UNIVERSITY OF
SOUTHERN CALIFORNIA

COCOTS (COnstructive COTS) Software Integration Cost Model: An Overview

Dr. Barry Boehm

USC Center for Software Engineering

July 23, 1998

UNIVERSITY OF
SOUTHERN CALIFORNIA

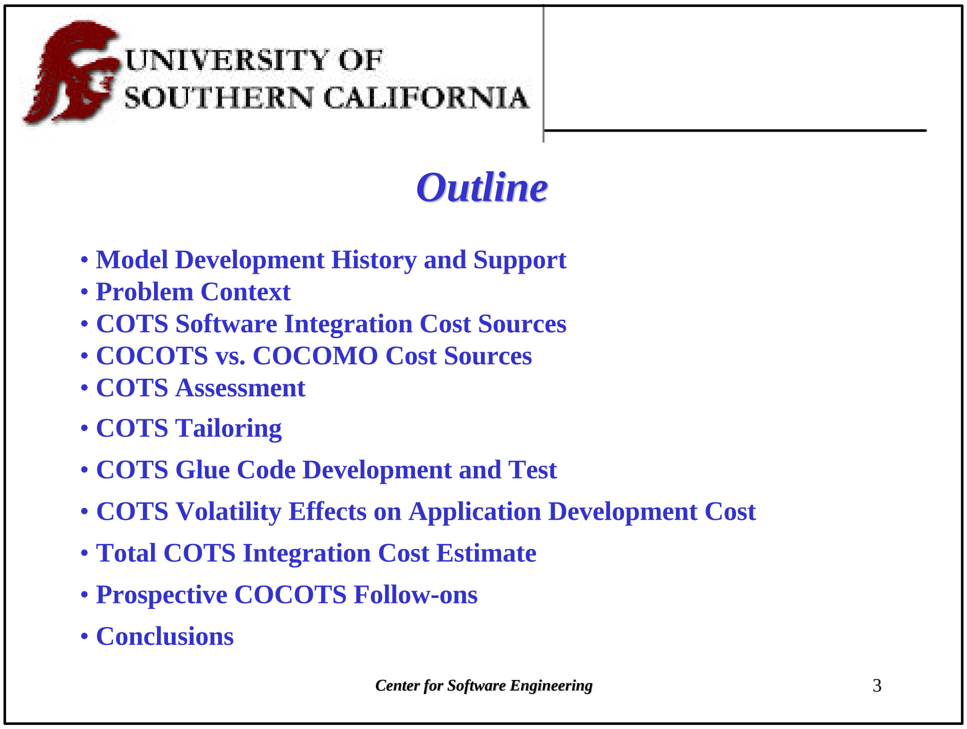
USC Center for Software Engineering

Points of Contact at USC-CSE in Los Angeles

Mr. Christopher Abts (primary graduate researcher)……………	(213) 740-6470
Ms. Jennifer Browning (CSE Office Administrator)…..………..	(213) 740-5703
Dr. Barry W. Boehm (CSE Director)…………………………….	(213) 740-8163
USC Center for Software Engineering FAX line……..………...	(213) 740-4927
COTS Software Research Effort E-Mail….……………	*cots-info@sunset.usc.edu*
World Wide Web page…………..….	*http://sunset.usc.edu/COCOTS /cocots.html*

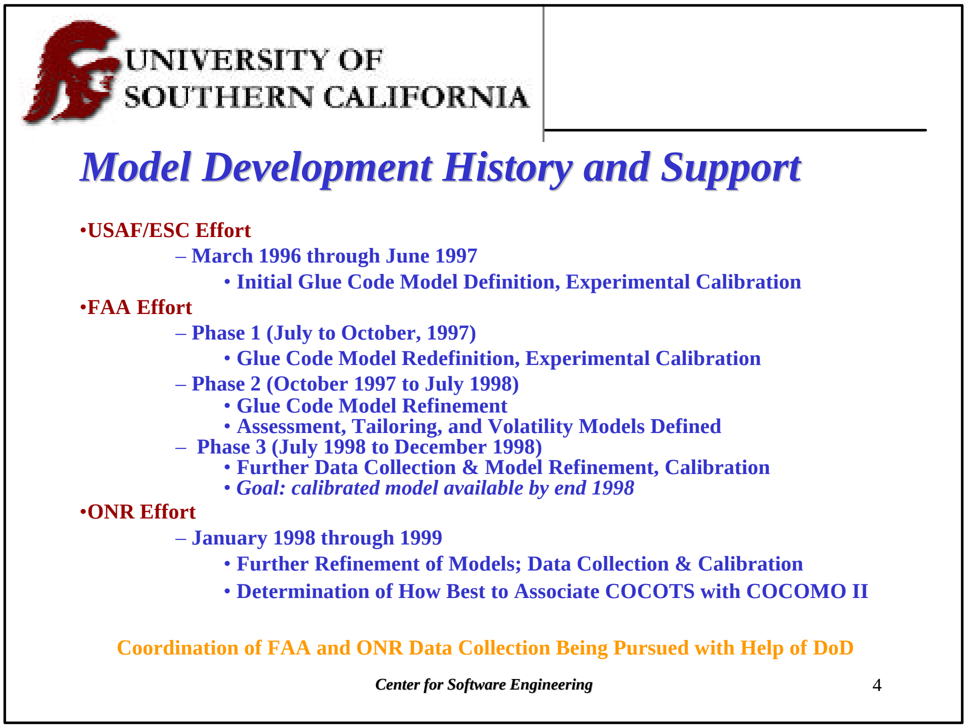
Additional Contact at Software Metrics, Inc. in Virginia (near D.C.)

Dr. Elizabeth (Betsy) Bailey…………..………………………….	(703) 754-0115
FAX line…………………………………………..……….…..	(703) 754-0115
E-Mail……………………………………………………….	*bkbailey@erols.com*



Outline

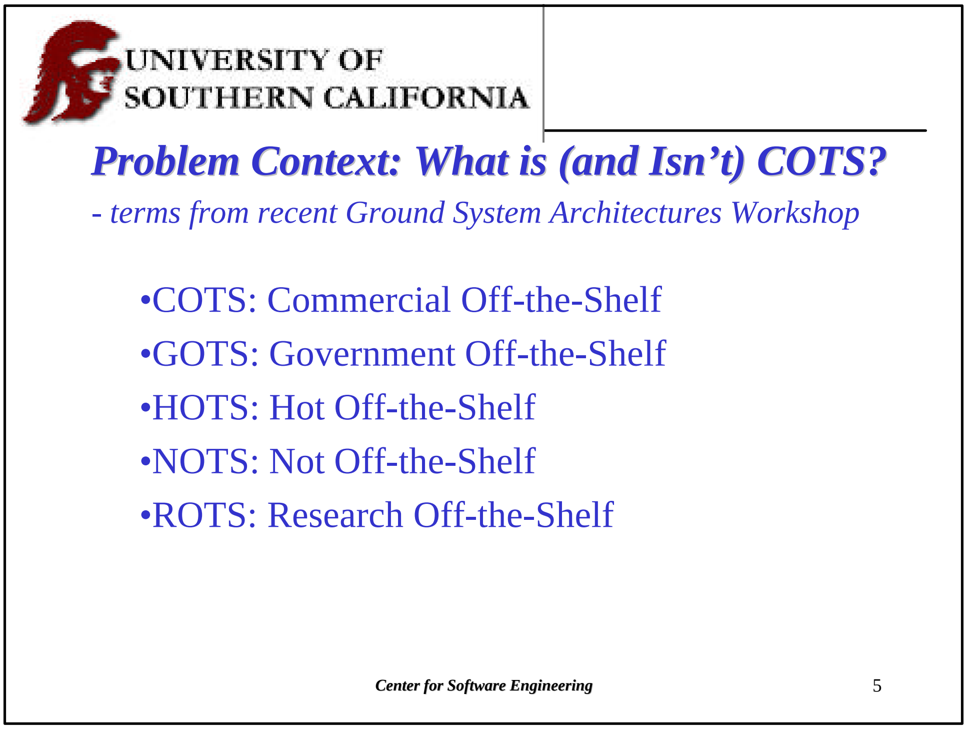
- Model Development History and Support
- Problem Context
- COTS Software Integration Cost Sources
- COCOTS vs. COCOMO Cost Sources
- COTS Assessment
- COTS Tailoring
- COTS Glue Code Development and Test
- COTS Volatility Effects on Application Development Cost
- Total COTS Integration Cost Estimate
- Prospective COCOTS Follow-ons
- Conclusions



