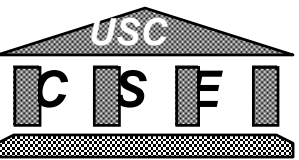


# Developing COTS-Based Applications

Barry Boehm, Dan Port, Ye Yang, Jesal Buhta  
*Center for Software Engineering*  
*University of Southern California*  
*yangy@sunset.usc.edu*





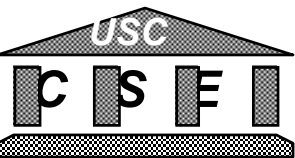
# Introduction

- **Increasing COTS-composing Trend**
- **Traditional sequential processes do not work for COTS-Based Applications (CBA's).**
- **We developed a CBA process decision framework and a set of composable process elements.**



# Definitions and Context

- **COTS (Commercial Off-The-Shelf)**
  - A product that is ---- by SEI
    - Sold, leased, or licensed to general public
    - Offered by a vendor trying to profit from it
    - Supported and evolved by the vendor, which retains the intellectual property rights
    - Available in multiple, identical copies
    - Used without internal modification by a consumer
  - No vendor guarantees!
- **COTS classification by role**
  - COTS infrastructure
  - COTS tool
  - COTS Application component ---- by COCOTS



# CBS and CBA

- **COTS-Based System:**
  - A system that uses COTS ---- by SEI
- **CBA:**
  - A system for which at least 30% of the end-user functionality (in terms of functional elements: inputs, outputs, queries, external interfaces, internal files) is provided by COTS products, and at least 10% of the development effort is devoted to COTS considerations. ---- by CSE



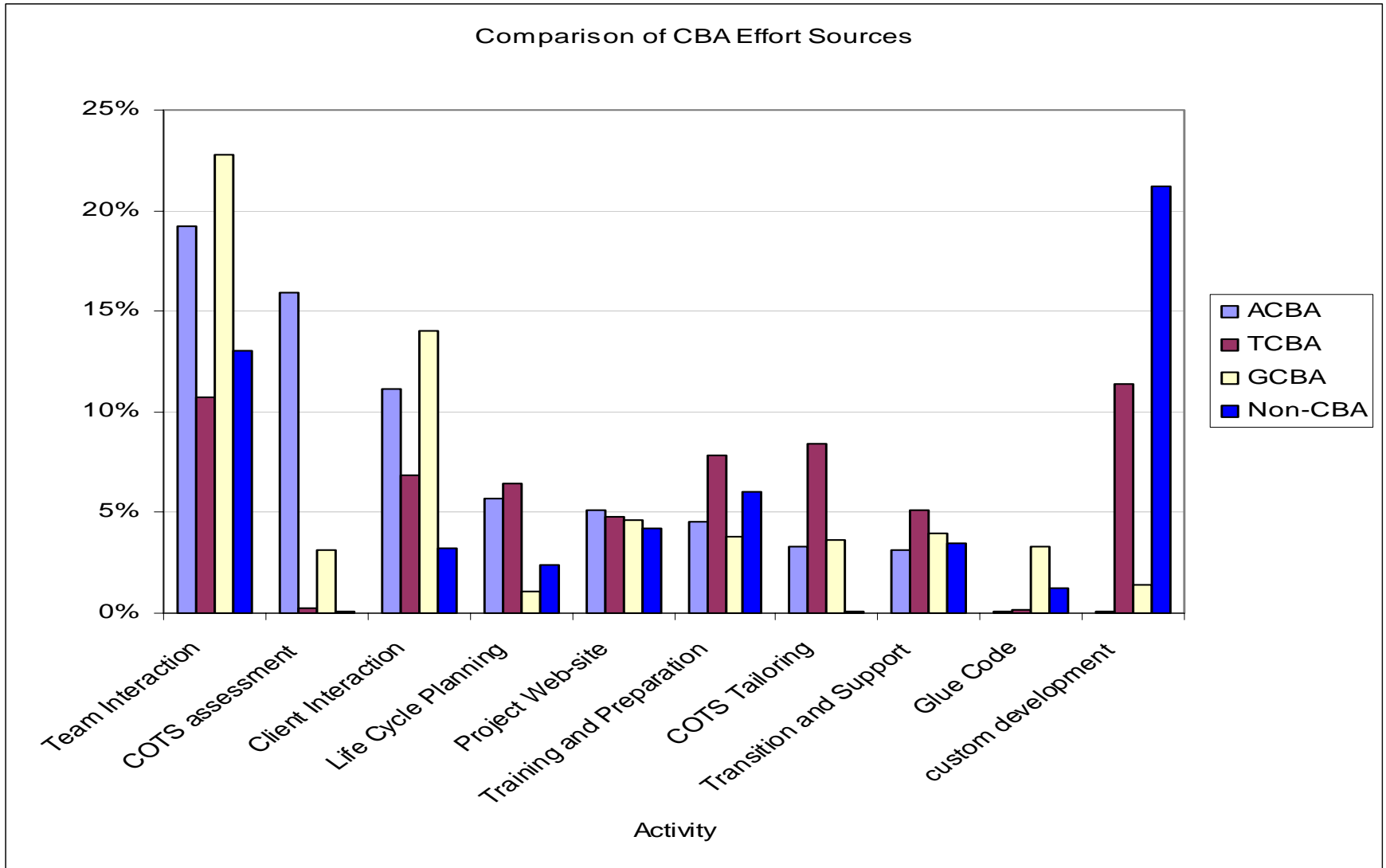
# COTS Related Activities

- **COTS assessment**
  - Activity of determining the appropriateness of feasibility of using specific COTS to fulfill required system functions
- **COTS tailoring**
  - Activity associated with setting or defining shell parameters or configuration options for a COTS, but which do not require modification of source code, including defining I/O report formats, screens, etc.
- **Glue code**
  - Custom development needed to integrate COTS packages within an application external to the packages themselves



# CBS Types

- **Four types:**
  - **Assessment Intensive CBS**
  - **Tailoring Intensive CBS**
  - **Glue Code Intensive CBS**
  - **Non-COTS Intensive CBS**
- **Effects of CBS Types:**
  - **MBASE development guidelines**
  - **Glue code, assessment, tailoring effort**
  - **Risk factors**
  - **Requirements flexibility**
  - **System evolution and maintenance**
  - **Cost factors**

