



University of Southern California
Center for Software Engineering

Effect of CASE Tools on Software Development Effort

15th International Forum on COCOMO
and Software Cost Estimation

Jongmoon Baik

Oct 25, 2000



Outline

- Motivation
- Productivity Impact in COCOMO II
- Tool Classification, Rating Scales and Research Model
- Project Activity Difference
- Delphi, Sample, and Posterior Results
- Cross Validation – Data Splitting & Bootstrap
- Contributions
- Future Works



Motivation

- **Very Simple Tool Rating Scale in COCOMOII Model**
- **Strong Statistical Significance of TOOL Effect on Effort & Schedule**
- **No Consideration of Correlations and Overlaps with Other Parameters**
- **No Consideration of tool integration, maturity, and user support**

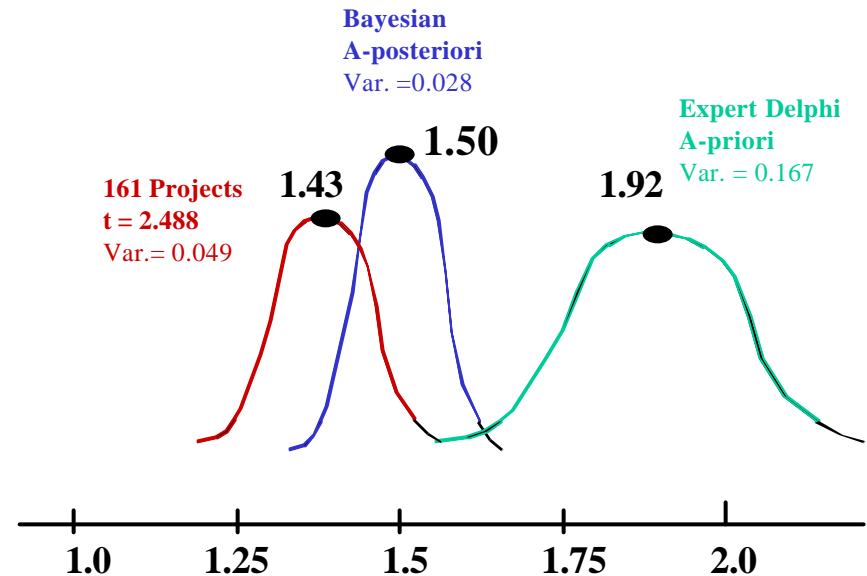
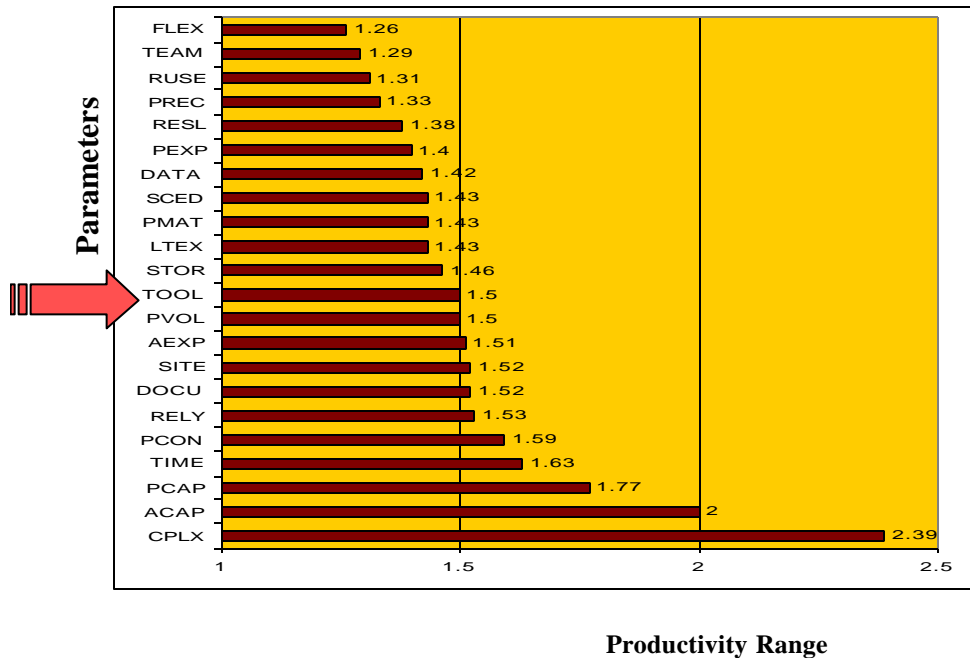