Model Development History and Support

- **USAF/ESC Effort**
  - March 1996 through June 1997
    - Initial Glue Code Model Definition, Experimental Calibration
- **FAA Effort**
  - Phase 1 (July to October, 1997)
    - Glue Code Model Redefinition, Experimental Calibration
  - Phase 2 (October 1997 to July 1998)
    - Glue Code Model Refinement
    - Assessment, Tailoring, and Volatility Models Defined
  - Phase 3 (July 1998 to December 1998)
    - Further Data Collection & Model Refinement, Calibration
    - *Goal: calibrated model available by end 1998*
- **ONR Effort**
  - January 1998 through 1999
    - Further Refinement of Models; Data Collection & Calibration
    - Determination of How Best to Associate COCOTS with COCOMO II

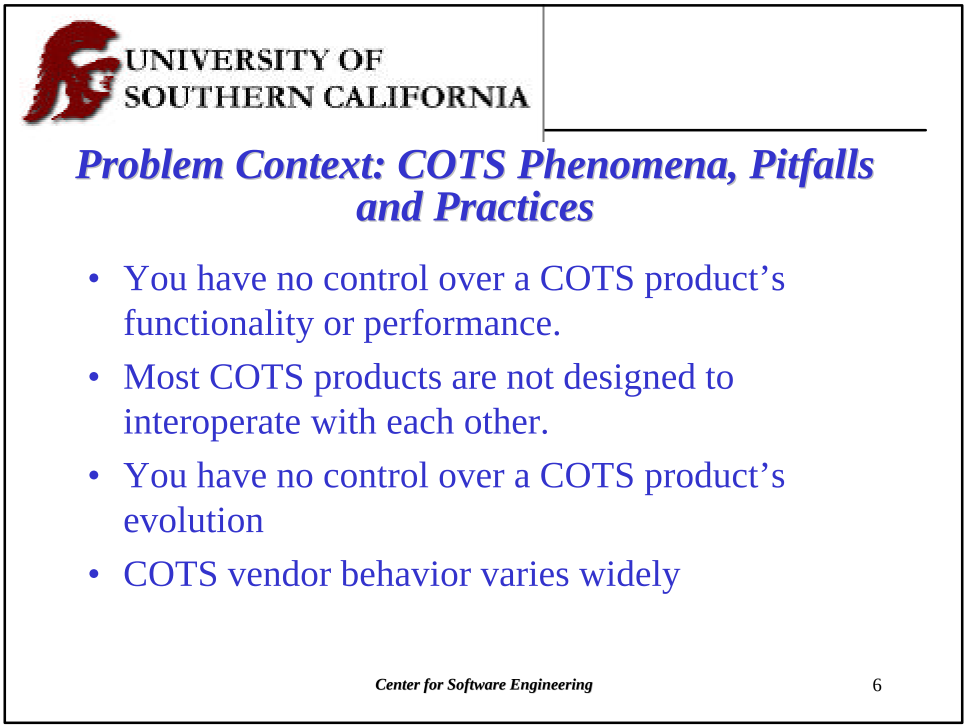
Coordination of FAA and ONR Data Collection Being Pursued with Help of DoD



Problem Context: What is (and Isn't) COTS?

- terms from recent Ground System Architectures Workshop

- •COTS: Commercial Off-the-Shelf
- •GOTS: Government Off-the-Shelf
- •HOTS: Hot Off-the-Shelf
- •NOTS: Not Off-the-Shelf
- •ROTS: Research Off-the-Shelf



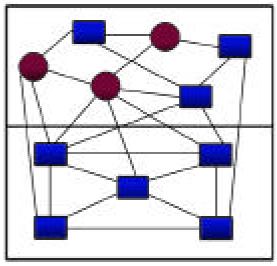
Problem Context: COTS Phenomena, Pitfalls and Practices

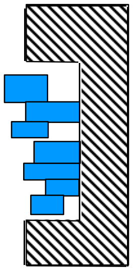
- You have no control over a COTS product's functionality or performance.
- Most COTS products are not designed to interoperate with each other.
- You have no control over a COTS product's evolution
- COTS vendor behavior varies widely

Problem Context: Modeling

COTS ■ and Custom ●
Applications Components

New **COCOTS**
Modeling
Problem





COTS Infrastructure
COCOMO II: PVOL, PEXP

COTS Tools
LTEX, TOOL

Cost Modeling Currently Addressed

COTS Software Integration Cost Sources*

1. COTS Assessment

- *Initial Filtering*
- *Final Selection*

2. COTS Tailoring

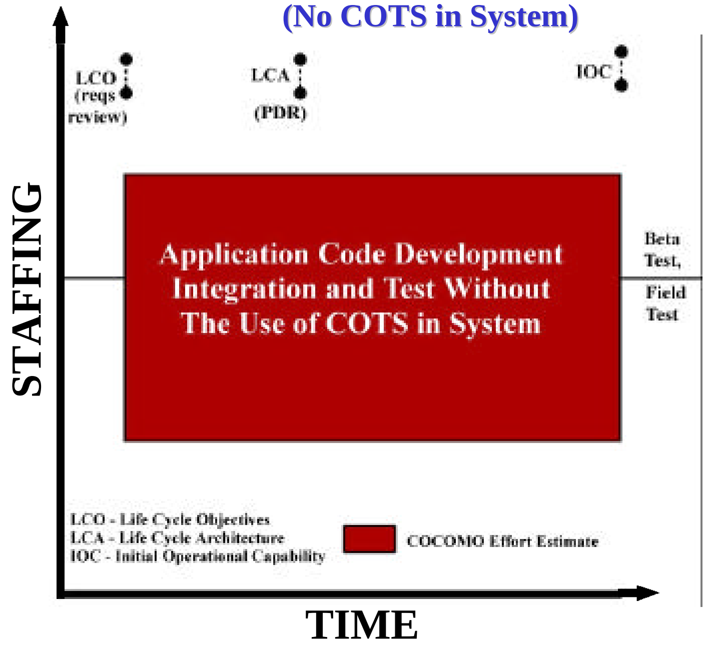
3. COTS Application Glue Code Development and (System) Test

4. COTS Volatility Effects on Application Development Cost

***Initial COCOTS Focus: Software Development;
Operations & Maintenance to be addressed later**

COCOMO Cost Sources

(No COTS in System)



STAFFING

LCO (reqs review)

LCA (PDR)

IOC

Application Code Development Integration and Test Without The Use of COTS in System

Beta Test, Field Test

LCO - Life Cycle Objectives
LCA - Life Cycle Architecture
IOC - Initial Operational Capability