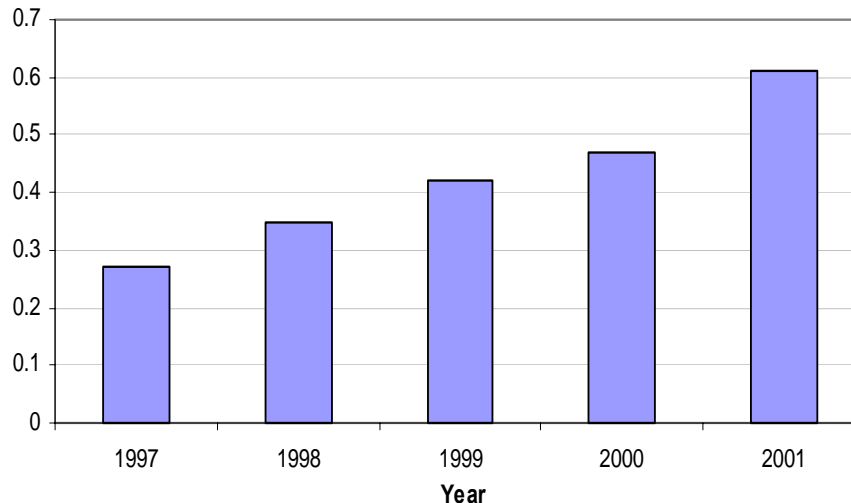


Source: CS577 2001-2002 Project Effort Data

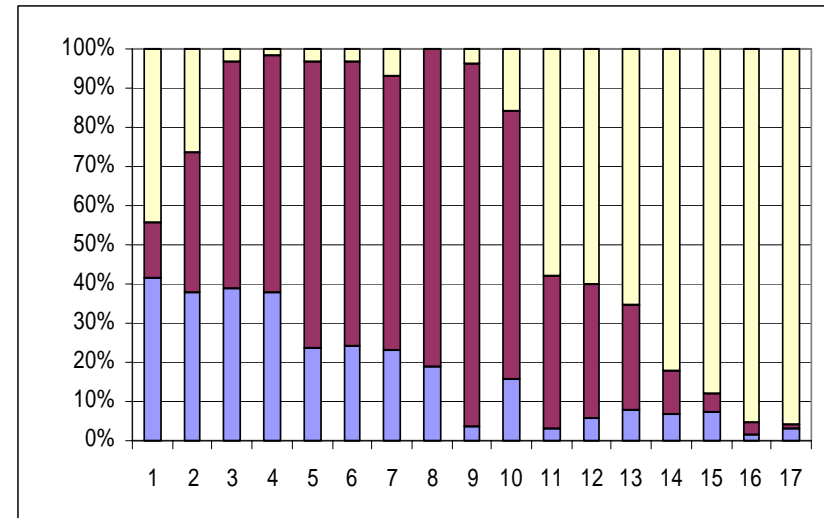
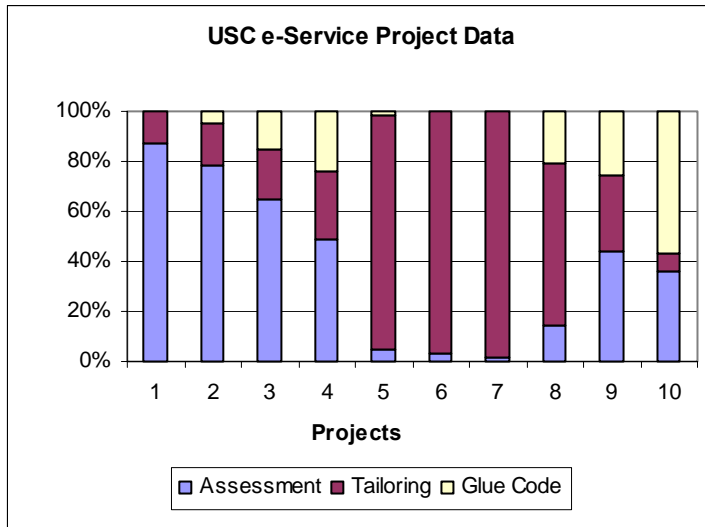
# CBA Growth Trend

- **CBA Definition again**
  - A system for which at least **30% of the end-user functionality (in terms of functional elements: inputs, outputs, queries, external interfaces, internal files) is provided by COTS products, and at least 10% of the development effort is devoted to COTS considerations.**
  - A **5-year longitudinal analysis of cs577 e-services applications showed a growth from 28% CBA's in 1997 to 60% in 2001**

fraction of projects satisfying (30%, 10%) CBA criteria

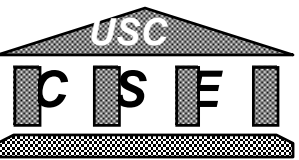


# CBA Effort Distributions



- **USC e-Service project data**
  - **24 weeks fixed schedule**

- **COCOTS Calibration Data**
  - **Small to large business mgmt, analysis, and control app**



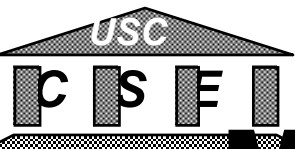
# CBA Effort Sequences

- **Some Observations from CS577 Projects**

- **Assessment (A)**
- **Tailoring (T)**
- **Glue code Development (G)**
- **Custom Development (C)**

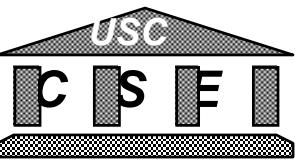
No.	Effort Sources Map
1	ATGC
2	ATA
3	A(TG)AG
4	A(TG)A(TG)

**Table 1.** CBA Effort Sequences



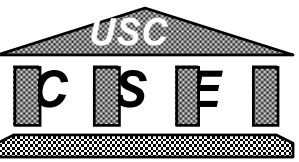
# **MBASE Guideline vs. CBA Process**

- **No one-size-fits-all process**
- **Reconcile conflict between CBA process needs and our UML-based MBASE process and documentation guidelines**
- **Example:**
  - **CS577 01-02: USC Collaborative Services**



# USC Collaborative Services

- **Assessment Intensive CBA**
  - Four COTS packages to be assessed in 577a01
  - Two new candidates introduced at the beginning of 577b02
- **Volatile requirements and COTS candidates**
- **Stick to process-mandated UML based MBASE Guideline?**

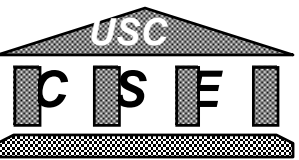


# USC Collaborative Services

- Resulted in a good deal of confusion, frustrating re-work!

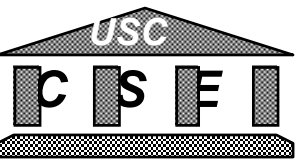


<<boundary>>  
Admin Login  
(from Enterprise Model)



# SSAD Re-work Cause

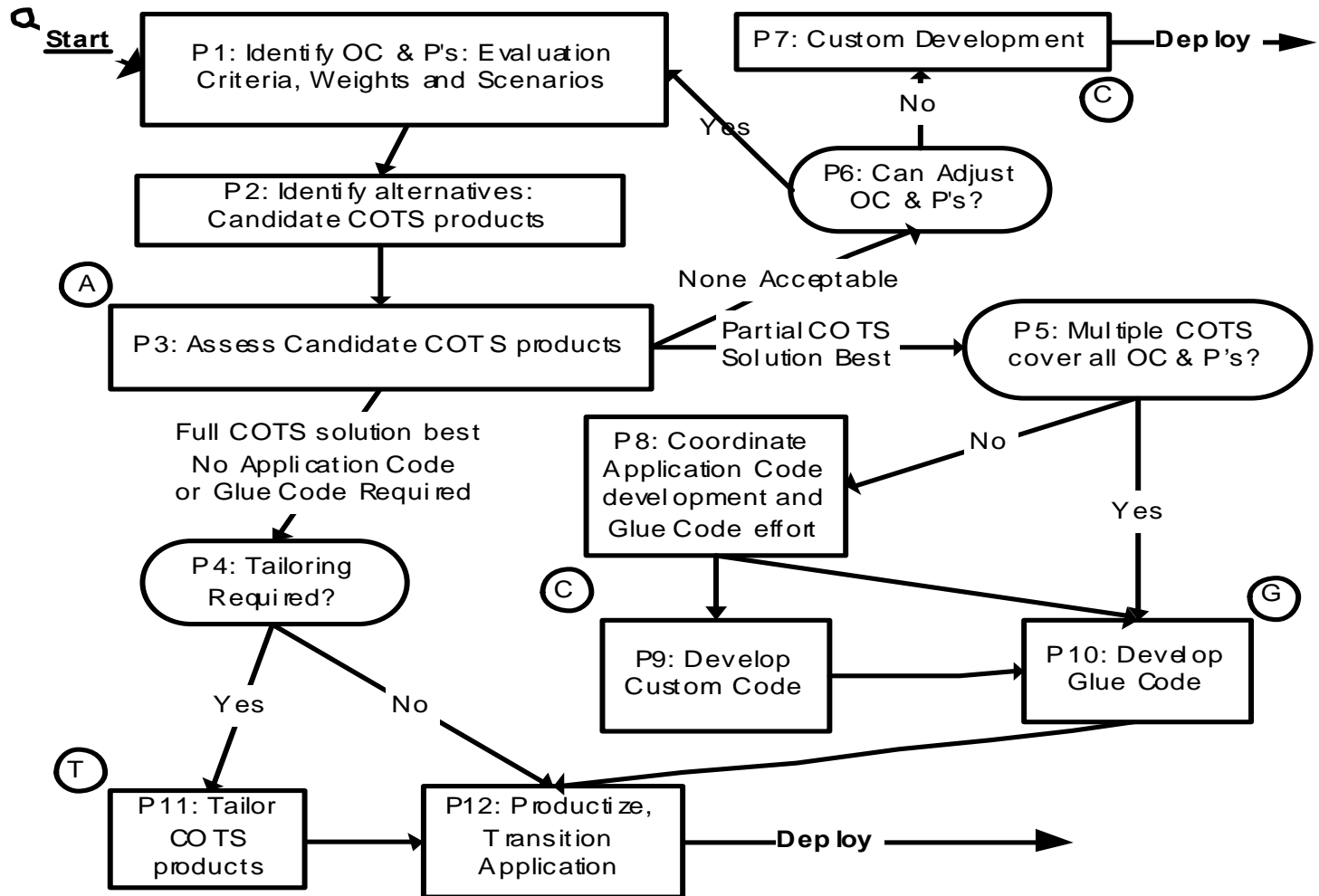
- **System architecture is focused on a COTS package configuration model, since the class model, object model and interaction model are not available for most COTS products.**

