

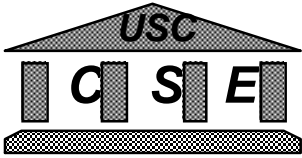
University of Southern California  
Center for Software Engineering

---

# **COCOMO II and Model-Based (System) Architecting and Software Engineering (MBASE)**

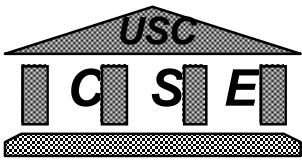
**Barry Boehm, USC  
COCOMO/SCM Forum Tutorial  
October 24, 2000**

**boehm@sunset.usc.edu  
<http://sunset.usc.edu/research/MBASE>**



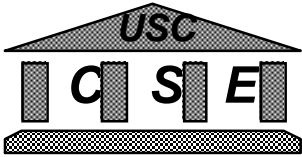
# Outline

- **Future software trends and challenges**
- **Relations to spiral model, COCOMO II, and MBASE**
- **MBASE principles and key practices**
  - Relation to new CMMI process areas
  - Relation to Benefits Realization
- **MBASE project usage experience**
- **COCOMO II MBASE/Rational Unified Process phase and activity distributions**



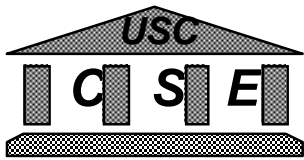
# Software Trends and Challenges

- **Trends and challenges**
  - Rapid change; COTS integration; Web/Net applications; systems of systems
- **Commercial responses**
  - Rapid application development; synch & stabilize; spiral and adaptive processes; benefits realization
- **Government responses**
  - Evolutionary acquisition (new DoD 5000 series); integrated capability maturity models (CMM's)



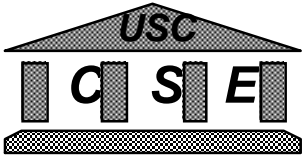
# The SW-CMM to CMMI Paradigm Shift

- **Focus on the system vs. on the software**
  - **SW-CMM: “System requirements analysis is not the responsibility of the SW group, but is a prerequisite for their work.”**
- **New CMMI process areas added**
  - **Shared vision; collaborative leadership; integrated team; customer and product reqts.; decision analysis and resolution; risk management; others**



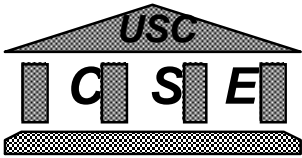
# Resulting Implementation Challenge

- **Need a model that is**
  - Well-grounded in software experience
  - Supportive of all CMMI process areas
  - Tailorable to individual situations
  - Specific about what to do
  - Supportive of future software/system trends
    - Rapid Application Development (RAD)
    - Cost/Schedule as Independent Variable (CAIV/SAIV)
    - COTS Integration
    - Web, Internet, Agents, et al.
- **Spiral Model? Almost. Needs some refinements**
- **MBASE? Provides key refinements**



# Outline

- **Future software trends and challenges**
- ➔ • **Relations to spiral model, COCOMO II and MBASE**
- **MBASE principles and key practices**
  - Relation to new CMMI process areas
  - Relation to Benefits Realization
- **MBASE project usage experience**
- **COCOMO II MBASE/Rational Unified Process phase and activity distributions**



# Spiral Model Refinements

- Where do objectives, constraints, alternatives come from?

- Win Win extensions

- Lack of intermediate milestones

- Need in COCOMO II

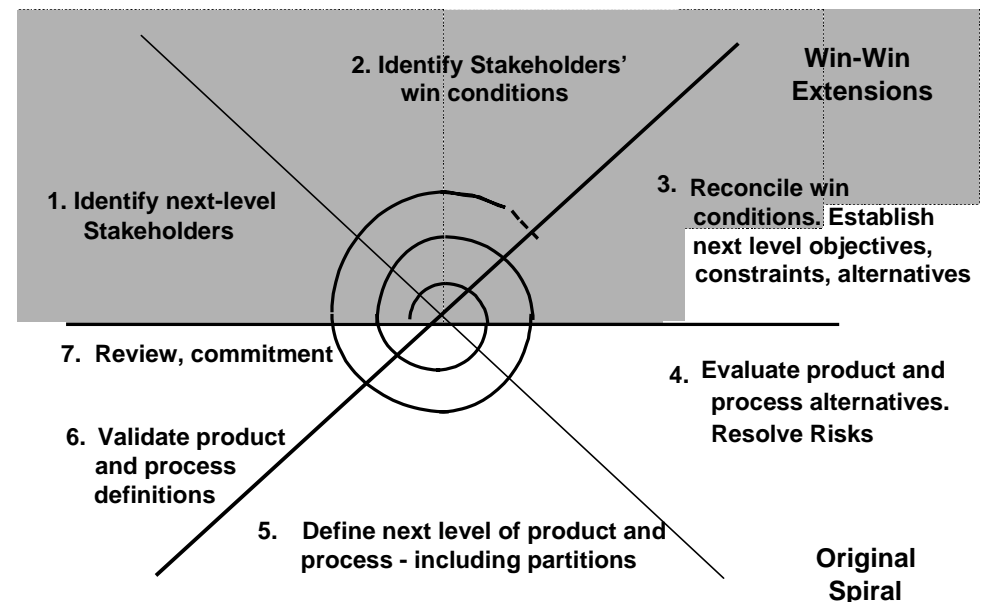
- Anchor Points: LCO, LCA, IOC

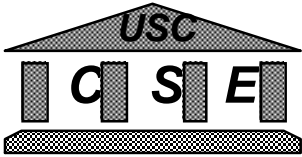
- Concurrent-engineering spirals between anchor points

- Need to avoid model clashes, provide more specific guidance

- MBASE

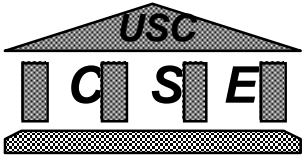
## The WinWin Spiral Model





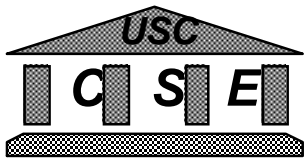
# **COCOMO II Need for Milestones**

- **COCOMO II oriented toward new processes**
  - Spiral, evolutionary, incremental, adaptive
- **No way to ground and calibrate COCOMO II**
- **Convened two Affiliates' workshops to try to define common stakeholder commitment milestones**
- **Resulting life cycle anchor points became foundations for MBASE (and Rational Unified Process)**



# 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

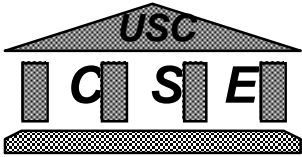


# Win Win Spiral Anchor Points

(Risk-driven level of detail for each element)

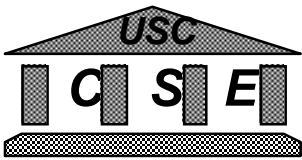
Milestone Element	Life Cycle Objectives (LCO)	Life Cycle Architecture (LCA)
<b>Definition of Operational Concept</b>	<ul style="list-style-type: none"> <li>• Top-level system objectives and scope               <ul style="list-style-type: none"> <li>- System boundary</li> <li>- Environment parameters 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</li> <li>- Identification of TBD's (to-be-determined items)</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>

\*WWWWWHH: Why, What, When, Who, Where, How, How Much



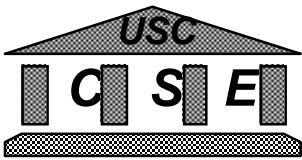
# MBASE Overview

- **Nature of Model Clashes**
- **MBASE Integration Framework**
- **MBASE Process Framework**
- **Relations to Commercial Best Practices**
  - **Rational Unified Process**
  - **AT&T/Lucent Architecture Review Boards**
- **MBASE Electronic Process Guide**



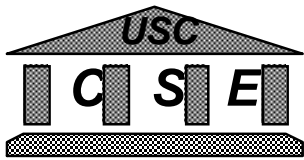
# Nature of Model Clashes

- **Model (Webster):** *A description or analogy used to help visualize or analyze something; a pattern of something to be made.*
  - Includes product models, process models, property models, success models
- **Model Clash:** *An incompatibility among the underlying assumptions of a set of models.*
  - Produces conflicts, confusion, mistrust, frustration, rework, throwaway systems
- **Model Integration:** *Choosing and/or reengineering models to reconcile their underlying assumptions.*



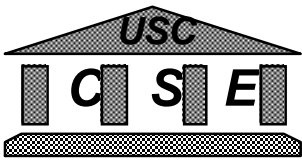
# Clashes Among MBASE Models

	Product Model	Process Model	Property Model	Success Model
Product Model	<ul style="list-style-type: none"> <li>• Structure clash</li> <li>• Traceability clash</li> <li>• Architecture style clash</li> </ul>	<ul style="list-style-type: none"> <li>• COTS-driven product vs. Waterfall (requirements-driven) process</li> </ul>	<ul style="list-style-type: none"> <li>• Interdependent multiprocessor product vs. linear performance scalability model</li> </ul>	<ul style="list-style-type: none"> <li>• 4GL-based product vs. low development cost and performance scalability</li> </ul>
Process Model		<ul style="list-style-type: none"> <li>• Multi-increment development process vs. Single-increment support tools</li> </ul>	<ul style="list-style-type: none"> <li>• Evolutionary development process vs. Rayleigh-curve cost model</li> </ul>	<ul style="list-style-type: none"> <li>• Waterfall process model vs. "I'll know it when I see it" (IKIWISI) prototyping success model</li> </ul>
Property Model			<ul style="list-style-type: none"> <li>• Minimize cost and schedule vs. maximize quality (Quality is free)</li> </ul>	<ul style="list-style-type: none"> <li>• Fixed-price contract vs. easy-to-change, volatile requirements</li> </ul>
Success Model				<ul style="list-style-type: none"> <li>• Golden Rule vs. stakeholder win-win</li> </ul>