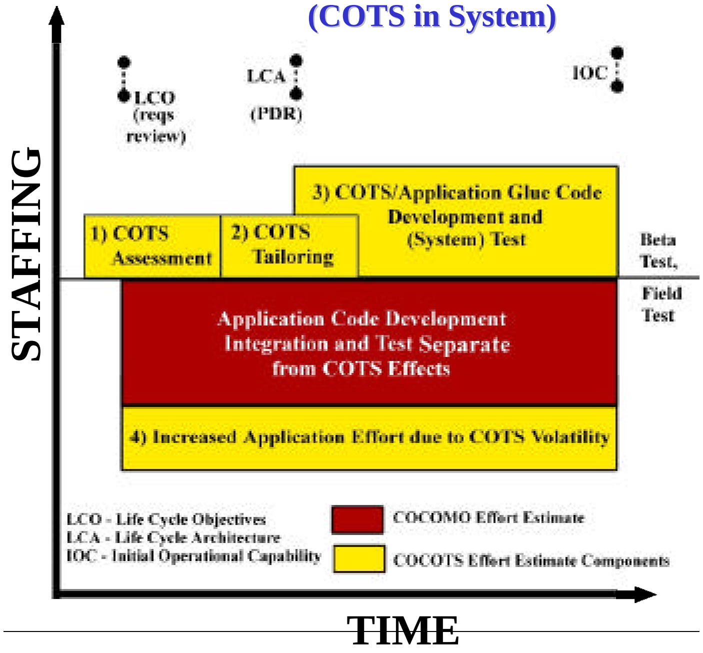
🟥 COCOMO Effort Estimate

TIME

9

COCOMO vs. COCOTS Cost Sources

(COTS in System)



STAFFING

LCO (reqs review)

LCA (PDR)

IOC

1) COTS Assessment

2) COTS Tailoring

3) COTS/Application Glue Code Development and (System) Test

Beta Test, Field Test

Application Code Development Integration and Test Separate from COTS Effects

4) Increased Application Effort due to COTS Volatility

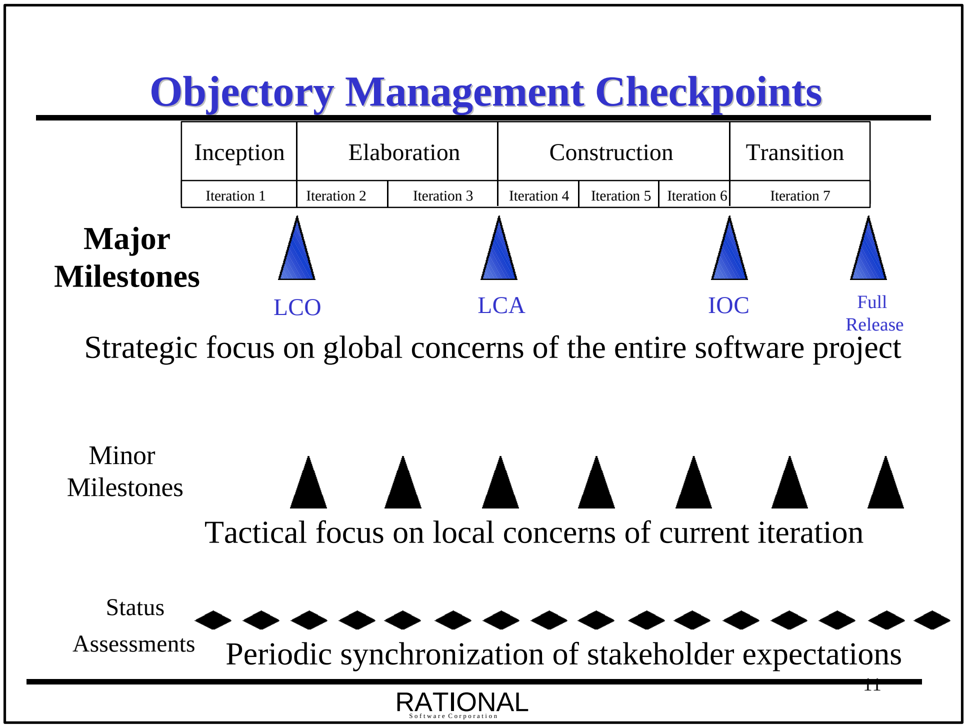
LCO - Life Cycle Objectives
LCA - Life Cycle Architecture
IOC - Initial Operational Capability

🟥 COCOMO Effort Estimate

🟨 COCOTS Effort Estimate Components

TIME

10



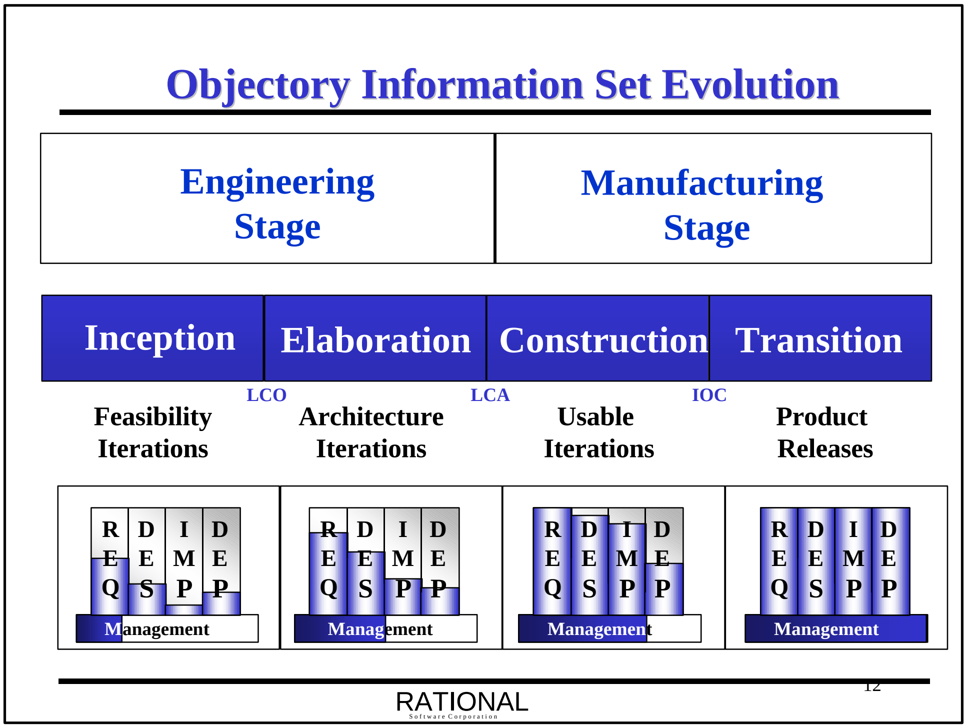
Objectory Management Checkpoints

Inception	Elaboration		Construction			Transition
Iteration 1	Iteration 2	Iteration 3	Iteration 4	Iteration 5	Iteration 6	Iteration 7

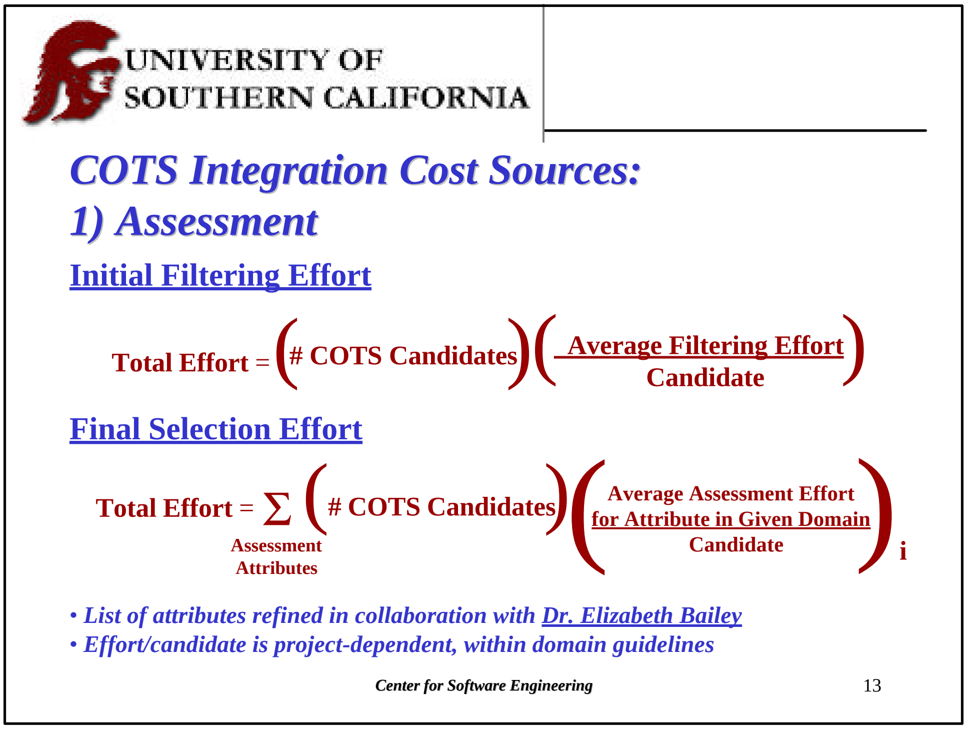
Strategic focus on global concerns of the entire software project