COCOMO II **TOOL** Rating Scale & Value

	Very Low	Low	Nominal	High	Very High
Use of Software Tools (TOOL)	Edit, code, debug	Simple front-end CASE, back-end CASE; Little Integration	Basic lifecycle tools; Moderately integrated	Strong, mature, lifecycle tools; Moderately integrated	Strong, mature, proactive lifecycle tools; Well integrated with process, methods, reuse
	1.17	1.09	1.00	0.90	0.78

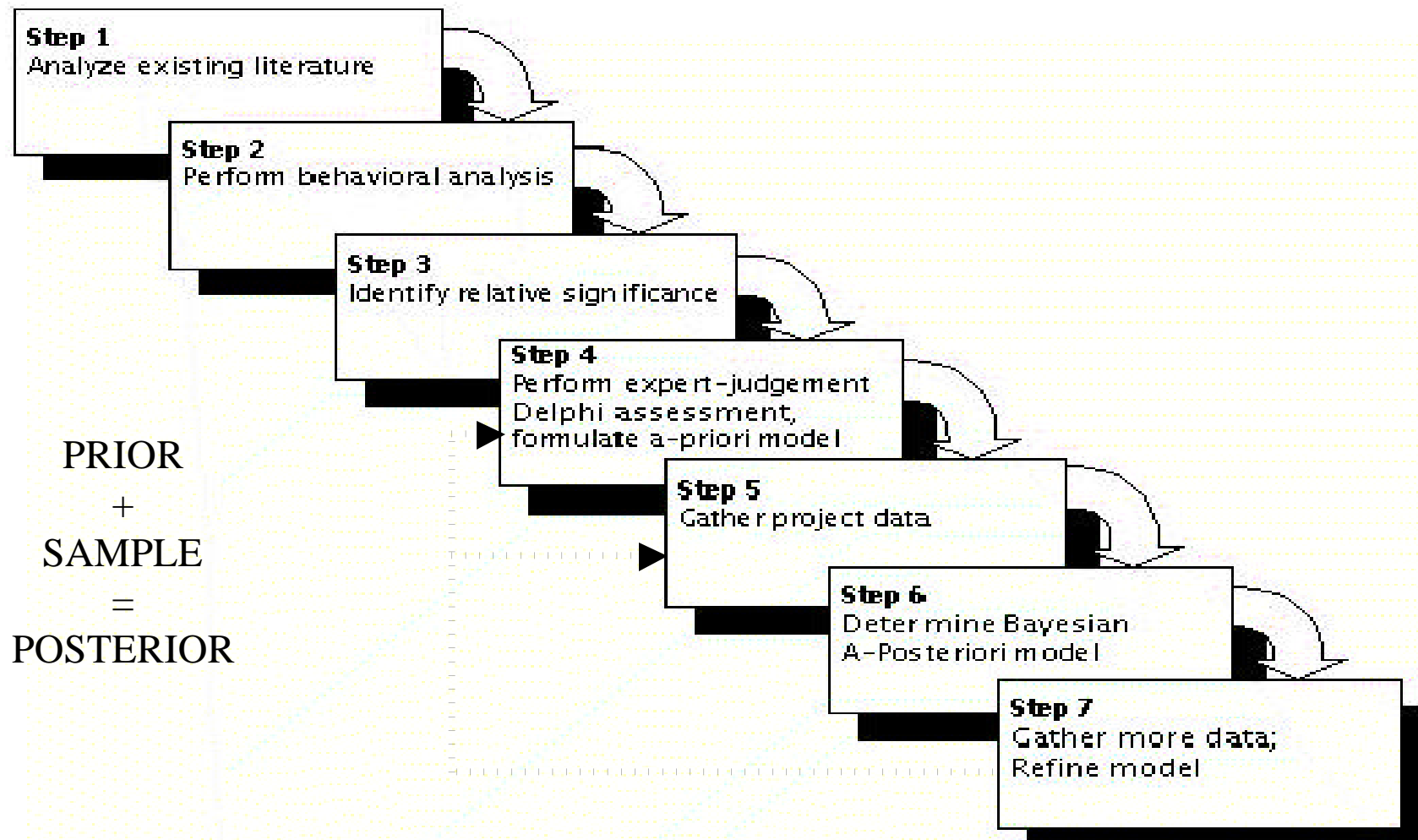
- **No Comparison of the Same Kind of Tools**
- **No Clear Definition of Tools and Integration**
- **No Consideration of Tool Maturity and User Support**
- **No Considerations of Interactions with Other Factors**

