Improved Method for Predicting Software Effort and Schedule

Wilson Rosa
Barry Boehm
Ray Madachy
Brad Clark
Joseph P Dean
Cheryl Jones and John McGarry

2014 ICEAA Professional Development & Training Workshop
June 11, 2014

Changing the Culture to Win the War on Cost

Purpose

- Present a set of effort and schedule estimating relationships for predicting software development projects using empirical data from 317 very recent US DoD programs.
  - Equations are simpler and more viable to use for early estimates than traditional parametric cost models.
  - Provides the statistics and regression models upon which detailed estimates are based.
  - The methods are applicable to all industry sectors.

✓ Analysis results will be discussed in this presentation.

Changing the Culture to Win the War on Cost 2

OUTLINE

- Research Method
- Data Demographics
- Software Productivity Benchmarks
- Effort and Schedule Estimation Models
- Conclusion
- Backup

Research Method

Instrumentation

- Questionnaire:
  - Software Resource Data Report" (SRDR) (DD Form 2630)
- Source:
  - Defense Cost Analysis Resource Center (DCARC) website:
    http://dcarc.cape.osd.mil/Files/Policy/2011-SRDRFinal.pdf
    http://dcarc.cape.osd.mil/Files/Policy/Final_Developer_Report.xlsx
- Content:
  - Allows for the collection of project context, company information, requirements, product size, effort, schedule, and quality

Data Collection and Validation

- Initial Dataset
  - 474 records fully reviewed using GAO Best Practices
  - 157 were excluded based on the following limitations:
    - Inadequate information on reused and modified code
    - Projects cancelled or terminated before delivery
    - Inaccurate effort and schedule data
    - Same duration (start and end dates) across software projects/components
    - Missing effort or schedule data on more than 2 activities
    - Duplicate records or submissions
    - Estimates At Completion vice Actual Data
    - Data reported at project level vice CSCI
- Final Dataset
  - 317 projects included in the analysis as these passed quality inspection

Data Normalization and Analysis Workflow

- Data was normalized to “account for cost and sizing units, mission or application, technology maturity, and content so they are consistent for comparisons” (source: GAO)

Segment Data by Operating Environment
↓
Segment Data by Application Type
↓
Convert Size to Logical Count
↓
Normalize to Equivalent Size
↓
Analyze with Descriptive Statistics
↓
Select Best Model Form

Changing the Culture to Win the War on Cost 7

Segment Data by Operating Environment

Operating Environment		Examples
Ground Site	Fixed	Command Post, Ground Operations Center, Ground Terminal, Test Faculties
	Mobile	Intelligence gathering stations mounted on vehicles, Mobile missile launcher
Ground Vehicle	Manned	Tanks, Howitzers, Personnel carrier
	Unmanned	Robots
Maritime Vessel	Manned	Aircraft carriers, destroyers, supply ships, submarines
	Unmanned	Mine hunting systems, Towed sonar array
Aircraft	Manned	Fixed-wing aircraft, Helicopters
	Unmanned	Remotely piloted air vehicles
Ordinance and Missile	Unmanned	Air-to-air missiles, Air-to-ground missiles, Smart bombs, Strategic missiles
Space	Manned	Passenger vehicle, Cargo vehicle, Space station
	Unmanned	Orbiting satellites (weather, communications), Exploratory space vehicles





Changing the Culture to Win the War on Cost 8

Segment Data by Application Type: Overview

- Application types are groups of application domains that are environment independent, technology driven, and are characterized by 13 COCOMO product attributes.
- SRDR dataset was segmented into 14 Application Types to increase the accuracy of estimating cost and schedule:

1. Sensor Control and Signal Processing
2. Vehicle Control
3. Real Time Embedded
4. Vehicle Payload
5. Mission Processing
6. System Software
7. Telecommunications
8. Process Control
9. Scientific Systems
10. Mission Planning
11. Training
12. Test Software
13. Software Tools
14. Intelligence & Information Systems

