

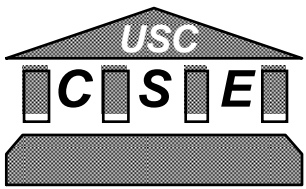
Integrating Industrial and Academic Software Engineering Through Software Project Courses

Barry Boehm, USC

<http://csse.usc.edu>

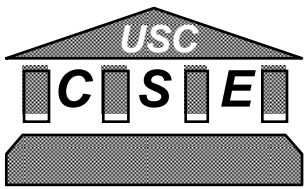
CSUF SwE Workshop Keynote Address

April 25, 2008



Outline

- ➔ • **Academic and Industrial SwE vs. Bloom Taxonomy**
 - Future of industrial software engineering
- **Nature of real-client project course**
 - Primarily USC campus, neighborhood e-services
 - Stakeholder win-win approach
 - MS-level; 2 semester; 6-8 person teams
- **Research/education integration via project experiments**
 - Validate new methods and tools via project usage
 - Partial basis of 11 PhD dissertations
 - Rqts. negotiation, formalization (3), COTS integration (2), Value-based methods (2), Agile methods (1), Quality tradeoffs (1), Risk analysis (1), Cost estimation (1)
- **Conclusions**



Academic SwE vs. Bloom Taxonomy

Bloom Taxonomy

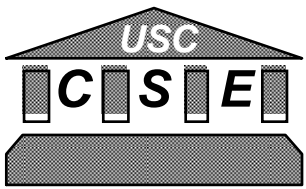
- **Create**
- **Evaluate**
- **Analyze**
- **Apply: Real World**

- **Apply: Set Piece**
- **Understand**
- **Remember**

Software Engineering

- **Industrial software engineering**
- **Real-client project courses**

- **Much of academic software engineering**



Software Engineering Trends

Traditional

- Standalone systems
- Mostly source code
- Requirements-driven
- Control over evolution
- Focus on software
- Stable requirements
- Premium on cost

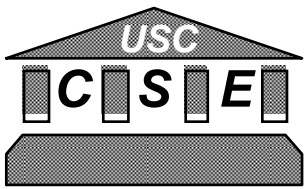
Future

- Everything connected-maybe
- Mostly COTS components
- Requirements are emergent
- No control over COTS evolution
- Focus on systems and software
- Rapid change
- Premium on value, speed, quality



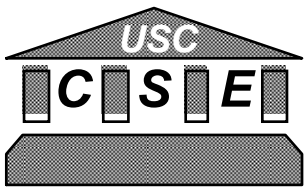
Architected Agile Approach

- **Management commitment, with incremental feasibility checkpoints**
 - Clear message about objectives, scope, and strategy
 - Involve top people from stakeholder organizations
 - Build in growth to expansion sites
 - Lead through early successes
- **Thoroughly prepare the ground**
 - Architecture, infrastructure, policies, practices, roles, training
 - Customer buy-in and expectations management
 - Get help from experts
- **Make clear what's essential, optional**
 - Most frequently, Scrum plus organizational essentials
 - Precede Development Sprints by Architecting Sprint
 - Follow by Release Sprint, beta testing
 - Where needed, work compliant mandate interpretations
- **Monitor, reflect, learn, evolve**



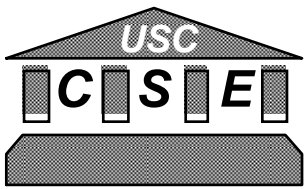
Medical Case Study -- USA

- **1400 software people; 7M SLOC; 7 sites**
 - 4 in Europe, 2 in India
- **500 medical applications; 500 financial; others**
- **Survivability-critical software problems**
 - Reliability, productivity, performance, interoperability
 - Sarbanes-Oxley requirements
 - Management receptive to radical change
- **Some limited experimental use of agile methods**
 - Led by top software technologist/manager
- **Committed to total change around Scrum and XP**



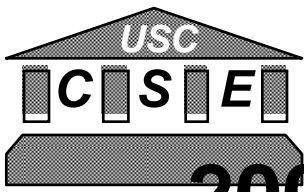
Key Practices – USA Medical

- **Include customers and marketers**
 - New roles; do's/don'ts/opportunities; CRACK personnel; full collaboration and teamwork; expectations management
- **Scrum; most XP practices; added company practices**
 - 6-12 person teams with team rooms, dedicated servers
 - Hourly smoke test; nightly build and regression test
 - Just-in-time analysis; story-point estimates; fail fast; detailed short-term plans; company architecture compliance
 - Embrace change in applications and practices
 - Global teams: wikis, daily virtual meetings, act as if next-door
- **Release management**
 - 2-12 week architecting Sprint Zero; 3-10 1-month Sprints; Release Sprint; 1-6 month beta test
 - Next Sprint Zero concurrent with Release Sprint
- **Initiative manager and team**
 - Define practices; evolve infrastructure; provide training; guide implementation; evaluate compliance/usage; continuous improvement



Outline

- **Academic and Industrial SwE vs. Bloom Taxonomy**
 - Future of industrial software engineering
- ➔• **Nature of real-client project course**
 - Primarily USC campus, neighborhood e-services
 - Stakeholder win-win approach
 - MS-level; 2 semester; 6-8 person teams
- **Research/education integration via project experiments**
 - Validate new methods and tools via project usage
 - Partial basis of 11 PhD dissertations
 - Rqts. negotiation, formalization (3), COTS integration (2), Value-based methods (2), Agile methods (1), Quality tradeoffs (1), Risk analysis (1), Cost estimation (1)
- **Conclusions**



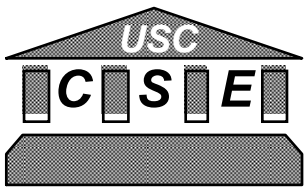
2007-08 Software Engineering Projects

Box Office Database System	Jay McAdams	24th Street Theatre
Online Peer Review System	Stephen Bucher	USC, Engineering Writing Program
Thai CDC Software Tool	Pheel Wang	Thai Community Development Center
USC COINCOMO	Ed Colbert, Vu Nguyen	USC CSSE
Crystal Stairs Dashboard	Rick Capella	Crystal Stairs, Inc.
Family & Homeless Well-Being Project	Jill Remelski	St. Francis Center (SFC)
BTI Appraisal Projects	Megan Tunnell O'Rourke	BTI Appraisal
LAMAS Customer Service Application	Celia Castellanos	Los Angeles Music and Art School
Web-based service for TPC Foundation	Pat Means	Turning Point URBAN Magazine
REEO Database	Scott Stimpfel	Resources for Educational and Employment Opportunities
BID review System	Keith Culling	Background Investigation Division, LA city
THSA Website Project	Poonchana Thitametakul	USC THSA
EEO Section Database	Art Irigoyen	LA city
Conference Room Reservation System	Linda Lanier	LA city
Applicant-processing systems for CLAPD	Dee Dee Coultas	LA city
E-Mentoring program	Carlene Davis	Los Angeles Urban League
Social Networking Tools for Librarians	Julie Kwan	Social Networking Tools for Librarians
EZSIM II	Ray Madachy	USC-CSSE
Los Angeles County Generation Web Initiative	Frank Cheng, Jeramy Gray	LA County
Development Of Hunter-gatherer Database	Chris Boehm	USC



Stakeholder Win-Win Approach

Stakeholders	Win Conditions
<ul style="list-style-type: none">• Students, Employers	<ul style="list-style-type: none">• Full range of SW Engr. skills• Real-client project experience• Non-outsourcable skills• Advanced SW tech. experience
<ul style="list-style-type: none">• Project clients	<ul style="list-style-type: none">• Useful applications• Advanced SW tech. understanding• Moderate time requirements
<ul style="list-style-type: none">• Faculty, PhD Students	<ul style="list-style-type: none">• Educate future SW Engr. leaders• Research results• Applied on real-client projects

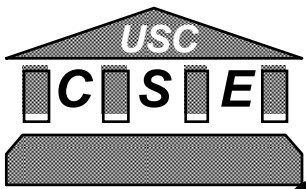


CS 577 Learning Objectives

“ Software Engineering:” The disciplines which distinguish the coding of a computer program from the development of a software product.