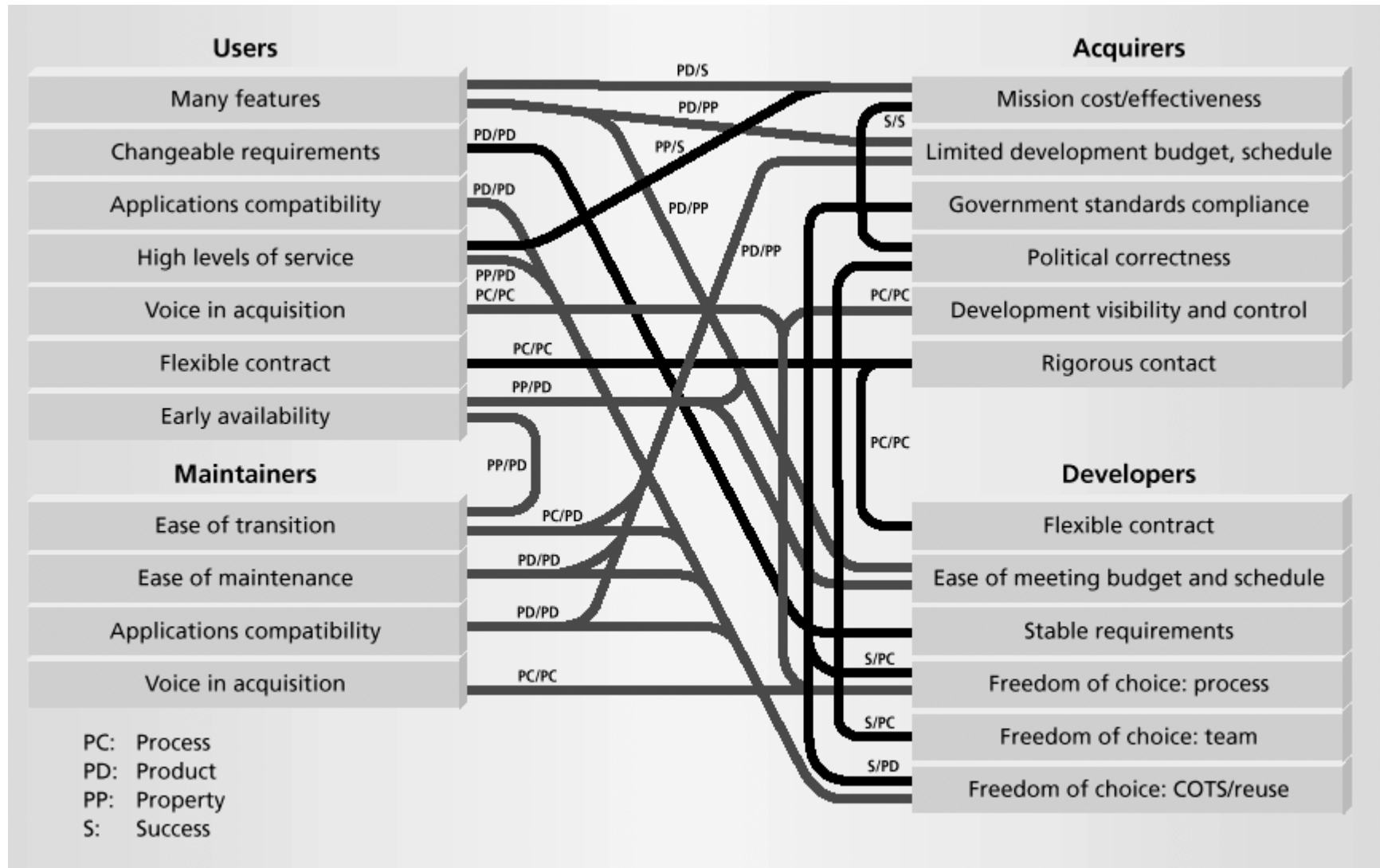


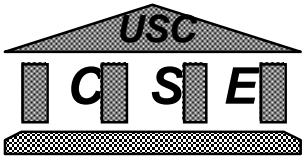
# Where do Models (and Clashes) Come From?

- **Childhood training**
  - Golden Rule, easiest - first
- **Past experience**
  - Waterfall, Add people to speed up
- **Exaggerating for effect**
  - Quality is free, COTS marketing
- **Government/Corporate policy**
  - Use waterfall, use COTS, use Ada, use 4GL's, Cost as Independent Variable
- **Built-in conflicts among stakeholder success models**

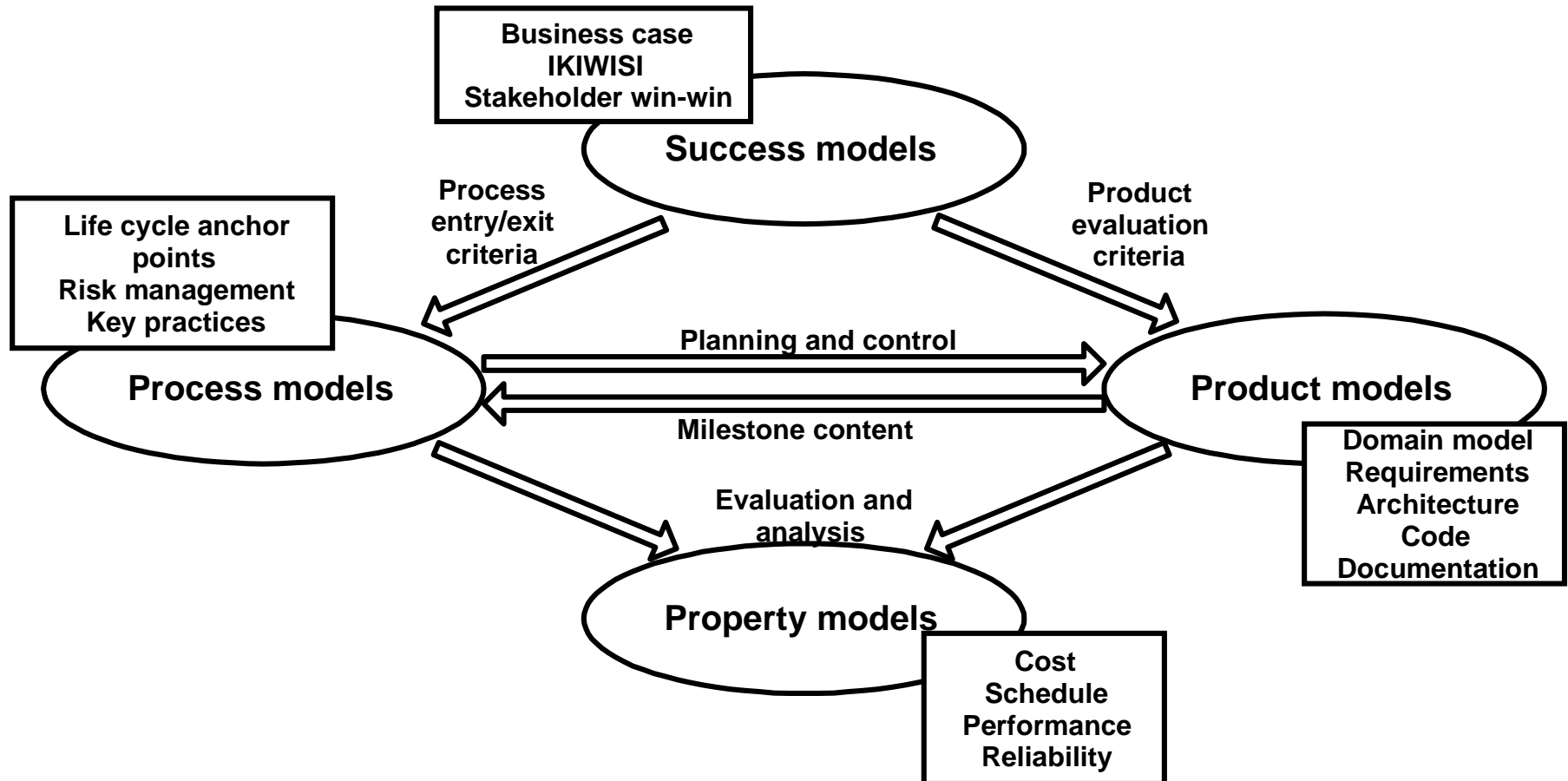


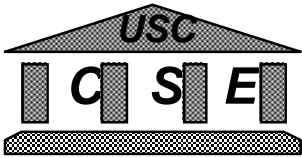
# Success Model-Clash Profiles: MasterNet