If you segment your dataset by "Application Type", you have already captured most of the COCOMO Effort Multipliers

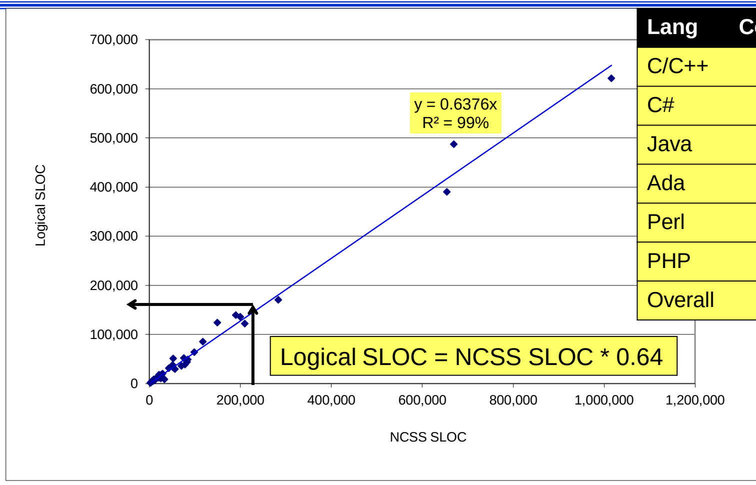
Segment Data by Application Type: Taxonomy

- 37 SEER-SEM application domains were stratified into 14 general complexity zones called Application Types

SEER-SEM Application Domains	Application Type
Radar, Signal Processing	Sensor Control & Signal Processing
Flight Systems (Controls)	Vehicle Control
Flight Systems (Payload)	Vehicle Payload
Embedded Electronics/Appliance, GUI (cockpit displays), Robotics	Real Time Embedded
Command/Control	Mission Processing
Communications, Message Switching	Telecommunications
Process Control	Process Control
Device Driver, System & Device Utilities, OS/Executive	System Software
Training / CBT / CAI	Training
Business Analysis Tool, CAD, Software Development Tools	Software Tools
Diagnostics, Testing Software	Test Software
Expert System, Math & Complex Algorithms, Simulation, Graphics	Scientific Systems
Mission Planning & Analysis	Mission Planning
Database, Data Mining, Data Warehousing, Financial Transactions, GUI, MIS, Multimedia, Relational/Object-Oriented Database, Transaction Processing, Internet Server Applet, *Report Generation,* Office Automation	Intelligence & Information Systems

See slides 36& 37 for Application Type definitions

Convert Size (NCSS*) to Logical Count



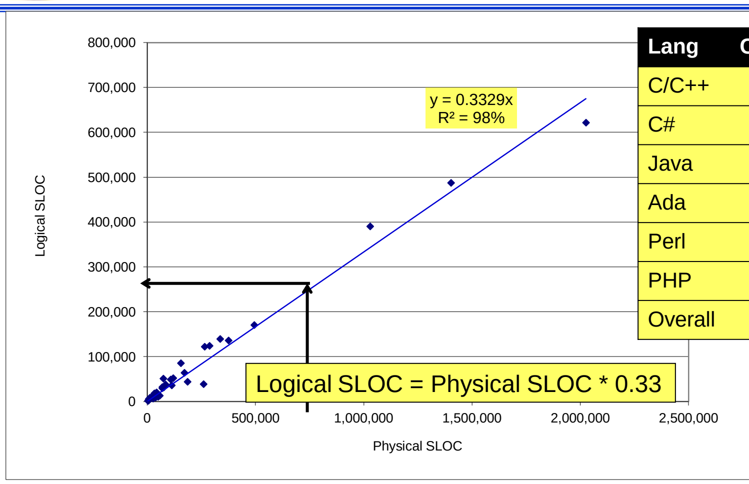
Lang	Conversion
C/C++	0.61
C#	0.61
Java	0.72
Ada	0.52
Perl	0.70
PHP	0.66
Overall	0.64

*NCSS = Non-commented source statement

Rosa W., Boehm B., Clark B., and Madachy R. (2012). Domain-Driven Software Cost Estimation. *27th International Forum on COCOMO® and Systems/Software Cost Modeling.* University of Southern California.