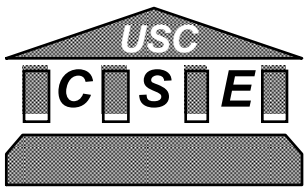
Issues \ Stages	Requirements, Architecture	Design, Code	Test, Implement, Maintain
Computer Science		CS Focus	
User Applications			
Economics			
People			

- **Prepare you for software leadership careers through the 2040's**
 - Agility , discipline, COTS/OSS, service-based systems, new processes, leading-edge research and commercial methods and tools
- **Integrate all these considerations**
 - Via Incremental Commitment Model



e-Services Projects Overview

- **Clients identify prospective projects**
 - Operational capabilities or feasibility explorations
 - Fall: 12 weeks to prototype, analyze, design, plan, validate
 - Spring: 12 weeks to develop, test, transition
 - MS-level, 6-8 person, CS 577 project course (2 off-campus)
- **Clients, CSSE negotiate workable projects**
 - Useful results within time constraints
 - Operationally supportable as appropriate
- **Clients work with teams to define, steer, evaluate projects**
 - Exercise prototypes, negotiate requirements, review progress
 - Mutual learning most critical success factor



Architected Agile Approach

- **Fall: Develop Life Cycle Architecture Packages**
 - Ops. Concept, Requirements, Prototype, Architecture, Plan
 - Feasibility Evidence, including business case
 - Results chain linking project results to desired outcomes
 - 20 projects; 120 students; about 20 clients
- **Spring: Agile Initial Operational Capability**
 - 6-10 projects; 30-50 students; 6-10 clients
 - Prioritized features; timeboxed increments
 - 2-week transition period
 - Then the student teams disappear
- **Tools and techniques: WikiWinWin; Results Chain; Rational RSA; Subversion; USC COCOMO II; MS Project; USC Incremental Commitment Model**
 - Reworked annually based on student & client feedback


Digital Manuscript Archive Home Page - Netscape

File Edit View Go Communicator Help

Back Forward Reload Home Search Guide Print Security Stop

Bookmarks Location: http://sunset.usc.edu:80/cs577_96/dma/home/

Internet New and Cool Look Up



Digital Manuscript Archive Manuscript Index Page - Netscape

File Edit View Go Communicator Help

Back Forward Reload Home Search Guide Print Security Stop

Bookmarks Location: http://sunset.usc.edu:80/cs577_96/dma/scripts/detail

Internet New and Cool Look Up

Antiphonarium

Title	Antiphonarium
Author	Catholic Church
Date	15th Century
Type	Liturgical & Ritual
Style	Not Available
Physical Characteristics	On vellum, red staves with black Gregorian capitals and rubrication. Dimension of leaves is 57cm x 41cm.

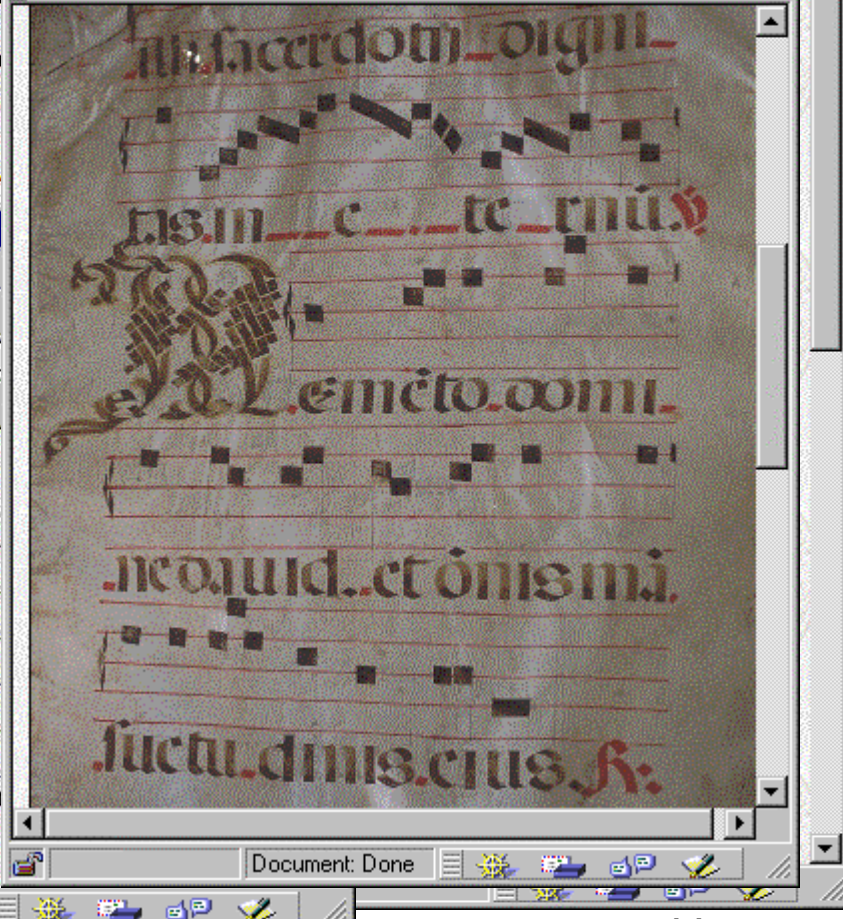
Digital Manuscript Archive Display Leaf Image Page - ...