Tactical focus on local concerns of current iteration

Periodic synchronization of stakeholder expectations



Objectory Information Set Evolution



COTS Integration Cost Sources: 1) Assessment

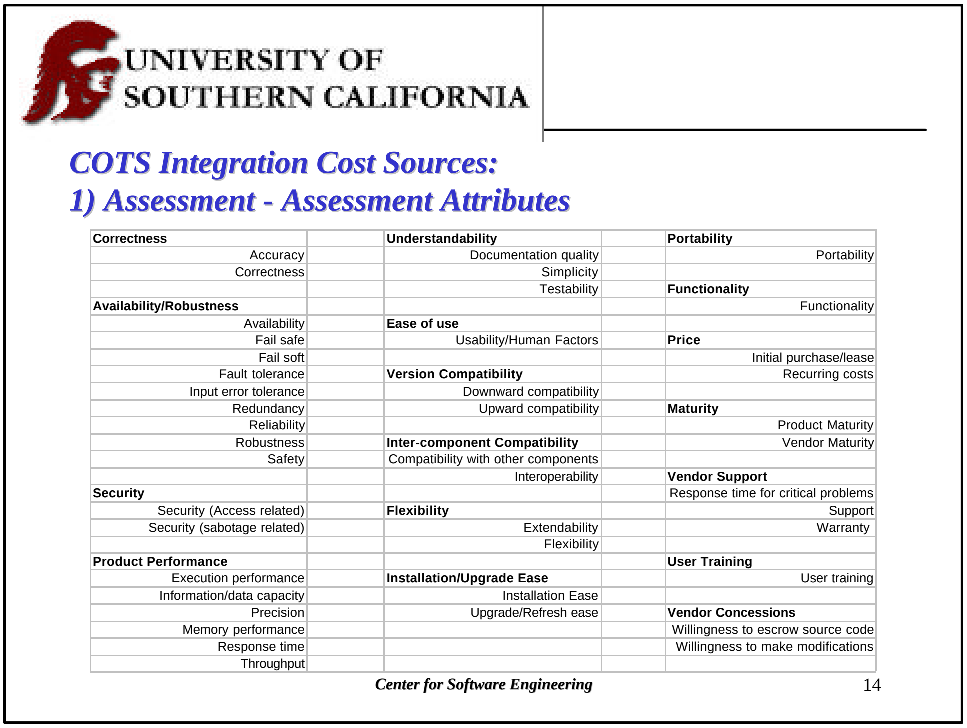
Initial Filtering Effort

$$\text{Total Effort} = \left(\#\ \text{COTS Candidates}\right)\left(\frac{\text{Average Filtering Effort}}{\text{Candidate}}\right)$$

Final Selection Effort

$$\text{Total Effort} = \sum_{\substack{\text{Assessment}\\ \text{Attributes}}} \left(\#\ \text{COTS Candidates}\right)\left(\frac{\text{Average Assessment Effort for Attribute in Given Domain}}{\text{Candidate}}\right)_i$$

- *List of attributes refined in collaboration with Dr. Elizabeth Bailey*
- *Effort/candidate is project-dependent, within domain guidelines*



COTS Integration Cost Sources: 1) Assessment - Assessment Attributes

Correctness		**Understandability**		**Portability**
Accuracy		Documentation quality		Portability
Correctness		Simplicity		
		Testability		**Functionality**
Availability/Robustness				Functionality
Availability		**Ease of use**		
Fail safe		Usability/Human Factors		**Price**
Fail soft				Initial purchase/lease
Fault tolerance		**Version Compatibility**		Recurring costs
Input error tolerance		Downward compatibility		
Redundancy		Upward compatibility		**Maturity**
Reliability				Product Maturity
Robustness		**Inter-component Compatibility**		Vendor Maturity
Safety		Compatibility with other components		
		Interoperability		**Vendor Support**
Security				Response time for critical problems
Security (Access related)		**Flexibility**		Support
Security (sabotage related)		Extendability		Warranty
		Flexibility		
Product Performance				**User Training**
Execution performance		**Installation/Upgrade Ease**		User training
Information/data capacity		Installation Ease		
Precision		Upgrade/Refresh ease		**Vendor Concessions**
Memory performance				Willingness to escrow source code
Response time				Willingness to make modifications
Throughput				

COTS Integration Cost Sources: 2) Tailoring

$$\text{Total Effort} = \sum_{\substack{\text{Tailoring}\\\text{Complexity}\\\text{Levels}}} \left(\begin{array}{c}\text{\# COTS Candidates}\\\text{Tailored at}\\\text{Complexity Level}\end{array}\right)_i \left(\begin{array}{c}\text{Average Effort at Tailoring}\\\text{Complexity Level in Domain}\end{array}\right)_i$$

–*Five tailoring effort complexity levels:*

Very Low, Low, Nominal, High, Very High

– *Differentiated based on number tailored parameters, difficulty of needed scripts, API iterations, etc.*

COTS Integration Cost Sources: 2) Tailoring - Dimensions of Tailoring Difficulty

	Individual Activity & Aid Complexity Ratings					
Tailoring Activities & Aids	**Very Low** (point value = 1)	**Low** (point value = 2)	**Nominal** (point value = 3)	**High** (point value = 4)	**Very High** (point value = 5)	**Corre-sponding Points**
Parameter Specification	Zero to 50 parms to be initialized.	51 to 100 parms to be initialized.	101 to 500 parms to be initialized.	501 to 1000 parms to be initialized.	1001 or more parms to be initialized.	-------

Script Writing	Menu driven;
	1 to 5 line scripts;
	1 to 5 scripts needed.
		Menu driven;
		6 to 10 line scripts;
		6 to 15 scripts needed.
			Hand written;
			11 to 25 line scripts;
			16 to 30 scripts needed.
				Hand written;
				26 to 50 line scripts;
				31 to 50 scripts needed.
					Hand written;
					51 or more line scripts;
					51 or more scripts needed.
						------- 	

I/O Report & GUI Screen Specification & Layout	Automated or standard templates used; 1 to 5 reports/screens needed.	Automated or standard templates used; 6 to 15 reports/screens needed.	Automated or standard templates used; 16 to 25 reports/screens needed.	Hand written or custom designed; 26 to 50 reports/screens needed.	Hand written or custom designed; 51 or more reports/screens needed.	-------

Security/Access Protocol Initialization & Set-up	1 security level;
	1 to 20 user profiles;
	1 input screen/user.
		2 security levels
		21 to 50 user profiles;
		2 input screens/user.
			3 security levels
			51 to 75 user profiles;
			3 input screens/user.
				4 security levels
				76 to 100 user profiles;
				4 input screens/user.
					5 or more security levels
					101 or more user profiles;
					5 or more input screens/user.
						------- 	

Availability of COTS Tailoring Tools	No tools available.	**N/A**	**N/A**	**N/A**	Tools are available.	-------

Total Point Score = ___________

UNIVERSITY OF
SOUTHERN CALIFORNIA

COTS Integration Cost Sources:
3) Glue Code Development and Test

$$\text{Total Effort} = A \bullet [(\text{size})(1+\text{breakage})]^{B} \bullet \prod (\text{effort multipliers})$$