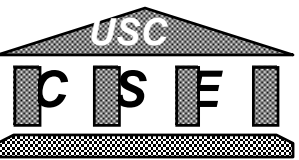


# CBA Decision Framework

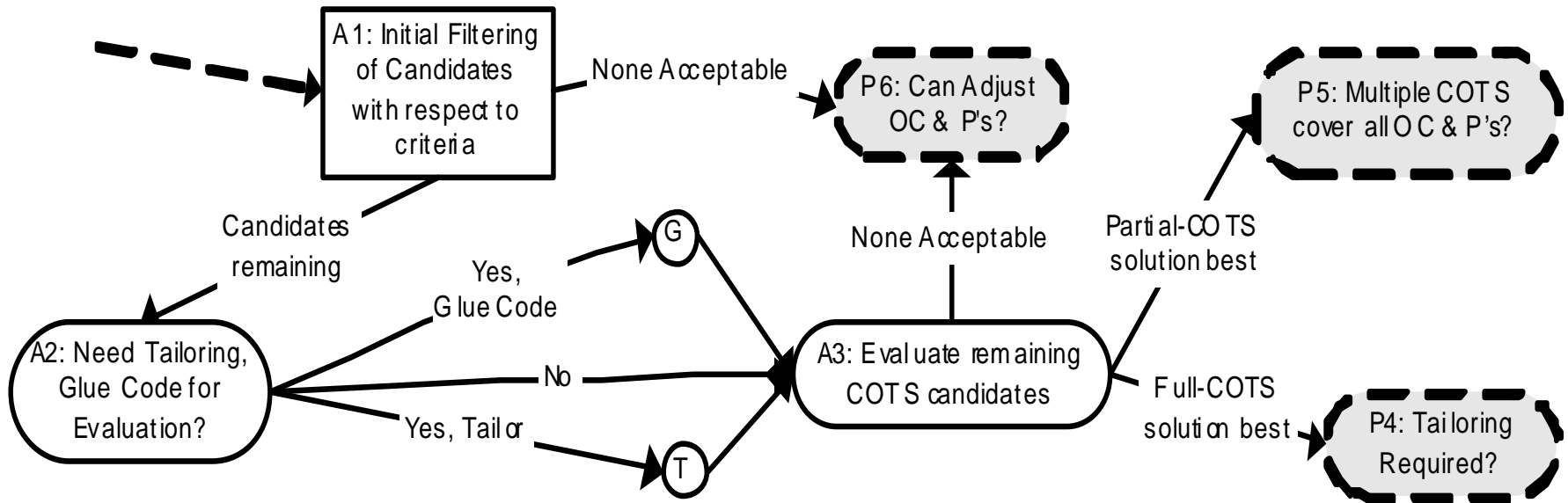
- **For CBAs, no clear requirements at front, just OC&P's (Objectives, Characteristics, and Priorities)**
- **COTS assessment, tailoring, glue code activities are critical development process elements**
- **CBA Decision Framework is a recursive and reentrant process to accommodate concurrent CBA activities and frequent go-backs based on new and evolving OC&P's and COTS considerations**

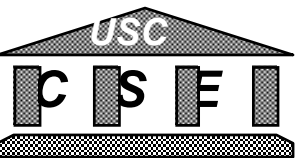
# CBA Decision Framework





# Assessment Process Elements





# Assessment Example

- **USC Collaborative Services (USCCS):**
  - **OC&P's**
    - Project management
    - File management
    - Discussion board
    - Project calendaring
    - Chat room
  - **COTS Candidates**
    - eProject (577a and 577b)
    - Dot Project (577a)
    - eStudio (577a)
    - eRoom (577a)
    - Blackboard (577b)
    - iPlanet (577b)



# USCCS COTS Assessment steps

- **Plan**
  - Set the evaluation effort scope and provide a baseline for evaluation process.
- **COTS Identification**
  - Search for candidate COTS packages based on the set of DC&P's
- **Establish evaluation criteria**
  - Based on functional, performance, architectural, and financial characteristics of DC&P's
- **Conduct multiple cycle of evaluation**
  - Collect evaluation data against the defined evaluation criteria.
- **Perform Hand-on Experiments**
  - To verify vendor's claim
- **Data analysis**
  - Compare and contrast the evaluation data and make COTS final selection.



# Evaluation Criteria

No.	Criteria	Description	Weight
1	Functionality	The degree to which a COTS component has the functional capability needed by the system. For determinators, referring to the feature checklist in table 8.	500
2	Ease of Use	The ease with which a user can learn to operate, prepare inputs for and interpret outputs of a system	150
3	Product Performance	The degree to which a COTS component perform its functions within given timing constraints.	150
4	Security	The degree to which a system or component prevents unauthorized access to or modification of data.	150
...			
17	Maturity of product	The degree that a COTS component has been commercially available	50

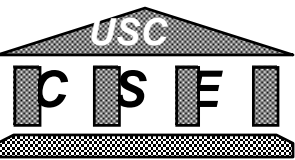
# Table 8 – Feature Checklist

Weight	Feature	Sub-weight	Sub-feature
0.2	Project management	0.03	Has scheduling function?
		0.02	Has task notification function? (new task)
		0.03	Has task progress report function?
		0.03	Has task reminder function? (when login)
		0.03	Has workflow control feature?
		0.03	Has support for printing out task report?
		0.03	Has change schedule function?
0.2	File management	0.03	Has Configuration management support?
		0.02	Has Access control?
...	...	...	...
	total	1	



