



# **Developing and Integrating Software/System Product, Process, Property, and Success Models**

**Barry Boehm, USC-CSE**  
**IBM T. J. Watson Presentation**  
**August 13, 1999**

**Boehm@sunset.usc.edu**  
**<http://sunset.usc.edu>**



# Outline

- **Developing Software/System Models at USC-CSE**
  - Process Models
  - Product Models
  - Property Models
  - Success Models
- **Integrating Software/System Models**
- **Conclusions**
- **Further information**



# Process Model Research at USC-CSE

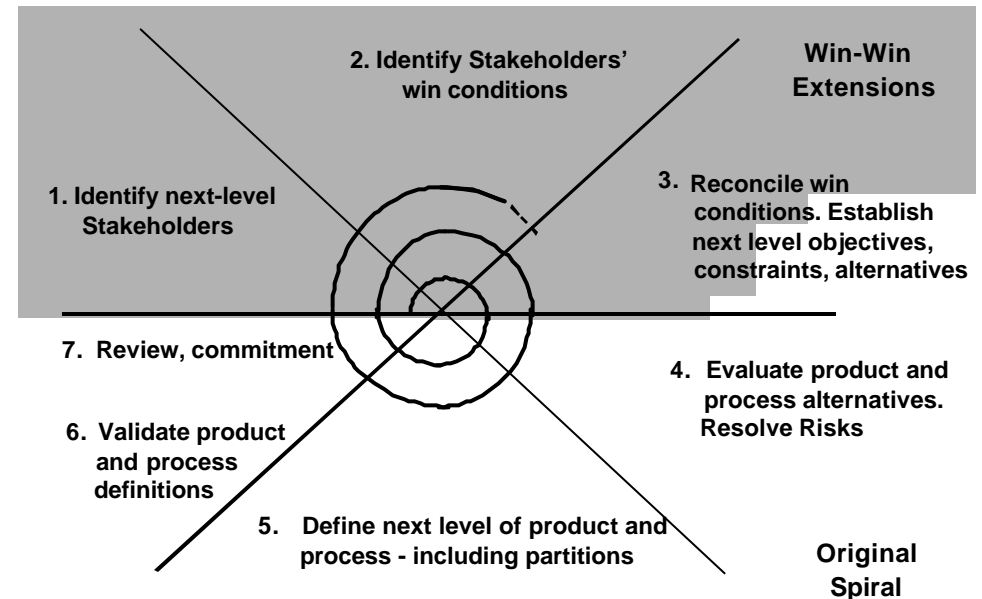
- **Spiral Model Extensions**
  - WinWin spiral model
  - Life cycle anchor points
- **WinWin Negotiation Model**
  - IPT, JAD guidelines
  - Groupware support tools
- **Object-oriented processes with COTS, reuse**
- **RAD-SAIV process model**
- **System dynamics models**



# Spiral Model Experience

- Where do objectives, constraints, alternatives come from?
  - Win Win extensions
- Lack of intermediate milestones
  - Anchor Points: LCO, LCA, IOC
  - Concurrent-engineering spirals between anchor points

## The WinWin Spiral Model





# Life Cycle Anchor Points

- **Common System/Software stakeholder commitment points**
  - Defined in concert with Government, industry affiliates
  - Coordinated with Rational's Unified Software Development Process
- **Life Cycle Objectives (LCO)**
  - Stakeholders' commitment to support system architecting
  - Like getting engaged
- **Life Cycle Architecture (LCA)**
  - Stakeholders' commitment to support full life cycle
  - Like getting married
- **Initial Operational Capability (IOC)**
  - Stakeholders' commitment to support operations
  - Like having your first child



# Win Win Spiral Anchor Points

(Risk-driven level of detail for each element)

Milestone Element	Life Cycle Objectives (LCO)	Life Cycle Architecture (LCA)
<b>Definition of Operational Concept</b>	<ul style="list-style-type: none"> <li>• Top-level system objectives and scope               <ul style="list-style-type: none"> <li>- System boundary</li> <li>- Environment parameters and assumptions</li> <li>- Evolution parameters</li> </ul> </li> <li>• Operational concept               <ul style="list-style-type: none"> <li>- Operations and maintenance scenarios and parameters</li> <li>- Organizational life-cycle responsibilities (stakeholders)</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• Elaboration of system objectives and scope of increment</li> <li>• Elaboration of operational concept by increment</li> </ul>
<b>System Prototype(s)</b>	<ul style="list-style-type: none"> <li>• Exercise key usage scenarios</li> <li>• Resolve critical risks</li> </ul>	<ul style="list-style-type: none"> <li>• Exercise range of usage scenarios</li> <li>• Resolve major outstanding risks</li> </ul>
<b>Definition of System Requirements</b>	<ul style="list-style-type: none"> <li>• Top-level functions, interfaces, quality attribute levels, including:               <ul style="list-style-type: none"> <li>- Growth vectors and priorities</li> <li>- Prototypes</li> </ul> </li> <li>• Stakeholders' concurrence on essentials</li> </ul>	<ul style="list-style-type: none"> <li>• Elaboration of functions, interfaces, quality attributes, and prototypes by increment               <ul style="list-style-type: none"> <li>- Identification of TBD's (to-be-determined items)</li> </ul> </li> <li>• Stakeholders' concurrence on their priority concerns</li> </ul>
<b>Definition of System and Software Architecture</b>	<ul style="list-style-type: none"> <li>• Top-level definition of at least one feasible architecture               <ul style="list-style-type: none"> <li>- Physical and logical elements and relationships</li> <li>- Choices of COTS and reusable software elements</li> </ul> </li> <li>• Identification of infeasible architecture options</li> </ul>	<ul style="list-style-type: none"> <li>• Choice of architecture and elaboration by increment               <ul style="list-style-type: none"> <li>- Physical and logical components, connectors, configurations, constraints</li> <li>- COTS, reuse choices</li> <li>- Domain-architecture and architectural style choices</li> </ul> </li> <li>• Architecture evolution parameters</li> </ul>
<b>Definition of Life-Cycle Plan</b>	<ul style="list-style-type: none"> <li>• Identification of life-cycle stakeholders               <ul style="list-style-type: none"> <li>- Users, customers, developers, maintainers, interoperators, general public, others</li> </ul> </li> <li>• Identification of life-cycle process model               <ul style="list-style-type: none"> <li>- Top-level stages, increments</li> </ul> </li> <li>• Top-level WWWWWHH* by stage</li> </ul>	<ul style="list-style-type: none"> <li>• Elaboration of WWWWWHH* for Initial Operational Capability (IOC)               <ul style="list-style-type: none"> <li>- Partial elaboration, identification of key TBD's for later increments</li> </ul> </li> </ul>
<b>Feasibility Rationale</b>	<ul style="list-style-type: none"> <li>• Assurance of consistency among elements above               <ul style="list-style-type: none"> <li>- via analysis, measurement, prototyping, simulation, etc.</li> <li>- Business case analysis for requirements, feasible architectures</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• Assurance of consistency among elements above</li> <li>• All major risks resolved or covered by risk management plan</li> </ul>

7/28/99

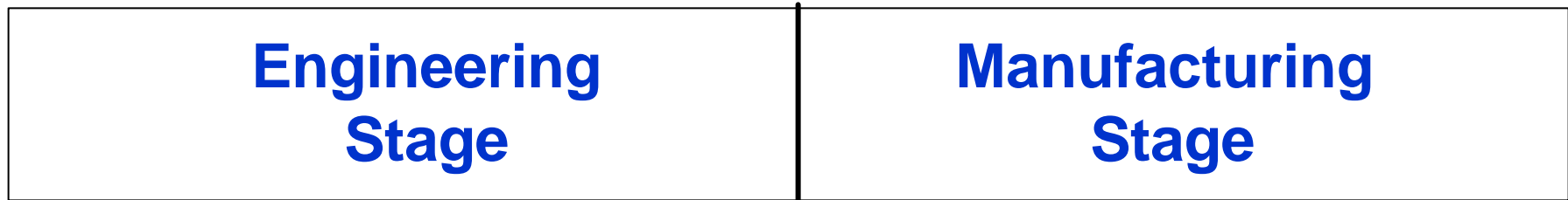


# Initial Operational Capability (IOC)

- **Software preparation**
  - Operational and support software
  - Data preparation, COTS licenses
  - Operational readiness testing
- **Site preparation**
  - Facilities, equipment, supplies, vendor support
- **User, operator, and maintainer preparation**
  - Selection, teambuilding, training