COCOMOII.2000 Productivity Ranges



- **Productivity Range (PR) = Highest Rating Value / Lowest Rating Value**
- **Statistically Significant : $t = 2.488 > 1.96$**

COCOMO II Modeling Methodology





Tool Impact in Software Cost Models

	SLIM	Jensen	Checkpoint	PRICE-S	Softcost	COCOMO
“Use of Software Tools”	Yes	Yes	Yes	Yes	Yes	Yes
Completeness of Activity Coverage	No	No	Yes	No	Yes	Yes
Degree of Tool Integration	No	No	Yes	No	No	Yes
Tool Maturity and User Support	No	No	No	No	No	No



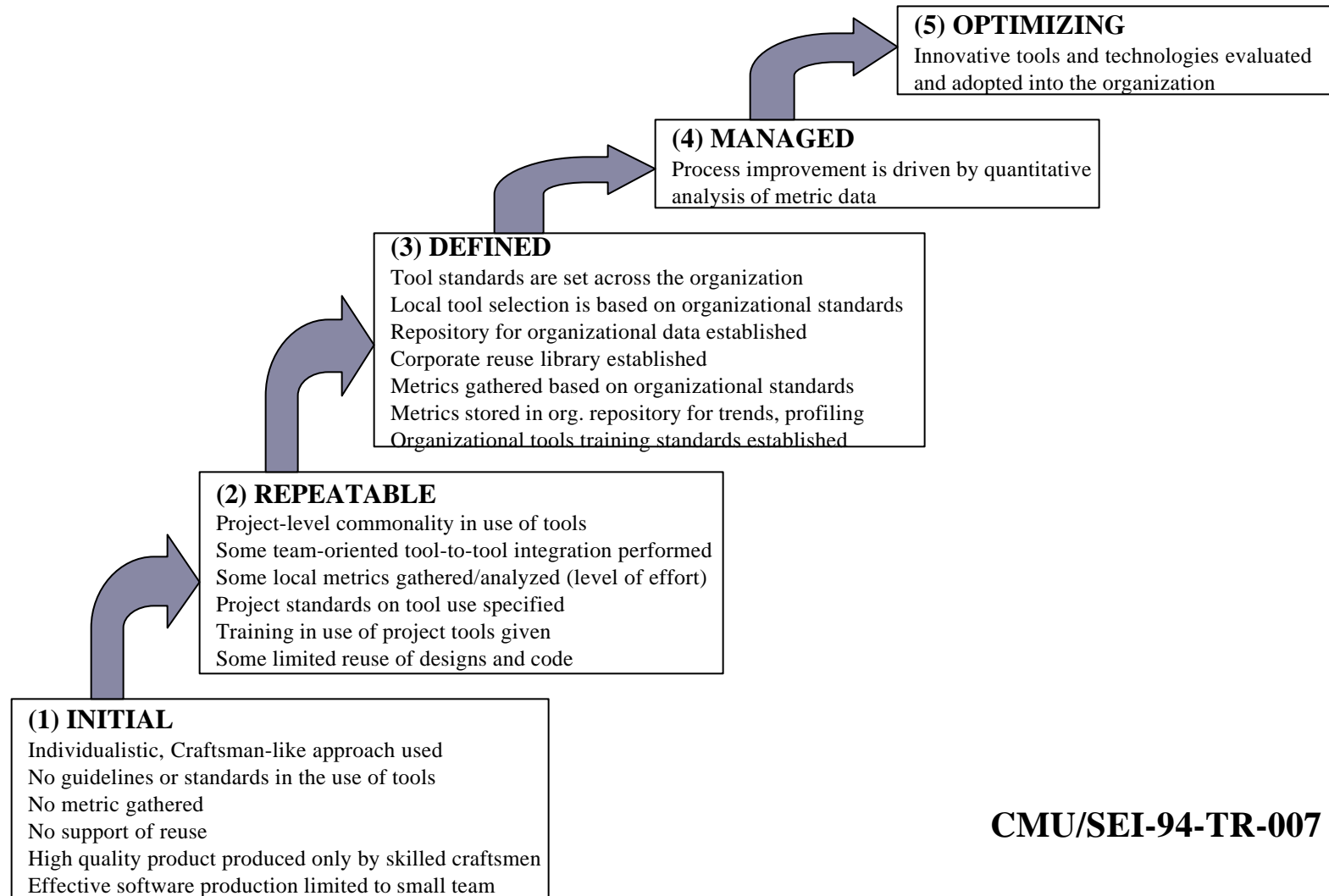
Completeness of Activity Coverage (TCOV)