Changing the Culture to Win the War on Cost 11

Convert Size (Physical) to Logical Count



Lang	Conversion
C/C++	0.32
C#	0.35
Java	0.52
Ada	0.25
Perl	0.25
PHP	0.44
Overall	0.33

Rosa W., Boehm B., Clark B., and Madachy R. (2012). Domain-Driven Software Cost Estimation. *27th International Forum on COCOMO® and Systems/Software Cost Modeling.* University of Southern California.

Changing the Culture to Win the War on Cost 12

Normalize to Equivalent Size

- **Logical SLOC normalized to Equivalent SLOC (ESLOC) to reflect the actual degree of work involved:**

Formula:

$$ESLOC = New\ SLOC + Modified\ SLOC*AAF_M + Reused\ SLOC*AAF_R + Generated\ SLOC*AAF_G + Converted\ SLOC*AAF_C$$

Where:

$$AAF_i = 0.4*DM + 0.3*CM + 0.3*IM$$

And:

AAF	=	Adaptation Adjustment Factor
i	=	Refers to the size type: Modified (M), Reuse (N), Generated (R), Converted (C)
DM	=	Design Modified (DM), also known as re-design
CM	=	Code Modified (CM), also known as re-code
IM	=	Integration Modified (IM), also known as re-test

- ✓ Formula adapted from COCOMO II Reuse Model
- ✓ Model Input Parameters (DM, CM, IM) provided by Data Sources (System Developers)

Analyze the Data

- Data is analyzed using the following taxonomy

Operating Environment

Application Type	Ground Fixed	Ground Vehicle	Space unmanned	Maritime Vessel	Aircraft Manned	Aircraft Unmanned	Ordinance & Missile	**Total**
Software Tools	1	0	0	0	5	2	0	8
Mission Planning	20	0	0	0	0	0	0	20
Intel and Information Systems	11	2	0	0	1	0	0	14
Scientific	10	1	0	1	6	0	1	19
System	13	3	0	3	6	0	0	25
Telecommunications	22	2	0	22	1	0	0	47
Test Software	6	0	0	4	1	0	0	11
Real-Time Embedded	21	3	0	5	20	3	5	57
Mission Processing	16	0	0	3	9	1	5	34
Vehicle Control	0	14	0	0	9	1	3	27
Vehicle Payload	0	0	1	1	9	2	5	18
Sensor Ctrl & Signal Processing	14	1	1	3	3	9	6	37
Total	134	26	2	42	70	18	25	**317**

When the dataset is grouped by Application Type and Operating Environment, the impact accounted for by many COCOMO II model drivers are considered

Model Reliability and Validity

- Accuracy of the Models verified using seven different measures:

Measure	Symbol	Description
Standard Error	SEE	Standard Error of the Estimate is a measure of the difference between the observed and CER estimated effort. The SEE is to linear models as the standard deviation is to a sample mean.
Coefficient of Variation	CV	Percentage expression of the standard error compared to the mean of dependent variable. A relative measure allowing direct comparison among models.
Mean Absolute Deviation	MAD	Measures the average percentage by which the regression overestimates or underestimates the observed actual value. Mitigates against the “cancellation” effect from the sign and magnitude of a single % error.
Anderson-Darling test's p-value	AD p-value	Examines whether the dataset follows a normal distribution. The use of non-linear regression is appropriate when AD p-value is greater than 0.05, as there is evidence that the data do not follow a normal distribution.
Variance Inflation Factor	VIF	Indicates whether multicollinearity (correlation among predictors) ispresent in a multi-regression analysis. Multicollinearity is problematic because it can increase the variance of the regression coefficients, making them unstable and difficult to interpret.
Coefficient of Determination	R^2	The Coefficient of Determination shows how much variation in dependent variable is explained by the regression equation. Not applicable for Non-Linear regression.
F-test	F-test	The value of the F test is the square of the equivalent t test; the bigger it is, the smaller the probability that the difference could occur by chance. Not applicable for Non-Linear regression.