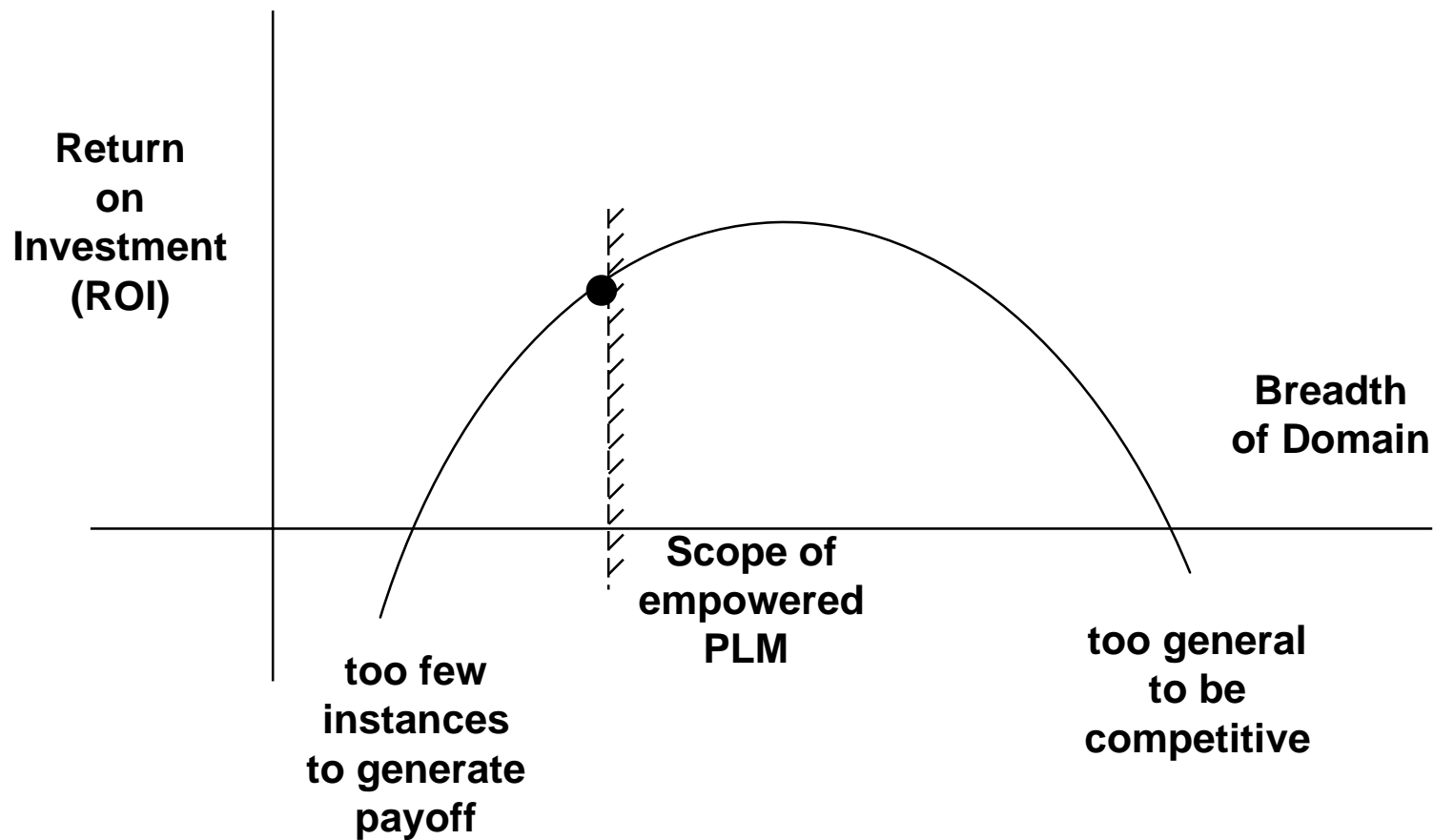


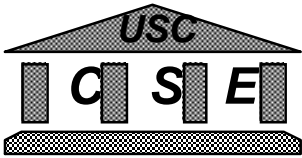
# MBASE Integration Framework



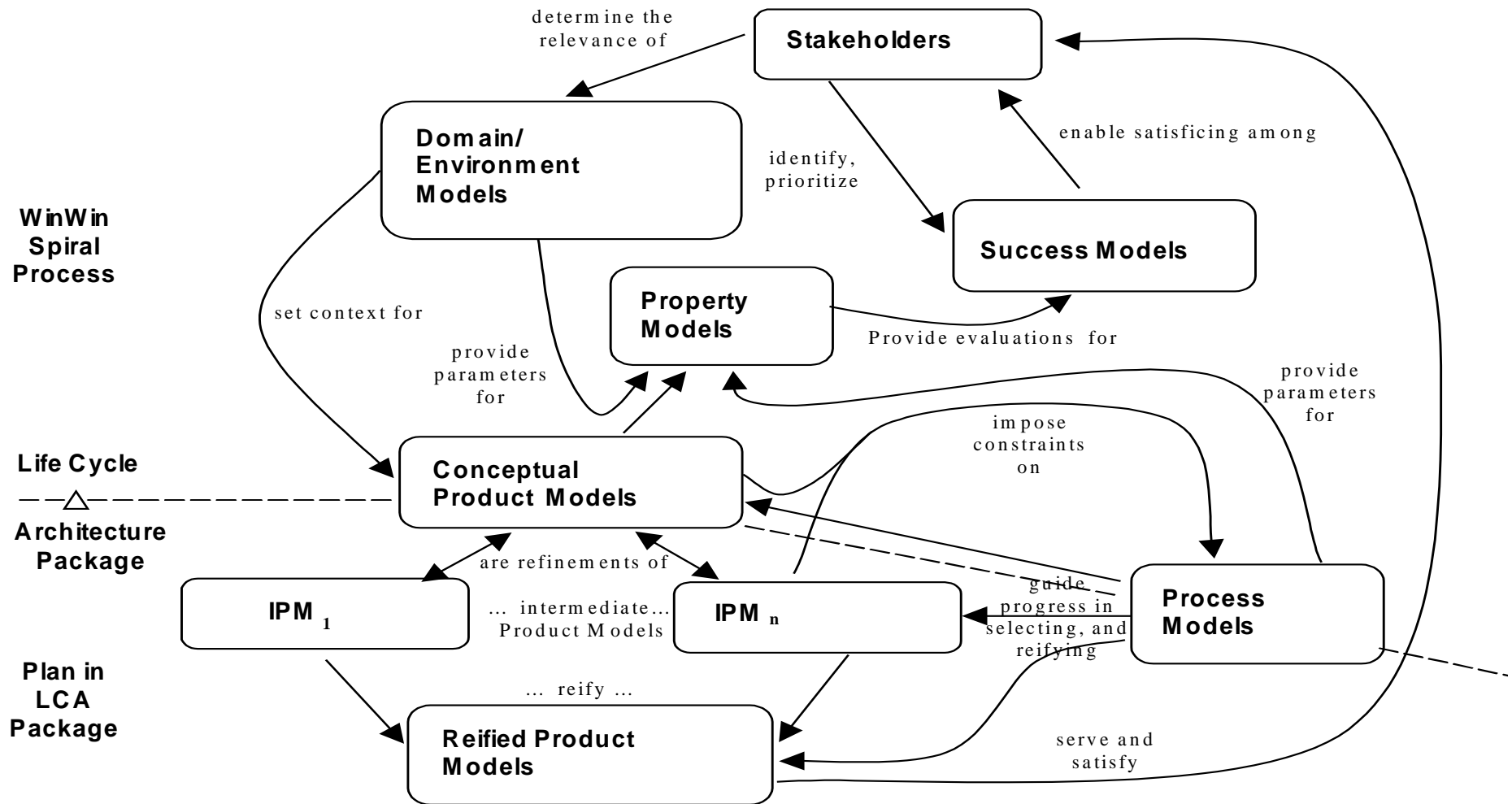


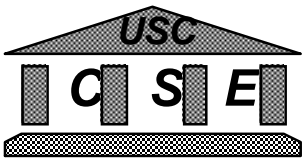
## Product Line Domain Scope a Function of ROI, Scope of Empowered PL Manager





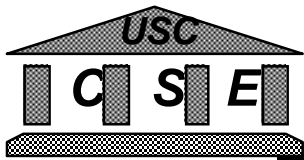
# MBASE Process Framework





# 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>

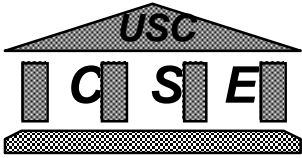


# Elements of Critical Front End Milestones

(Risk-driven level of detail for each element)

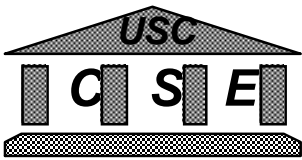
Milestone Element	Life Cycle Objectives (LCO)	Life Cycle Architecture (LCA)
<b>Definition of Operational Concept</b>	<ul style="list-style-type: none"> <li>• Top-level system objectives and scope               <ul style="list-style-type: none"> <li>- System boundary</li> <li>- Environment parameters 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</li> <li>- Identification of TBD's (to-be-determined items)</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>

\*WWWWWHH: Why, What, When, Who, Where, How, How Much



# 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



# Objectory Management Checkpoints

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

## Major Milestones



LCO



LCA



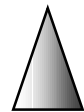
IOC



Full Release

Strategic focus on global concerns of the entire software project

## Minor Milestones



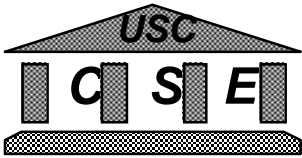
Tactical focus on local concerns of current iteration

## Status

## Assessments



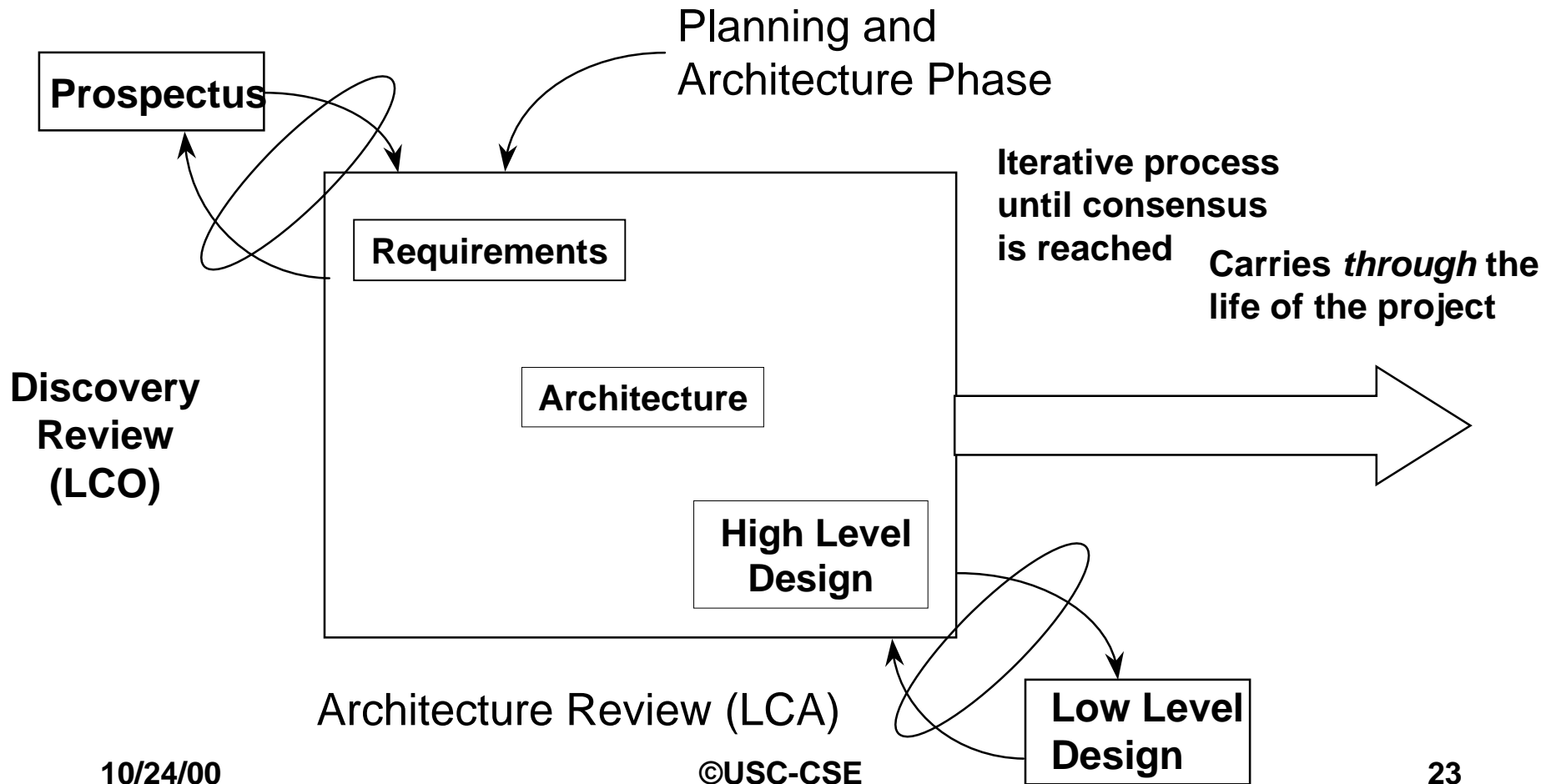
Periodic synchronization of stakeholder expectations

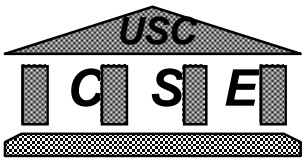


AT&T Architectural Review Boards:

# Architecture in a Project's Life Cycle

It encompasses the requirements, architecture and high level design phases of the typical waterfall diagram. It also continues throughout the life of the project (someone continues to wear the architect's hat).