Very Low	Text-Based Editor Basic 3GL Compiler Basic library Aids Basic Text-based Debugger Basic Linker
Low	Graphical Interactive Editor Simple Design Language Simple Programming Support Library Simple Metrics/Analysis Tool
Nominal	Local Syntax Checking Editor Standard Template Support Document Generator Simple Design Tools Simple Standalone Configuration Management Tool Standard Data Transformation Tool Standard Support Metrics Aids with Repository Simple Repository, Basic Test Case Analyzer
High	Local Semantics Checking Editor Automatic Document Generator Requirement Specification Aids and Analyzer Extended Design Tools Automatic Code Generator from Detailed Design Centralized Configuration Management Tool Process Management Aids Partially Associative Repository (Simple Data Model Support) Test Case Analyzer with Spec. Verification Aids Basic Reengineering & Reverse Engineering Tool
Very High	Global Semantics Checking Editor Tailorable Automatic Document Generator Requirement Specification Aids and Analyzer with Tracking Capability Extended Design Tools with Model Verifier Code Generator with Basic Round-Trip Capability Extended Static Analysis Tool Basic Associative, Active Repository (Complex Data Model Support) Heterogeneous N/W Support Distributed Configuration Management Tool Test Case Analyzer with Testing Process Manager, Oracle Support Extended Reengineering & Reverse Engineering Tools
Extra High	GroupWare systems Distributed Asynchronous Requirement Negotiation and Tradeoff tools Code Generator with Extended Round-Trip Capability Extended Associative, Active Repository Spec-based Static and Dynamic Analyzers Pro-active Project decision Assistance

- **Based on**
 - **Breadth of Activity Coverage**
 - **Specification, Analysis, Design, Programming, Test, CM, QA, Collaboration, Management, etc.**
 - **Tool's Functionality Support**
 - **SEI- CMM Tool Characteristics**
- **“Clear relationship b/w tool functionality and the needs of a particular KPA of CMM”**



Tool-Use Characteristics at CMM Levels





Degree of Tool Integration (TINT)