Select Best Fit Model: Effort

- Three effort model forms were examined for each dataset

$PM = A*Size^B$ — Log-Linear Model

$PM = C + A*Size^B$ — Non-Linear Model 1

$PM = C + Size^B$ — Non-Linear Model 2

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	=	B-exponent (Normally greater than 1, indicating diseconomies of scale)
C	=	Fixed level of effort support activities (in Person-Months)

- Rules of Thumb for Selecting Best Model

Measure	Rules of Thumb
# Observations	> 12
CV	≤40%
MAD	≤40%
R^2	> 60%

Select Best Fit Model: Schedule

- Two schedule model forms were examined for each dataset

$$TDEV = A * PM^{F}$$

COCOMO 81 Model

$$TDEV = A * Size^{B} * FTE^{C}$$

Non-Linear Model

Where

TDEV	=	Time (in months) to develop the Software Product
Size	=	Software Size in Equivalent Source Lines of Code (ESLOC)
FTE	=	Full Time Equivalent (FTE) Staffing Levels
PM	=	Total Estimated Effort in Person-Months (PM)
A	=	is a duration constant
B	=	Scaling factor to account for changing productivity as size increases,
C	=	C-Scaling Factor accounts for the non-linear relationship between increasing staffing levels and shortening development time, TDEV
F	=	Scaling factor for effort changes

- Rules of Thumb for Selecting Best Model

Measure	Rules of Thumb
# Observations	> 10
C-Scaling Factor	< 0.0
MAD	≤40%
CV	≤40%