# MBASE Electronic Process Guide (1)

**Process Explorer - Netscape**  
File Edit View Go Communicator Help

### MBASE 577 Process Guide

PROCESS EXPLORER | GLOSSARY | BEHAVIORAL DIAGRAMS | FUNCTIONAL DIAGRAMS | GUIDE HELP | PROCESS HELP | (C) 1999 Info | CLOSE WINDOW

#### Activities

- MBASE 577 a Process
  - Inception Phase
    - Risk-driven Analysis
      - Identify Critical Risks
      - Identify Frequent Risks
      - Develop Prototype
    - Domain Analysis
      - Tailor WinWin Taxonomy
      - Negotiate System Capabilities
      - Consider Product Line Opportu
      - Define System Boundary and I
      - Describe Current System and F
    - Success Analysis
      - Identify Stakeholders
      - Identify Primary Win Condition
      - Create WinWin Agreements
      - Develop Business Case
    - Product Analysis
      - Describe System Requirements
      - Identify Viable Architecture Op
      - Assess and analyze COTS proc
    - Process Analysis
      - Identify Lifecycle Strategy
      - Identify Top Level WWWW/WH
    - Property Analysis

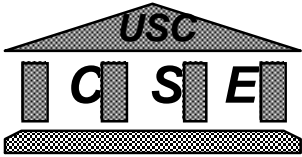
#### Artifacts

- Engineering Documents
  - LCA Package
    - Operational Concept Description
      - Domain Description
      - Proposed System
      - Common Definition Language
    - System and Software Requirement
      - Project Requirements
      - Capability Requirements
      - System Interface Requirement
      - Level of Service Requirements
      - Evolution Requirements
      - Common Definition Language
    - System and Software Architecture I
      - Architectural Analysis
      - System Design
      - Common Definition Language
    - Life Cycle Plan
      - Milestones and Products
      - Responsibilities
      - Approach
      - Resources
    - Feasibility Rationale Description
      - Product Rationale
      - Process Rationale

#### Agents

- Participating Agent
  - Customer
  - User
  - Domain Expert
- Performing Agent
  - Project Manager
  - Architect
  - System Analyst
  - Designer
  - Developer

Reset Process Explorer



# MBASE Electronic Process Guide (2)

Activity: MBASE 577a Process - Netscape

File Edit View Go Communicator Help

**MBASE 577 Process Guide**

PROCESS EXPLORER GLOSSARY BEHAVIORAL DIAGRAMS FUNCTIONAL DIAGRAMS GUIDE HELP PROCESS HELP (C) 1999 Info CLOSE WINDOW

Activities:

- MBASE 577a Process
  - Inception Phase
  - Elaboration Phase
  - Record Project Effort
  - Methods for MBASE

**MBASE 577a Process**

**Overview**

The process followed by students of CS 577a for Digital Library projects

**Purpose**

The objectives for the MBASE 577a process are:

- To define a life cycle process for the students of CS 577a for use in Digital Library and similar projects
- To provide guidance to process enactors about the inter-dependence of MBASE process elements

**Decomposition**

The activity MBASE 577a Process is decomposed into the following:

- Inception Phase
- Elaboration Phase
- Record Project Effort

**Description**

Model-Based (System) Architecting and Software Engineering (MBASE) is an approach for developing software intensive systems. The MBASE approach integrates the four common development models (success, product, process and property) around the creation and use of a software architecture package. Using MBASE, model clashes can be recognized and reconciled as a matter of fact rather than after the fact.

The MBASE 577 process is a variant of the MBASE framework used by students in the course CS 577 offered at USC. The process consists of four distinct phases Inception, Elaboration, Construction and Transition. Each phase is completed with a commitment from the stakeholders during a review.

The tasks to be performed during the MBASE 577 process are:

- Identifying and resolving the critical risks
- Analyzing the problem domain
- Identifying the feasible architecture
- Developing a prototype to test the validity of the architecture and satisfaction of stakeholders' concerns
- Understanding the requirements of the life cycle and obtaining concurrence from the system stakeholders

**Tools and Techniques**

Electronic Process Guide for MBASE

Guidelines for Model-Based Architecting and Software Engineering (MBASE) deliverables: Inception and Elaboration

**Pitfalls**

Activity: MBASE 577a Process

- Overview
- Purpose
- Decomposition
- Description
- Tools and Techniques
- Pitfalls

Artifact: Operational Concept Description - Netscape

File Edit View Go Communicator Help

**MBASE 577 Process Guide**

PROCESS EXPLORER GLOSSARY BEHAVIORAL DIAGRAMS FUNCTIONAL DIAGRAMS GUIDE HELP PROCESS HELP (C) 1999 Info CLOSE WINDOW

Artifacts:

- ring Documents
  - Package
    - Operational Concept Description**
      - Domain Description
      - Proposed System
      - Common Definition Language for System and Software Requirements Description and Software Architecture Description
      - Life Cycle Plan
      - Feasibility Rationale Description
      - Prototype
      - WinWin Negotiation Report
      - System and Software Architecture Description
      - System and Software Requirements Description
      - Effort Reports
      - Structure Patterns

**Operational Concept Description**

**Overview**

Provides the overall context of the proposed system and its operational concept

**Purpose**

- Describe the overall context of the system to be developed, why it's being built, what exists now, and where the project is starting from
- Describe to the stakeholders of the system to be developed ('developed' is meant to include such terms as 'enhanced', 'updated', 're-engineered', 'automated'), how the system will work in practice once it is deployed
- Enable the operational stakeholders to evolve knowledgeably from their current operational concept to the new operational concept, and to collaboratively adapt the operational concept as developments arise, to make clear the value of developing the new system

**Owner**

The artifact Operational Concept Description is owned by the agent System Analyst .

**Decomposition**

The artifact Operational Concept Description is decomposed into the following:

- Domain Description
- Proposed System
- Common Definition Language for Domain

**Description**

**OCD Audience and Participants**

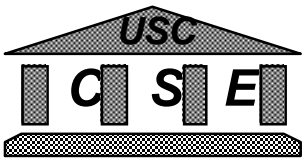
- Audience
  - Customer for Domain Description Domain
  - Domain Expert for System Analysis
  - Use language and define CDL appropriately for intended audience
- Participants
  - Same stakeholders as WinWin negotiation
  - Establish concept of operation agreed on by all stakeholders

**OCD High-Level Dependencies**

- WinWin Negotiations Give
- System Responsibilities
- Changes Considered But Not Included
- Domain Description Terms
- Project Goals, Quality Goals
- OCD Yields
  - Project, System and Quality Reqs for SSRD
  - Domain Description and Initial Analysis for SSAD
  - Stakeholder and Organizational Responsibilities
  - Business Case Analysis appropriate for CDL

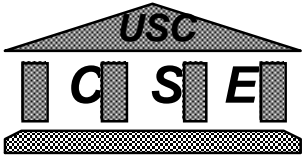
Artifact: **Operational Concept Description**

- Overview
- Purpose
- Owner
- Decomposition
- Description
- Storage Location
- Periodicity
- Flows
- Behavior
- Behavior Notes
- Retention Period
- Templates
- Examples



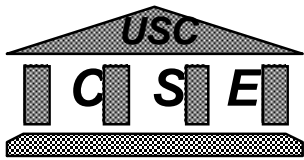
## MBASE Invariants and Variants

<b>Invariants</b>	<b>Variants</b>
<ol style="list-style-type: none"><li data-bbox="317 464 915 594">1. Defining and sustaining a stakeholder win-win relationship through the system's life-cycle.</li><li data-bbox="317 659 842 740">2. Using the MBASE Model Integration Framework.</li><li data-bbox="317 805 879 886">3. Using the MBASE Process Integration Framework.</li><li data-bbox="317 1000 926 1081">4. Using the LCO, LCA, and IOC Anchor Point milestones.</li><li data-bbox="317 1154 999 1276">5. Ensuring that the content of MBASE artifacts and activities is risk-driven.</li></ol>	<ol style="list-style-type: none"><li data-bbox="1100 464 1766 545">1. Use of particular success, process, product, or property models.</li><li data-bbox="1100 643 1703 724">2. Choice of process or product representation.</li><li data-bbox="1100 797 1671 919">3. Degree of detail of process, product, property, or success modeling.</li><li data-bbox="1100 992 1766 1073">4. Number of spiral cycles or builds between anchor points.</li><li data-bbox="1100 1146 1734 1268">5. Mapping of activities onto Inception-Elaboration-Construction-Transition phases.</li><li data-bbox="1100 1341 1671 1422">6. Mapping of staff levels onto activities.</li></ol>



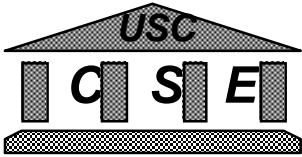
# MBASE Overview Outline

- **Future software trends and challenges**
- **Relations to spiral model and MBASE**
- **MBASE principles and key practices**
- ➔ **– Relation to new CMMI process areas**
  - Relation to Benefits Realization**
- **MBASE project usage experience**



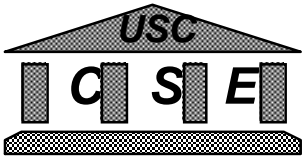
# MBASE and New CMMI Process Areas - I

- **Shared vision; collaborative leadership; integrated team**
  - **Stakeholder win-win management (Theory W)**
  - **WinWin spiral model**
  - **Easy WinWin groupware tool**
  - **DMR Benefits Realization Approach**



# WinWin Definition

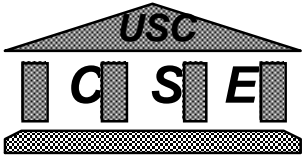
- **The win-win approach is a set of principles, practices, and tools**
  - which enable a set of interdependent *stakeholders*
  - to work out a *mutually satisfactory* (win-win)
  - set of *shared commitments*.