- **A** - a linear scaling constant
- **Size** - of the glue code in SLOC or FP
- **Breakage** - of the glue code due to change in requirements and/or COTS volatility
- **Effort Multipliers** - 13 parameters, each with settings ranging VL to VH
- **B** - an architectural scale factor with settings VL to VH

UNIVERSITY OF
SOUTHERN CALIFORNIA

COTS Integration Cost Sources:
3) Glue Code Development and Test - Glue Code Cost Drivers

Personnel Drivers

1) ACIEP - COTS Integrator Experience with Product
2) ACIPC - COTS Integrator Personnel Capability
3) AXCIP - Integrator Experience with COTS Integration Processes
4) APCON - Integrator Personnel Continuity

COTS Component Drivers

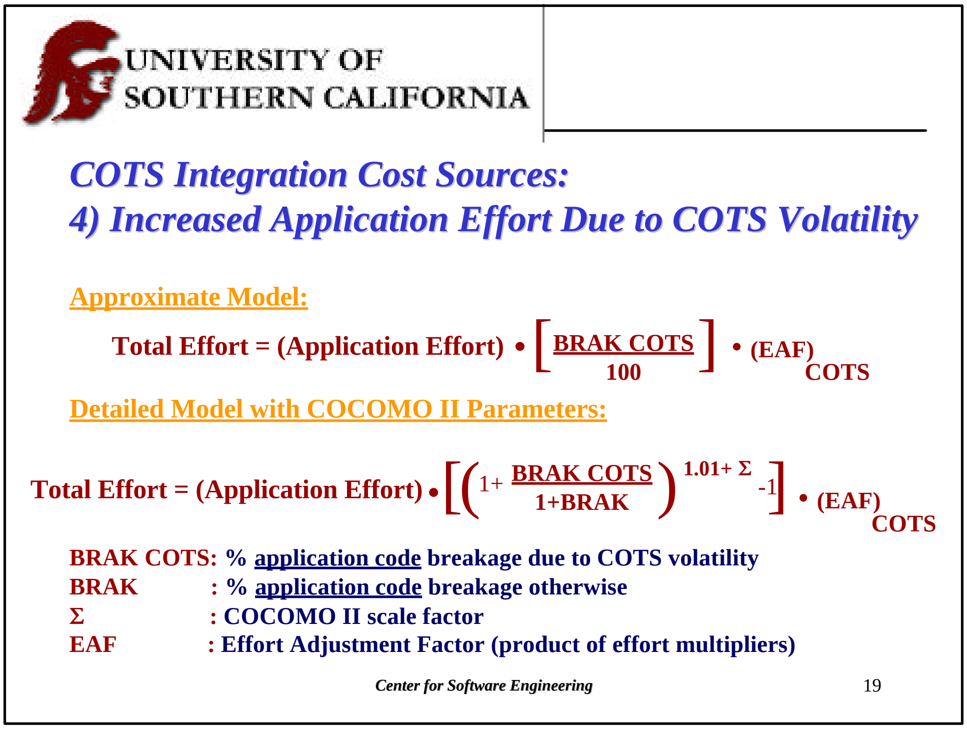
5) ACPMT - COTS Product Maturity
6) ACSEW - COTS Supplier Product Extension Willingness
7) APCPX - COTS Product Interface Complexity
8) ACPPS - COTS Supplier Product Support
9) ACPTD - COTS Supplier Provided Training and Documentation

Application/System Drivers

10) ACREL - Constraints on Application System/Subsystem Reliability
11) AACPX - Application Interface Complexity
12) ACPER - Constraints on COTS Technical Performance
13) ASPRT - Application System Portability

Nonlinear Scale Factor

1) AAREN - Application Architectural Engineering



UNIVERSITY OF SOUTHERN CALIFORNIA

COTS Integration Cost Sources:
4) Increased Application Effort Due to COTS Volatility

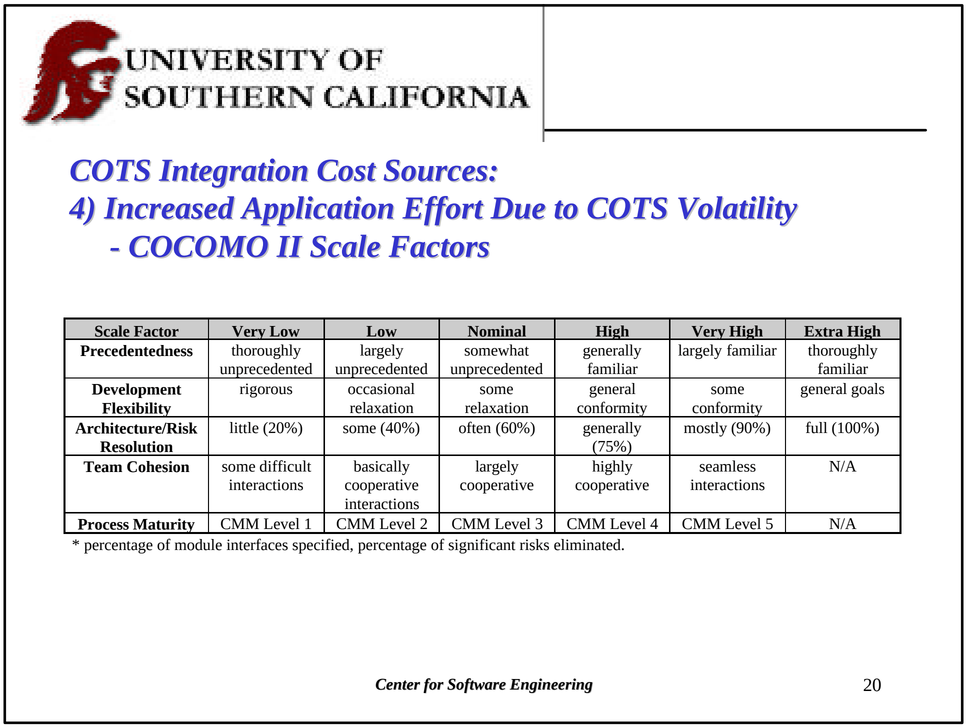
Approximate Model:

$$\text{Total Effort} = (\text{Application Effort}) \bullet \left[\frac{\text{BRAK COTS}}{100}\right] \bullet (\text{EAF})_{\text{COTS}}$$

Detailed Model with COCOMO II Parameters:

$$\text{Total Effort} = (\text{Application Effort}) \bullet \left[\left(1+ \frac{\text{BRAK COTS}}{1+\text{BRAK}}\right)^{1.01+ \Sigma} -1\right] \bullet (\text{EAF})_{\text{COTS}}$$

BRAK COTS: % application code breakage due to COTS volatility
BRAK : % application code breakage otherwise
Σ : COCOMO II scale factor
EAF : Effort Adjustment Factor (product of effort multipliers)