Changing the Culture to Win the War on Cost 17

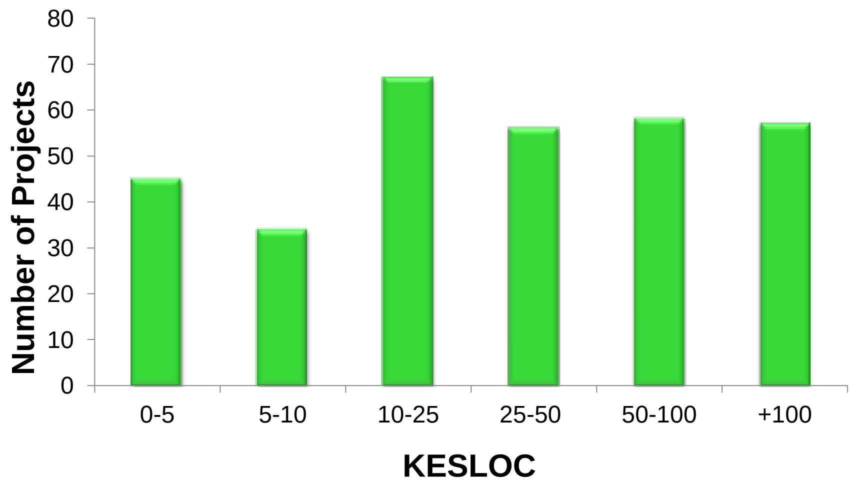
DATA DEMOGRAPHICS

Changing the Culture to Win the War on Cost

Age of Data

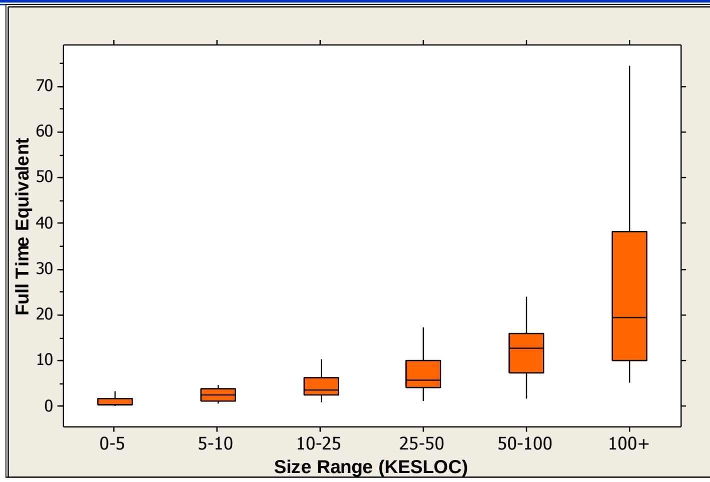
Changing the Culture to Win the War on Cost 19

Software Size



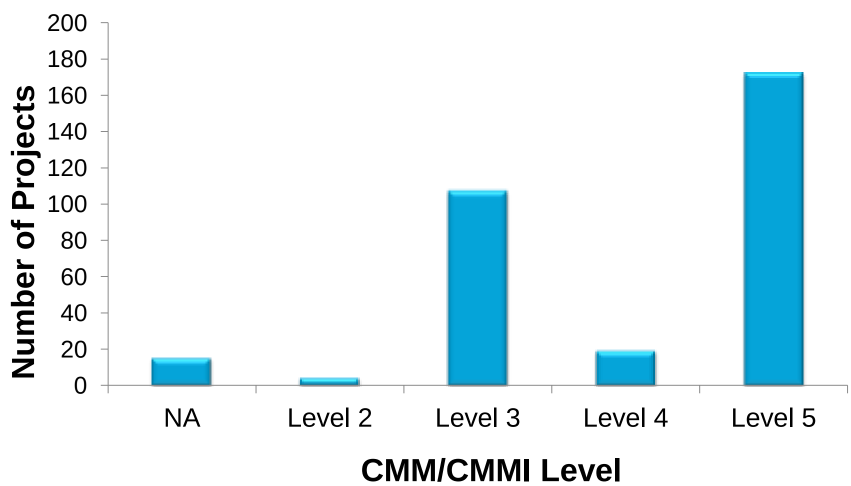
Changing the Culture to Win the War on Cost 20

Staff Level: Full Time Equivalent



- Appropriate for crosschecking staff levels prior to contract award

Software Process Maturity



Effort Distribution (%)

Effort Distribution by Activity by Application Type

Application Type	Req. Analysis	Archit. & Design	Coding & Testing	SW/SYS Integration	Qualification Testing	DT&E	Other Develop. Effort
Mission Planning	9%	12%	30%	6%	5%	10%	28%
Intelligence and Information Systems	6%	18%	29%	6%	5%	9%	26%
Scientific	9%	13%	27%	15%	8%	5%	23%
System	10%	16%	26%	17%	5%	2%	24%
Telecommunications	20%	14%	21%	12%	11%	2%	20%
Real-Time Embedded	10%	15%	27%	18%	6%	4%	20%
Vehicle Control	14%	14%	19%	15%	12%	7%	18%
Mission Processing	9%	12%	29%	13%	11%	4%	21%
Sensor Control and Signal Processing	6%	12%	21%	17%	12%	5%	27%
Composite	10%	14%	25%	13%	8%	5%	23%

Application/Uses

- **Appropriate for allocating resources across Software Activities**
- **Appropriate for normalizing inconsistent effort data**