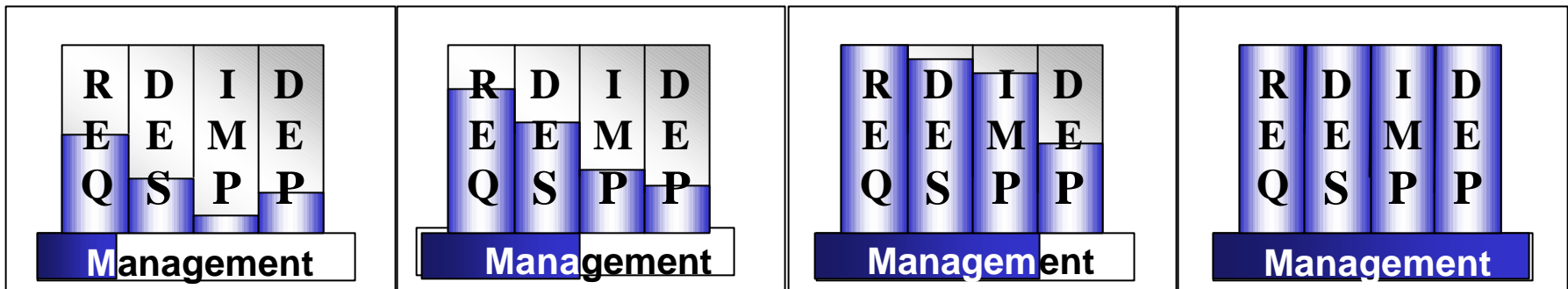


# Anchor Points and Rational USDP Phases

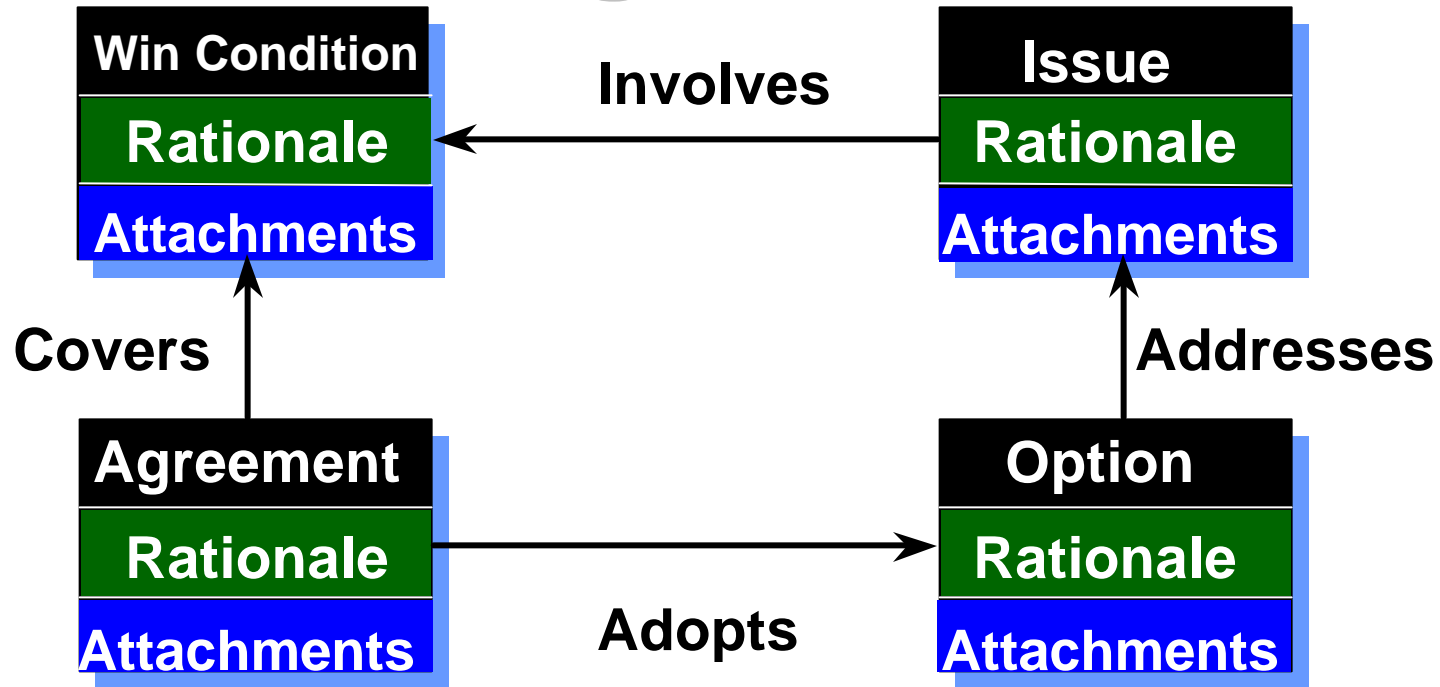


LCO
LCA
IOC

<b>Feasibility Iterations</b>	<b>Architecture Iterations</b>	<b>Usable Iterations</b>	<b>Product Releases</b>
-------------------------------	--------------------------------	--------------------------	-------------------------



# WinWin Negotiation Model



- **Win-Win equilibrium**

- All Win Conditions covered by Agreements
- No outstanding Issues



# WinWin Look and Feel

The screenshot displays the WinWin software interface for a project named "USC/WINWIN 1.2 (adamwu) - team6". The main window is divided into four columns: WIN CONDITIONS, ISSUES, OPTIONS, and AGREEMENTS. Each column contains several artifacts represented as boxes with arrows indicating relationships between them.

- WIN CONDITIONS:** swong-WINC-1 (user friendly), swong-WINC-2 (online system), swong-WINC-3 (current web site), swong-WINC-4 (maintained by USC), swong-WINC-5 (upgrading system), swong-WINC-6 (Retrieving data), swong-WINC-7 (online help).
- ISSUES:** swong-ISSU-1 (Access to librar\*\*), swong-ISSU-3 (necessary equipm\*\*), swong-ISSU-4 (Customer will no\*\*), swong-ISSU-5 (SQL in user inte\*\*), swong-ISSU-6 (Database cannot), tinglich-ISSU (Customers are t), tinglich-ISSU (Developer shou).
- OPTIONS:** swong-OPTN-1 (condition for Ac\*\*), and others partially visible.
- AGREEMENTS:** swong-AGRE-12 (Developer can us\*\*), swong-AGRE-13.

Overlaid on the main window is a "Win Condition" detail window for artifact "swong-WINC-5" (upgrading system). This window includes the following fields:

- ID:** swong-WINC-5
- CREATION DATE:** 10/17/96 13:24
- REVISION DATE:** 10/21/96 19:09
- ROLE:** customer
- STATUS:** Active
- PRIORITY:** Very High
- STATE:** (empty)
- Name:** upgrading system
- Choices=>** Body/Rationale
- Body:** System upgrades should be allowed, because library may have images in other format in the future. System should be reusable. Library doesn't want to spend money later on changing the system all
- Rationale:** (empty)

At the bottom of the main window, a log of recent changes is visible, including entries like "swong-ISSU-6 has been modified-(RefersTo) at 10/19/96" and "swong-WINC-5 has been modified-(Body) at 10/21/96".



# Product Model Research at USC-CSE

- **Guidelines for LCO, LCA product deliverables**
  - Operational Concept Description
    - Domain model, stakeholder roles, use case scenarios
  - Requirements Description
    - Priorities, evolution requirements, WinWin traceability
  - Architecture description
    - Extended UML, COTS integration
  - Feasibility Rationale
    - Assurance of consistency, feasibility, risk resolution
- **Architecture Research**



# Architecture Research

- **SAAGE (Software Architecture Analysis, Generation, and Evolution)**
- **Architecture style and view integration**
  - UML view integration