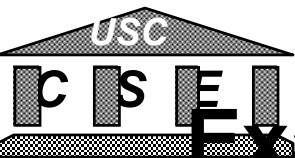
# Example: COTS Evaluation Results

Weight	Evaluation Criteria	eProject		iplanet		Blackboard	
		Average	Score	Average	Score	Average	Score
90	Inter-compatibility	6.4	576	7	630	8	720
150	Product Performance	8.8	1320	9	720	8	1200
500	Functionality	8.239	4119.5	5	500	7.28	3640
60	Documentation	8.2	492	9	540	9	540
80	Flexibility	6.6	528	8	640	8	640
50	Maturity of Development	9.8	490	10	500	9	450
80	Vendor support	9.2	736	9	720	9	720
150	Security	9.2	1380	9	810	9	1350
150	Ease of use	8.4	1260	8	720	7	1050
80	Training	7	560	7	560	8	640
60	Ease of Installation	7	420	7	420	8	480
60	Ease of maintenance	6.2	372	8	480	8	480
100	Scalability	8.4	840	9	900	8	800
60	vendor viability	8.6	516	8	480	9	540
100	Compatibility	6.6	660	10	1000	9	900
80	Evolution and Upgradeability	7.2	576	8	640	9	720
90	Ease of Integration	9	810	8	720	8	720
<b>1850</b>	<b>Total</b>		<b>15655.5</b>		<b>14630</b>		<b>15590</b>
	<b>Average</b>		<b>8.06985</b>		<b>7.5412</b>		<b>8.0361</b>



# Business Case Analysis

- **A business case analysis for each COTS solution is highly recommended for the detailed COTS assessment**
  - **COCOTS Model for effort and schedule estimation**
    - **Assessment, glue code, tailoring effort tradeoffs**
  - **Return on Investment analysis**
    - **Value, risk, cost tradeoffs**
  - **COTS specific risks and mitigation costs**
    - **vendor volatility, upgrade lifecycle, etc.**
- **Example: Dot-Project infeasibility in CS577A 2001 team 8**

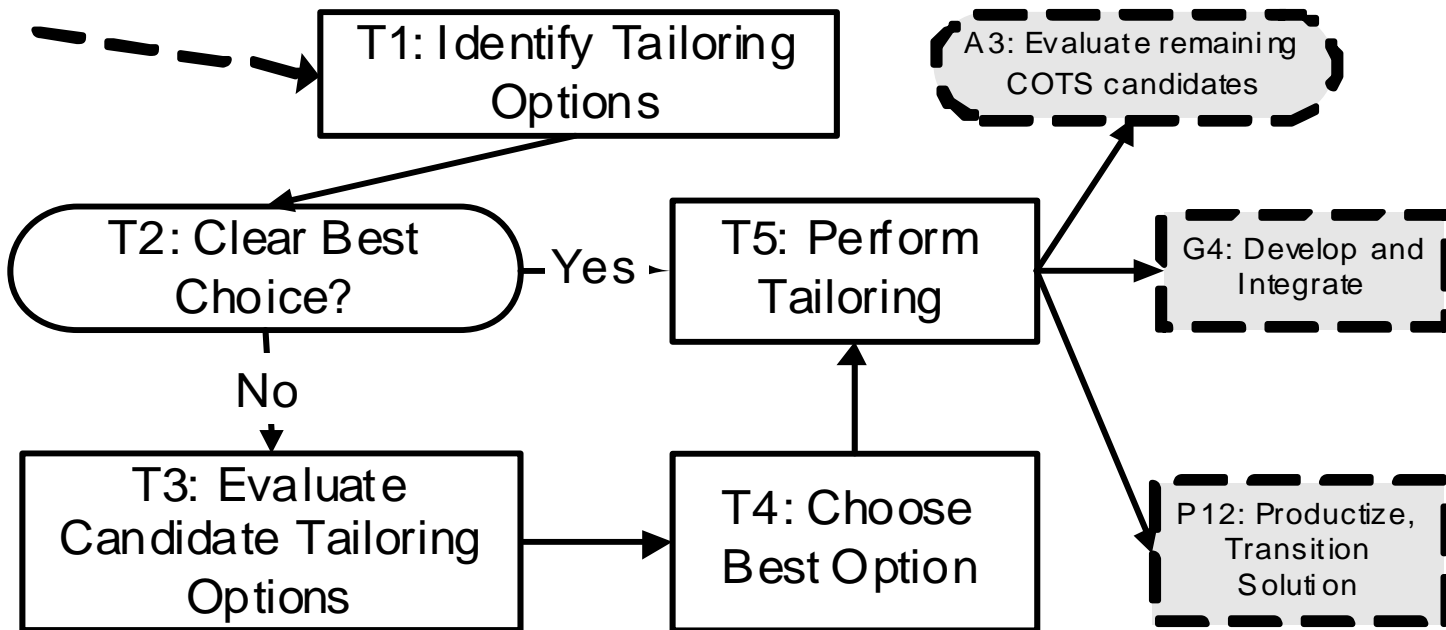


# Example - Comparison of results from initial assessment and detailed assessment

Evaluation Requirement	COTS candidate 1- eProject		COTS candidate 2- Blackboard		COTS candidate 3- iPlanet collaborative package	
	Initial	Detailed	Initial	Detailed	Initial	Detailed
Users can upload/modify/download/delete authorized files	**	*****	**	**	*	None
Project management, including planning group tasks, timelines and tracking	**	*****	**	***	None	None
Message Board	**	*****	**	****	None	None
User Interface	**	*****	**	**	**	*
Admin Interface	**	*****	**	****	**	None
Group Calendar	**	*****	**	**	**	*

Detailed analysis provides greater assurance of COTS characteristics with respect to vendor documentation (although at significant effort)

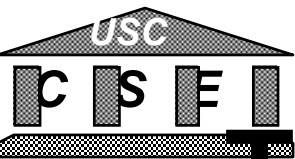
# Tailoring Process Elements





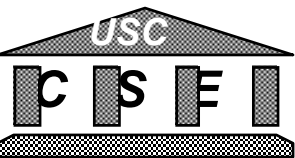
# Tailoring Types

- **The focus of the system architecture depends upon the type of the COTS package tailoring**
  - **GUI Based Tailoring (ex. Spearmint)**
    - No focus on system architecture
  - **Parameter Based Tailoring (ex. Windows Media Player)**
    - Little or no focus on system architecture
  - **Programmable based Tailoring Interface (ex. Hyperwave)**
    - Moderate focus on system architecture