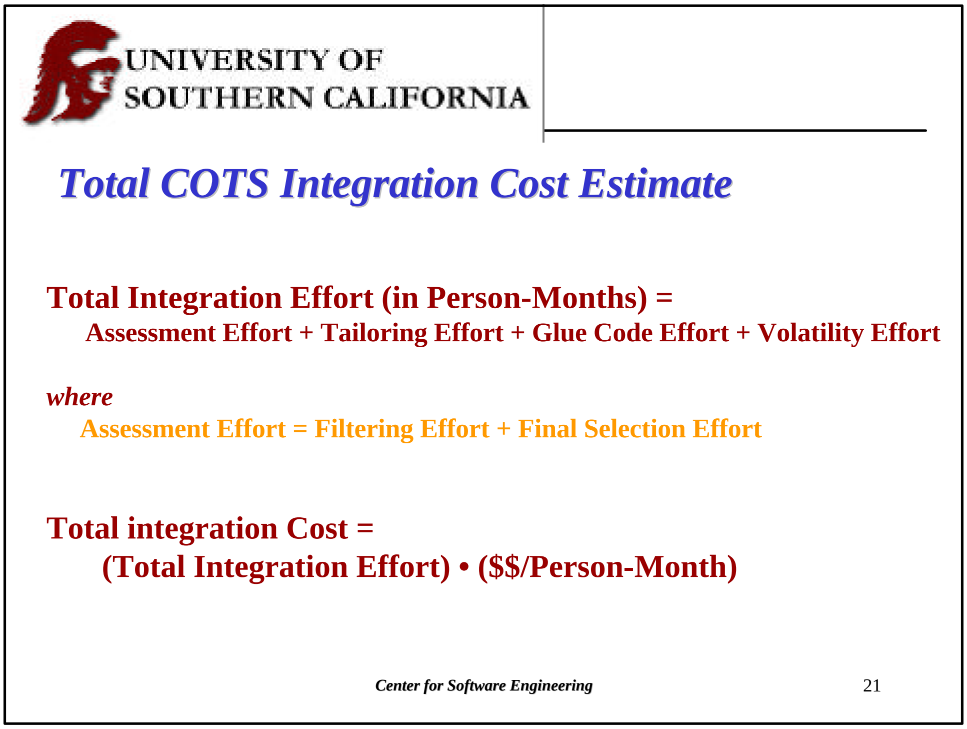


UNIVERSITY OF SOUTHERN CALIFORNIA

COTS Integration Cost Sources:
4) Increased Application Effort Due to COTS Volatility
- COCOMO II Scale Factors

Scale Factor	Very Low	Low	Nominal	High	Very High	Extra High
Precedentedness	thoroughly unprecedented	largely unprecedented	somewhat unprecedented	generally familiar	largely familiar	thoroughly familiar
Development Flexibility	rigorous	occasional relaxation	some relaxation	general conformity	some conformity	general goals
Architecture/Risk Resolution	little (20%)	some (40%)	often (60%)	generally (75%)	mostly (90%)	full (100%)
Team Cohesion	some difficult interactions	basically cooperative interactions	largely cooperative	highly cooperative	seamless interactions	N/A
Process Maturity	CMM Level 1	CMM Level 2	CMM Level 3	CMM Level 4	CMM Level 5	N/A

* percentage of module interfaces specified, percentage of significant risks eliminated.

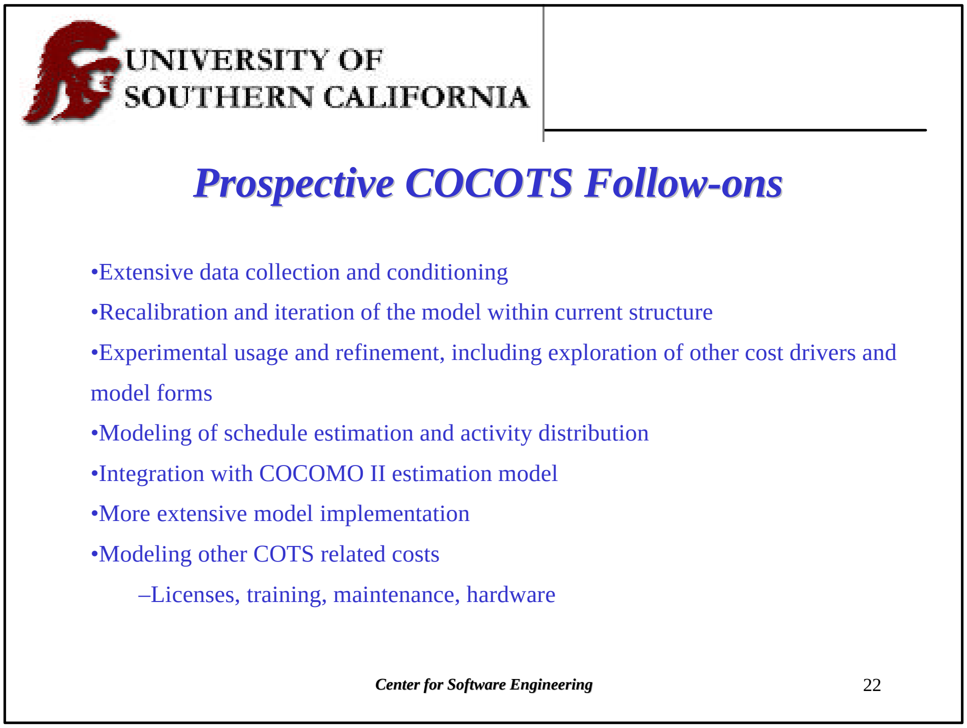


Total COTS Integration Cost Estimate

Total Integration Effort (in Person-Months) =
Assessment Effort + Tailoring Effort + Glue Code Effort + Volatility Effort

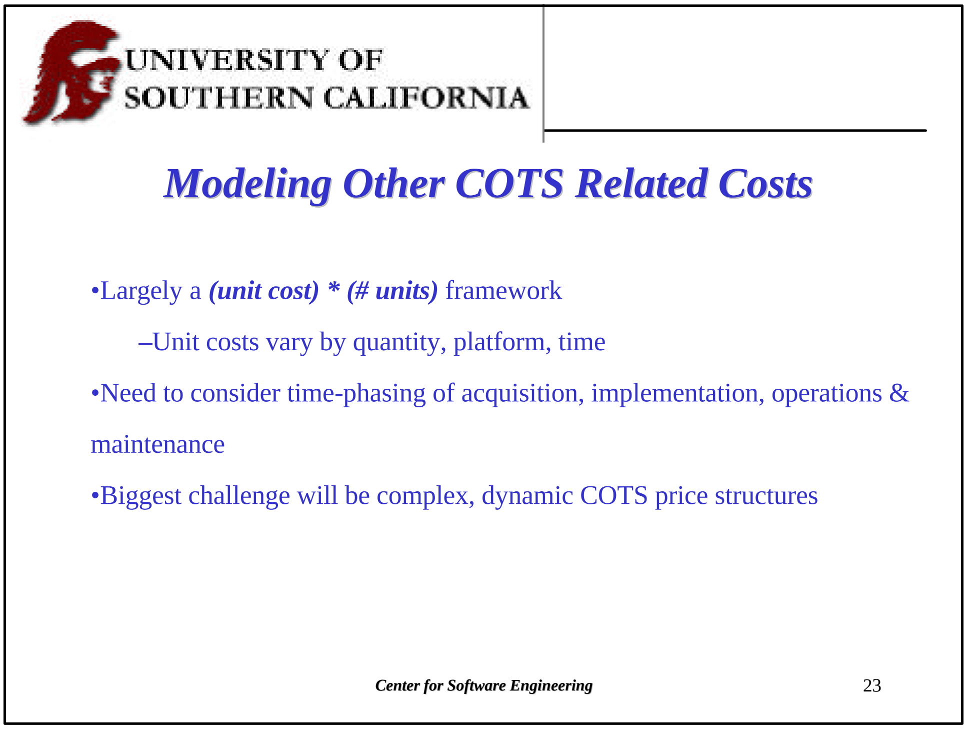
where
Assessment Effort = Filtering Effort + Final Selection Effort

Total integration Cost =
(Total Integration Effort) • ($$/Person-Month)