## Software Architecture, Analysis, Generation, and Evolution (SAAGE)

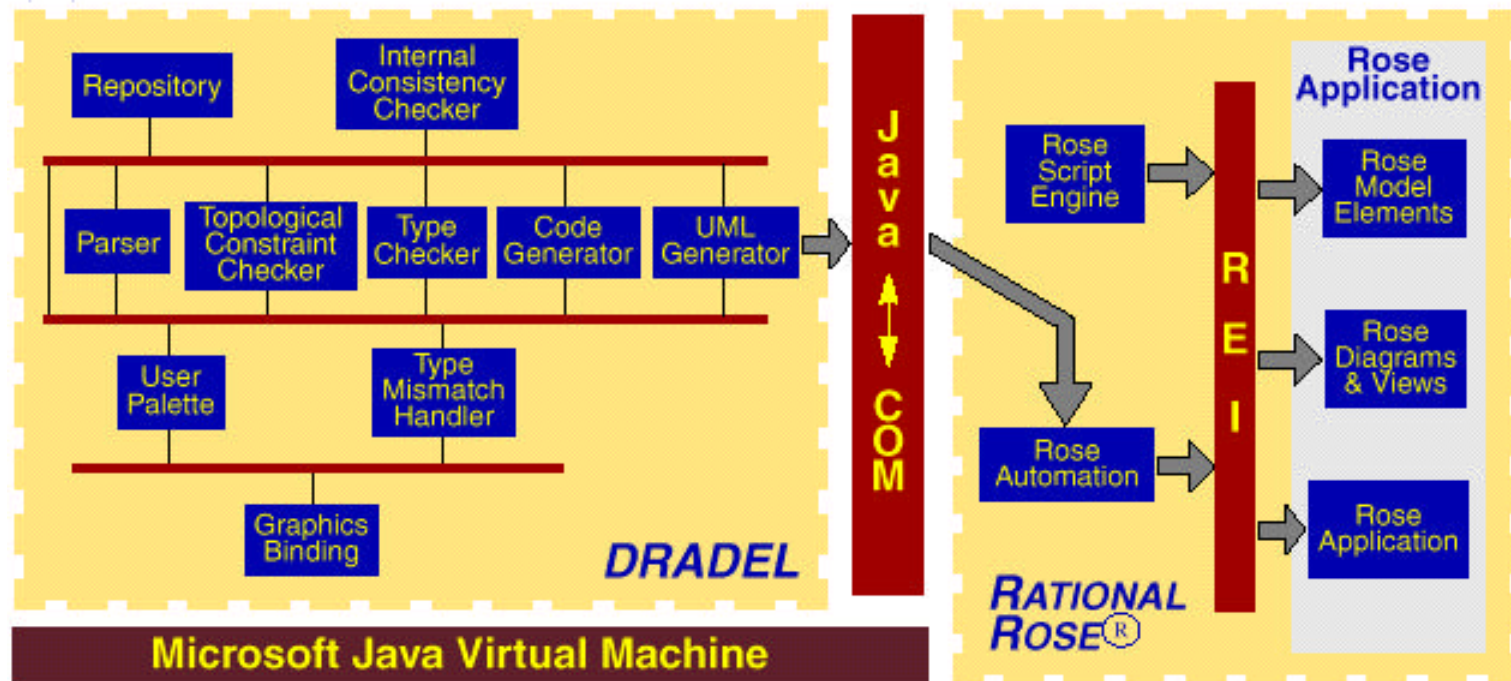
Research focus:

- style-based application design and implementation
- component- and connector-based architectural composition
- architecture modeling and analysis
- architecture-based software evolution
- consistent refinement of architecture into design
- system generation

Supported by an integrated toolset composed of in-house and third-party tools

# SAAGE Overview

- Integrated environment for transforming C2-style architectures into UML



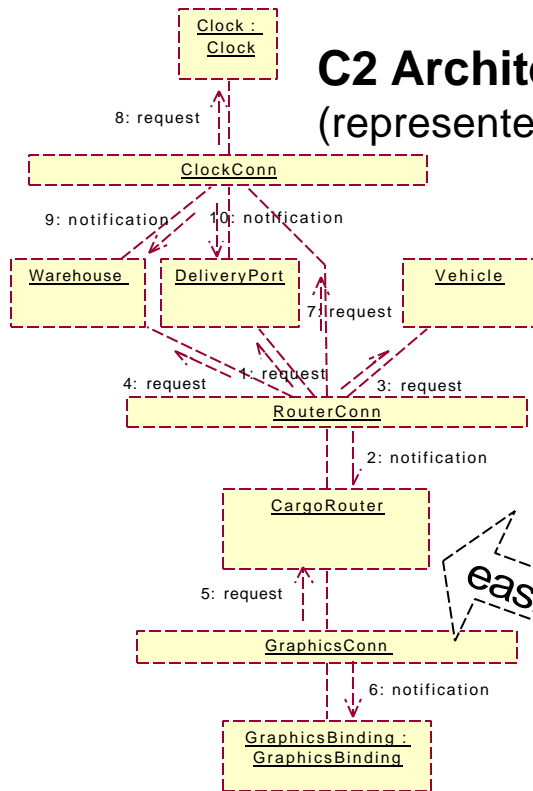


## View Integration Model (UML/Analyzer Tool)

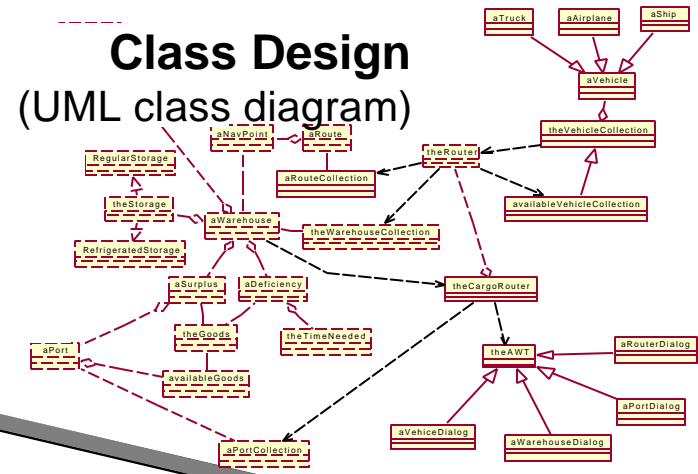
A System for Defining and Analyzing the Conceptual Integrity of UML Models

- implements generic view integration model
- describes and identifies causes of architectural and design mismatches across UML views
- is integrated with Rational Rose®
- complements pure view comparison with transformation (abstraction, translation, and unification) and mapping techniques
- currently supports class, object, sequence, collaboration, state, and C2 diagrams
- uses mismatch rules and transformation rules

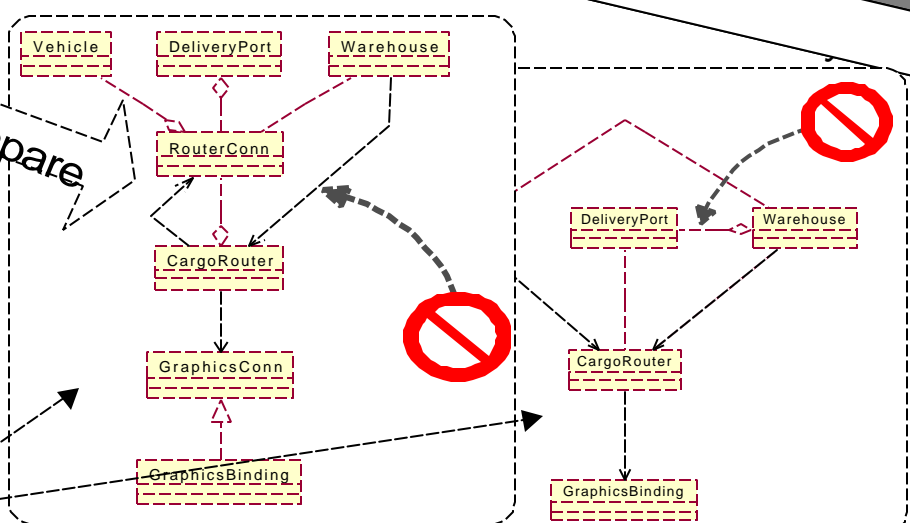
# Consistency between Architecture and Design



hard to compare



easier to compare



Automatically derived Views



# Property Model Research at USC-CSE

- **Constructive Cost Model (COCOMO) II**
- **COCOMO II Extensions**
  - COCOTS: COTS Integration
  - COQUALMO: Delivered Defect Density
  - CORADMO: Rapid Development Cost and Schedule
  - COPSEMO: Phase/Activity Distribution
  - COPROMO: Productivity Improvement Analysis
- **Property Tradeoff Assistance: QARCC, S-COST**



# COCOMO II Summary

- **Extended Modeling Approach**

**Effort ~ (Environment) \* (Size)<sup>Process</sup>**

**Schedule ~ (Effort)<sup>Process</sup>**

- **Tailorable Family of Models**

- **Keyed to information available and development approach**



# COCOMO II Family

<b>Model</b>	<b># Drivers</b>		<b>Sizing</b>
	<b>Envir.</b>	<b>Process</b>	
<b>Applications Composition</b>	<b>2</b>	<b>0</b>	<b>Object Points</b>
<b>Early Design</b>	<b>7</b>	<b>5</b>	<b>Function Points or SLOC</b>
<b>Post-Architecture</b>	<b>17</b>	<b>5</b>	<b>Function Points or SLOC</b>
<b>COCOMO 81</b>	<b>15</b>	<b>1</b>	<b>SLOC (FP Extensions)</b>



# Calibration: COCOMO II.1997 Vs. .1999

	1981	1997	1999
<b>Project Data Points</b>	<b>63</b>	<b>83</b>	<b>161</b>
<b>Calibration</b>		<b>10% Data, 90% Experts</b>	<b>Bayesian</b>
<b>PRED (.30) Values</b>			
• <b>Effort</b>	<b>81%</b>	<b>52%</b>	<b>75%</b>
– by Org'n		<b>64%</b>	<b>80%</b>
• <b>Schedule</b>	<b>65%</b>	<b>61%</b>	<b>72%</b>
– by Org'n		<b>62%</b>	<b>81%</b>