# Types of Tailoring options and Characteristics

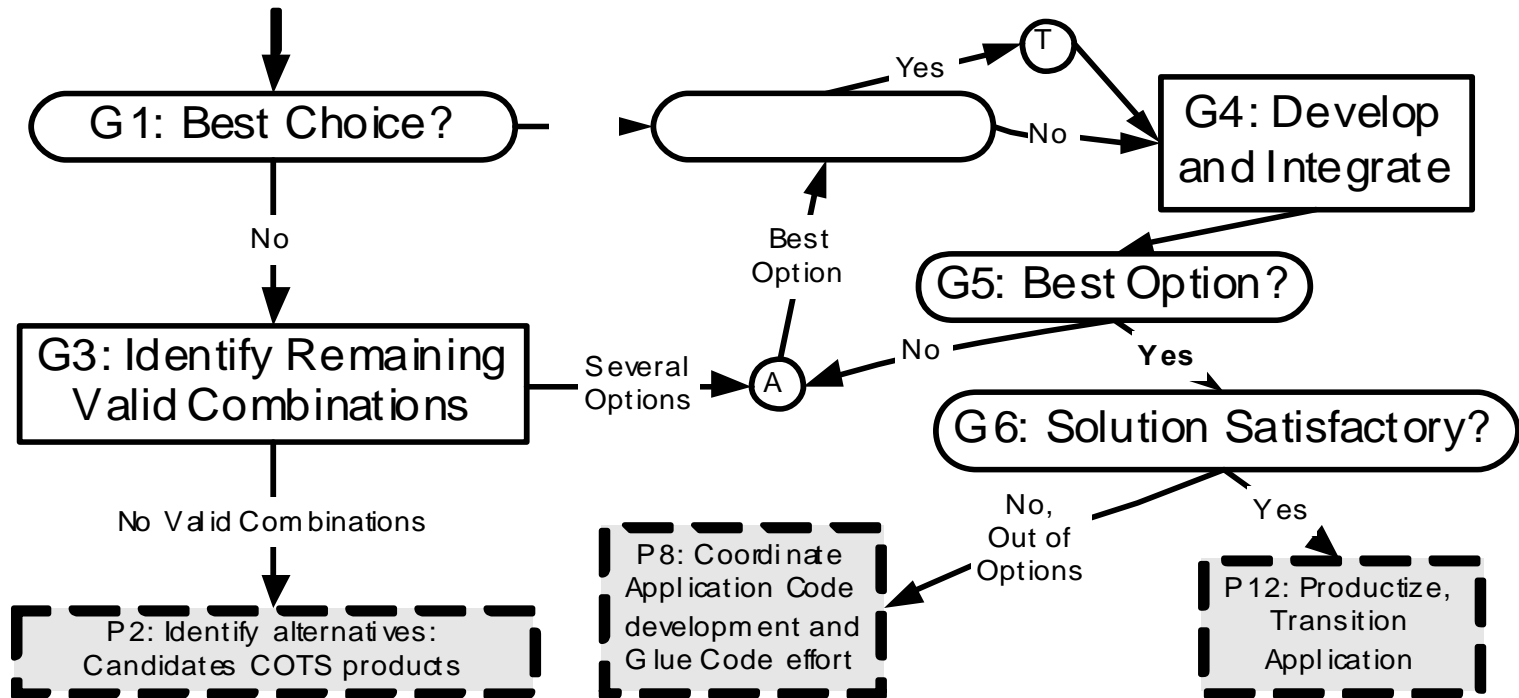
Options Eval. Parameters	GUI based Interface	Parameter Based Interface	Programmable Interface
<b>Design Details Required</b>	Low - None	Low	Detailed
<b>Complexity</b>	Low - Moderate	Moderate	High
<b>Adaptability</b>	Low	Low - Moderate	High
<b>Developer Resources</b>	Low	Low - Moderate Mr. Sid /	Moderate - High
Example	Real Player	Windows Media Player	Browsers - Java Scripts
		J2EE	
	Windows Media Encoder		
	Microsoft Word 2000		



# Tailoring Example

- **Team 21 in cs577'01-02:**
  - **Quality Mgmt. in BORE**
  - **BORE is the main COTS**
    - **Hosted at Univ. of Nebraska at Lincoln**
    - **Building an Organizational Repository of Experiences**
    - **“There's often time that BORE's down. We'r frustrating by the fact that we can't work on BORE whenever we want to, and we afraid things we've put in BORE might just disappear ..” -by the team leader**
  - **Tailoring Activities**
    - **Put content, create rules & question, etc.. into BORE, including updates (because periodically BORE upgrades )**

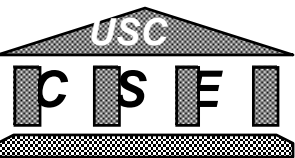
# GlueCode Process Elements





# Glue Code examples

- **Team 15 DART 2001-2002**
  - Tomcat, MySQL
- **Team 17 Web-mail 2000-2001**
  - Tomcat
- **Team 8 Fulltext Title Search 2000-2001**
  - MySQL, Tomcat, Zope
- **Team 9 Student/Staff Directories 2001-2002**
  - Zope

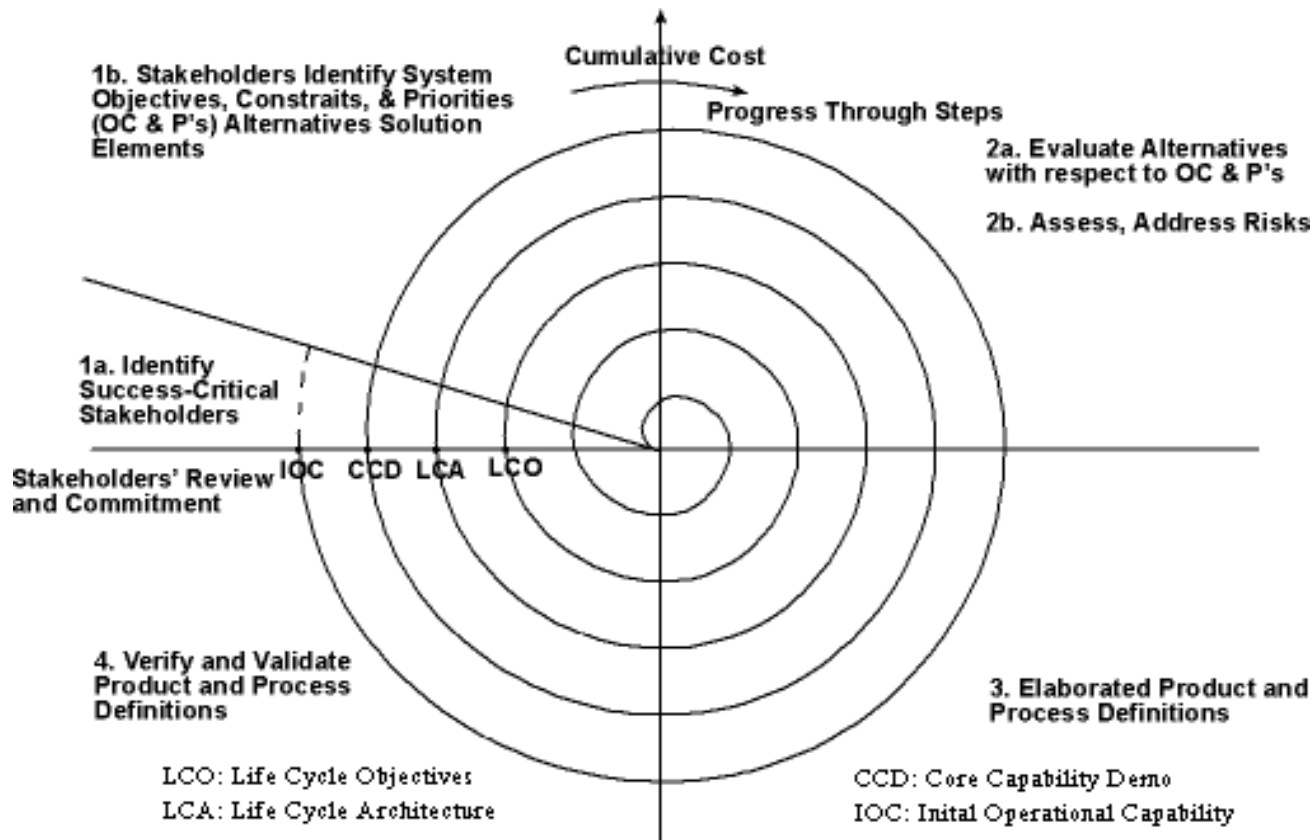


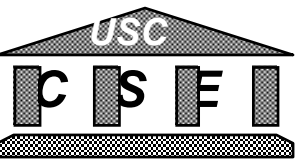
# Top CBA Project Risks

No.	Risk Items	No.	Risk Items
1	Requirements Changes and Mismatches	9	Difficulty in coordinating meetings with key personnel may result in significant delays
2	Many new non technical activities are introduced in the Software Engineering Process	10	Inadequate vendor support may result in significant project delays
3	Miss possible COTS candidates within the COTS process	11	COTS package incompatibilities may result in feature loss and significant project delays (Integration Clash)
4	Too much time spent in assessment due to too many requirements and too many COTS candidates	12	Added complexity of unused COTS features
5	Might not include all key aspects for establishing evaluation criteria set. (Inadequate COTS assessment)	13	Overly optimistic expectations of COTS quality attributes
6	Introducing new COTS candidates is likely and requires re-planning	14	Overly optimistic COTS package learning curve
7	Faulty Vendor Claims may result in feature loss and/or significant delays	15	A version upgrade may result in re-tailoring of COTS package
8	Ability or willingness of the organization to accept the impact of COTS requirements	16	Imposed black box testing of COTS components