	Very Low	Low	Nominal	High	Very High	Extra High
Degree of Tool Integration	Individual File Formats for Tools (No Conversion Aid), No Activation Control for Other Tools, Different User Interface for each Tools, Fundamental Incompatibilities among Process Assumptions and Object Semantics	Various File Formats for Each Tools (File Conversion Aids), Message Broadcasting to Tools, Some Standardized User Interfaces among Tools, Difficult Incompatibilities among Process Assumptions and Object Semantics	Shared-Standard Data Structure, Message Broadcasting through Message Server, Standard User Interface Usage among Tools, Workable Incompatibilities among Process Assumptions and Object Semantics	Shared Repository, Point-to-Point Message Passing, Customizable User Interface Support, Largely Workable Incompatibilities among Process Assumptions and Object Semantics	Highly Associative Repository, Point-to-Point Message Passing Using reference for Parameters, Some level of Different User Interface, Largely Consistent among Process Assumption and Object Semantics	Distributed-Associative Repository, Extended Point-to-Point Message Passing for Tool Activation, Complete Set of User Interface for different level of Users, Fully Consistent among Process Assumption and Object Semantics

- Based on
 - **Wasserman's five level model of Integration**
 - *Platform Integration: Tools run on the same hardware/operating system platform*
 - *Data Integration: Tools operate using the shared data model*
 - *Presentation Integration: Tools offer a common user interface*
 - *Control Integration: Tools activate and control the operation of other tools*
 - *Process Integration: Tool usage guided by an explicit process model and associated process engine*



Tool Maturity and User Support (TMAT)

	Very Low	Low	Nominal	High	Very High	Extra High
Tool Maturity and User Support	Version in pre-release beta-test, Simple documentation and help	Version on market/available less than 6 month, Up-dated documentation, help available	Version on market/available between 6 months and 1 year, On-line help, tutorial available	Version on market/available between 1 and 2 years, On-line User Support Group	Version on market/available between 2 and 3 years, On-Site Technical User Support Group	Version on market/available more than 3 years

- Based on
 - Correlation between maturity and years after they are released in CASE market
 - Increasing level of vendor support services



Project Activity Difference (TCOV)

Phase Rating	Plans & Requirements	Product Design	Programming	Test	Implementation & Modification		
Very Low	More effort for simulation, prototyping, requirement tracing, configuration management, document generation, and cooperation/synchronization of team members	More requirements errors;	harder detection and removal	More effort for transforming requirements to design	More design errors;	harder detection and removal	
	More requirements errors;	harder detection and removal	More effort for transforming requirements to design	More design errors;	harder detection and removal	More effort for transforming design to code	
	harder detection and removal	More effort for transforming requirements to design	More design errors;	harder detection and removal	More effort for transforming design to code	More code errors;	
	More effort for transforming requirements to design	More design errors;	harder detection and removal	More effort for transforming design to code	More code errors;	harder detection and removal	
	More design errors;	harder detection and removal	More effort for transforming design to code	More code errors;	harder detection and removal	More effort for generating test cases, capturing project specific metrics, and QA	
	harder detection and removal	More effort for transforming design to code	More code errors;	harder detection and removal	More effort for generating test cases, capturing project specific metrics, and QA	More effort for linking, exercising subsystems, reverse/re-engineering	
	More effort for transforming design to code	More code errors;	harder detection and removal	More effort for generating test cases, capturing project specific metrics, and QA	More effort for linking, exercising subsystems, reverse/re-engineering		
	More code errors;	harder detection and removal	More effort for generating test cases, capturing project specific metrics, and QA	More effort for linking, exercising subsystems, reverse/re-engineering			
	harder detection and removal	More effort for generating test cases, capturing project specific metrics, and QA	More effort for linking, exercising subsystems, reverse/re-engineering				
	More effort for generating test cases, capturing project specific metrics, and QA	More effort for linking, exercising subsystems, reverse/re-engineering					
Low	Intermediate level of above effects						
	No change						
	Nominal	Easier simulation, prototyping, requirement tracing, configuration management, document generation, and cooperation/synchronization of team members	Fewer requirements errors;	easier detection and removal	Less effort for transforming requirements to design	Fewer design errors;	easier detection and removal
		Fewer requirements errors;	easier detection and removal	Less effort for transforming requirements to design	Fewer design errors;	easier detection and removal	Reduced design effort
		easier detection and removal	Less effort for transforming requirements to design	Fewer design errors;	easier detection and removal	Reduced design effort	Less effort for transforming design to code
		Less effort for transforming requirements to design	Fewer design errors;	easier detection and removal	Reduced design effort	Less effort for transforming design to code	Fewer code errors;
		Fewer design errors;	easier detection and removal	Reduced design effort	Less effort for transforming design to code	Fewer code errors;	easier detection and removal
		Reduced design effort	Less effort for transforming design to code	Fewer code errors;	easier detection and removal	Less effort for generating test cases, capturing project specific metrics, and QA	Less effort for linking, exercising subsystems, reverse/re-engineering
		Less effort for transforming design to code	Fewer code errors;	easier detection and removal	Less effort for generating test cases, capturing project specific metrics, and QA	Less effort for linking, exercising subsystems, reverse/re-engineering	
		Fewer code errors;	easier detection and removal	Less effort for generating test cases, capturing project specific metrics, and QA	Less effort for linking, exercising subsystems, reverse/re-engineering		
easier detection and removal		Less effort for generating test cases, capturing project specific metrics, and QA	Less effort for linking, exercising subsystems, reverse/re-engineering				
Less effort for generating test cases, capturing project specific metrics, and QA		Less effort for linking, exercising subsystems, reverse/re-engineering					
High	Higher level of above effects						
	Highest level of above effects						
Very High	Highly reduced effort for project management	Much easier reqt. spec. & update	Much easier reqt. spec. & update	Highly reduced errors,	Much faster detection & removal		
	Much easier reqt. spec. & update	Much easier reqt. spec. & update	Highly reduced errors,	Much faster detection & removal			
	Much easier reqt. spec. & update	Highly reduced errors,	Much faster detection & removal				
	Highly reduced errors,	Much faster detection & removal					
	Much faster detection & removal						
Extra High	Much faster communication among team members						



