

University of Southern California
Center for Software Engineering

Value-Based Feedback in Software/IT Systems

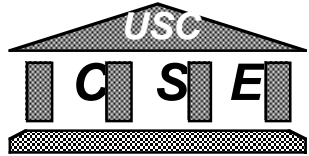
Barry Boehm, USC

FEAST/ProSim/SOCE 2000 Keynote

July 12, 2000

boehm@sunset.usc.edu

<http://sunset.usc.edu>



Outline

- **COCOMO II 4-Cycle feedback model**
- **Value-based feedback approach**
 - **DMR Benefits Realization Approach (BRA)**
 - **Implementing DMR-BRA via MBASE**
 - **Model-Based (System) Architecting and Software Engineering**
- **Example application: Y2K**
 - **Relation to Laws of Evolution anomalies**
- **Conclusions**

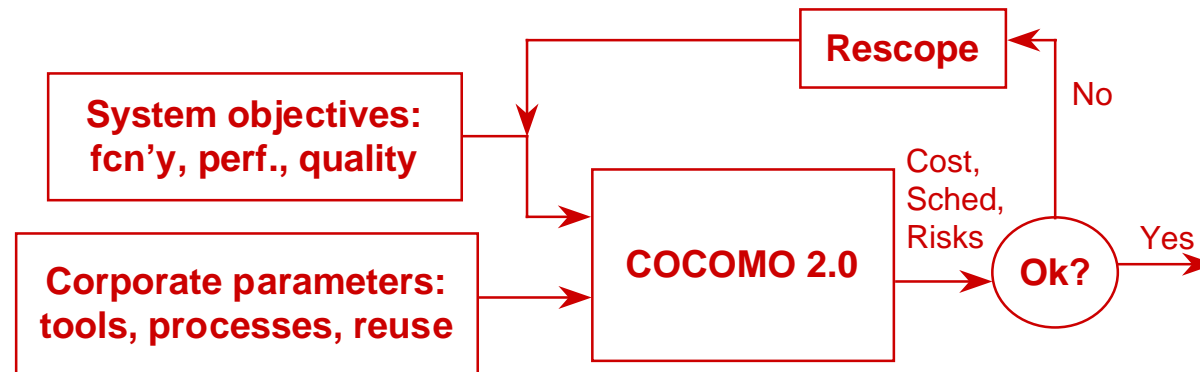


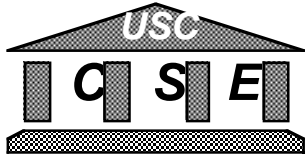
COCOMO II Overview

- **Re-engineering of 1981 Constructive Cost Model (COCOMO)**
 - Focused on 2000's software projects
 - New approaches to processes, scaling, reuse
- **Calibrated to 161 project data points**
 - Estimates within 30% of actuals, 75% of time
 - 80% of time when calibrated to organization
- **COCOMO II book due 1 August 2000 (Prentice Hall)**
 - Includes CD with Freeware model
 - Demo versions of 3 commercial versions