**Source: USC e-service projects 2000-2002**

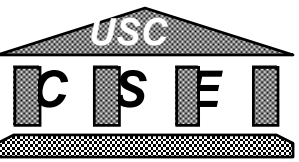
# Elaborated Win Win Spiral Model





# Example CBA: Oversize Image Viewer

- **Development of a viewing capability for oversized images.**
- **Several COTS products were available for the image processing functions, each with its strengths and weaknesses. None could cover the full system capability.**



# Applying the Decision

## Framework and the Win Win

### Spiral Model

- **First three spiral cycles**
- **Each cycle description begins with its use of the Win Win Spiral Model, as the primary sequencing of tasks is driven by the success-critical stakeholders' win conditions and the project's major risk items.**



# Spiral Model Application to Oversize Image Viewer (1)

	Cycle 1	Cycle 2 (LCO)	Cycle 3 (LCA)
<b>Stakeholders</b>	Developer, customer, library-user client, COTS vendors	Additional user representatives (Unix, Mac communities)	Additional end-users (staff, students) for usability evaluation
<b>OC&amp;P's</b>	Image navigation, cataloguing, search, archive and access administration COTS cost $\leq$ \$25K, $\geq$ 5 user organizations IOC developable in 24 weeks	System usable on Windows, Unix, and Mac platforms	Detailed GUI's satisfy representative users
<b>Alternatives</b>	ER Mapper, Mr SID, Systems ABC, XYZ	ER Mapper, Mr SID	Many GUI alternatives
<b>Evaluation; Risks</b>	XYZ $>$ \$25K; ABC $<$ 5 user org's ER Mapper, Mr SID acceptable Risk picking wrong product without exercise	ER Mapper Windows-only; plans to support Unix, Mac; schedule unclear Mr SID supports all 3 platforms Risk of Unix, Mac non-support	Risk of developing wrong GUI without end-user prototyping Mr SID/My SQL/Java interoperability risks
<b>Risk Addressed</b>	Exercise ER Mapper, Mr SID	Ask ER Mapper for guaranteed Unix, Mac support in 9 months	Prototype full range of system GUI's, Mr SID/My SQL/Java interfaces



# Spiral Model Application to Oversize Image Viewer (2)

	Cycle 1	Cycle 2 (LCO)	Cycle 3 (LCA)
<b>Risk Resolution</b>	ER Mapper image navigation, display stronger	ER Mapper: no guaranteed Unix, Mac support even in 18 months	Acceptable GUI's, Mr SID/My SQL/Java glue code
<b>Product Elaboration</b>	Use ER Mapper for image navigation, display	Use Mr SID for image navigation, MySQL for catalog support, Java for admin/GUI support	Develop production Mr SID/My SQL/Java glue code
<b>Process Elaboration</b>	Tailor ER Mapper for library-user Windows client	Prepare to tailor Mr SID, My SQL to support all 3 platforms	Use Schedule as Independent Variable (SAIV) process to ensure acceptable IOC in 24 weeks
<b>Product Process</b>	Customer: want campus-wide usage, support of Unix, Mac platforms	Need to address Mr SID/My SQL/Java interoperability, glue code issues; GUI usability issues	Need to prioritize desired capabilities to support SAIV process
<b>Commitment</b>	Customer will find Unix, Mac user community representatives	Customer will buy Mr SID Users will support GUI prototype evaluations	Customer will commit to post-deployment support of software Users will commit to support training, installation, operations



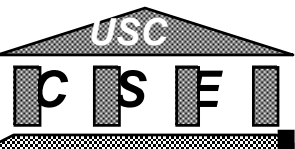
# Use of the CBA Decision Framework in Cycle 1

- **Entrance:**
  - The first three steps establish the preconditions for entering the CBA Assessment decision framework
    - Such as top-level evaluation criteria, weights, and scenarios; candidate COTS products
  - Spiral step 4 establishes the entry into *Assessment process element*
- **Assessment Process Element**
  - Detailed assessment involved *Tailoring* to accommodate the newspaper image files, and identified ER Mapper as the best OIV solution.
- **Exit:**
  - In the “Partial COTS solution best” direction.



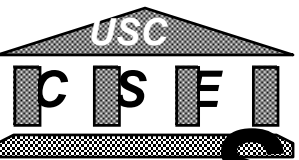
# Use of the CBA Decision Framework in Cycle 2

- **Go-back:**
  - The new stakeholders and OC&P's in cycle 2 required the project to backtrack to the beginning of the *Assessment process element*.
- **Assessment Process Element (1)**
  - ER Mapper was filtered out when it declined to guarantee early Unix and Mac versions.
- **Tailoring Process Element**
  - *Tailoring* was required to verify that Mr. SID performed satisfactorily on Unix and Mac platforms.
- **Assessment Process Element (1)**
  - Concurrently, *Assessment* filtering and evaluation tasks were being performed for the cataloguing and GUI functions.



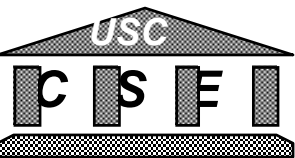
# Use of the CBA Decision Framework in Cycle 3

- **Assessment Process Element:**
  - The Assessment of detailed interoperability characteristics of Mr. SID, MySQL, and the GUI software on the Windows, Unix, and Mac platforms.
  - The interoperability assessment involved:
    - **Tailoring Process Element**
      - *Tailoring* was required to verify that Mr. SID performed satisfactorily on Unix and Mac platforms.
    - **Glue Code Process Element**
      - To enable interoperability



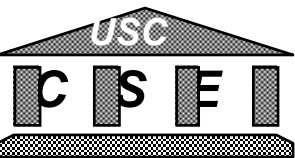
# Summary of CBA Decision Framework Use

- **Subsequent spiral cycles to develop the core capability and the IOC did not involve further Assessment, but involved concurrent use of the Tailoring, Glue Code, and custom development processes.**
- **The use of the CBA decision framework during the three spiral system definition cycles and the subsequent development activity can be summarized by the sequence A, T; (AA); A, (TG); (TGC).**



# Conclusion

- **CBA Definition**
- **CBA Process Decision Framework**
- **CBS Process Elements**
- **Applying the Decision Framework and Spiral Model**



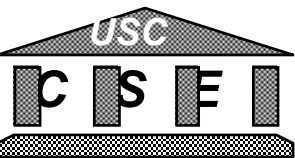
# Constructive COTS Integration Cost Model (COCOTS)

Ye Yang

*Center for Software Engineering*

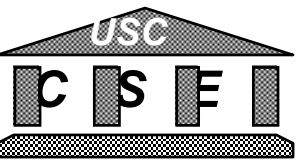
*University of Southern California*

*yangy@sunset.usc.edu*



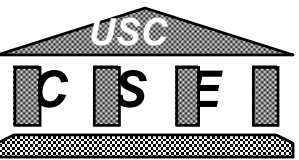
# Outlines

- **Introduction**
- **Iterations**
- **Forth Iteration Model**
- **Application example**
- **Model Calibration**