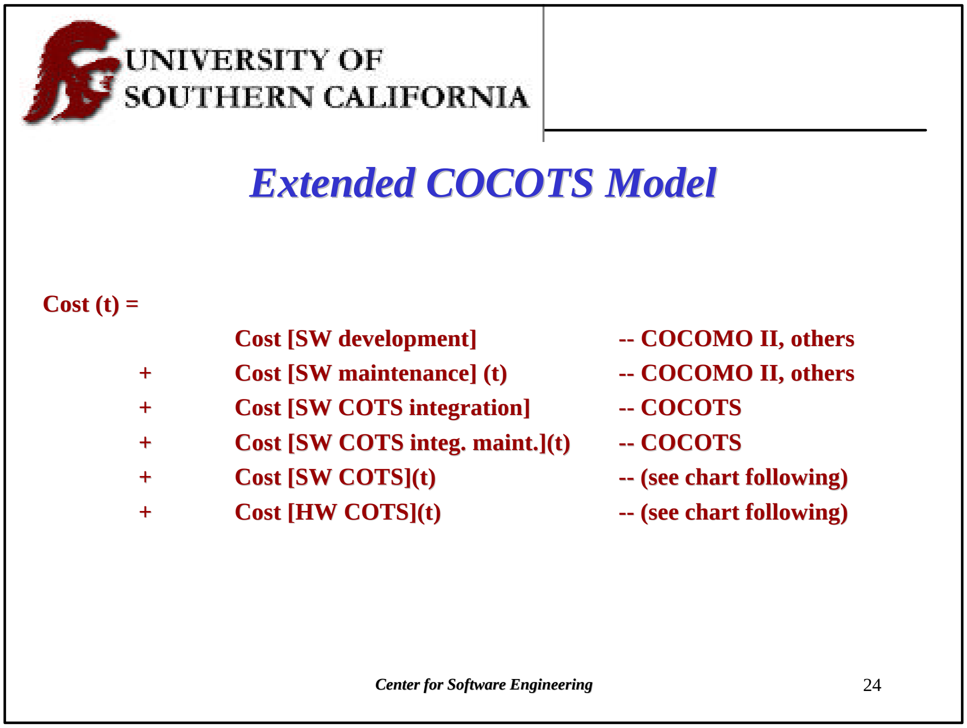
Prospective COCOTS Follow-ons

- Extensive data collection and conditioning
- Recalibration and iteration of the model within current structure
- Experimental usage and refinement, including exploration of other cost drivers and model forms
- Modeling of schedule estimation and activity distribution
- Integration with COCOMO II estimation model
- More extensive model implementation
- Modeling other COTS related costs
  - Licenses, training, maintenance, hardware



Modeling Other COTS Related Costs

- Largely a ***(unit cost) * (# units)*** framework
  - –Unit costs vary by quantity, platform, time
- Need to consider time-phasing of acquisition, implementation, operations & maintenance
- Biggest challenge will be complex, dynamic COTS price structures



Extended COCOTS Model

Cost (t) =

	Cost [SW development]	**-- COCOMO II, others**
+	**Cost [SW maintenance] (t)**	**-- COCOMO II, others**
+	**Cost [SW COTS integration]**	**-- COCOTS**
+	**Cost [SW COTS integ. maint.](t)**	**-- COCOTS**
+	**Cost [SW COTS](t)**	**-- (see chart following)**
+	**Cost [HW COTS](t)**	**-- (see chart following)**

Estimating Cost of Software COTS

Cost [SW COTS](t) =

Cost[acquisition office](t)

+ Cost[licenses](t) {= Cost[#licenses,#features,#platforms](t)}

+ Cost[implementation](t) {= Cost[training](t) + Cost[install](t)}

+ Cost[Op. & Mnt.](t) {= Cost[maint. lic’s.](t) + Cost[support](t)}

Estimating Cost of Hardware COTS

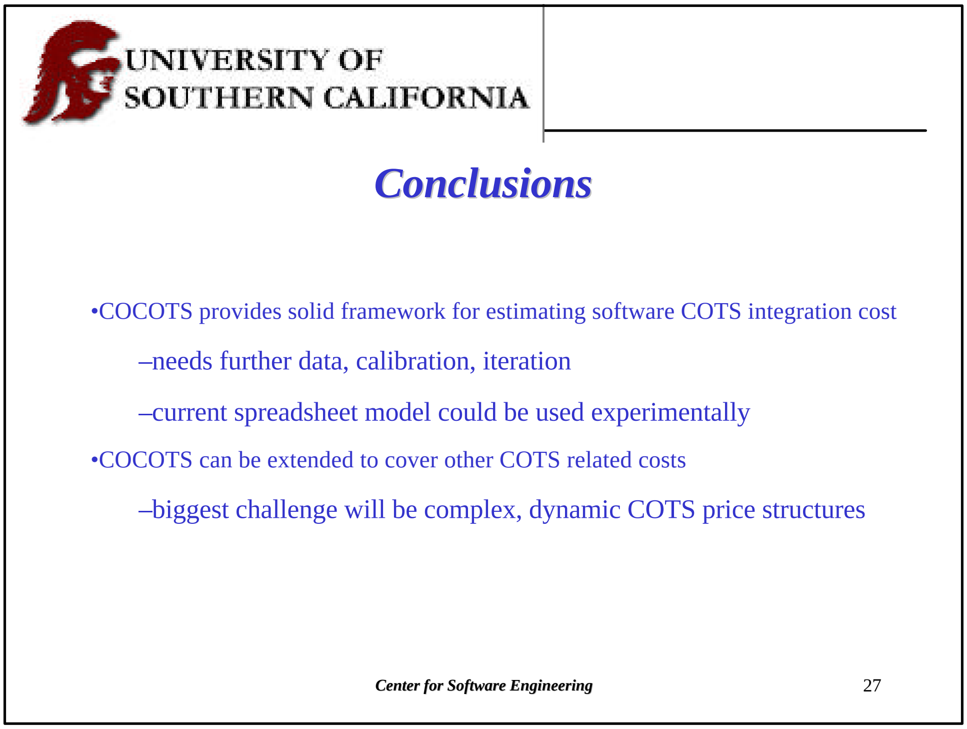
Cost[HW COTS](t) =

Cost[acquisition office](t)

$+ \sum_{i} \{\text{Cost[acquistion]}_i(t) + \text{Cost[implementation]}_i(t) + \text{Cost[O\&M]}_i(t)\}$

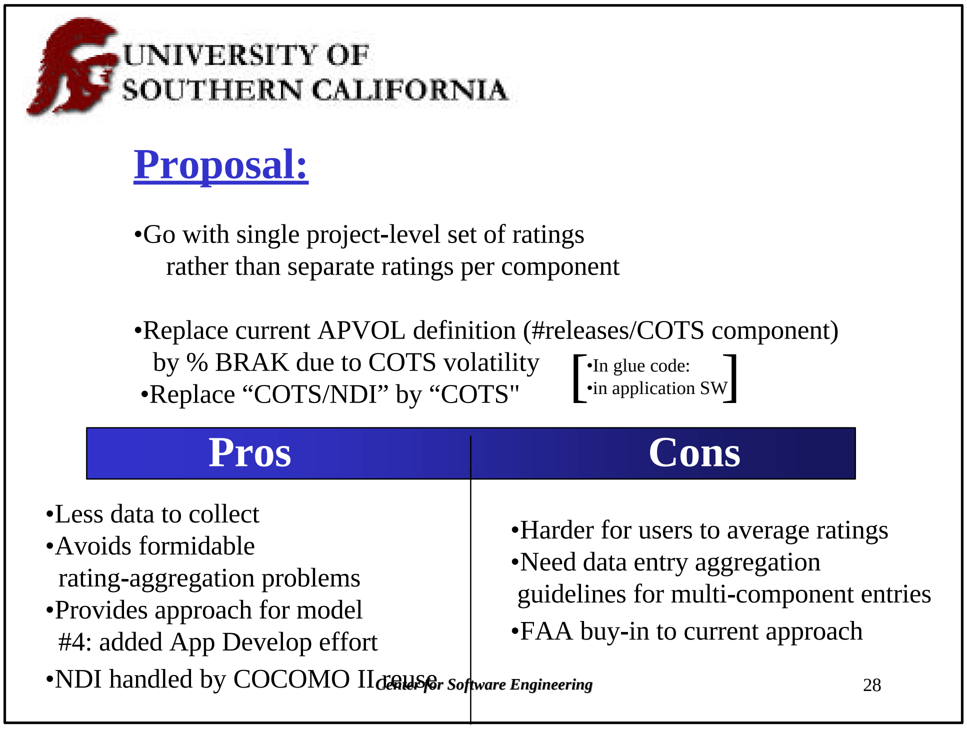
for

i = processors, storage, workstation, communications



Conclusions

- COCOTS provides solid framework for estimating software COTS integration cost
  - needs further data, calibration, iteration
  - current spreadsheet model could be used experimentally
- COCOTS can be extended to cover other COTS related costs
  - biggest challenge will be complex, dynamic COTS price structures