File Edit View Go Communicator Help

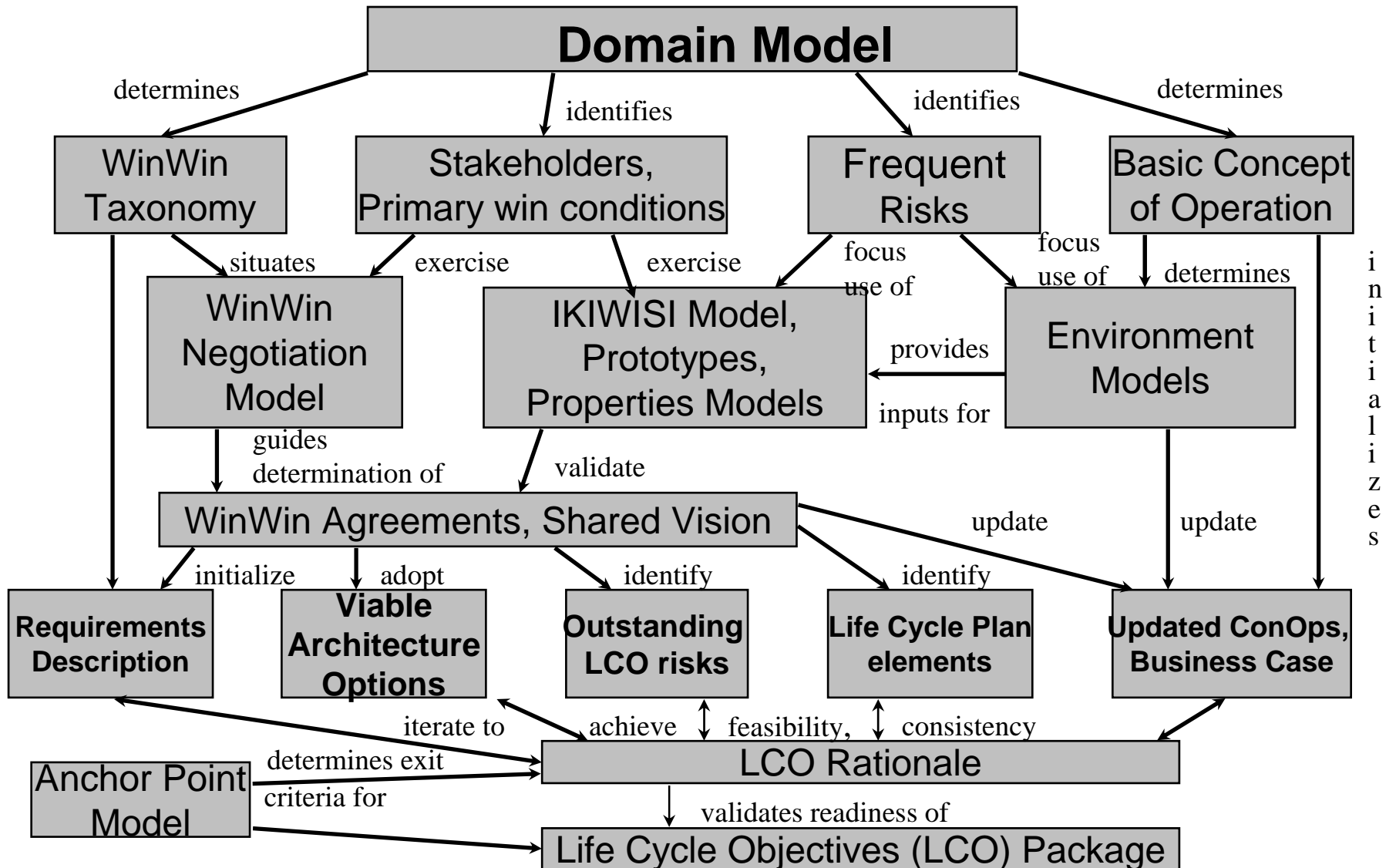
Back Forward Reload Home Search Guide Print Security Stop

Bookmarks Location: [tiphonarium&pageno=10&m=1&p=0](#)

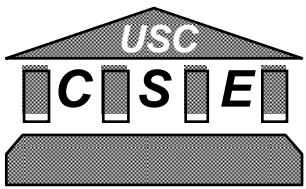
Internet New and Cool Look Up



MBASE Model Integration: LCO Stage

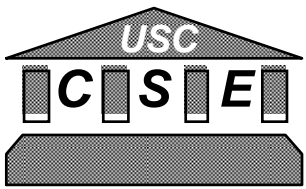


initializes



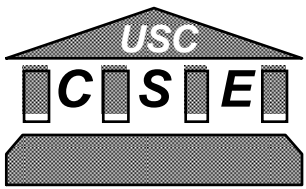
S&C Subdomain (General)

Type of Application	Simple Block Diagram	Examples (project nos.)	Developer Simplifiers	Developer Complicators
<p>Multimedia Archive</p>	<pre> graph TD QueryIn[query] --> Catalog Catalog -- "MM asset info" --> Out1[] Catalog -- query --> MMArchive[MM Archive] MMArchive -- "update notification" --> Catalog UpdateIn[update] --> MMArchive MMArchive -- "MM asset" --> Out2[] </pre>	<p>1, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 20, 31, 32, 35, 36, 37, 39</p>	<ul style="list-style-type: none"> • Use standard query languages • Use standard or COTS search engine • Uniform media formats 	<ul style="list-style-type: none"> • Natural language processing • Automated cataloging or indexing • Digitizing large archives • Digitizing complex or fragile artifacts • Rapid access to large Archives • Access to heterogeneous media collections • Automated annotation/description/ or meanings to digital assets • Integration of legacy systems



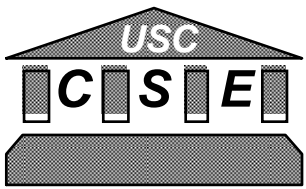
The Results

- **Projects That Failed LCO Criteria**
 - **1996: 4 out of 16 (25%)**
 - **1997: 4 out of 15 (27%)**
 - **1998: 1 out of 19 (5%)**
 - **1999: 1 out of 22 (5%)**
 - **2000: 2 out of 20 (10%)**
- **Post-1997 failures due to non-S&C causes**
 - **Team cohesion, client outages**



Outline

- **Academic and Industrial SwE vs. Bloom Taxonomy**
 - Future of industrial software engineering
- **Nature of real-client project course**
 - Primarily USC campus, neighborhood e-services
 - Stakeholder win-win approach
 - MS-level; 2 semester; 6-8 person teams
- ➔ • **Research/education integration via project experiments**
 - Validate new methods and tools via project usage
 - Partial basis of 11 PhD dissertations
 - Rqts. negotiation, formalization (3), COTS integration (2), Value-based methods (2), Agile methods (1), Quality tradeoffs (1), Risk analysis (1), Cost estimation (1)
- **Conclusions**