# Win-lose Generally Becomes Lose-lose

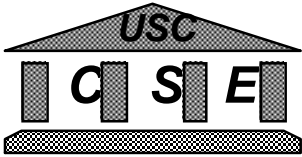
<b>Proposed Solution</b>	<b>“Winner”</b>	<b>Loser</b>
<b>Quick, Cheap, Sloppy Product</b>	<b>Developer &amp; Customer</b>	<b>User</b>
<b>Lots of “bells and whistles”</b>	<b>Developer &amp; User</b>	<b>Customer</b>
<b>Driving too hard a bargain</b>	<b>Customer &amp; User</b>	<b>Developer</b>

**Actually, nobody wins in these situations**

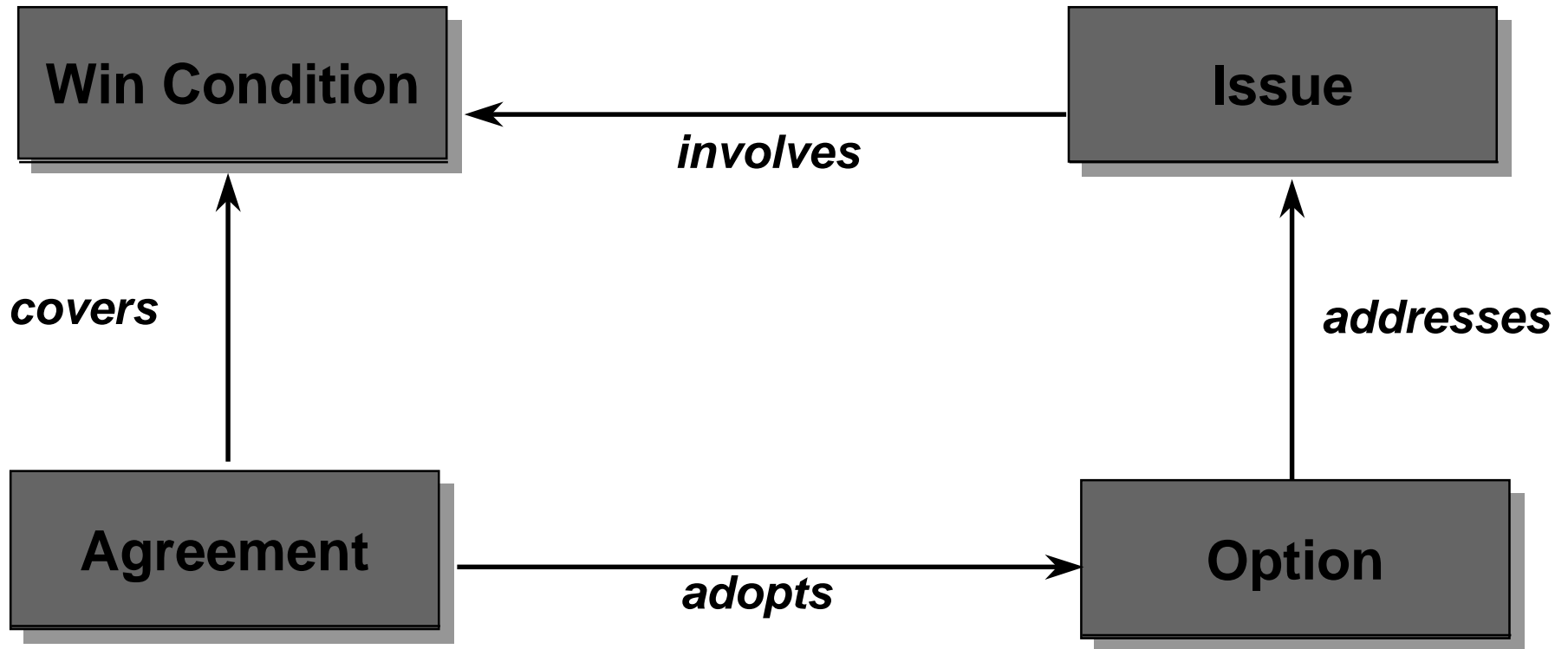


# Key Concepts

- **Win Condition:** objective which makes a stakeholder feel like a winner
- **Issue:** conflict or constraint on a win condition
- **Option:** A way of overcoming an issue
- **Agreement:** mutual commitment to an option or win condition

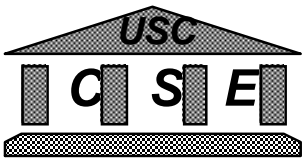


# WinWin Negotiation Model



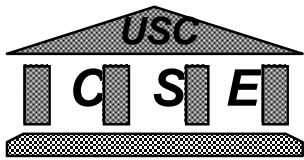
## WinWin Equilibrium State

- All Win Conditions covered by Agreements
- No outstanding Issues



## Why Use WinWin ?

- **The alternatives don't work**
  - Win-lose often leads to lose-lose
- **Avoids costly rework**
  - 100X cost to fix requirements after delivery
- **Builds trust and manages expectations**
  - Looking out for other's needs builds trust
  - Balancing needs leads to realistic expectations
- **Helps stakeholders adapt to change**
  - Shared vision and the flexibility of quick re-negotiation



# Easy WinWin Tool Support



## Review and Expand Negotiation Topics (Group Outliner)

Jointly review and define the scope of the negotiation. Identify the negotiation topics for your EasyWinWin activity.



## Brainstorm Stakeholder Interests (Electronic Brainstorming)

Collect ideas about Win Conditions for your EasyWinWin activity



## Converge on Win Conditions (Categorizer)

Jointly craft and organize a succinct list of win conditions.



## Capture Glossary of Terms (Topic Commenter)

Define important terms of the domain.



## Prioritize Win Conditions (Alternative Analysis)

Determine the business importance and the ease of implementation of all win conditions.  
Reveal issues and constraints.



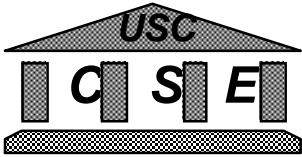
## WinWin Tree (Group Outliner)

Identify Issues and Options. Negotiate Agreements.



## Organize Negotiation Results (Categorizer)

Categorize the results using the negotiation topics.

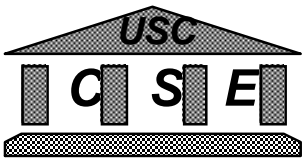


# The Information Paradox (Thorp)

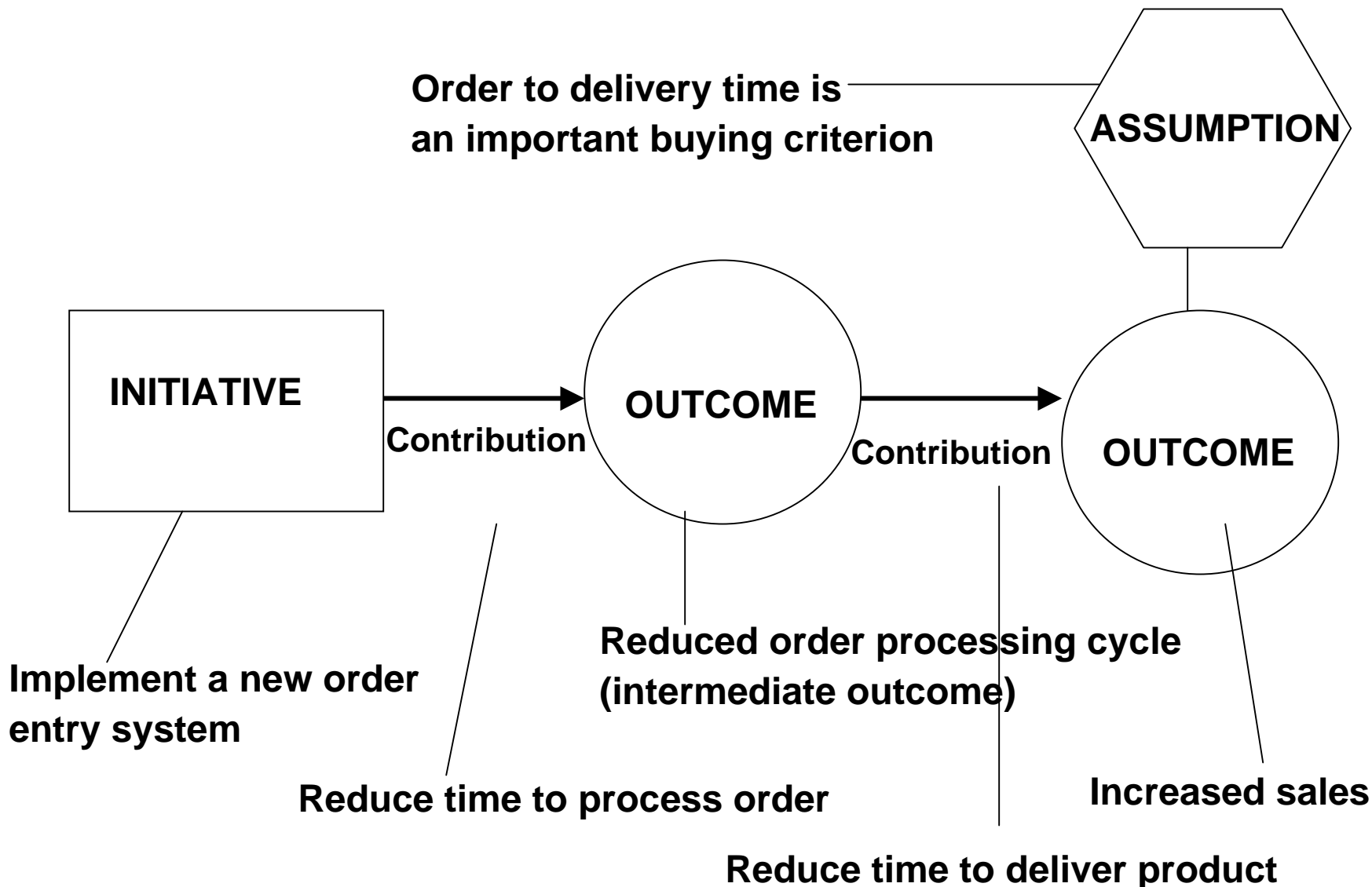
- **No correlation between companies' IT investments and their market performance**

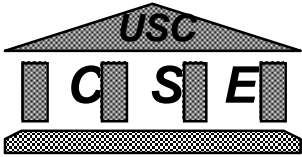


- **Field of Dreams**
  - **Build the (field; software)**
  - **and the great (players; benefits) will come**
- **Need to integrate software and systems initiatives**