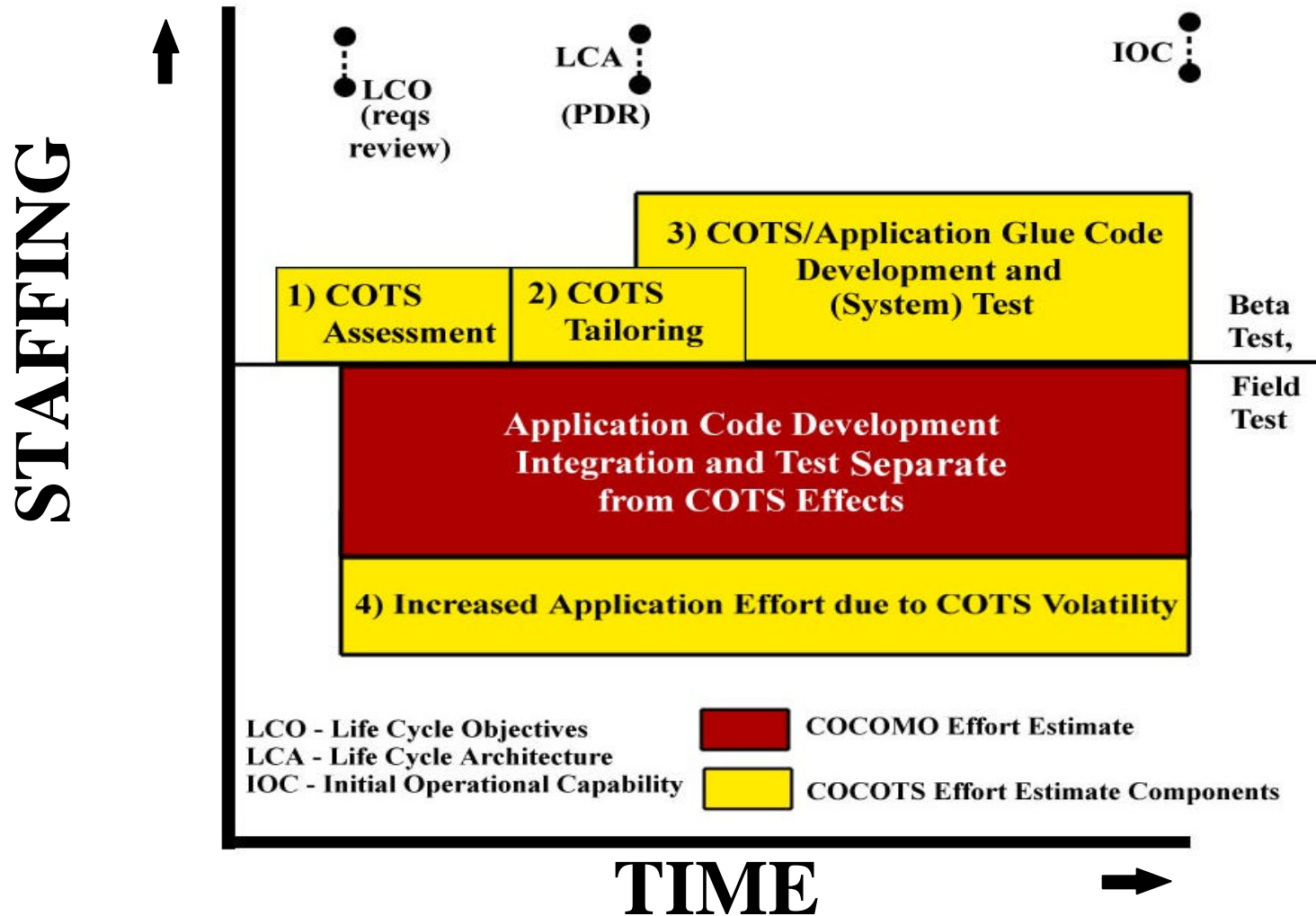


# Process Maturity (PMAT) Effects

- Effort reduction per maturity level, 100 KDSI project
- Normalized for effects of other variables
- **Clark Ph.D. dissertation (112 projects)**
  - Research model: 12-23% per level
  - COCOMO II subsets: 9-29% per level
- **COCOMO II.1999 (161 projects)**
  - 4-11% per level
- **PMAT positive contribution is statistically significant**