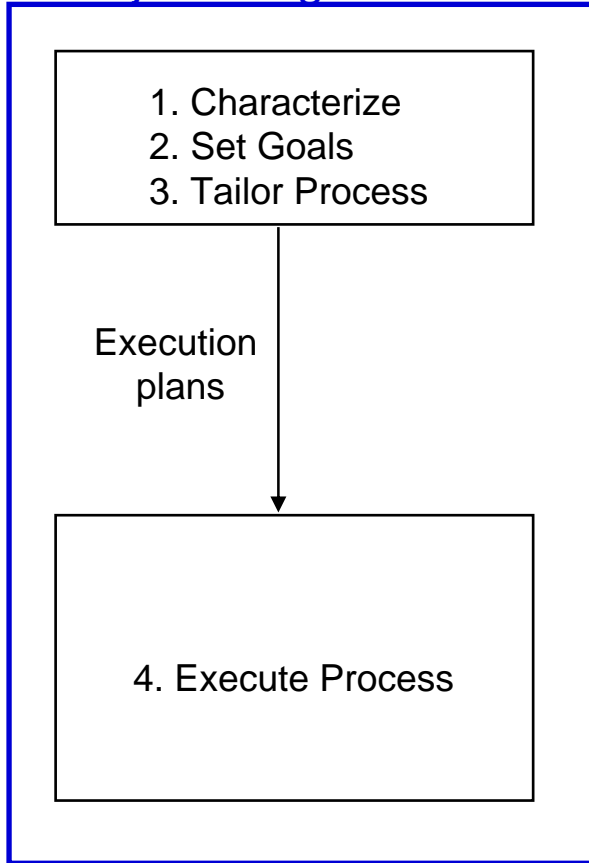


Empirical Software Engineering Research

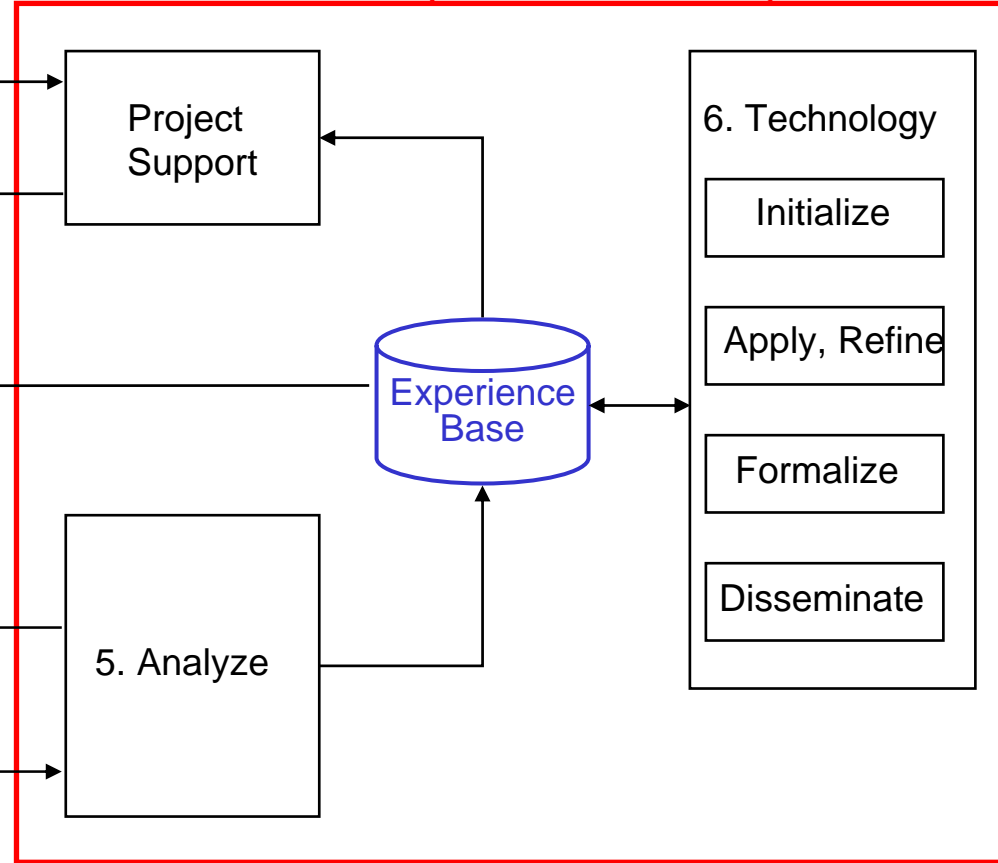
- **Empirical software engineering research generally slow**
 - Projects take 2-5 years to complete
 - Improvements confounded with other factors
 - Data generally sparse, hard to compare across projects
- **Team projects the ESE equivalent of the fruit fly**
 - 20 per year, real clients, using industrial-grade processes
 - Teams include 2 off-campus working professionals
 - 1-2 of 6 on-campus students have 1-2 years work experience
 - Extensive data, consistently collected
 - Opportunities to run (partially) controlled experiments
 - Projects, teams not identical

Project Course Experience Factory

Projects Organization



Experience Factory



WikiWinWin – Tool (3)

Brainstorm Win Conditions ([ComputerSoftwareWinConditions](#))

The system is developed on Windows Server 2003, and [Content Management System\(CMS\)](#) is implemented along with the necessary extensions.

The database used for the system would be [MySQL?](#)

[Joomla](#) is an ideal open source CMS. It offers various plug-ins that cater to the requirements of the project

Joomla CMS provides a calendar event feature plug-in which is not available in Drupal CMS. There are various other features like these which Joomla supports and Drupal does not. Hence, [Joomla](#) would be our best CMS choice.

Initial ideas surfaced at the meeting



Converge on Win Conditions (Win Condition List)

State	Name	Timestamp	Creator	Role
Not Agreed	Content Management Specification	05 Oct 2007 06:20	KalyaniSoniminde?	Shaper

Shaper organize them into a prospective win condition after the meeting



Description: Joomla which is a open source content management system shall be used

Comment on Current Win Condition:

After research and analysis of various CMS, joomla is the best fit [NiravChheda?](#) 07 Oct 2007 - 18:34

Add comment

Stakeholders engage in a further discussion

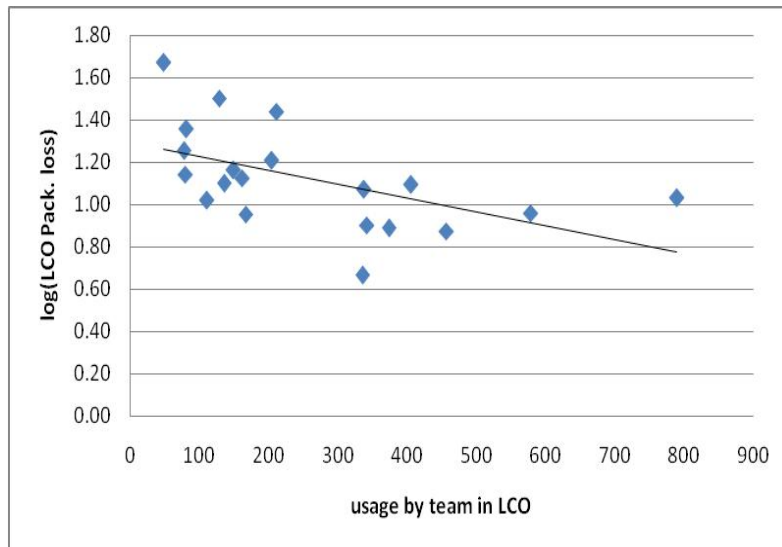
Identify Issues

State	Name	Timestamp	Creator	Role	Options
Resolved	Inflexible if requirements change	07 Oct 2007 18:35	KalyaniSoniminde?	Shaper	enter

Shaper facilitate negotiation in WikiWinWin

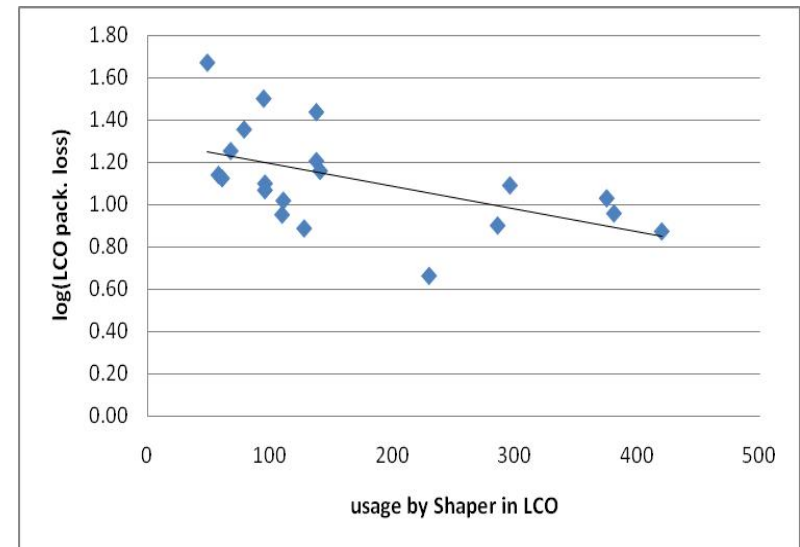
WikiWinWin – Current Progress (2)

- **Initial Results (fall 07)**
 - Correlation between usage aspect and outcome
 - Not all positive feedbacks