Changing the Culture to Win the War on Cost 23

SOFTWARE PRODUCTIVITY BENCHMARKS

Changing the Culture to Win the War on Cost

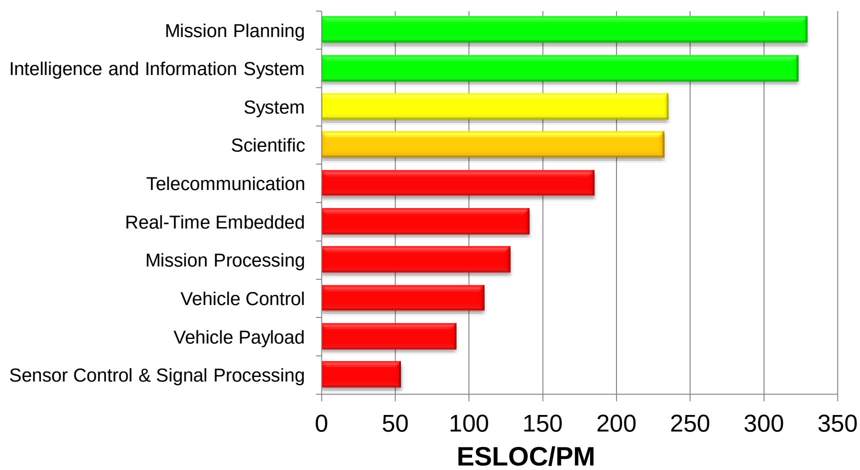
Software Productivity Benchmarks

- Software productivity refers to the ability of an organization to generate outputs using the resources that it currently has as inputs. Inputs typically include facilities, people, experience, processes, equipment, and tools. Outputs generated include software applications and documentation used to describe them.

$$\mathrm{PROD} = \frac{ESLOC}{PM}$$

- Metric used to express software productivity is equivalent source lines of code (ESLOC) per person-month (PM) of effort. While many other measures exist, ESLOC/PM will be used because most of the data collected by the Department of Defense (DoD) on past projects is captured using these two measures. While controversy exists over whether or not ESLOC/PM is a good measure, consistent use of this metric provides for meaningful comparisons of productivity.

Productivity Comparison (Median)



See slide 38 for associated productivity boxplot

Productivity Benchmark: All Operating Environments

Productivity (ESLOC/PM) across Operating Environments

Application Type	ESLOC/PM			Obs.	Std. Dev. (%)	CV (%)	KESLOC	
	1st Quartile	Median	3rd Quartile				MIN	MAX
Mission Planning	207	329	427	20	153	46	10	570
Information and Intelligence Systems	292	323	407	14	82	23	18	417
System	168	235	260	25	86	38	6	842
Scientific	129	232	260	19	101	48	2	226
Telecommunications	140	185	243	47	74	39	1	532
Real-Time Embedded	84	141	172	57	66	46	2	201
Mission Processing	103	128	178	34	57	40	1	229
Vehicle Control	70	110	126	27	52	45	1	330
Vehicle Payload	43	91	120	18	41	46	1	221
Sensor Control and Signal Processing	40	54	79	37	25	42	1	193

Application/Uses
- Applicable for manned aircraft, UAV, ground system, missiles, ordnance, ship and ground vehicle platforms

Changing the Culture to Win the War on Cost 27

Effort and Schedule Estimation Models

Changing the Culture to Win the War on Cost

Effort Estimation Models: All Operating Environments

Effort Models (in Person-Month) across 6 Operating Environments

Application Type	Model Form	Obs.	MAD (%)	CV (%)	R^2 (%)	KESLOC MIN	KESLOC MAX
Mission Planning	47.78 + KESLOC ^ 1.193	20	35	38	**	10	570
Information and Intelligence Systems	2.643 * KESLOC ^ 1.024	14	18	24	97	18	417
System	33.58 + KESLOC ^ 1.276	25	36	43	**	6	842
Scientific	31 + KESLOC ^ 1.334	17	40	39	**	2	226
Telecommunications	7.3 * KESLOC ^ 0.9133	47	35	32	88	1	532
Real-Time Embedded	60.14 + KESLOC ^ 1.44	57	36	39	**	2	201
Mission Processing	6.602 * KESLOC ^ 1.045	33	36	40	88	1	229
Vehicle Control	9.048 * KESLOC ^ 1.018	27	37	35	92	1	330
Vehicle Payload	22.27 * KESLOC ^ 0.804	18	35	15	89	1	221
Sensor Control and Signal Processing	26.43 * KESLOC ^ 0.8668	37	34	29	91	1	193