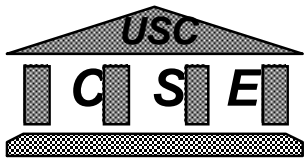
# DMR/BRA Results Chain





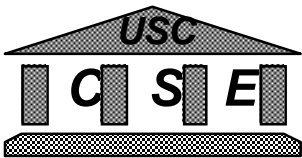
## MBASE Usage Examples

- **Digital Library Applications**
  - Over 100 real-client, rapid-development web and multimedia applications
- **CCPDS-R**
  - Over 1M lines of Ada for mission-critical command and control application

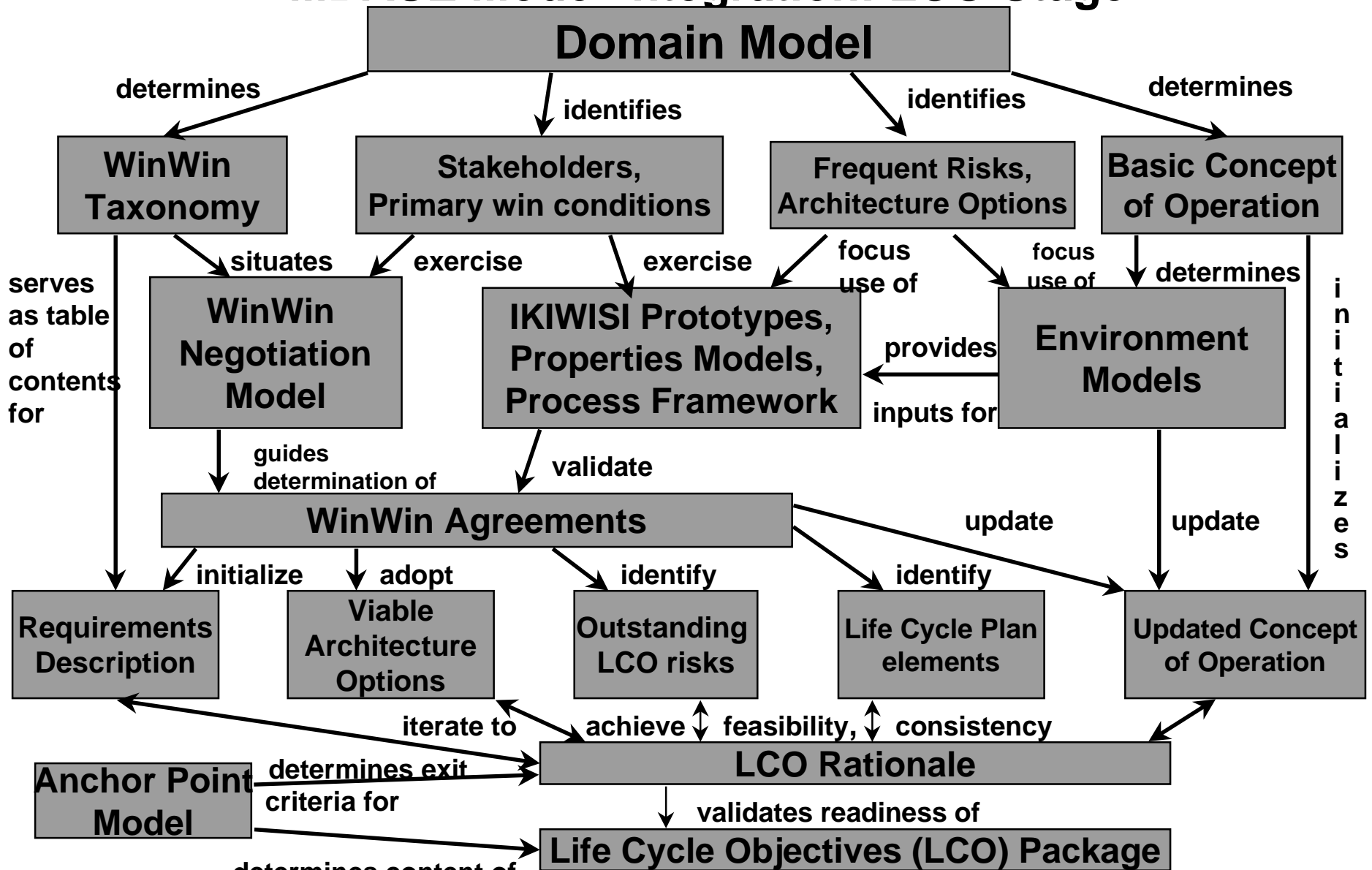


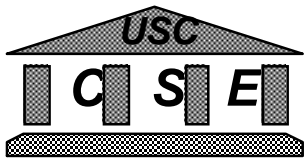
# The Challenge

- **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 1-week beta test



# 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/](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](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.

Document: Done

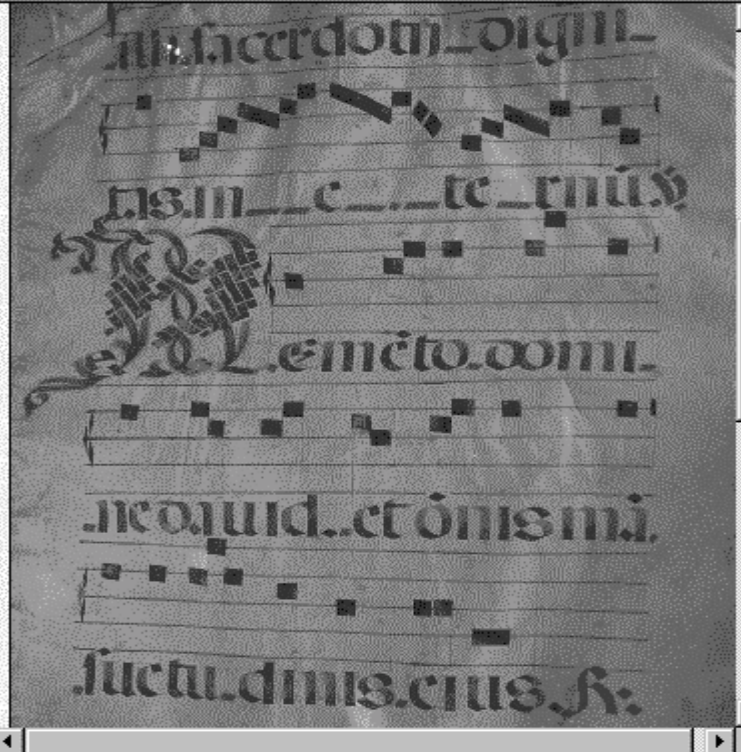
**Digital Manuscript Archive Display Leaf Image Page - Netscape**

File Edit View Go Communicator Help

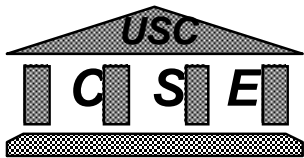
Back Forward Reload Home Search Guide Print Security Stop

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

Internet New and Cool Look Up

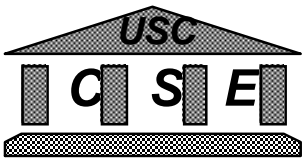


Document: Done



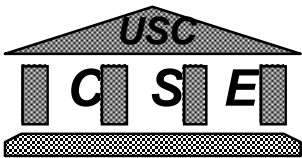
## MBASE Project Experience at USC/Columbia

Metric	USC 1996-97	USC 1997-98	USC 1998-99	Columbia U-grad. 99	Columbia Grad. 99
<b>Fall Semester: LCA Package</b>					
Teams	15	16	20	20	13
Students	86	80	102	107	59
Applications	12	15	17	10	10
Teams failing LCO review	4	4	1	10	6
Teams failing LCA review	0	0	0	0	1
Pages, LCO package	160	103	114	124	116
Pages, LCA package	230	154	167	142	142
Client Evaluation (1-5, 5 best)	4.46	4.67	4.74	-	-
<b>Spring Semester: IOC Package</b>					
Teams	6	5	6	Remained the same since projects were only one semester long	
Students	28	23	28		
Applications	8	5	6		
Teams failing IOC acceptance review	0	0	0	0	0
Applications satisfying clients	5	5	6	20*	12*
Applications not overtaken by events	6	4	4	10	9
Applications continued	3	3	TBD	-	-
Applications used	1	3	TBD	10	5
Client evaluation	-	4.15	4.3	4.44	4.21