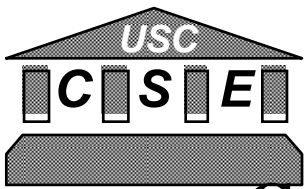


LCO package quality shortfall vs. usage by team

March 2008

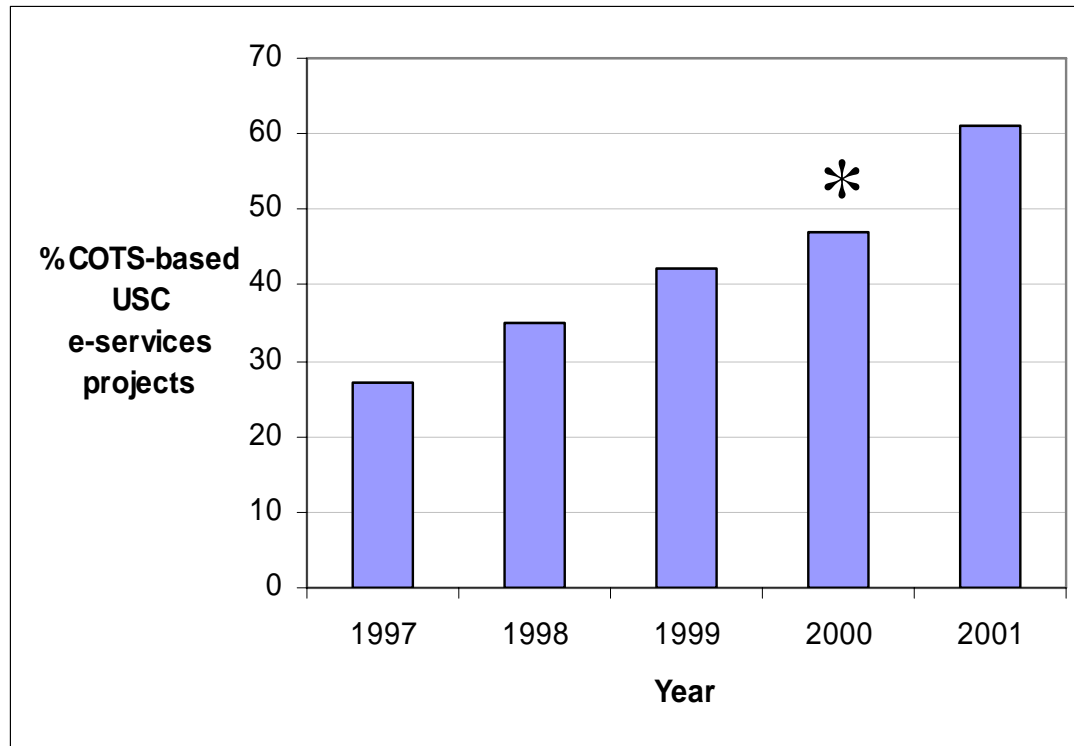


LCO package quality shortfall vs. usage by shaper



Growth of COTS-Based USC e-Services Projects

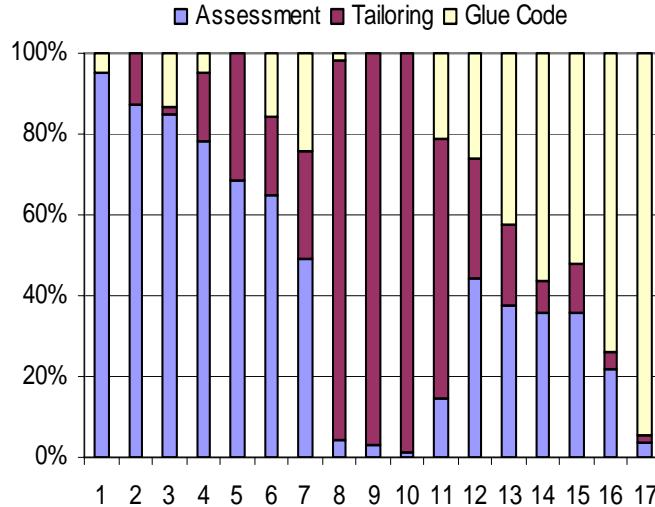
- Requires new processes, architectures, methods
- It's not “all about programming” anymore
- Similar trends at Motorola, other USC-CSE Affiliates