Changing the Culture to Win the War on Cost 29

Schedule Estimation Models: All Operating Environments

Schedule Models (in Months) across all Operating Environments

Application Type	Model Form	Obs.	SEE (%)	MAD (%)	CV (%)	KESLOC MIN	KESLOC MAX
Mission Planning	2.657 * KESLOC ^ 0.9995*FTE ^ (-0.9854)	10	0.3	19	19	10	570
Information and Intelligence Systems	6.034 * KESLOC ^ 0.6622 * FTE ^ (-0.6002)	19	0.4	31	31	18	417
System	7.681 * KESLOC ^ 0.8363 * FTE ^ (-0.9489)	14	0.4	27	24	7	764
Scientific	16.87 * KESLOC ^ 0.3082 * FTE ^ (-0.2603)	14	0.4	25	24	2	120
Telecommunications	14.78 * KESLOC ^ 0.4512 * FTE ^ (-0.4881)	42	0.4	28	22	1	312
Real-Time Embedded	18.08 * KESLOC ^ 0.5201 * FTE ^ (-0.5695)	44	0.3	23	23	5	201
Mission Processing	9.934 * KESLOC ^ 0.731 * FTE ^ (-0.5978)	24	0.4	35	34	1	225
Vehicle Control	8.288 * KESLOC ^ 0.8527 * FTE ^ (-0.772)	18	0.4	34	34	1	330
Sensor Control and Signal Processing	30.6 * KESLOC ^ 0.4982 * FTE ^ (-0.4895)	28	0.3	25	21	1	193

Application/Uses
Appropriate for crosschecking schedule, given product size and staff level**

Changing the Culture to Win the War on Cost 30

CONCLUSION

Changing the Culture to Win the War on Cost

Primary Findings

1. Analysis results indicate that the effect of software size on software development effort shall be interpreted along with Application Type.
2. Application Type again was shown as a valid predictor of software development duration when used in combination with staffing levels (full time equivalents) and software size
3. Software development duration can be shortened by decreasing software size and/or increasing staffing levels.

Changing the Culture to Win the War on Cost 32

Minimizing Threats to Validity

- *Always have a valid reason for removing outliers*
- *Discarding data based on arbitrary grounds is referred to as "Sampling Bias"*
- *Data Normalization is a critical step to increase statistical validity and reliability*
  - *Helps reduce noise in the data*

Adapted from: W. Rosa, B. Fersch and J. Bilbro, "Software Cost Estimating: QMT 290 Beta Test", AFIT, 2012

Summary

- A valid software cost model should account for the impact of fixed costs and diseconomies of scale.
- The models in this presentation focus on the prime contractor's implementation team, and therefore should be applicable to all sectors.
- Prime contractors can use these models to validate their Implementation Team's cost proposals or estimates.

BACKUP

Application Type Definitions (1 of 2)