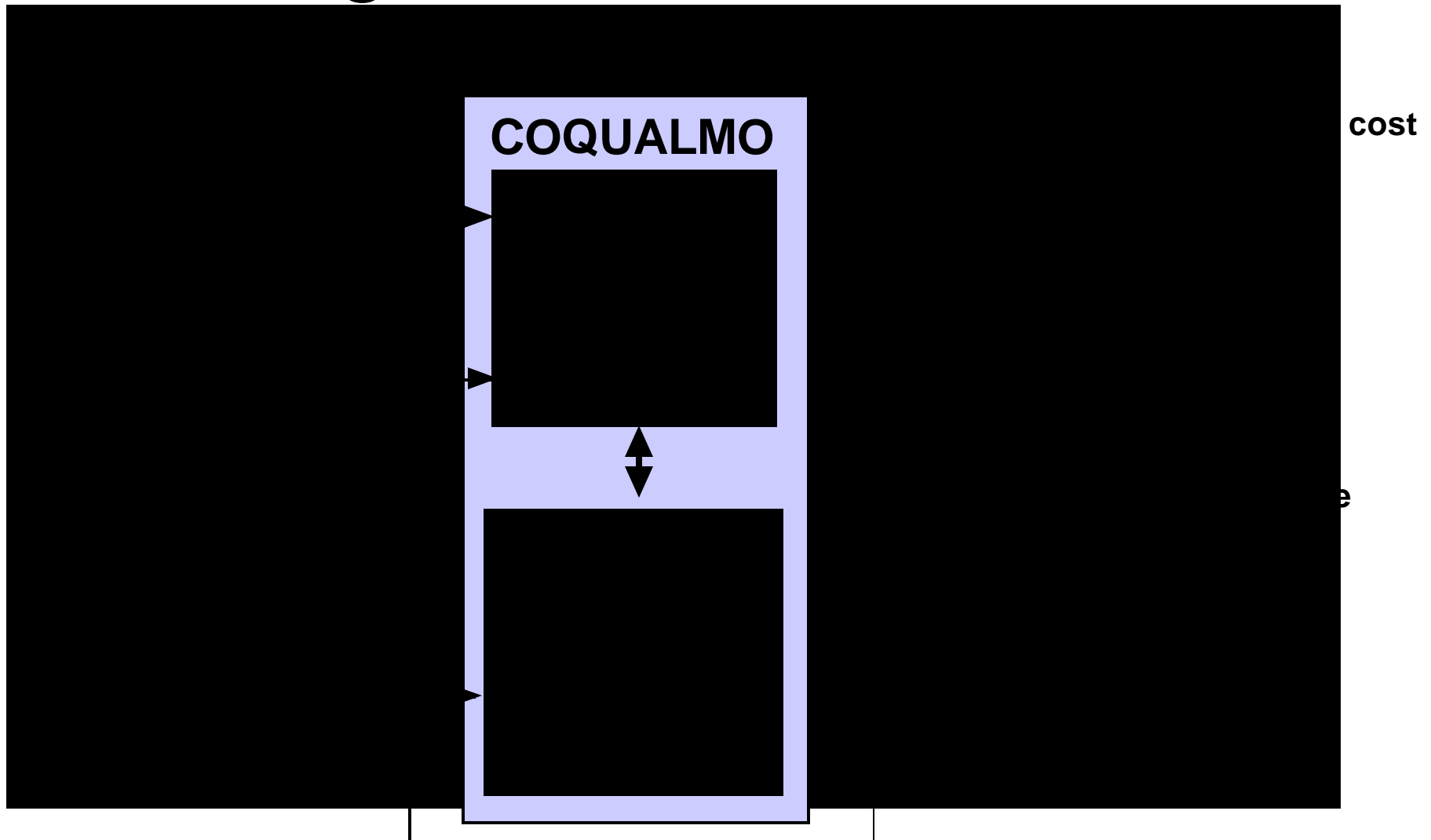


# COCOMO vs. COCOTS Cost Sources



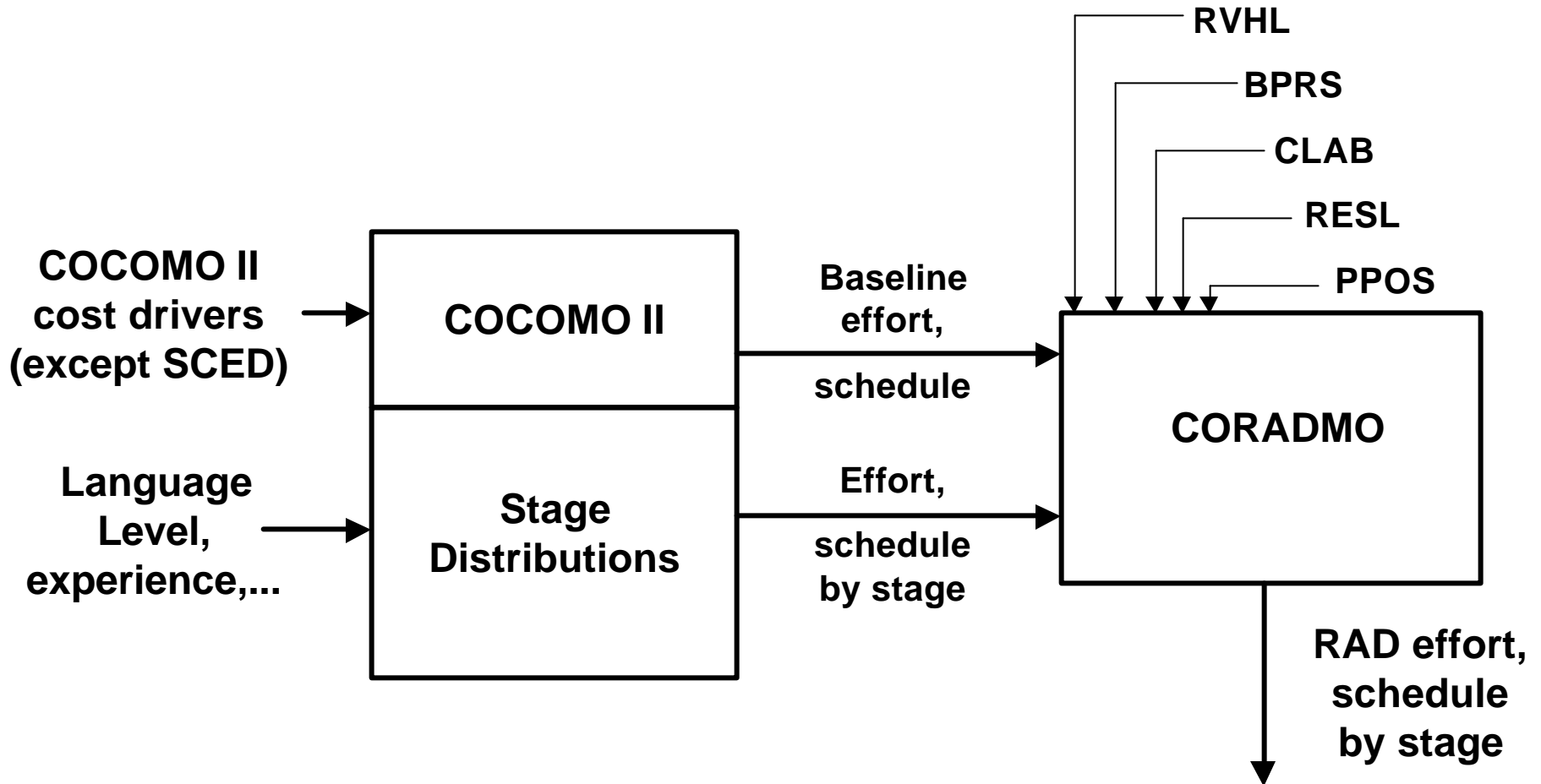


# Integrated COQUALMO





# COCOMO II RAD Extension (CORADMO)





# COCOMO II Book

- Publication date now early-2000
  - Full draft by August 1
- Will use current calibration values as COCOMO II.2000
- Providing USC COCOMO II.2000 on CD included with book
  - Also manuals, tutorials, commercial model demos
- Plan new edition, recalibration every 2 years
  - Affiliates - only recalibration in interim



# Success Model Research at USC-CSE

- **Stakeholder Win-Win (Theory W)**
  - Win win negotiation model and tools
  - Two Cultures and expectations management
- **Success Model profiles and model clashes**



# Unmet Expectations Problems

- **LCO success condition**
  - Describes at least one feasible architecture
  - Satisfying requirements within cost/schedule/resource constraints
  - Viable cost-effective business case
  - Stakeholder concurrence on key system parameters
- **Projects That Failed LCO Criteria**
  - 1996: 4 out of 16 (25%)
  - 1997: 4 out of 15 (27%)

**why?**



# Requirements and Expectations: Domain Model Clashes

- **Easy/hard things for software people**

**“If you can do queries with all those ands, ors, synonyms, data ranges, etc., it should be easy to do natural language queries.”**

**“If you can scan the document and digitize the text, it should be easy to digitize the figures too.”**

- **Easy/hard things for librarians**

**“It was nice that you could add this access feature, but it overly (centralizes, decentralizes) control of our intellectual property rights.”**

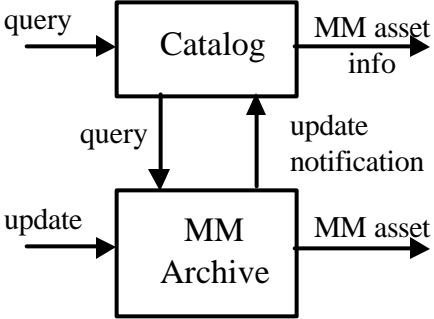
**“It was nice that you could extend the system to serve the medical people, but they haven’t agreed to live with our usage guidelines.”**



# 1998 Simplifier/Complicator Experiment

- **Identify application simplifiers and complicators**
  - For each digital library sub-domain
  - For both developers and clients
- **Provide with explanations to developers and clients**
  - Highlight relation to risk management
- **Homework exercise to analyze simplifiers and complicators**
  - For two of upcoming digital library projects
- **Evaluate effect on LCO review failure rate**

# Example S&C's

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

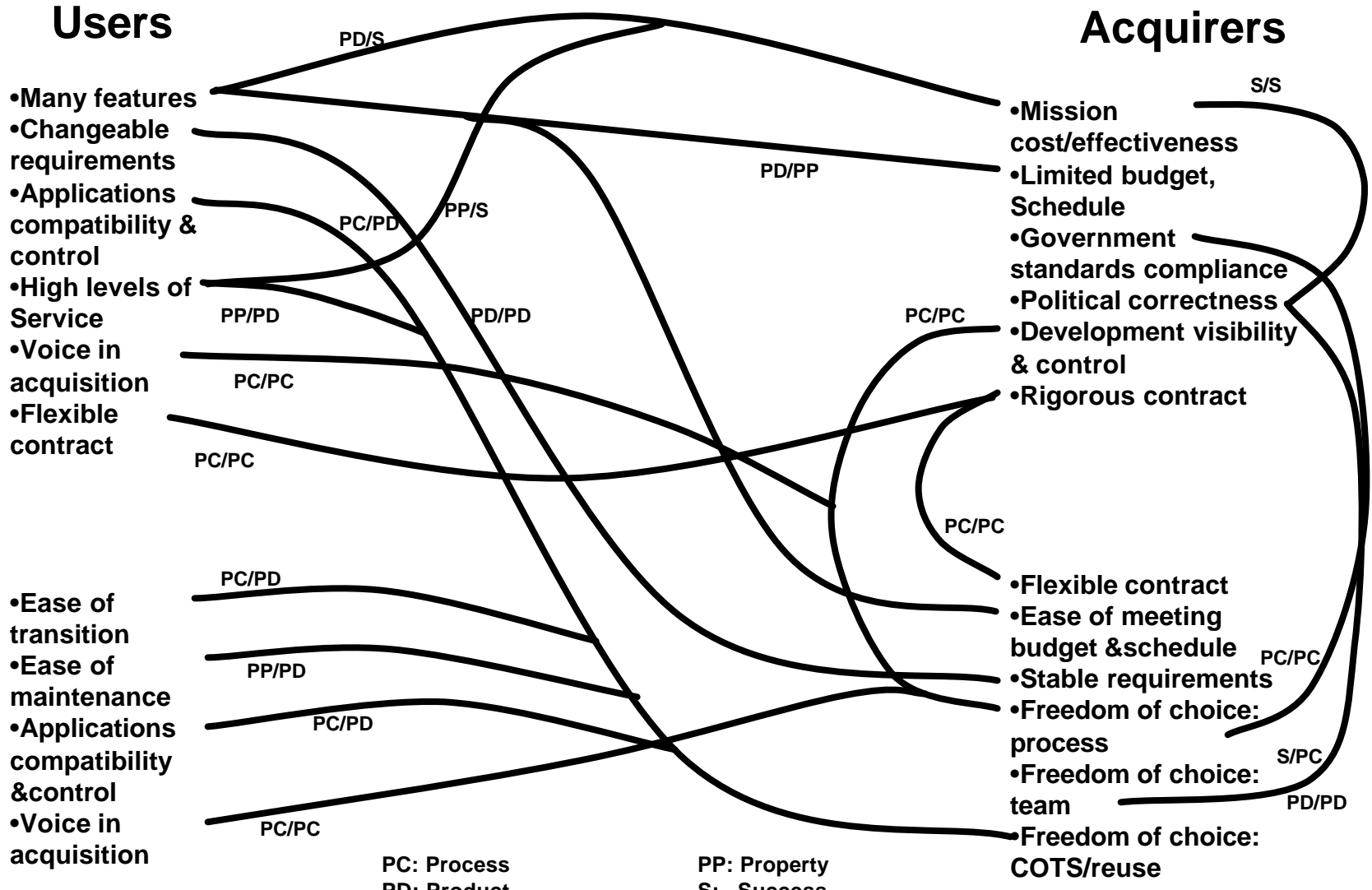


# The Results

- **Projects That Failed LCO Criteria**
  - **1996: 4 out of 16 (25%)**
  - **1997: 4 out of 15 (27%)**
  - **1998: 1 out of 20 (5%)**
- **40% of Student critiques cited S&C's as helpful**
  - **In focusing on achievable requirements set within tight schedule**
  - **In understanding project risks and tradeoffs**



