0040	0.0113	0.0042	1.2930	0.9249

****Additive error CER and is expressed as Standard Error, not %SEE***

***Combination of the data used in the separate CERs*

NOTE: Data in published Aerospace Corp. TOR 2004 (8311)-1; not shown in this briefing