# Introduction

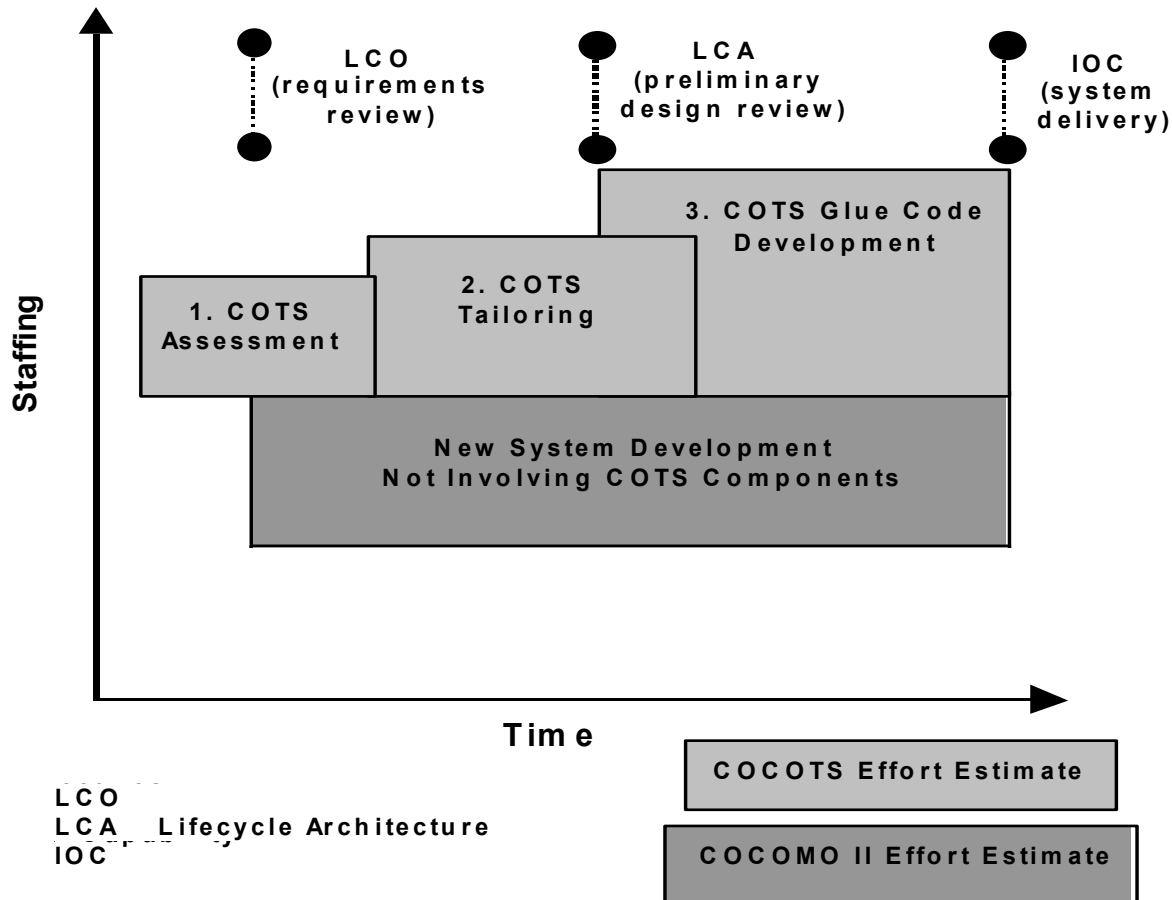
- **COTS integration becomes the most significant software development practices in the last 20 years**
- **COCOMO and COCOMO II: missing CBS development modeling**
- **COCOTS: an extension to COCOMO II to model COTS integration cost**

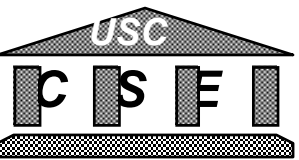


# Iterations

- **Dr. Chris Abts, USC-CSE**
- **Four iterations through the COCOMO II 7-steps waterfall modeling methodology**
  - **Exploration of concept**
  - **Refinement of model in “real world” context**
  - **Major revision of model**
  - **Simplification of model**

# Forth Iteration Model





# Assessment Submodel

- **COTS Assessment - the activity of determining the appropriateness or feasibility of using specific COTS products to fulfill required system functions**

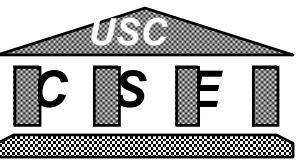
**Initial Filtering Effort (IFE) =**

**$\Sigma$  (#COTS Filtered in class) (Mean Filtering Effort per candidate for class)  
over all classes**

**Detailed Assessment Effort (DAE) =**

**$\Sigma$  (#COTS Detailed in class) (Mean Detailed Effort per candidate for class)  
over all classes**

**Final Project Assessment Effort (FPAE) = IFE + DAE**



# Tailoring Submodel

- **COTS Tailoring - the activity associated with setting or defining shell parameters or configuration options available for a COTS product, but which do not require modification of COTS source code, including defining I/O report formats, screens, etc**

**Project Tailoring Effort (PTE) =**

**$\Sigma$  (#COTS Tailored in class)(Mean Tailoring Effort for class)\*TCQ**  
**over all classes**

**Where**

**TCQ = Tailoring Complexity Qualifier with settings ranging from VL,  
L, N, H to VH**

# Assessment and Tailoring Sub-Models Parameters

COCOTS Classes	Assessment Parameters		Tailoring Parameters					
	Median Filtering Effort	Median Assessment Effort	Median Tailoring Effort	Tailoring Complexity Qualifier (TCQ)				
				VL	L	N	H	VH
1 Generic Component	0.63	1.33	4.00	0.69	0.83	1.00	1.21	1.45
2 Back office retail	0.00	0.00	3.00	0.69	0.83	1.00	1.21	1.45
3 Communication protocols/packages	0.00	0.00	1.00	0.69	0.83	1.00	1.21	1.45
4 Compilers	0.00	0.00	0.00	0.69	0.83	1.00	1.21	1.45
5 Configuration mgmt/build tools	0.00	0.00	0.00	0.69	0.83	1.00	1.21	1.45
6 Data conversion packages	0.75	0.00	0.00	0.69	0.83	1.00	1.21	1.45
7 Databases	1.91	5.75	38.29	0.69	0.83	1.00	1.21	1.45
8 Device drivers	0.00	0.00	3.00	0.69	0.83	1.00	1.21	1.45
9 Disk arrays	0.00	0.00	4.00	0.69	0.83	1.00	1.21	1.45
10 Emulators	0.08	0.00	3.00	0.69	0.83	1.00	1.21	1.45
11 Engineering tools (req mgmt, design)	0.00	0.00	0.00	0.69	0.83	1.00	1.21	1.45
12 Graphic information system	0.00	0.00	0.00	0.69	0.83	1.00	1.21	1.45
13 GUIs/GUI builders	0.10	0.67	14.00	0.69	0.83	1.00	1.21	1.45
14 Middleware	0.00	0.00	0.00	0.69	0.83	1.00	1.21	1.45
15 Network managers	0.27	7.17	12.67	0.69	0.83	1.00	1.21	1.45
16 Operating systems	0.50	0.50	2.00	0.69	0.83	1.00	1.21	1.45
17 Problem mgmt	0.00	0.00	0.00	0.69	0.83	1.00	1.21	1.45
18 Report generators	0.00	0.00	6.00	0.69	0.83	1.00	1.21	1.45
19 Software process tools	0.00	0.00	0.00	0.69	0.83	1.00	1.21	1.45
20 Telecommunication & infrastructure	0.00	0.00	0.00	0.69	0.83	1.00	1.21	1.45
21 Telemetry Analysis	0.00	0.00	0.00	0.69	0.83	1.00	1.21	1.45
22 Telemetry processing	2.00	2.37	0.00	0.69	0.83	1.00	1.21	1.45
23 Word processing	0.00	0.00	0.00	0.69	0.83	1.00	1.21	1.45
24	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
25	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
26	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
27	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
28	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00