**Success Model-Clash Profiles: General ...**





# Outline

- **Developing Software/System Models at USC-CSE**
  - Process, Product, Property, and Success Models
- • **Integrating Software/System Models**
  - Motivation: Model Clashes
  - Model-Based (System) Architecting and Software Engineering (MBASE)
- **Examples of Successful MBASE Use**
  - Over 50 Digital Library projects
  - TRW CCPDS-R project
- **Early Adopters**
- **Conclusions**
- **Further information**

# Understanding the Tar Pit: Model Clashes



- **Model Clash: An incompatibility among the underlying assumptions of a set of models**
  - Often unrecognized
  - Produces conflicts, confusion, mistrust, frustration, rework, throwaway systems

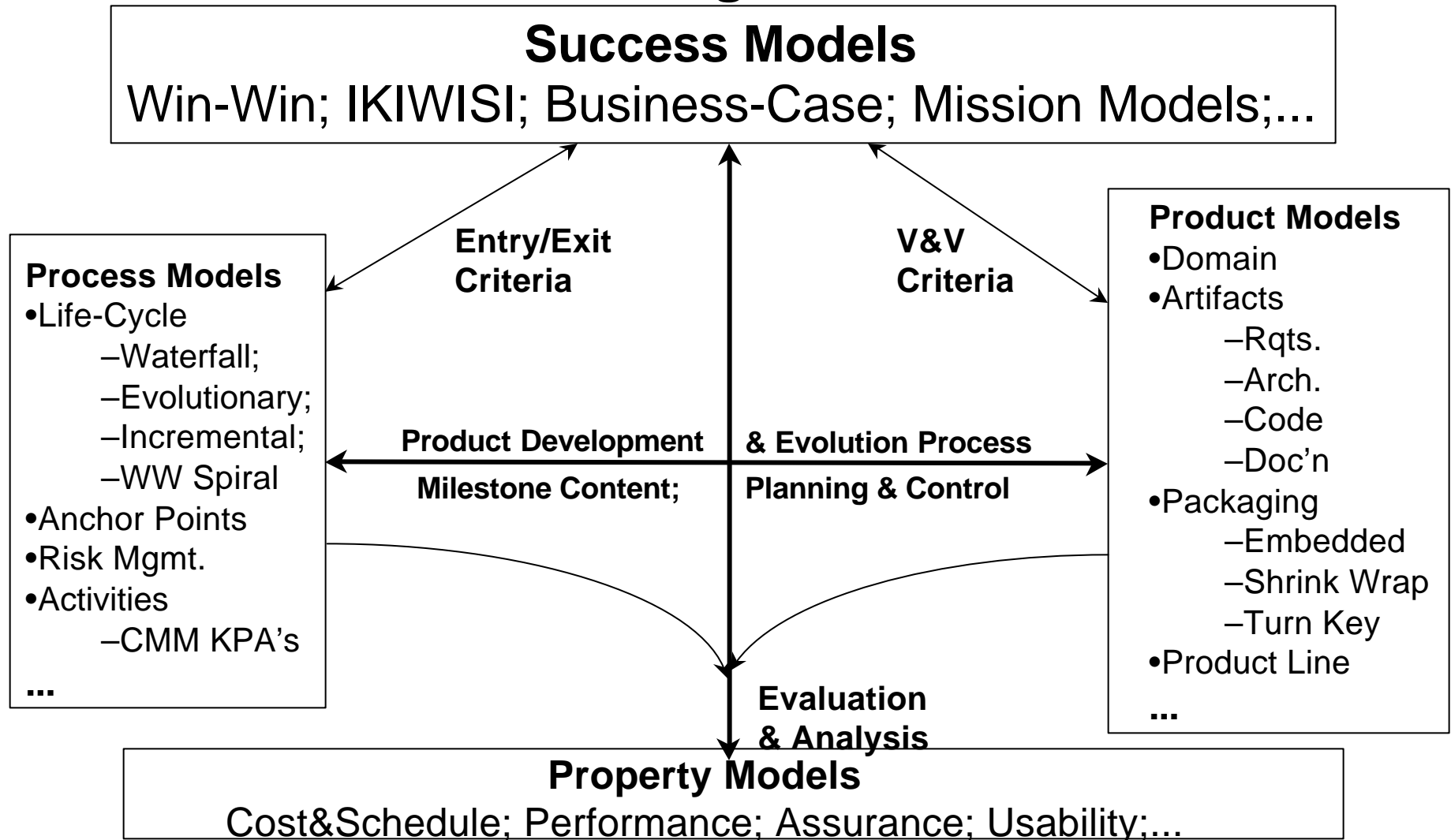


# Examples of Model Clashes

- **Design-to-schedule process, unprioritized requirements, and tightly-coupled architecture**
- **COTS-driven product and Waterfall process**
- **Risk-based process and spec-based progress payments**
- **Evolutionary development without life-cycle architecture**
- **Golden Rule and stakeholder win-win**
- **Spec-based process and IKIWISI success model**
  - **I'll know it when I see it**