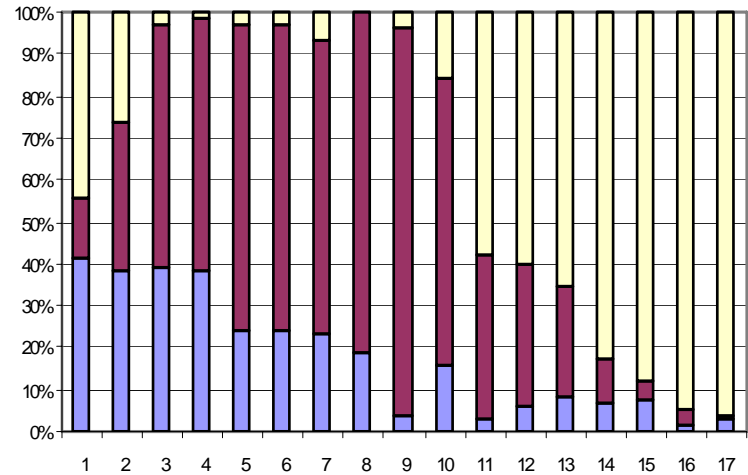
* Industry: 2001 Standish Report

Axiom 1. Process Happens Where the Effort Happens

- **COTS Assessment, Tailoring, Glue Code**

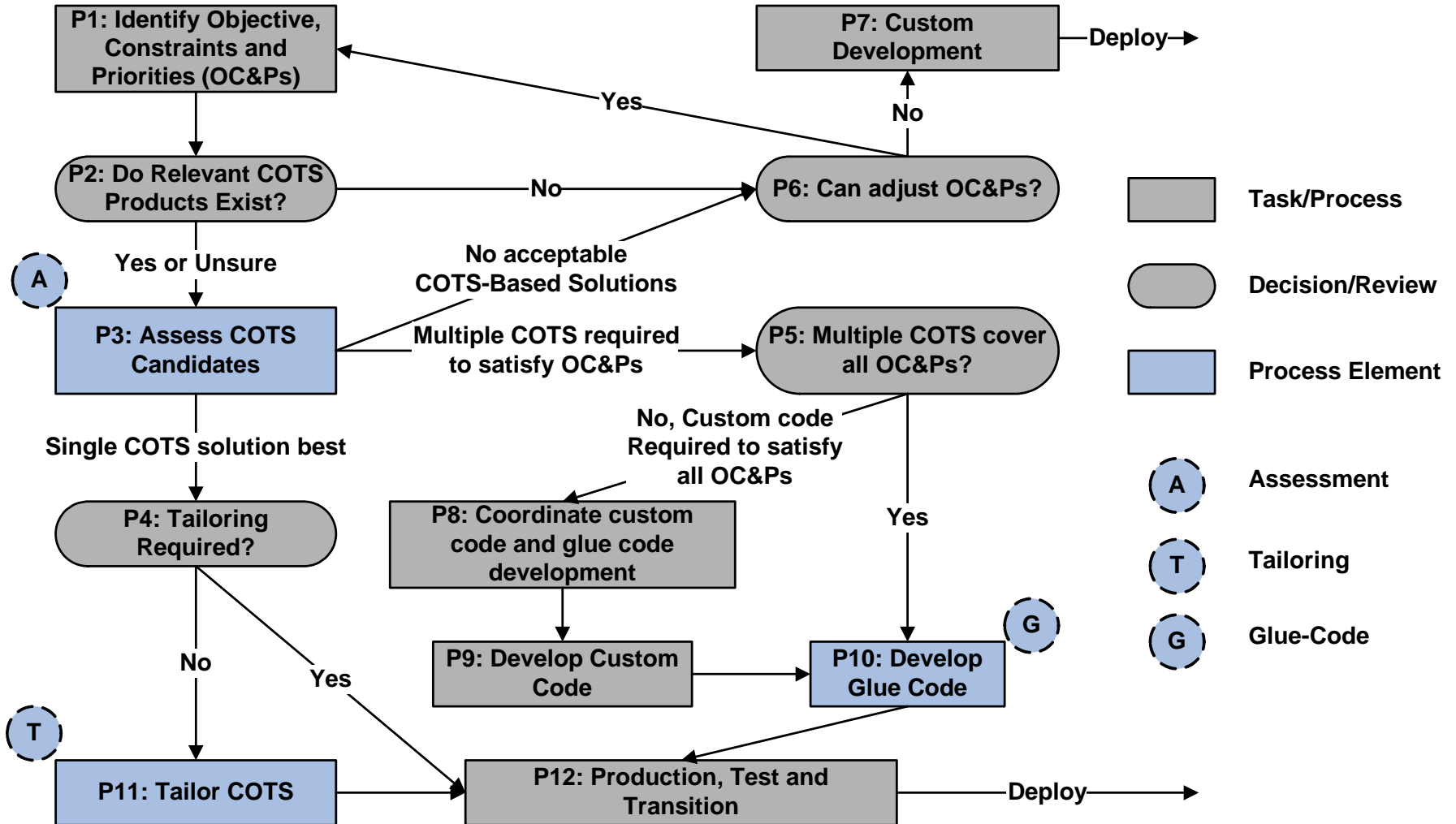


a. CBA Effort Distribution of USC E-Service Projects

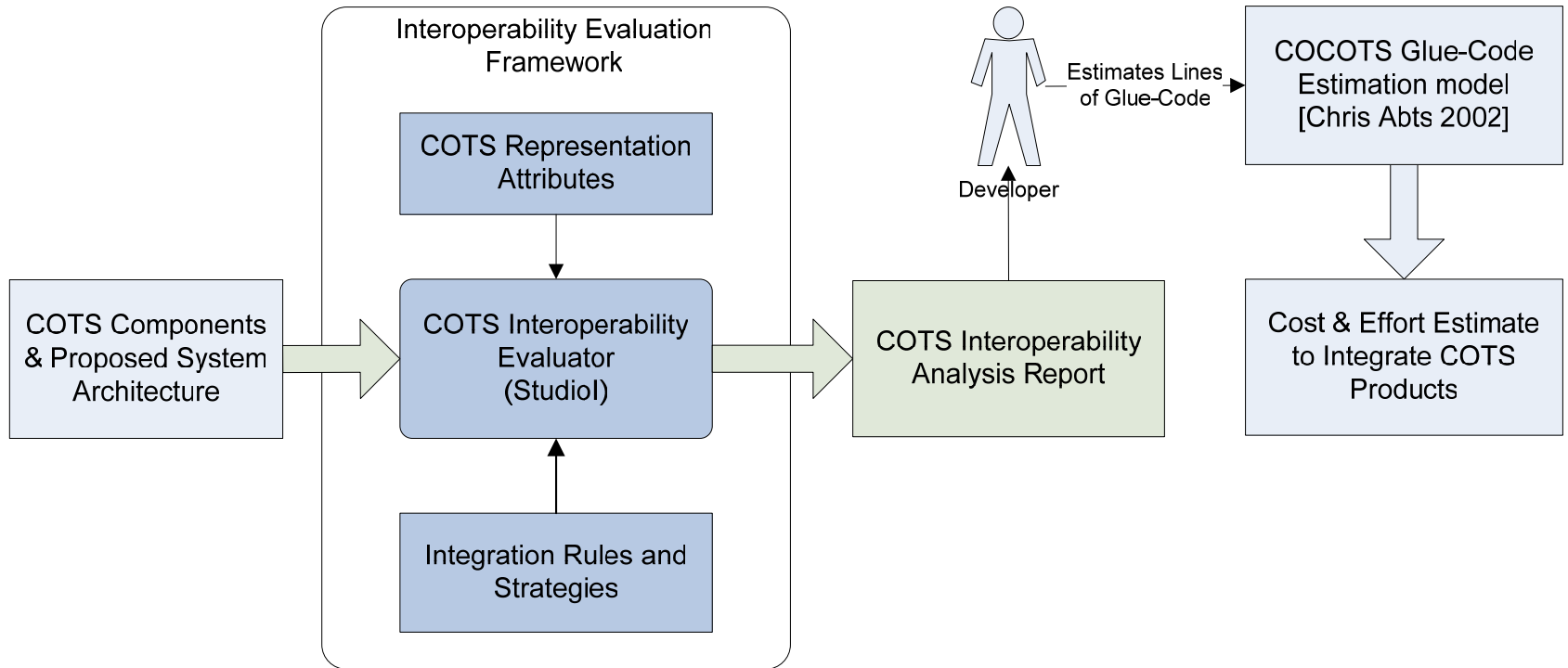


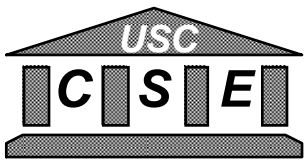
b. CBA Effort Distribution of Industry COCOTS Calibration Data

CBA Spiral Framework



Interoperability Evaluation Framework Interfaces





Experiment 1 Results

Data Set	Groups	Mean	Standard Deviation	P-Value
Dependency Accuracy	Pre Framework Application	79.3%	17.9	0.017
	Post Framework Application	100%	0	
Interface Accuracy	Pre Framework Application	76.9%	14.4	0.0029
	Post Framework Application	100%	0	
Actual Assessment Effort	Projects using this framework	1.53	1.71	0.053
	Equivalent projects that did not use this framework	5 hrs	3.46	
Actual Integration Effort	Projects using this framework	9.5 hrs	2.17	0.0003
	Equivalent projects that did not use this framework	18.2 hrs	3.37	

* **Accuracy of Dependency Assessment:**

1 – (number of unidentified dependencies/total number of dependencies)

** **Accuracy of Interface Assessment:**

1 – (number of interface interaction mismatches identified/total number of interface interactions)

Accuracy: a quantitative measure of the magnitude of error [IEEE 1990]