Research Model

- Extension of COCOMO II Post-Architecture Model
- 3 Productivity Dimensions instead of 1 Productivity Dimension in COCOMO II just based on Completeness of Tool Coverage
- Weighted Sum of Product of Weighting Values and 3 TOOL Rating Scales

$$Effort = A \cdot (Size)^B \cdot \left(\prod_{\substack{i=1 \\ i \neq 15}}^{17} EM_i \right) \cdot TOOL$$

Where

$$EM_{15} = TOOL$$

$$TOOL = b_1 \cdot TINT + b_2 \cdot TMAT + b_3 \cdot TCOV$$

$$\sum_{i=1}^3 b_i = 1$$



Delphi Assessment – Step 4 **(Prior Distribution)**

- **Expert-Judged relative weighting values for the TOOL rating scales**
- **An effective way of getting group consensus (Quantitative values)**
- **Alleviates the problem of individual biases and results in an improved consensus group estimate.**

	TCOV	TINT	TMAT
Mean	0.47	0.26	0.27
Var.	0.025694	0.005485	0.016875

2 Round Delphi Process Result



Sample Regression Fit - Step 5

- Sample: 15 Project data from COCOMO II database

$$\text{TOOL} = b_1\text{TCOV} + b_2\text{TCOV} + b_3\text{TMAT} \longrightarrow y = \mathbf{bX} + \mathbf{e}$$

Regression fit Using Arc

Data set = TOOL_15_data, Name of Fit = L1

Normal Regression

Kernel mean function = Identity

Response = TOOL

Terms = (TCOV TINT TMAT)

With no intercept.