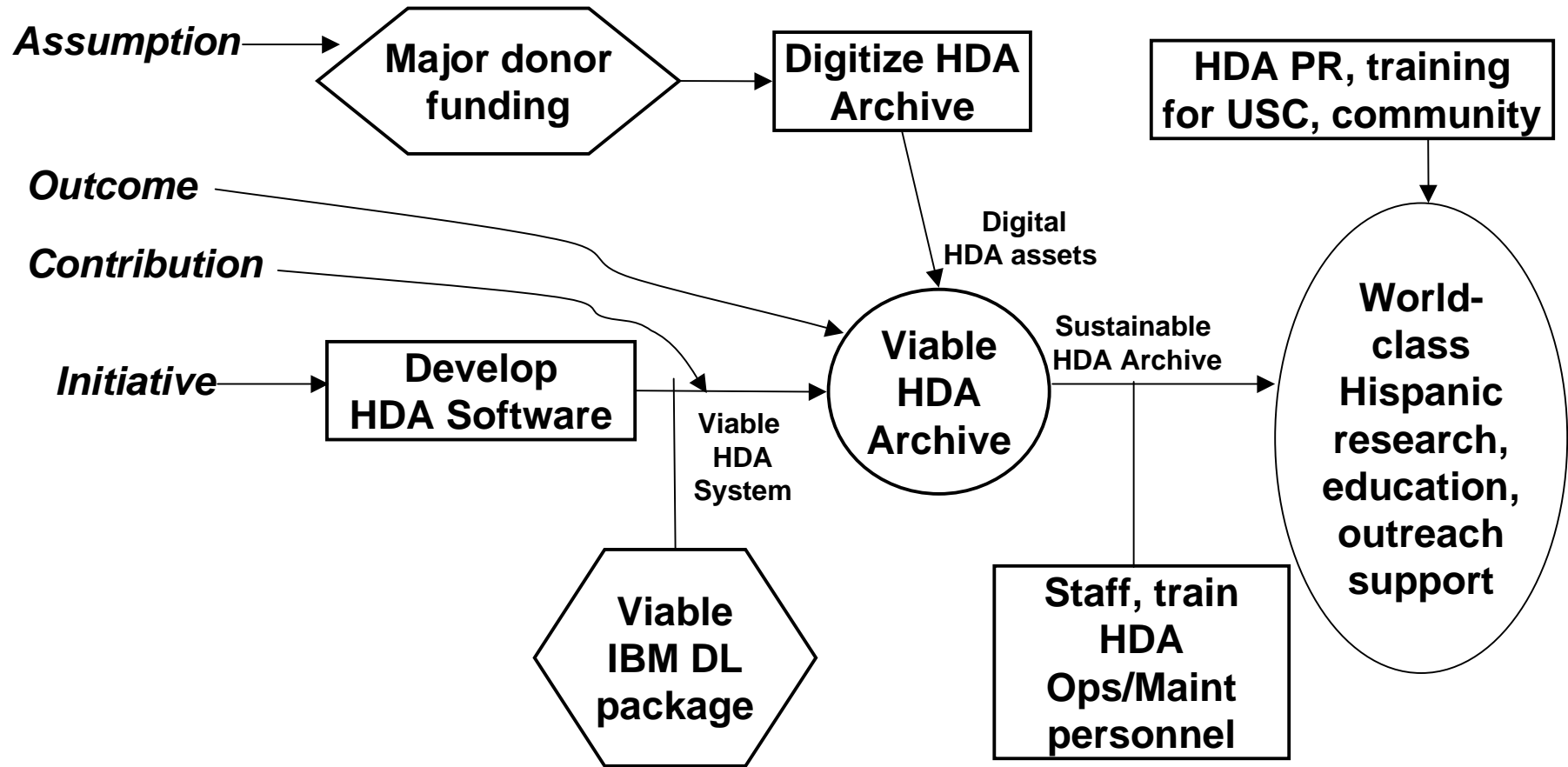


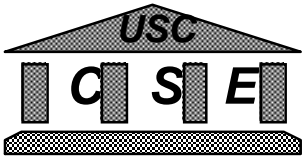
# Critical Success Factors for Adoption

Application	Client Characteristics					Transition Preparation			Outcome		
	Focused	Representative	O & M Resources	Collaborative	Domain Knowledge	Software	Site	People	Stable Envir.	Client Success	Adopted
1996-97											
EDGAR Business Data	+	+		+	+	+			+	+	
Medieval Manuscripts	+			+	+	+			+	+	
Technical Reports	+	+		+	+	+				+	
Latin American Pamphlets	+	+		+	+	+			+	+	
Cinema-TV	+	+	+	+	+	+	+	(+)		+	(+)
Image Archives				+		+			+		
1997-98											
S-Charts	+	+	+	+	+	+	(+)	+	(+)	+	(+)
Global Express	+	+	+	+	+	+		+		+	
Hancock Virtual Museum	+	+	(+)	+	+	+	+	+		+	
Serial Control Records	+	+	+	+	+	+	+	+	(+)	+	(+)
B-School Working Papers	+	+	+	+	+	+	+	+	+	+	+
1998-99											
Data Mining	+	+	+	+	+	+	+	+	(+)	+	(+)
Dissertations	+	+	+	+	+	+	+	(+)	+	+	(+)
Hispanic Archive	+	+	+	+	+	+	+	+		+	
WWI Archive	+	+	+	+	+	+	+	+		+	(+)



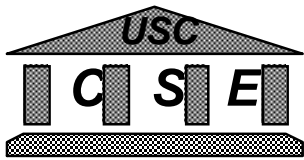
# Results Chain: Hispanic Digital Archive (HDA)





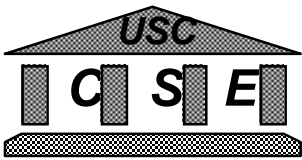
# Case Study: CCPDS-R Project Overview

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

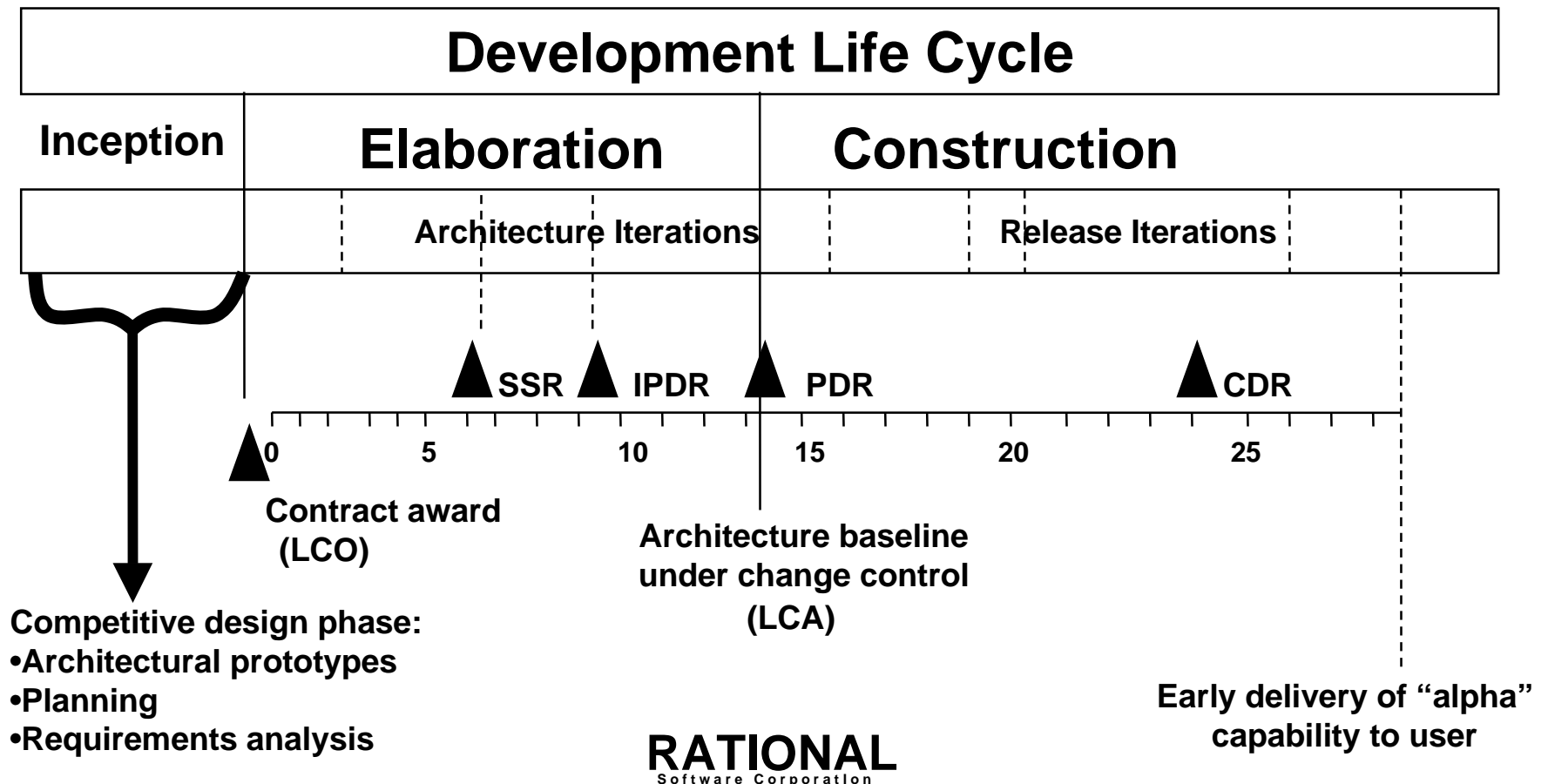


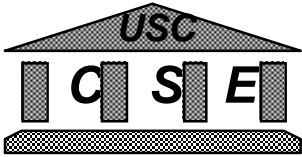
# CCPDS-R MBASE Models

- **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



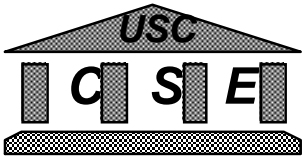
# Common Subsystem Macroprocess



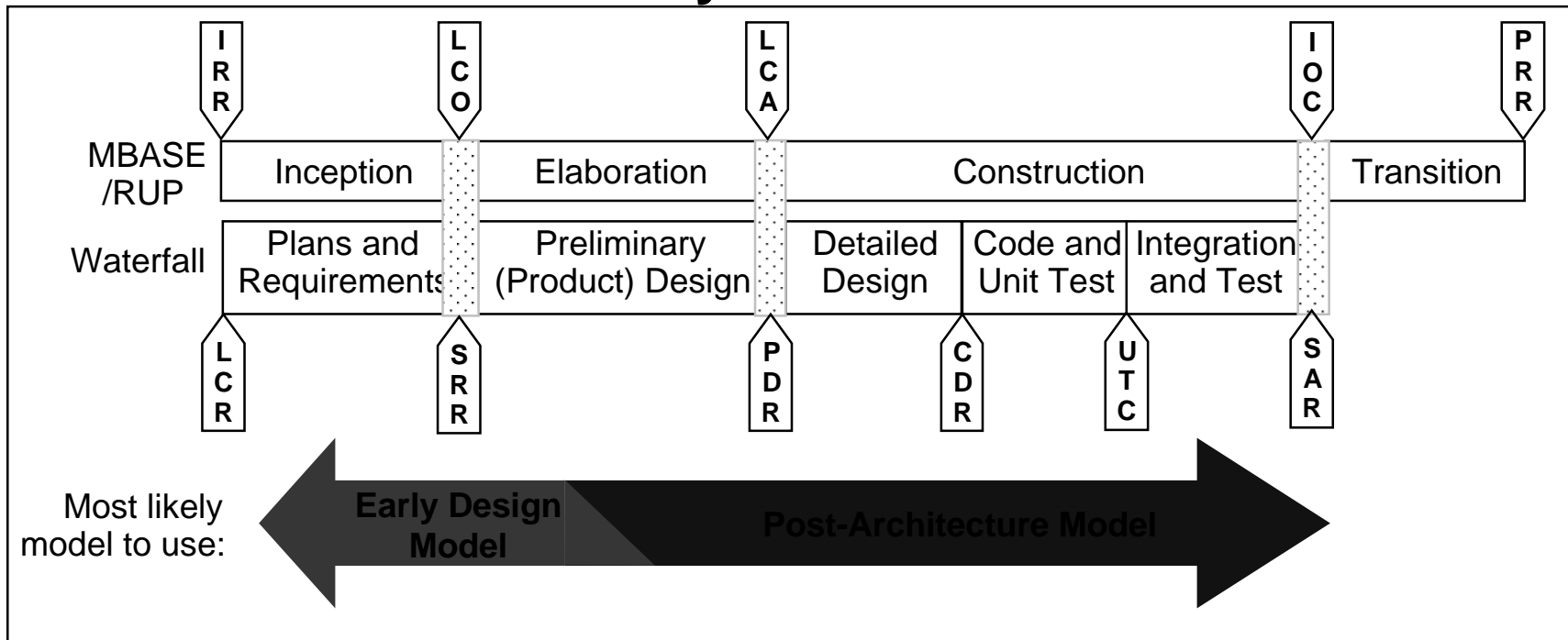


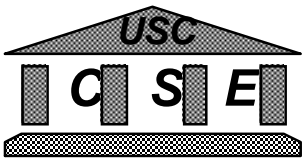
# Outline

- **Future software trends and challenges**
- **Relations to spiral model, COCOMO II and MBASE**
- **MBASE principles and key practices**
  - Relation to new CMMI process areas
  - Relation to Benefits Realization
- **MBASE project usage experience**
- ➔ • **COCOMO II MBASE/Rational Unified Process phase and activity distributions**