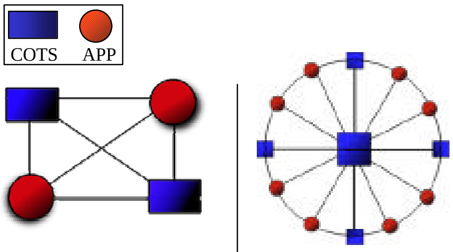
Proposal:

- Go with single project-level set of ratings rather than separate ratings per component

- Replace current APVOL definition (#releases/COTS component) by % BRAK due to COTS volatility [•In glue code: •in application SW]
- Replace “COTS/NDI” by “COTS"

Pros	Cons
•Less data to collect •Avoids formidable rating-aggregation problems •Provides approach for model #4: added App Develop effort •NDI handled by COCOMO II reuse	•Harder for users to average ratings •Need data entry aggregation guidelines for multi-component entries •FAA buy-in to current approach

Rating-Aggregation Problems



#COTS	2	4	1
# interactions with APP, COTS	2,1	2,1	8,4

- Can’t just average ratings
  - relative # interactions of COTS
  - relative interaction complexity
  - volatility effects
    - breakage per release
    - aggregation of release updates

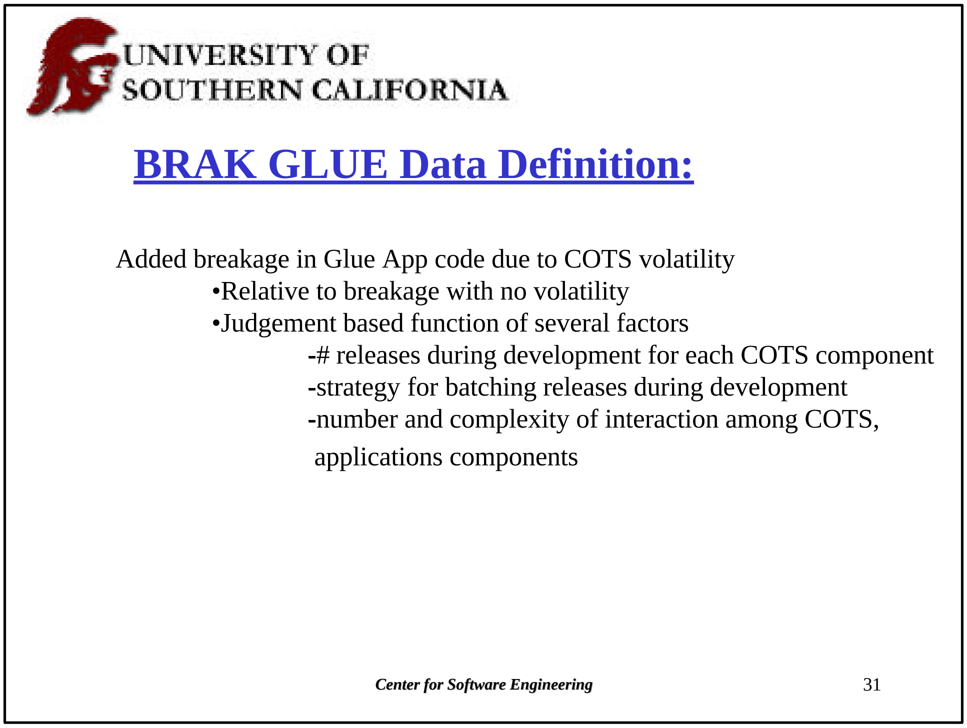
- No simple formulas for aggregating those effects

Disposition:

- Go with project-level BRAK GLUE parameter
  - avoid aggregation difficulties
  - includes effects of application volatility
  - compatible with BRAK COTS approach for added applications effort

- Leave other cost drivers at component level
  - easier user data/rating entry
  - start with simple averaging of ratings

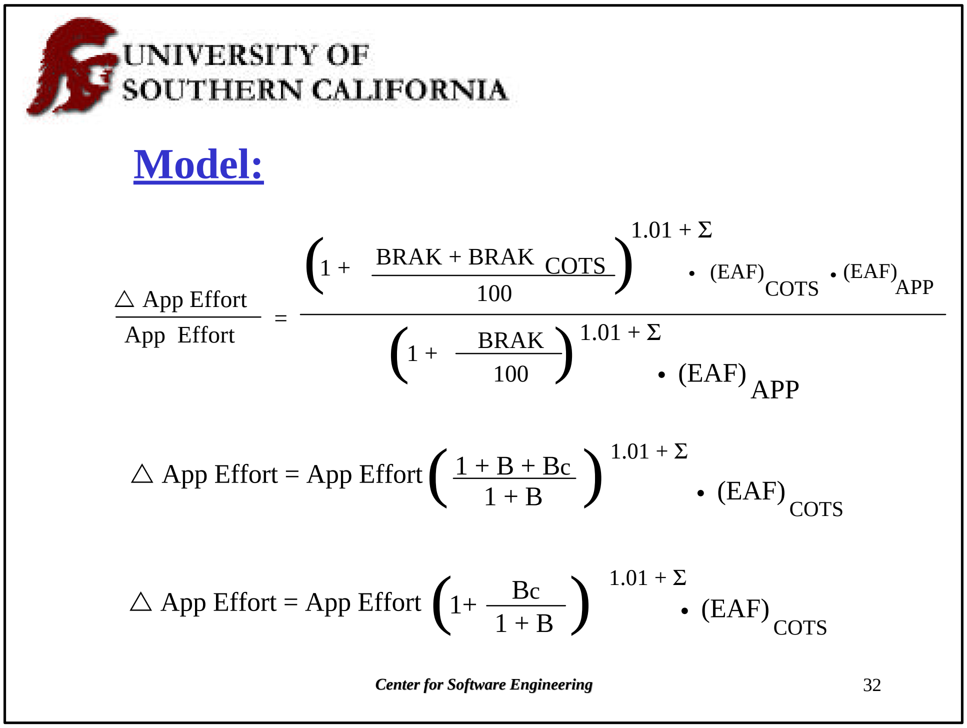
- Replace “COTS/NDI” by “COTS”



BRAK GLUE Data Definition:

Added breakage in Glue App code due to COTS volatility

- •Relative to breakage with no volatility
- •Judgement based function of several factors
  - -# releases during development for each COTS component
  - -strategy for batching releases during development
  - -number and complexity of interaction among COTS, applications components



Model:

$$\frac{\triangle \text{ App Effort}}{\text{App Effort}} = \frac{\left(1 + \frac{\text{BRAK} + \text{BRAK}_{\text{COTS}}}{100}\right)^{1.01 + \Sigma} \bullet (\text{EAF})_{\text{COTS}} \bullet (\text{EAF})_{\text{APP}}}{\left(1 + \frac{\text{BRAK}}{100}\right)^{1.01 + \Sigma} \bullet (\text{EAF})_{\text{APP}}}$$

$$\triangle \text{ App Effort} = \text{App Effort} \left(\frac{1 + B + Bc}{1 + B}\right)^{1.01 + \Sigma} \bullet (\text{EAF})_{\text{COTS}}$$

$$\triangle \text{ App Effort} = \text{App Effort} \left(1 + \frac{Bc}{1 + B}\right)^{1.01 + \Sigma} \bullet (\text{EAF})_{\text{COTS}}$$