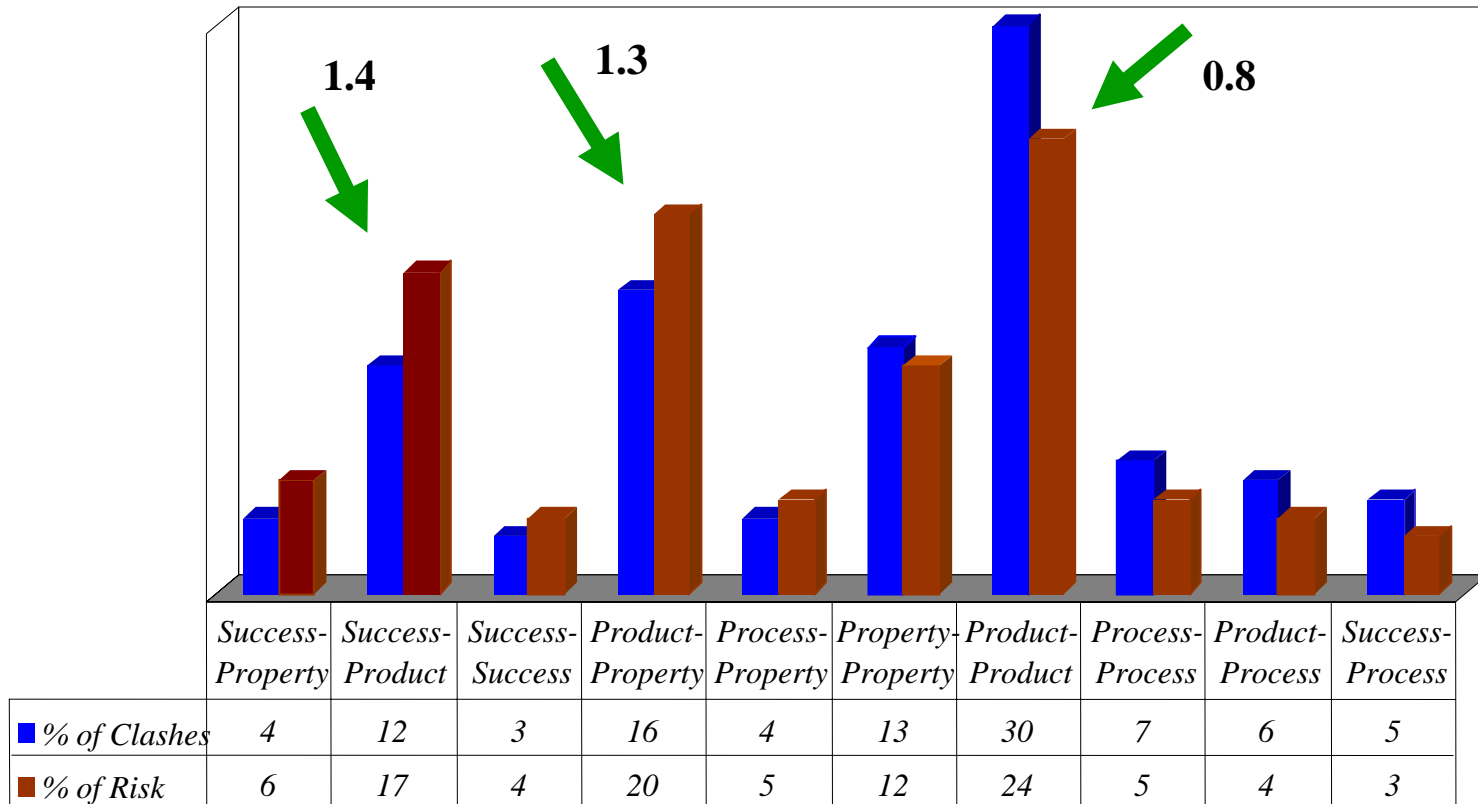


Model Clashes Occurrence Probability & Severity: 35 Projects RAD Sample

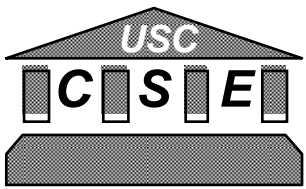
Assumptions	S	M	OP	SV
Developer needs extensive, on demand user/customer interaction	Dev	SS	0.86	H
User/Customer are available at limited times	User/ Cust	SS		
User/Customer are free to add/modify the system requirements during the system implementation	User/ Cust	PD	0.46	H
Changes/additions to system requirements require extra budget and schedule	Dev	PP		