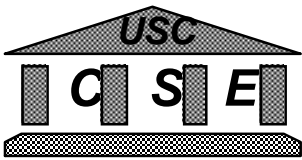
# Model-Based [System] Architecting and Software Engineering (MBASE) and Rational Unified Process (RUP) Life Cycle Phases





# MBASE & Rational Unified Process Milestones

1. *Inception Readiness Review (IRR)*
  - Candidate system objectives, scope, boundary
  - Key stakeholders identified
    - Committed to support Inception phase
  - Resources committed to achieve successful LCO package
2. *Life Cycle Objectives Review (LCO)*
3. *Life Cycle Architecture Review (LCA)*
4. *Initial Operational Capability (IOC)*
5. *Product Release Review (PRR)*



## MBASE & Rational Unified Process Milestones (cont.)

### 1. *Inception Readiness Review (IRR)*

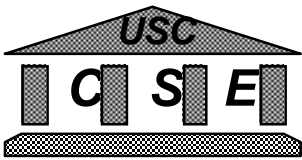
### 2. *Life Cycle Objectives Review (LCO)*

- Life Cycle Objectives (LCO) Package (see Table A.3)
  - Key elements of Operational Concept, Prototype, Requirements, Architecture, Life Cycle Plan, Feasibility Rationale
- Feasibility assured for at least one architecture, using the criteria:
  - Acceptable business case
  - A system developed from the architecture would support the operational concept, be compatible with the prototype, satisfy the requirements, and be buildable within the budgets and schedules in the life cycle plan.
- Feasibility validated by an Architecture Review Board (ARB)
  - ARB includes project-leader peers, architects, specialty experts, key stakeholders [Marenzano, 1995].
  - Key stakeholders concur on essentials, commit to support Elaboration phase
- Resources committed to achieve successful LCA package

### 3. *Life Cycle Architecture Review (LCA)*

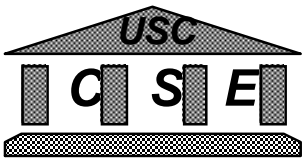
### 4. *Initial Operational Capability (IOC)*

### 5. *Product Release Review (PRR)*



## MBASE & Rational Unified Process Milestones (cont.)

1. *Inception Readiness Review (IRR)*
2. *Life Cycle Objectives Review (LCO)*
3. *Life Cycle Architecture Review (LCA)*
  - Life Cycle Architecture (LCA) Package (see Table A.3)
  - Feasibility assured for selected architecture (see above)
  - Feasibility validated by ARB
    - Stakeholders concur on their success-critical items, commit to support Construction, Transition, and phases maintenance.
    - All major risks resolved or covered by risk management plan
  - Resources committed to achieve Initial Operational Capability (IOC), life cycle support
4. *Initial Operational Capability (IOC)*
5. *Product Release Review (PRR)*



## MBASE & Rational Unified Process Milestones (cont.)

1. *Inception Readiness Review (IRR)*

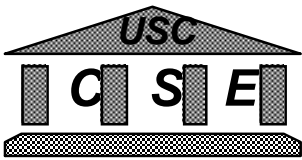
2. *Life Cycle Objectives Review (LCO)*

3. *Life Cycle Architecture Review (LCA)*

4. *Initial Operational Capability (IOC)*

- Software preparation, including both operational and support software with appropriate commentary and documentation; initial data preparation or conversion; the necessary licenses and rights for COTS and reused software, and appropriate operational readiness testing.
- Site preparation, including initial facilities, equipment, supplies & COTS vendor support arrangements.
- Initial user, operator and maintainer preparation, including selection, team building, training and other qualification for familiarization usage, operations, or maintenance.
- Successful Transition Readiness Review
  - Plans, preparations for full conversion, installation, training, and operational cut-over
  - Stakeholders confirm commitment to support Transition and Maintenance phases

5. *Product Release Review (PRR)*



## MBASE & Rational Unified Process Milestones (cont.)

1. *Inception Readiness Review (IRR)*

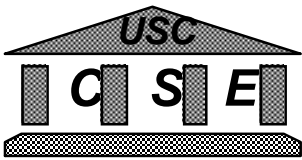
2. *Life Cycle Objectives Review (LCO)*

3. *Life Cycle Architecture Review (LCA)*

4. *Initial Operational Capability (IOC)*

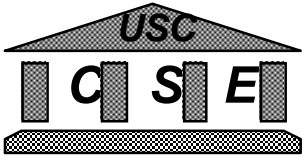
5. *Product Release Review (PRR)*

- Assurance of successful cutover from previous system for key operational sites
- Personnel fully qualified to operate and maintain new system
- Stakeholder concurrence that the deployed system operates consistently with negotiated and evolving stakeholder agreements
- Stakeholders confirm commitment to support Maintenance phase



## MBASE & Rational Unified Process MBASE and RUP Phase Distribution Percentages

Phase (endpoints)	MBASE		RUP	
	Effort% (range)	Schedule% (range)	Effort%	Schedule%
Inception (IRR to LCO)	6 (2-15)	12.5 (2-30)	5	10
Elaboration (LCO to LCA)	24 (20-28)	37.5 (33-42)	20	30
Construction (LCA to IOC)	76 (72-80)	62.5 (58-67)	65	50
Transition (IOC to TCR)	12 (0-20)	12.5 (0-20)	10	10
Totals:	118	125	100	100



# **COCOMO II MBASE/RUP Default Work Breakdown Structure**

## **A Management**

### **AA Inception phase management**

**AAA Top-level Life Cycle Plan (LCO version of LCP)**

**AAB Inception phase project control and status assessments**

**AAC Inception phase stakeholder coordination and business case development**

**AAD Elaboration phase commitment package & review (LCO package preparation & ARB review)**

### **AB Elaboration phase management**

**ABA Updated LCP with detailed Construction plan (LCA version of LCP)**

**ABB Elaboration phase project control and status assessments**

**ABC Elaboration phase stakeholder coordination and business case update**

**ABD Construction phase commitment package & review (LCA package preparation & ARB review)**

### **AC Construction phase management**

**ACA Updated LCP with detailed Transition and Maintenance plans**

**ACB Construction phase project control and status assessments**

**ACC Construction phase stakeholder coordination**

**ACD Transition phase commitment package & review (IOC package preparation & PRB review)**

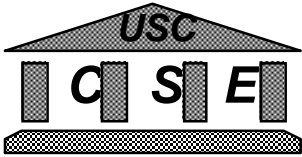
### **AD Transition phase management**

**ADA Updated LCP with detailed next-generation planning**

**ADB Transition phase project control and status assessments**

**ADC Transition phase stakeholder coordination**

**ADD Maintenance phase commitment package & review (PR package preparation & PRB review)**



# **COCOMO II MBASE/RUP Default Work Breakdown Structure (cont.)**

**A Management**

**B Environment and Configuration Management (CM)**

**BA Inception phase environment/CM scoping and initialization**

**BB Elaboration phase environment/CM**

**BBA Development environment installation and administration**

**BBB Elaboration phase CM**

**BBC Development environment integration and custom**

**toolsmithing**

**BC Construction phase environment/CM evolution**

**BCA Construction phase environment evolution**

**BCB Construction phase CM**

**BD Transition phase environment/CM evolution**

**BDA Construction phase environment evolution**

**BDB Transition phase CM**

**BDC Maintenance phase environment packaging and transition**

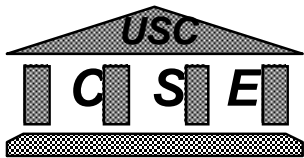
**C Requirements**

**D Design**

**E Implementation**

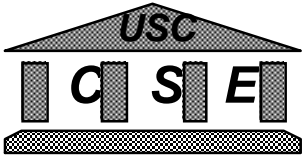
**F Assessment**

**G Deployment**



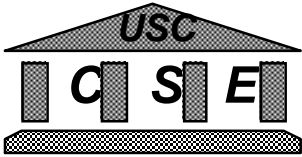
# COCOMO II MBASE/RUP Phase and Activity Distribution Values

	COCOMO II						
	Inception	Development		Transition	Total IECT		
		Elaboration	Construction		COCOMO II	Royce	
Rational schedule	10	30	50	10		100	
COCOMO II Schedule	12.5	37.5	62.5	12.5	125		
Rational Effort	5	20	65	10		100	
COCOMO II Effort	6	24	76	12	118		
Activity % of phase of IECT	100	100	100	100	118	100	
Management	14	12	10	14	13	12	
Environment/CM	10	8	5	5	7	12	
Requirements	38	18	8	4	13	12	
Design	19	36	16	4	22	18	
Implementation	8	13	34	19	32	29	
Assessment	8	10	24	24	24	29	
Deployment	3	3	3	30	7	6	



# Causes for Wide Variation

Factor	Inception	Transition
Complexity of LCO issues needing resolution	Very Large	Small
Involves major changes in stakeholder roles	Very Large	Large
Technical risk levels	Large	Some
Stakeholder trust levels	Large	Considerable
Heterogeneous stakeholder community	Large	Large
Degree of hardware/software integration	Large	Large
Complexity of transition from legacy system	Considerable	Large
Number and classes of installations	Some	Very Large



# References

**(MBASE material available at <http://sunset.usc.edu/MBASE>)**

**B. Blanchard, W. Fabrycky, Systems Engineering and Analysis (3rd ed.), Prentice Hall, 1998.**

**B. Boehm, D. Port, “When Models Collide: Lessons from Software Systems Analysis,” IEEE IT Professional, January/February 1999, pp. 49-56.**

**B. Boehm, D. Port, “Escaping the Software Tar Pit: Model Clashes and How To Avoid Them,” ACM Software Engineering Notes, January, 1999, pp. 36-48.**

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

**B. Boehm et al., “Developing Multimedia Applications with the WinWin Spiral Model,” Proceedings, ESEC/FSE 97, Springer Verlag, 1997.**

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

**J. Thorp and DMR, The Information Paradox, McGraw Hill, 1998.**