# Glue Code Submodel

- **Glue Code - software developed in-house and “hooking” the COTS component into the system or other COTS component**

**Total Glue Code Effort =**

$$A * [(Size)(1+CREVL)^{(B+(0.04*SF))}] * \Pi(EM_i)$$

**Where**

$$i = 1, 2, \dots, 13$$

**A – a multiplicative constant**

**Size – size of glue code in KSLOC**

**CREVL – percentage of glue code reworked due to change in requirements and/or COTS volatility**

**B – a constant accounting for nonlinear economies or diseconomies of scale**

**SF – an architectural scale factor characterizing project conditions having impact on nonlinear economies or diseconomies of scale**

**EM – effort multiplier parameters, each with settings ranging from VL to VH**

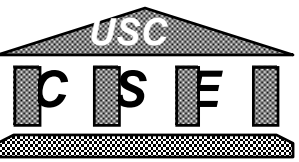


# Glue Code Sub-Model Parameters

- **Exponential Scale Factor**
  - AAREN - Application Architectural Engineering
- **Effort Multipliers - Personnel Drivers**
  - ACIEP - COTS Integrator Experience with Product
  - ACIPC - COTS Integrator Personnel Capability
  - AXCIP – Integrator Experience with COTS Integration Processes
  - APCON - Integrator Personnel Continuity
- **Effort Multipliers - COTS Component Drivers**
  - ACPMT - COTS Product Maturity
  - ACSEW - COTS Supplier Product Extension Willingness
  - APCPX - COTS Product Interface Complexity
  - ACPPS - COTS Supplier Product Support
  - ACPTD - COTS Supplier Provided Training and Documentation
- **Effort Multipliers - Application/System Drivers**
  - ACREL - Constraints on Application System/Subsystem Reliability
  - AACPX - Application Interface Complexity
  - ACPER - Constraints on COTS Technical Performance
  - ASPRT - Application System Portability

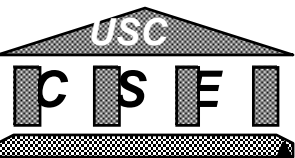
# Glue Code Sub-Model Parameters

Glue Code Parameters					
Nonlinear Scale Factor					
	VL	L	N	H	VH
AAREN	4.00	3.00	2.00	1.00	0.00
Glue Code Parameters					
Aggregate					
Cost Drivers	VL	L	N	H	VH
Personnel Drivers					
ACIEP	1.34	1.16	1.00	0.86	0.75
ACIPC	1.60	1.27	1.00	0.79	0.62
AXCIP		1.12	1.00	0.89	0.79
APCON	1.58	1.26	1.00	0.80	0.63
COTS Component Drivers					
ACPMT	1.45	1.20	1.00	0.83	0.69
ACSEW		1.07	1.00	0.94	0.88
APCPX		0.82	1.00	1.22	1.48
ACPPS		1.14	1.00	0.88	0.77
ACPTD	1.20	1.09	1.00	0.91	0.84
Application/System Drivers					
ACREL		0.88	1.00	1.14	1.30
AACPX		0.84	1.00	1.19	1.42
ACPER			1.00	1.11	1.22
ASPRT			1.00	1.07	1.14



University of Southern California  
Center for Software Engineering

---

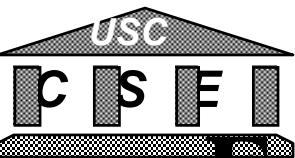


# Application Example - Team08

## CS577'01-02

- **Scale Factor:**

AAREN - Application Architectural Engineering					
	Very Low	Low	Nominal	High	Very High
Unknown	No architecture validation done.	Paper analysis performed.	Peer reviews of architectural design (including interface definitions).	Prototyping/ demos of the architecture performed.	Simulations of the architecture created.
	4.00	3.00	2.00	1.00	0.00
			<b>x</b>		
Rationale for rating:					
Though the architecture design of our project has to be kept very flexible, we have both peer rev					



# Example: Assessment Sub-model Estimation

- Add new COTS classes

COCOTS Assessment Model	
<b>Project Title :</b> <input type="text" value="USC Collaborative Services"/>	<b>Project ID No.</b> <input type="text" value="8"/>
<b>Date Prepared:</b> <input type="text" value="15-Jan-02"/>	<b>Rev No.</b> <input type="text" value="0"/>
<b>Originators</b> <input type="text" value="Ye"/>	
Classes of COTS Product Used	
<input type="checkbox"/> [A00] Aggregate Assessment Calculations	
<input type="checkbox"/> [A01] Generic Component <input type="checkbox"/> [A02] Back office retail <input type="checkbox"/> [A03] Communication protocols/packages <input type="checkbox"/> [A04] Compilers <input type="checkbox"/> [A05] Configuration mgmt/build tools <input type="checkbox"/> [A06] Data conversion packages <input type="checkbox"/> [A07] Databases <input type="checkbox"/> [A08] Device drivers <input type="checkbox"/> [A09] Disk arrays <input type="checkbox"/> [A10] Emulators <input type="checkbox"/> [A11] Engineering tools (req mgmt, design) <input type="checkbox"/> [A12] Graphic information system <input type="checkbox"/> [A13] GUIs/GUI builders <input type="checkbox"/> [A14] Middleware	<input type="checkbox"/> [A15] Network managers <input type="checkbox"/> [A16] Operating systems <input type="checkbox"/> [A17] Problem mgmt <input type="checkbox"/> [A18] Report generators <input type="checkbox"/> [A19] Software process tools <input type="checkbox"/> [A20] Telecommunication & infrastructure <input type="checkbox"/> [A21] Telemetry Analysis <input type="checkbox"/> [A22] Telemetry processing <input type="checkbox"/> [A23] Word processing <input checked="" type="checkbox"/> [A24] collaborative tools <input type="checkbox"/> [A25] <input type="checkbox"/> [A26] <input type="checkbox"/> [A27] <input type="checkbox"/> [A28]



# Example: Results of Assessment Sub-model Estimation

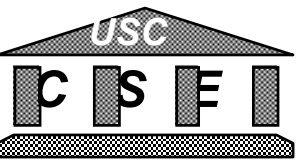
COCOTS			
Project Title :	USC Collaborative Services	Project ID No.	8
Date Prepared:	15-Jan-02	Rev No.	0
Originators	Ye		
CLASS:		<b>collaborative tools</b>	
Initial Filtering Effort by COTS Class			
Total number of COTS candidated filtering effort	2		
Median filtering effort per COTS candidate (person-month)	1.0		
<b>Total Initial Filtering Effort (person-months)</b>	<b>2.0</b>		
Attribute Assessment Effort by COTS Class			
Total number of COTS candidated detailed assessment	3		
Median Detailed Assessment Effort per COTS candidate (person-month)	4.5		
<b>Total Detailed Assessment Effort (person-months)</b>	<b>13.5</b>		
Class Assessment Effort			
<b>Assessment Class Effort =</b>	Initial Filtering Effort	+	Detailed Assessment Effort
=	<b>2.0</b>	+	<b>13.5</b>
=	<b>15.5</b>		<b>person-months</b>





# Model Calibration

- **Currently:**
  - 20 data points + 20 parameters = degree of freedom problem
- **Your contribution is needed:**
  - **Weekly effort report**
    - COTS Initial filtering
    - COTS Final Selection
    - COTS Tailoring
    - Glue Code
  - **Please help with good data!**



# Questions?