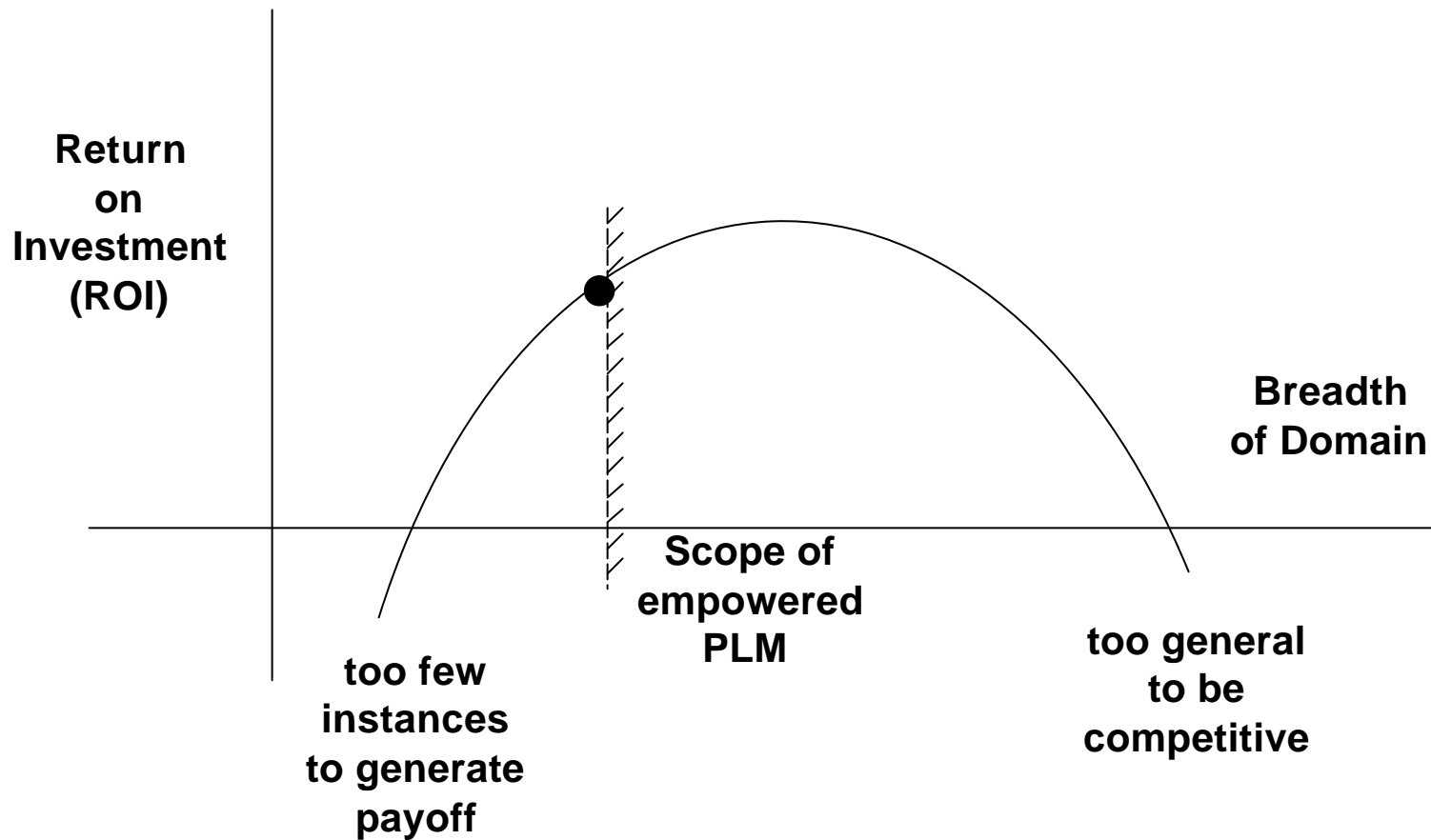


# MBASE Model Integration Framework



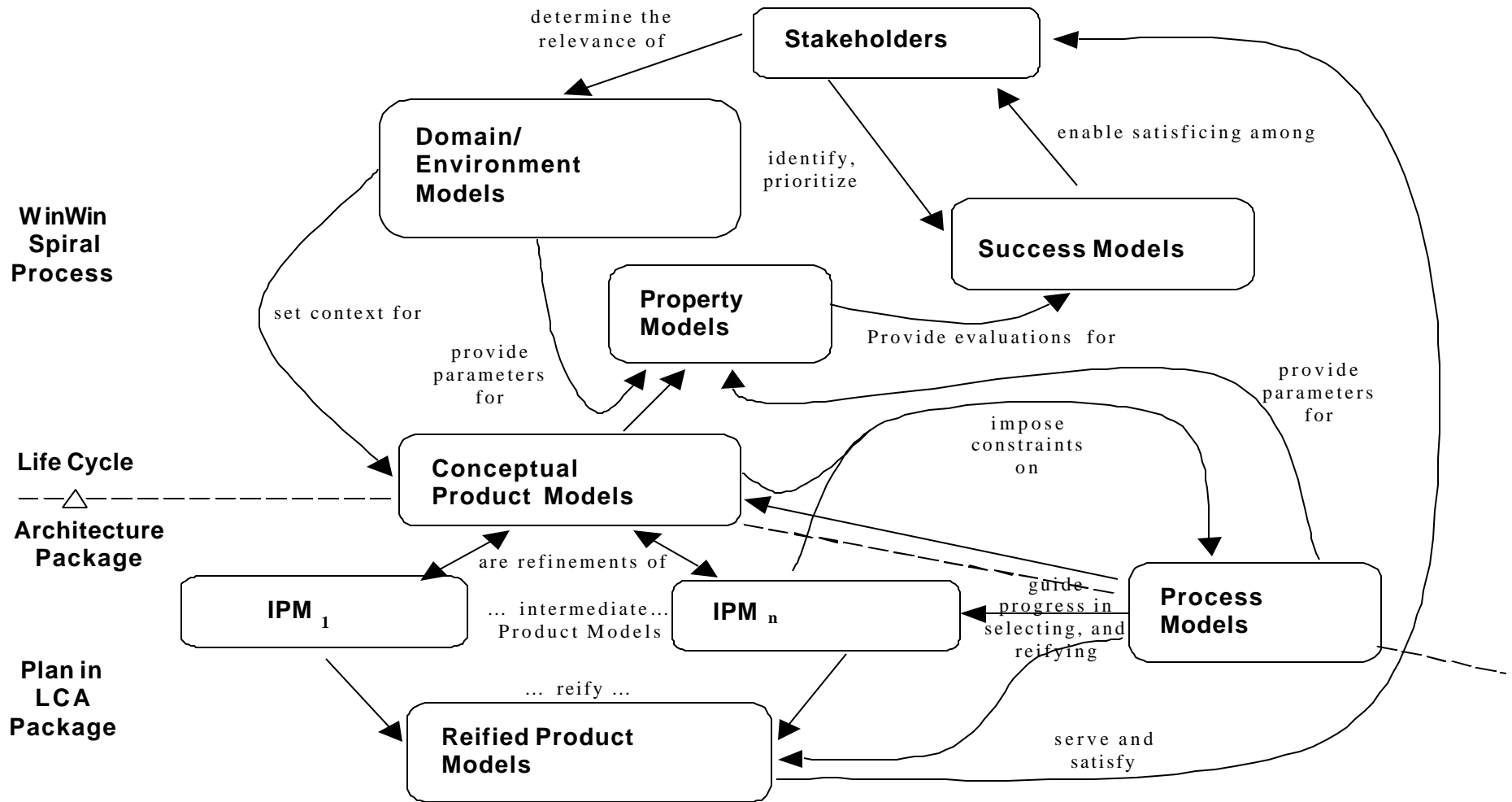


# Product Line Domain Scope a Function of ROI, Scope of Empowered PL Manager





# MBASE Process Framework





## Success Models Drive Other Model Choices

<b>Success Model</b>	<b>Demo agent-based E-commerce system at COMDEX in 9 months</b>	<b>Safe air traffic control system</b>
<b>Key Stakeholders</b>	<b>Entrepreneurs, venture capitalists, customers</b>	<b>Controllers, Govt. agencies, developers</b>
<b>Key Property Models</b>	<b>Schedule estimation</b>	<b>Safety models</b>
<b>Process Model</b>	<b>Design-to-schedule</b>	<b>Initial spiral to risk-manage COTS, etc.; Final waterfall to verify safety provisions</b>
<b>Product Model</b>	<b>Domain constrained by schedule; architected for ease in dropping features to meet schedule</b>	<b>Architected for fault tolerance, ease of safety verification</b>



# Outline

- **Developing Software/System Models at USC-CSE**
  - Process, Product, Property, and Success Models
- **Integrating Software/System Models**
  - Motivation: Model Clashes
  - Model-Based (System) Architecting and Software Engineering (MBASE)
- • **Examples of Successful MBASE Use**
  - Over 50 Digital Library projects
  - TRW CCPDS-R project
- **Early Adopters**
- **Conclusions**
- **Further information**



# The Challenge

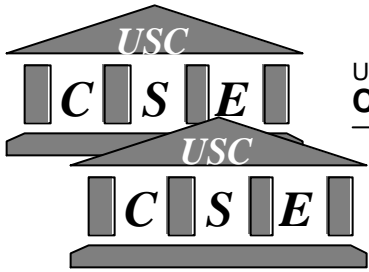
- **15 Digital Library Applications**
  - 2 sentence problem statements
  - Librarian clients
- **86 Graduate Students**
  - 30% with industry experience
  - Largely unfamiliar with each other, Library ops.
- \* **Develop LCA packages in 11 weeks**
- **Re-form teams from 30 continuing students**
- \* **Develop IOC packages in 12 more weeks**
  - Including 2-week beta test and transition



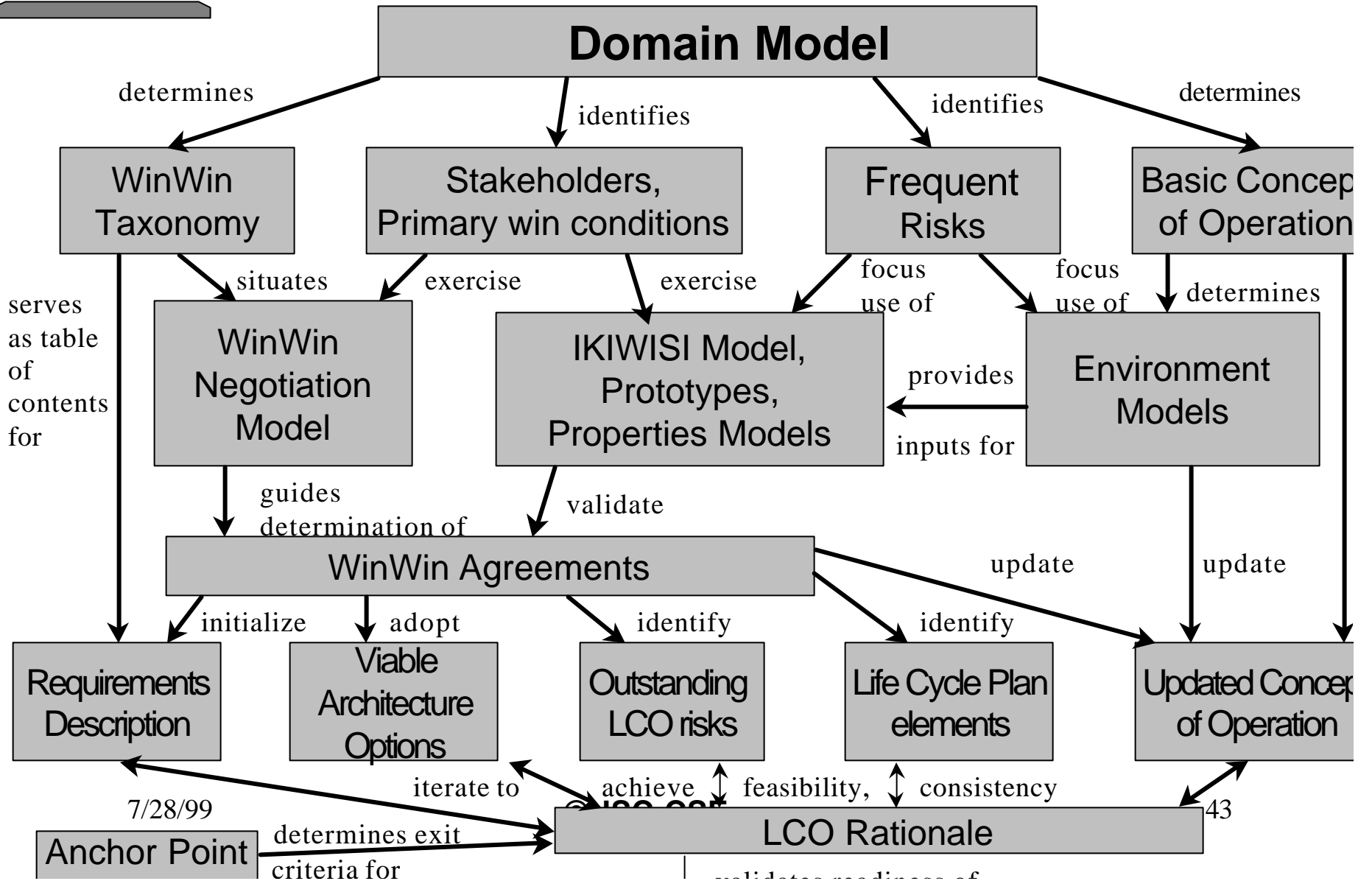
# Problem Statement #4: Medieval Manuscripts