Coefficient Estimates

Label	Estimate	Std. Error	t-value
TCOV	0.515982	0.0888635	5.806
TINT	0.282561	0.107657	2.625
TMAT	0.165480	0.111398	1.485

Sigma hat: 0.048214

Number of cases: 15

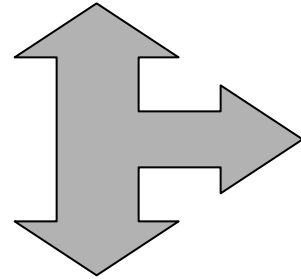
Degrees of freedom: 12

Summary Analysis of Variance Table

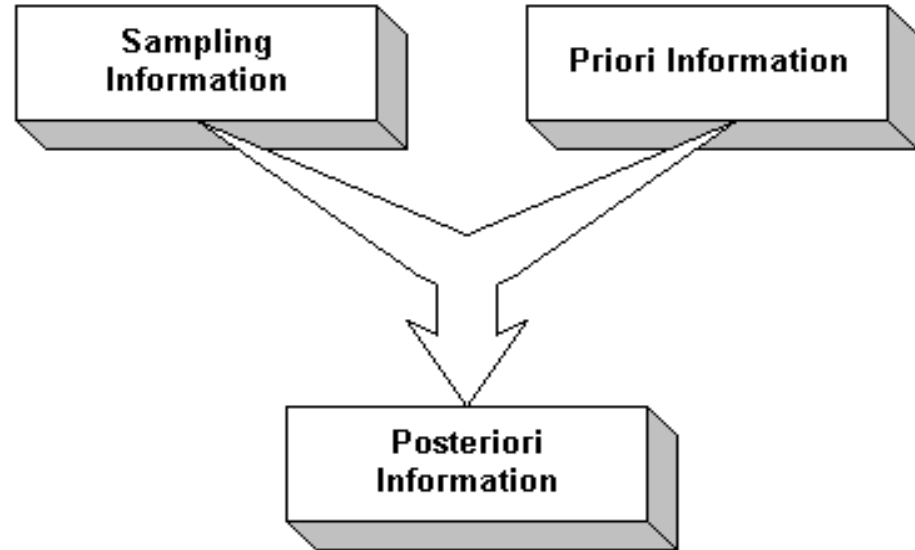
Source	df	SS	MS	F	p-value
Regression	3	20.5056	6.8352	2940.40	0.0000
Residual	12	0.027895	0.00232459		

Bayesian Analysis - Step 6

$$g(\mathbf{q} | Y) = \frac{f(Y | \mathbf{q})g(\mathbf{q})}{f(Y)}$$



$$g(\mathbf{q} | Y) \approx l(\mathbf{q} | Y)g(\mathbf{q})$$



$$\mathbf{b}^{**} = \left[\frac{1}{s^2} \mathbf{X}'\mathbf{X} + \mathbf{H}^* \right]^{-1} \times \left[\frac{1}{s^2} \mathbf{X}'\mathbf{X}\mathbf{b} + \mathbf{H}^* \mathbf{b}^* \right]$$

$$\text{Var}(\mathbf{b}^{**}) = \left[\frac{1}{s^2} \mathbf{X}'\mathbf{X} + \mathbf{H}^* \right]^{-1}$$



Point Estimates of Coefficients

Distribution		b_1	b_2	b_3
PRIOR	Mean	0.47	0.26	0.27
	Variance	0.025694	0.005485	0.016875
SAMPLE	Mean	0.515982	0.282561	0.165480
	Variance	0.0078967	0.011590	0.012409
POSTRIOR	Mean	0.495104	0.259691	0.211617
	Variance	0.00461028	0.00335019	0.005255464

- Normalized posterior weighting values
 - Sum of the weighting values = 1
 - $TOOL = 0.51 * TCOV + 0.27 * TINT + 0.22 * TMAT$



Comparison of Prediction Accuracies

- Magnitude of Relative Errors

$$MRE = \left| \frac{Estimated - Actual}{Actual} \right|$$

- Improved Prediction Accuracy over 15 project data
 - From 67 % (COCOMO II.2000) to 87% actuals within 10% MRE
- Same prediction accuracy in Sample & Posterior
 - But, Variances of Coefficient Estimates in Bayesian are Smaller

	COCOMO II.2000 (1 Dimensional TOOL)	Sample (3 Dimensional TOOL)	Bayesian Posterior (3 Dimensional TOOL)
PRED (.10)	67%	87%	87%