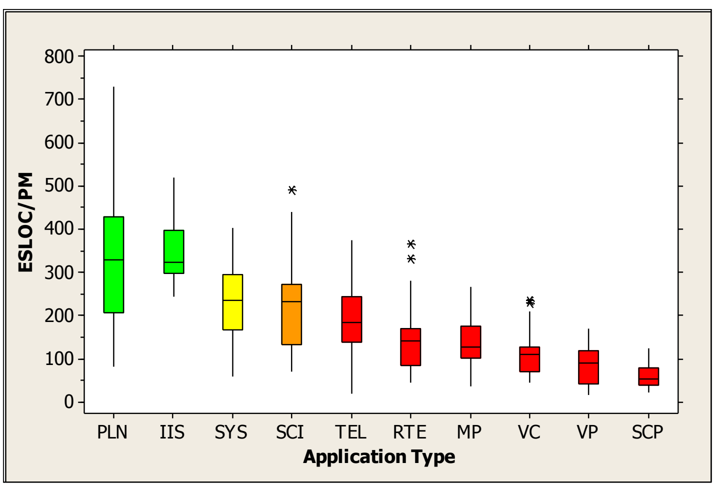
TYPE	DESCRIPTION
Sensor Control and Signal Processing (SCP)	Software that requires timing-dependent device coding to enhance, transform, filter, convert, or compress data signals. Examples: Beam steering controller, sensor receiver/transmitter control, sensor signal processing, sensor receiver/transmitter test. Examples. of sensors: antennas, lasers, radar, sonar, acoustic, electromagnetic.
Vehicle Control (VC)	Hardware & software necessary for the control of vehicle primary and secondary mechanical devices and surfaces. Examples: Digital Flight Control, Operational Flight Programs, Fly-By-Wire Flight Control System, Flight Software, Executive.
Vehicle Payload (VP)	Hardware & software which controls and monitors vehicle payloads and provides communications to other vehicle subsystems and payloads. Examples: Weapons delivery and control, Fire Control, Airborne Electronic Attack subsystem controller, Stores and Self-Defense program, Mine Warfare Mission Package.
Real Time Embedded (RTE)	Real-time data processing unit responsible for directing and processing sensor input/output. Examples: Devices such as Radio, Navigation, Guidance, Identification, Communication, Controls And Displays, Data Links, Safety, Target Data Extractor, Digital Measurement Receiver, Sensor Analysis, Flight Termination, Surveillance, Electronic Countermeasures, Terrain Awareness And Warning, Telemetry, Remote Control.
Mission Processing (MP)	Vehicle onboard master data processing unit(s) responsible for coordinating and directing the major mission systems. Examples: Mission Computer Processing, Avionics, Data Formatting, Air Vehicle Software, Launcher Software, Tactical Data Systems, Data Control And Distribution, Mission Processing, Emergency Systems, Launch and Recovery System, Environmental Control System, Anchoring, Mooring and Towing.
Process Control (PC)	Software that manages the planning, scheduling and execution of a system based on inputs, generally sensor driven.
System Software (SYS)	Layers of software that sit between the computing platform and applications. Examples: Health Management, Link 16, Information Assurance, Framework, Operating System Augmentation, Middleware, Operating Systems

Application Type Definitions (2 of 2)

TYPE	DESCRIPTION
Training (TRN)	Hardware and software that are used for educational and training purposes. Examples: Onboard or Deliverable Training Equipment & Software, Computer-Based Training
Telecommunications (TEL)	The transmission of information, e.g. voice, data, commands, images, and video across different mediums and distances. Primarily software systems that control or manage transmitters, receivers and communications channels. Examples: switches, routers, integrated circuits, multiplexing, encryption, broadcasting, protocols, transfer modes, etc.
Software Tools (TOOL)	Software that is used for analysis, design, construction, or testing of computer programs. Examples: Integrated collection of tools for most development phases of the life cycle, e.g. Rational development environment
Test Software (TST)	Hardware & Software necessary to operate and maintain systems and subsystems which are not consumed during the testing phase and are not allocated to a specific phase of testing. Examples: Onboard or Deliverable Test Equipment & Software
Intelligence & Information Systems (IIS)	An assembly of software applications that allows a properly designated authority to exercise control over the accomplishment of the mission. Humans manage a dynamic situation and respond to user-input in real time to facilitate coordination and cooperation. Software that manipulates, transports and stores information. Examples: Database, Data Distribution, Information Processing, Internet, Entertainment, Enterprise Services*, Enterprise Information**
Scientific Systems (SCI)	Non real time software that involves significant computations and scientific analysis. Examples: Environment Simulations, Offline Data Analysis, Vehicle Control Simulators
Training (TRN)	Hardware and software that are used for educational and training purposes. Examples: Onboard or Deliverable Training Equipment & Software, Computer-Based Training

Changing the Culture to Win the War on Cost 37

Software Productivity Boxplot



Changing the Culture to Win the War on Cost 38