**Ruth Wallach, Reference Center, Doheny Memorial Library**

**I am interested in the problem of scanning medieval manuscripts in such a way that a researcher would be able to both read the content, but also study the scribe's hand, special markings, etc. A related issue is that of transmitting such images over the network.**



# MBASE Model Integration: LCO Stage





**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 Home Page

**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

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

**Digital Manuscript Archive Display Leaf Image Page - Netscape**  
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

Document: Done



# MBASE Laboratory

- **15 software engineering projects/year**
  - 5-person USC Digital Library applications
- **Rapidly developing successful applications**
  - Multimedia, virtual assistants, data acquisition
- **Integrating models and tools**
  - DARPA-EDCS architecture and WinWin tools
  - Rational Rose, Unified Modeling Language
- **Rapidly improving artifact integration**
  - 1996 integrated specs, plans: 160 pages
  - 1997 integrated specs, plans: 110 pages
- **Results transitioning to early adopters**
- **Ultimate goal: Model-integrated SW Engr. agents**



# Case Study: CCPDS-R Project Overview

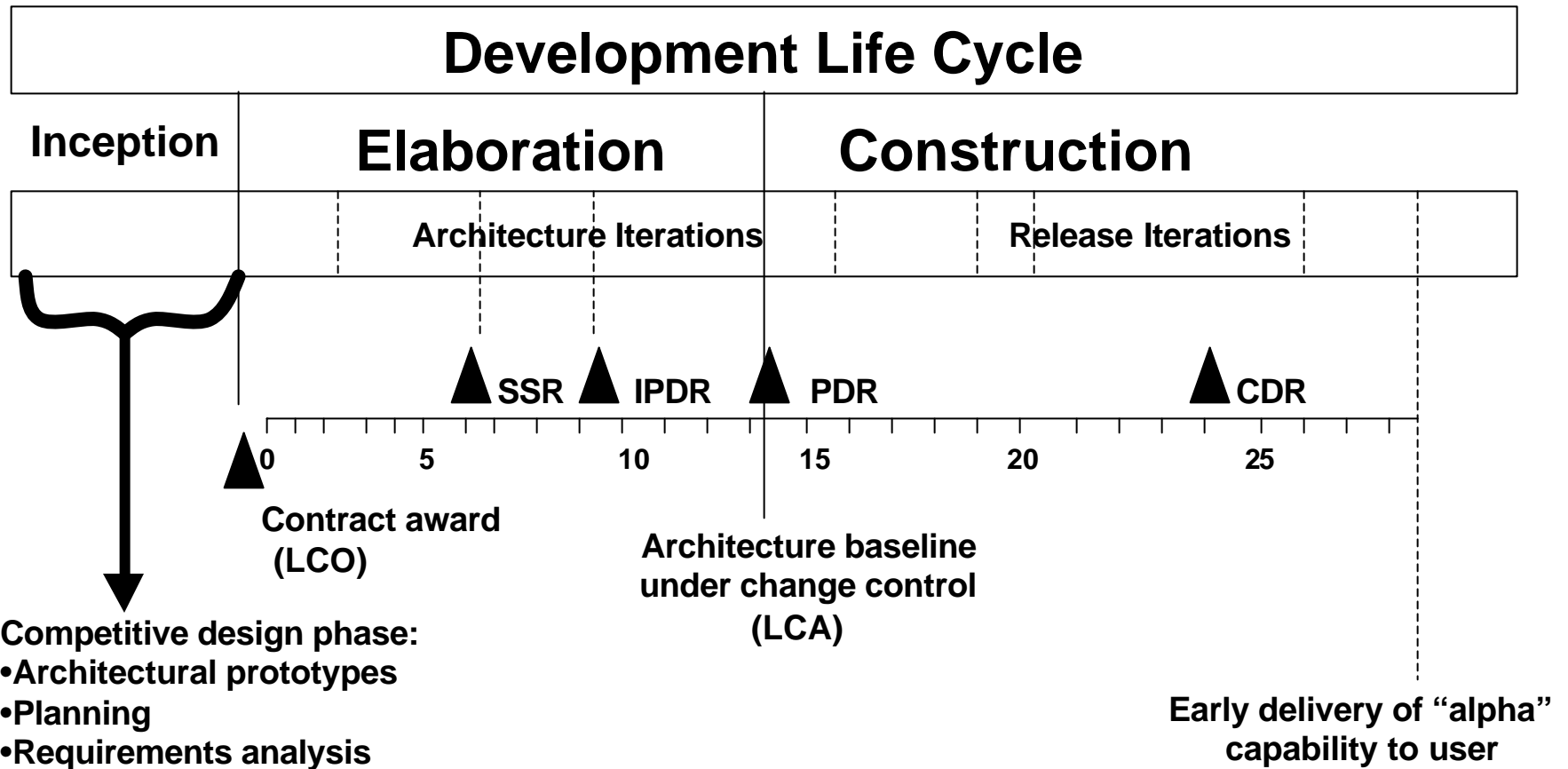
Characteristic	CCPDS-R
Domain	Ground based C3 development
Size/language	1.15M SLOC Ada
Average number of people	75
Schedule	75 months
Process/standards	DOD-STD-2167A Iterative development
Environment	Rational host DEC host DEC VMS targets
Contractor	TRW
Customer	USAF
Current status	Delivered On-budget, On-schedule



# CCPDS-R MBASE Models

- **Success Models**
  - Reinterpreted DOD-STD-2167a; users involved
  - Award fee flowdown to performers
- **Product Models**
  - Domain model and architecture
  - Message-passing middleware (UNAS)
- **Process Models**
  - Ada process model and toolset
  - Incremental builds; early delivery
- **Property Models**
  - COCOMO cost & schedule
  - UNAS - based performance modeling
  - Extensive progress and quality metrics tools

# Common Subsystem Macroprocess





# Early Adopters

- **Rational, Xerox, FAA**
- **Air Force C2ISR Center**
  - **Field initial new C2ISR capabilities in 18 months**
  - **Determine, support common spiral model**
  - **General Officers' Offsite Feb. 17-18, 1999**
    - **LG's Kadish, Donahue, Martin**
    - **MG's Cliver, Hawley, Carlson, Hess**
  - **Adopt WinWin Spiral Model as baseline**
  - **Revise draft AFI 63-123,**  
**“Evolutionary Acquisition for C2 Systems”**



# Conclusions

- **Research on several classes of models is synergetic**
  - COCOMO II, WinWin, spiral extensions, architecture research each helped by each other
- **MBASE proving successful in avoiding model clashes**
- **USC-CSE Affiliate support has been essential**



# Further Information

**B. Boehm et al., “Using the Win Win Spiral Model: A Case Study,” IEEE Computer, July 1998, pp. 33-44.**

**B. Boehm, D. Port, “Escaping the Software Tar Pit: Model Clashes and How to Avoid Them,” ACM Software Engineering Notes, January, 1999.**

**B. Boehm et al., Estimating Software Costs with COCOMO II, Prentice Hall, 2000 (to appear).**

**W.E. Royce, Software Project Management: A Unified Framework, Addison Wesley, 1998.**

**USC-CSE Web-Site. <http://sunset.usc.edu>**