Cross Validation by Data Splitting

- A widely used model checking method to validate a regression model
- Divides the original dataset into two parts
 - **Construction:** Exploration and model formulation
 - **Validation:** Model validation, formal estimation and testing
 - used in the same way as in the newly collected dataset
- Criteria Function
 - PRESS (Predicted Residual Sum of Squares)
 - Small PRESS indicates that the estimated regression model is a good model

$$PRESS = \sum \hat{e}_{(i)}^2$$

where

$$\hat{e}_{(i)}^2 = y_i - X_i^T \hat{\beta}_{(i)} = \frac{\hat{e}_i}{1 - h_{ii}}$$



Cross Validation Summary

1 st validation (TOOL)	<pre>Cross validation summary of cases not used to get estimates: Sum of squared deviations: 9.76793 Mean squared deviation: 0.212346 Sqrt(mean squared deviation): 0.460811 Number of observations: 46 > (/(sum (^ (/ (send L1 :residuals) (- 1 (send L1 :leverages))) 2)) (send L1 :num-included)) 0.262203</pre>
2 nd validation (TCOV, TINT, and TMAT) PRESS	<pre>Cross validation summary of cases not used to get estimates: Sum of squared deviations: 9.7475 Mean squared deviation: 0.211902 Sqrt(mean squared deviation): 0.460328 Number of observations: 46 >(/ (sum (^ (/ (send L1 :residuals) (- 1 (send L1 :leverages))) 2)) (send L1 :num-included)) 0.260493</pre>

- Different Construction datasets are used
 - 46 randomly selected project data excluded
 - 1st validation : used one dimension TOOL for both construction and validation sub-samples
 - 2nd validation: used three dimensional TCOV, TINT, and TMAT for only 15 projects out of 161 separated into both construction (11) and validation (4) sub-samples



Cross Validation - Bootstrap

- A statistical simulation methodology
 - Re-samples from the original dataset
- Used to solve two of the most important problems
 - Determination of an estimator for a particular parameter of interest
 - Evaluation of that estimator through the standard error
- Bootstrap Procedure
 - Generate a bootstrap sample of size n (where n is the original sample size) with replacement from the original distribution
 - Compute θ^* , the value of $\hat{\theta}$ obtained by using the bootstrap sample in place of the original sample
 - Repeat steps 1 and 2, k times



Bootstrap Summary

- COCOMO II log transformed regression model

$$\log PM = \beta_0 \log A + \beta_1 B \log Size + \beta_2 SF_1 \log Size + \dots + \beta_6 SF_5 \log Size + \beta_7 \log Size EM_1 + \beta_8 \log Size EM_2 + \dots + \beta_{23} \log Size EM_{17}$$

- Bootstrap Estimates for log[TOOL]

	Std-Error	Bias	Confidence Interval
One Dimension (TOOL)	0.392141	0.030489	(0.184884 1.67873)
Three Dimensions (TCOV, TINT, & TMAT)	0.368760	0.022662	(0.224192 1.63596)

- 1000 boots of same size (161): used in the program *Arc*
- Difference is not great. But, The results is stable and in the right direction



Contributions

- **Provided a classification of software tools based on the breadth of coverage**
- **Proposed three-dimensional TOOL rating scales to evaluate Software Tools**
- **Increased the Prediction Accuracy for an initial set of data points via Bayesian Approach**
- **Demonstrated a method to calibrate the individual contributions to a multidimensional parameter**
- **Validated the research model using cross-validation methods**
 - **Data Splitting and Bootstrapping methods**



Future Works

- **Refine classification and TOOL rating scales with more available tools and CASE technologies**
- **Refine behavioral analysis of TOOL effects**
- **Data Collection for interaction among TOOL rating scales**
- **Calibration of the next version of COCOMO II with and without TOOL rating scales**
- **Determine weights of individual CASE TOOL rating-scale effects on S/W development effort**
- **Tool Support for ROI Analysis of CASE tool adoption**
- **Integration of ROI Analyzer with USC-COCOMO II**