S: Stakeholder; M: Model; OP: Occurrence Probability; SV: Severity

Clash Types and their Contribution to Project Risk

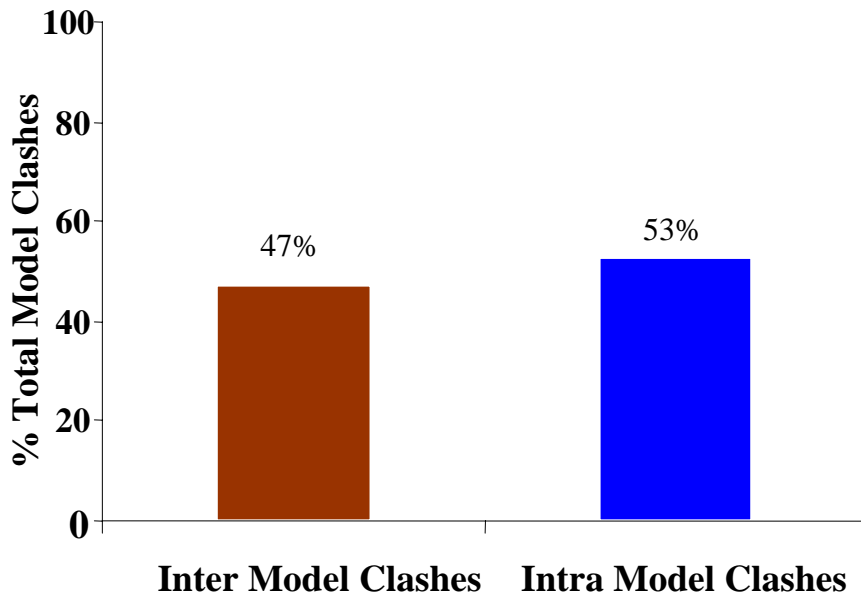


- Majority of research (product-product) addresses minority of risk

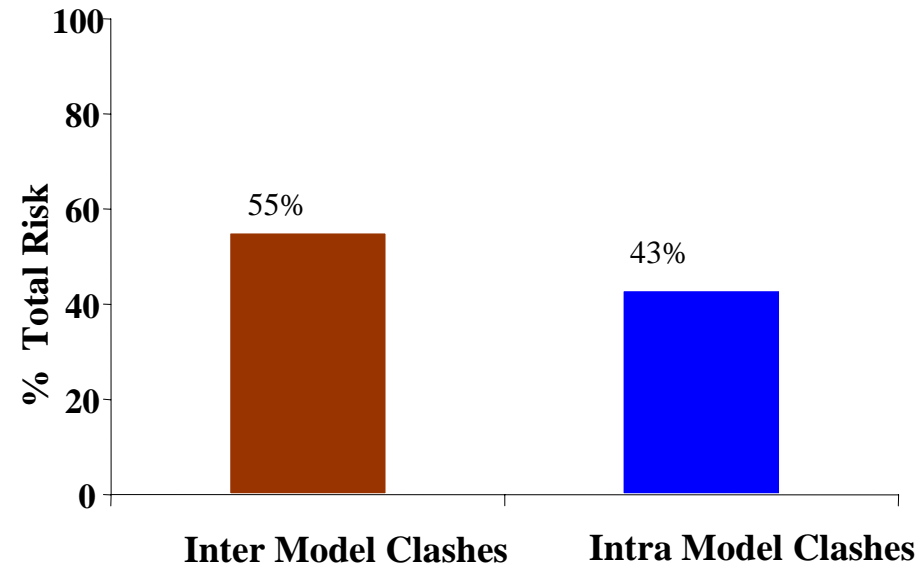


Inter and Intra Model Clashes and their Contribution to Project Risk

Distribution of Inter and Intra Model Clashes



Contribution of Inter and Intra Model Clashes to Risk



- Inter model clashes caused majority of risk

Model Clash Analysis Tool

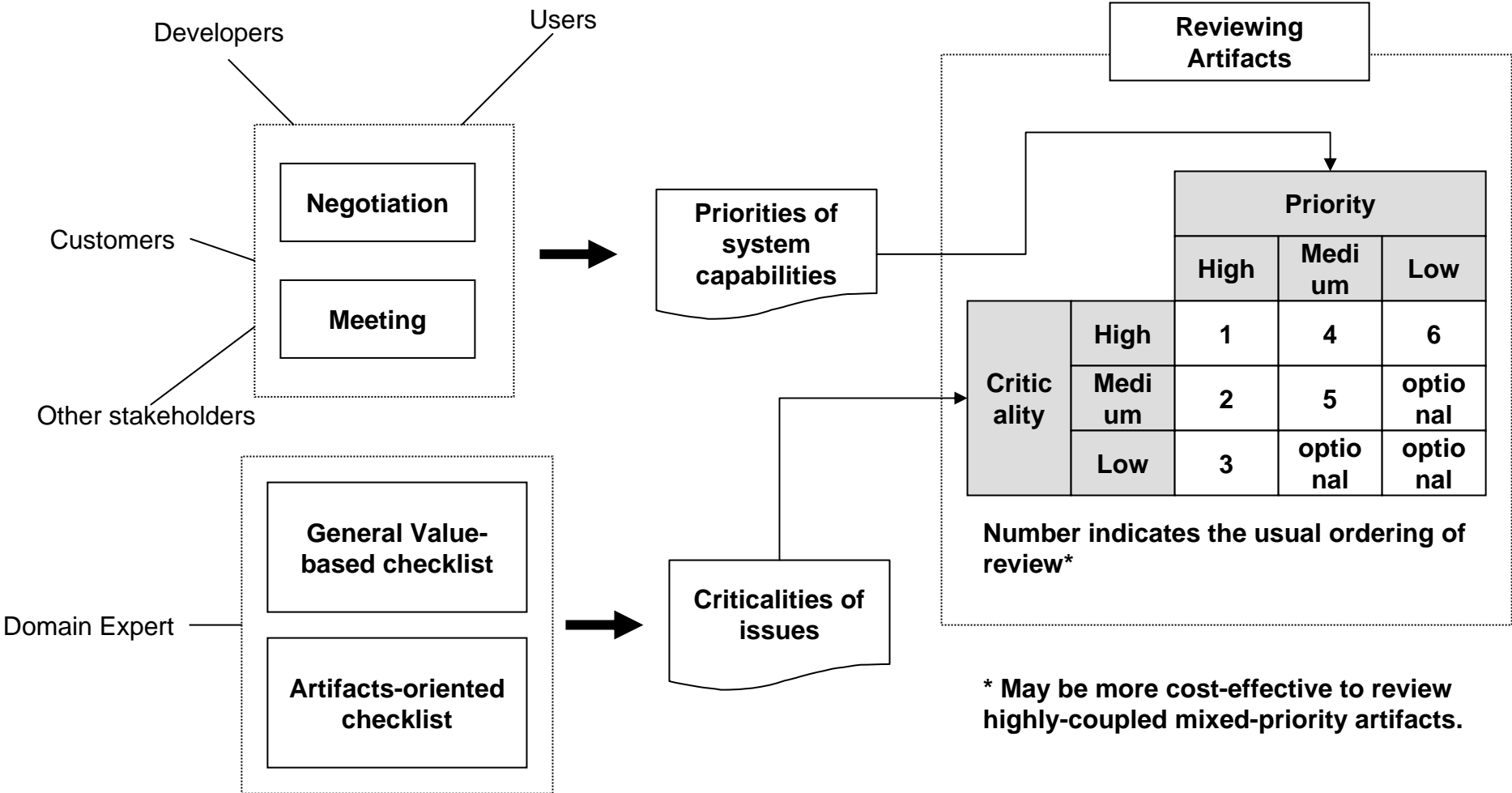
- Home
- Clash Detector
- Model Determination
- Model Comparison
- Model Management
- Contact Information
- Help

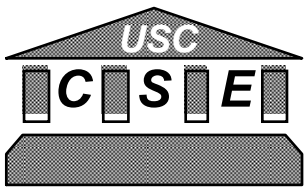
Add Question: Type Question and its properties

Model Comparison Results

Waterfall Assumption	COTS Assumption	Risk	Mitigation
Developer defines and documents a complete set of requirements for each system capability before starting the system detail design	System requirements and architectures are often driven by the capabilities and structures of the final set of COTS components selected for integration	Defining the complete requirements first and then looking for COTS product(s) to satisfy them has many problems. These problems are caused by the inflexibility of COTS. As an example, available COTS product (s) may not satisfy a requirement that demands a specific performance	I. Before starting the system design, prioritize the system requirements and then assess viable COTS products that satisfy these requirements II. All key stakeholders should be involved in assessing potential COTS III. Finalize the system requirements specifications after knowing COTS limitations

Value-Based Review Process (II)





Value-Based Checklist (I) <General Value-Based Checklist>

	High-Criticality Issues	Medium-Criticality Issues	Low-Criticality Issues
Completeness	<ul style="list-style-type: none"> •Critical missing elements: backup/ recovery, external interfaces, success-critical stakeholders; critical exception handling, missing priorities •Critical missing processes and tools; planning and preparation for major downstream tasks (development, integration, test, transition) •Critical missing project assumptions (client responsiveness, COTS adequacy, needed resources) 	<ul style="list-style-type: none"> •Medium-criticality missing elements, processes and tools: maintenance and diagnostic support; user help •Medium-criticality exceptions and off-nominal conditions; smaller tasks (review, client demos), missing desired growth capabilities, workload characterization 	<ul style="list-style-type: none"> •Easily-deferrable, low-impact missing elements: straightforward error messages, help messages, GUI details doable via GUI builder, project task sequence details
Consistency/ Feasibility	<ul style="list-style-type: none"> •Critical elements in OCD, SSRD, SSAD, LCP not traceable to each other •Critical inter-artifact inconsistencies: priorities, assumptions, input/output, preconditions/post-conditions •Missing evidence of critical consistency/feasibility assurance in FRD 	<ul style="list-style-type: none"> •Medium-criticality shortfalls in traceability, inter-artifact inconsistencies, evidence of consistency/feasibility in FRD 	<ul style="list-style-type: none"> •Easily-deferrable, low-impact inconsistencies or inexplicit traceability: GUI details, report details, error messages, help messages, grammatical errors
Ambiguity	<ul style="list-style-type: none"> •Vaguely defined critical dependability capabilities: fault tolerance, graceful degradation, interoperability, safety, security, survivability •Critical misleading ambiguities: stakeholder intent, acceptance criteria, critical user decision support, terminology 	<ul style="list-style-type: none"> •Vaguely defined medium-criticality capabilities, test criteria •Medium-criticality misleading ambiguities 	<ul style="list-style-type: none"> •Non-misleading, easily deferrable, low-impact ambiguities: GUI details, report details, error messages, help messages, grammatical errors
Conformance	<ul style="list-style-type: none"> •Lack of conformance with critical operational standards, external interfaces 	<ul style="list-style-type: none"> •Lack of conformance with medium-criticality operational standards, external interfaces •Misleading lack of conformance with document formatting standards, method and tool conventions 	<ul style="list-style-type: none"> •Non-misleading lack of conformance with document formatting standards, method and tool conventions, optional or low-impact operational standards
Risk	<ul style="list-style-type: none"> •Missing FRD evidence of critical capability feasibility: high-priority features, levels of service, budgets and schedules •Critical risks in top-10 risk checklist: personnel, budgets and schedules, requirements, COTS, architecture, technology 	<ul style="list-style-type: none"> •Missing FRD evidence of mitigation strategies for low-probability high-impact or high-probability, low-impact risks: unlikely disasters, off-line service delays, missing but easily-available information 	<ul style="list-style-type: none"> •Missing FRD evidence of mitigation strategies for low-probability, low-impact risks

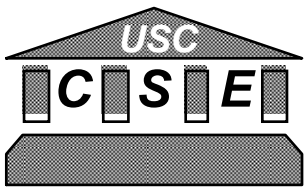
Value-Based Reading (VBR) Experiment

— Keun Lee, ISESE 2005

By Number	P-value	% Gr A higher	By Impact	P-value	% Gr A higher
Average of Concerns	0.202	34	Average Impact of Concerns	0.049	65
Average of Problems	0.056	51	Average Impact of Problems	0.012	89
Average of Concerns per hour	0.026	55	Average Cost Effectiveness of Concerns	0.004	105
Average of Problems per hour	0.023	61	Average Cost Effectiveness of Problems	0.007	108

- **Group A: 15 IV&V personnel using VBR procedures and checklists**
- **Group B 13 IV&V personnel using previous value-neutral checklists**
 - **Significantly higher numbers of trivial typo and grammar faults**

Experiment



Integrating Industrial and Academic SwE

- **Real-client team projects closer to industry practice**
 - Focused on projected future industry practice
 - **Architected agile approach; COTS integration**
 - Exercise full range of Bloom taxonomy
- **Projects provide the research equivalent of the fruit fly**
 - 20 per year, real clients, using industrial-grade processes
 - Extensive data, consistently collected
 - Opportunities to run (partially) controlled experiments
 - **Projects, teams not identical**
 - Results frequently correlate with industry experience
- **Results strengthen future educational experiences**