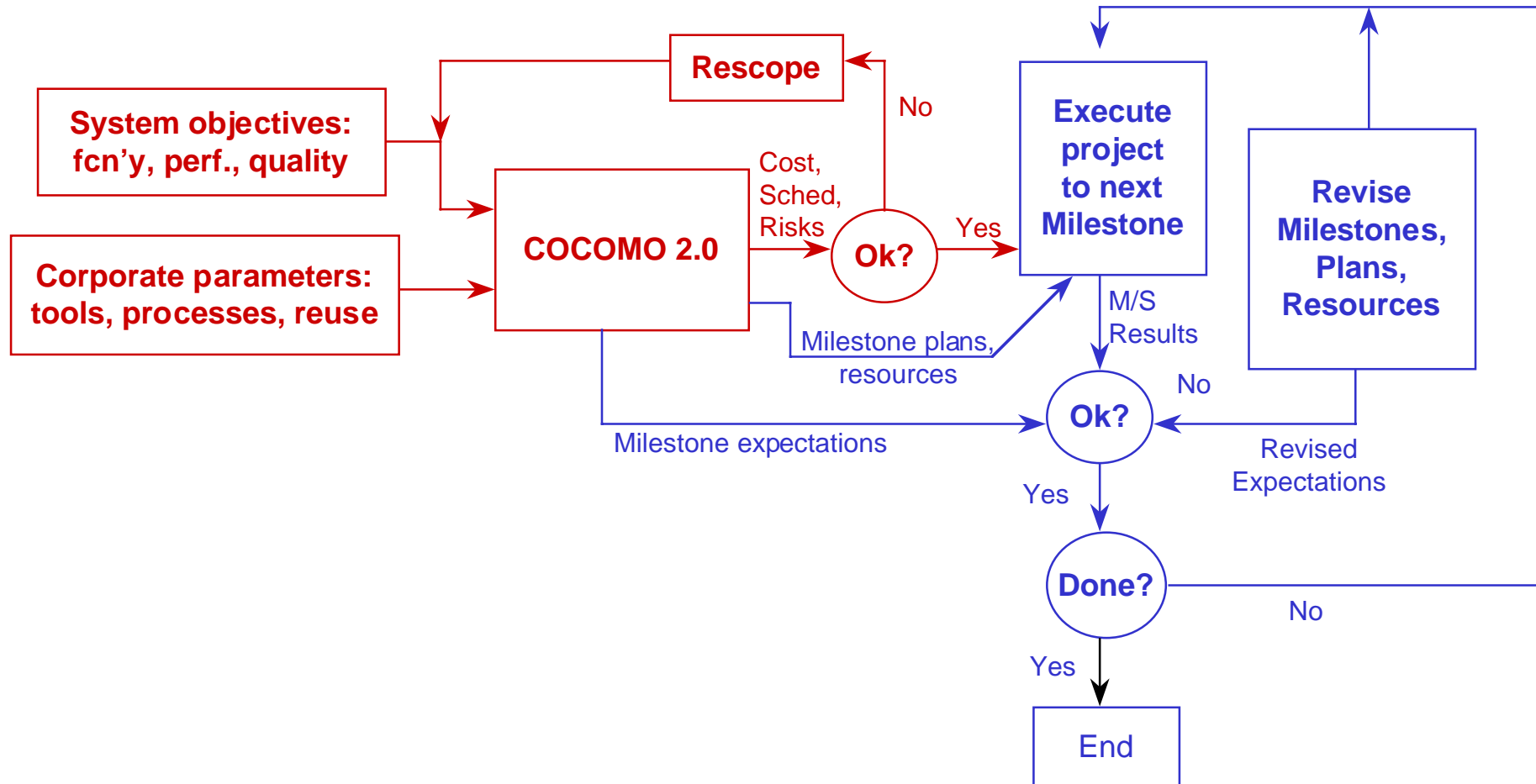


COCOMO II Long Term Vision



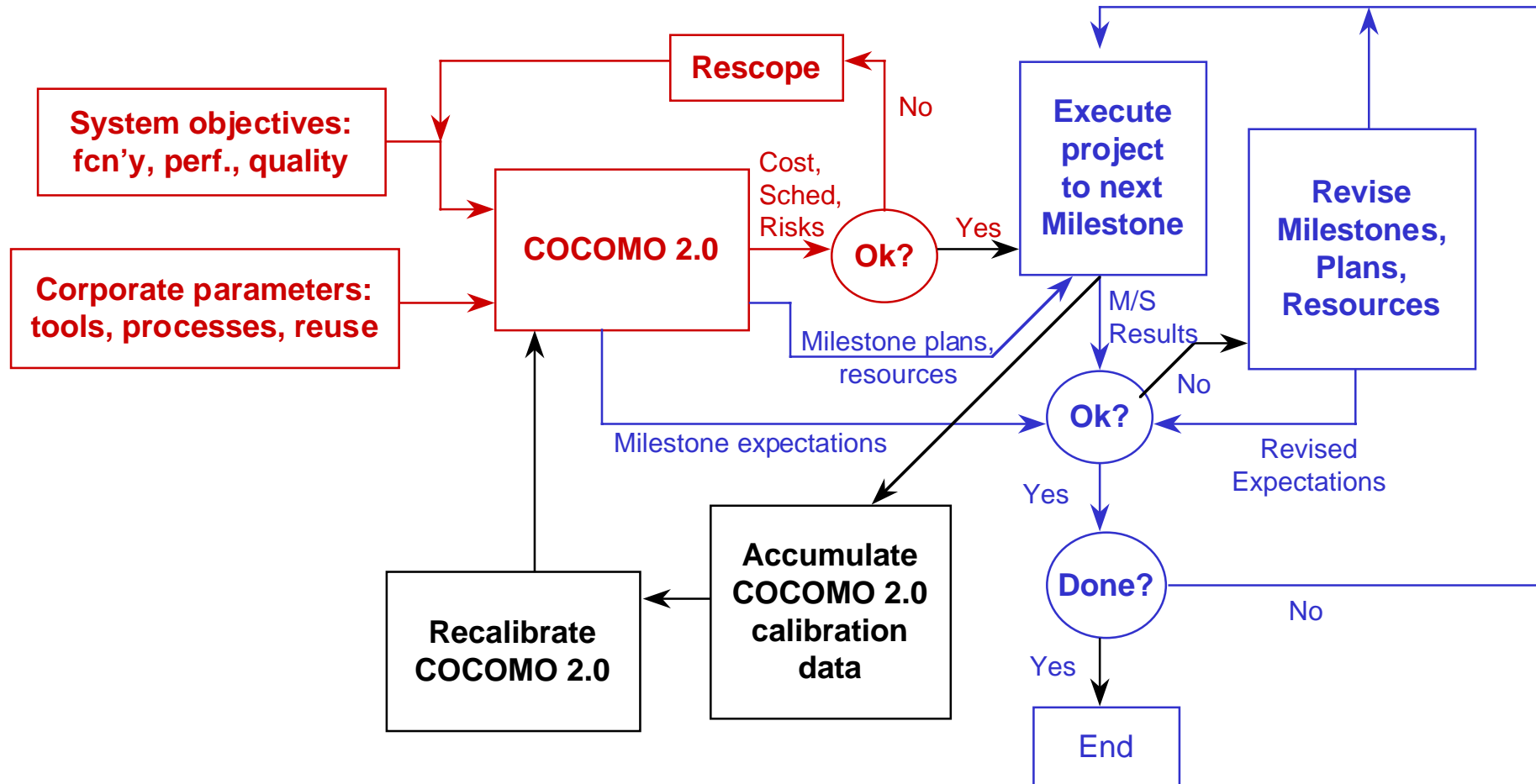


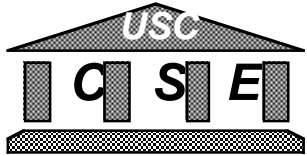
COCOMO II Long Term Vision



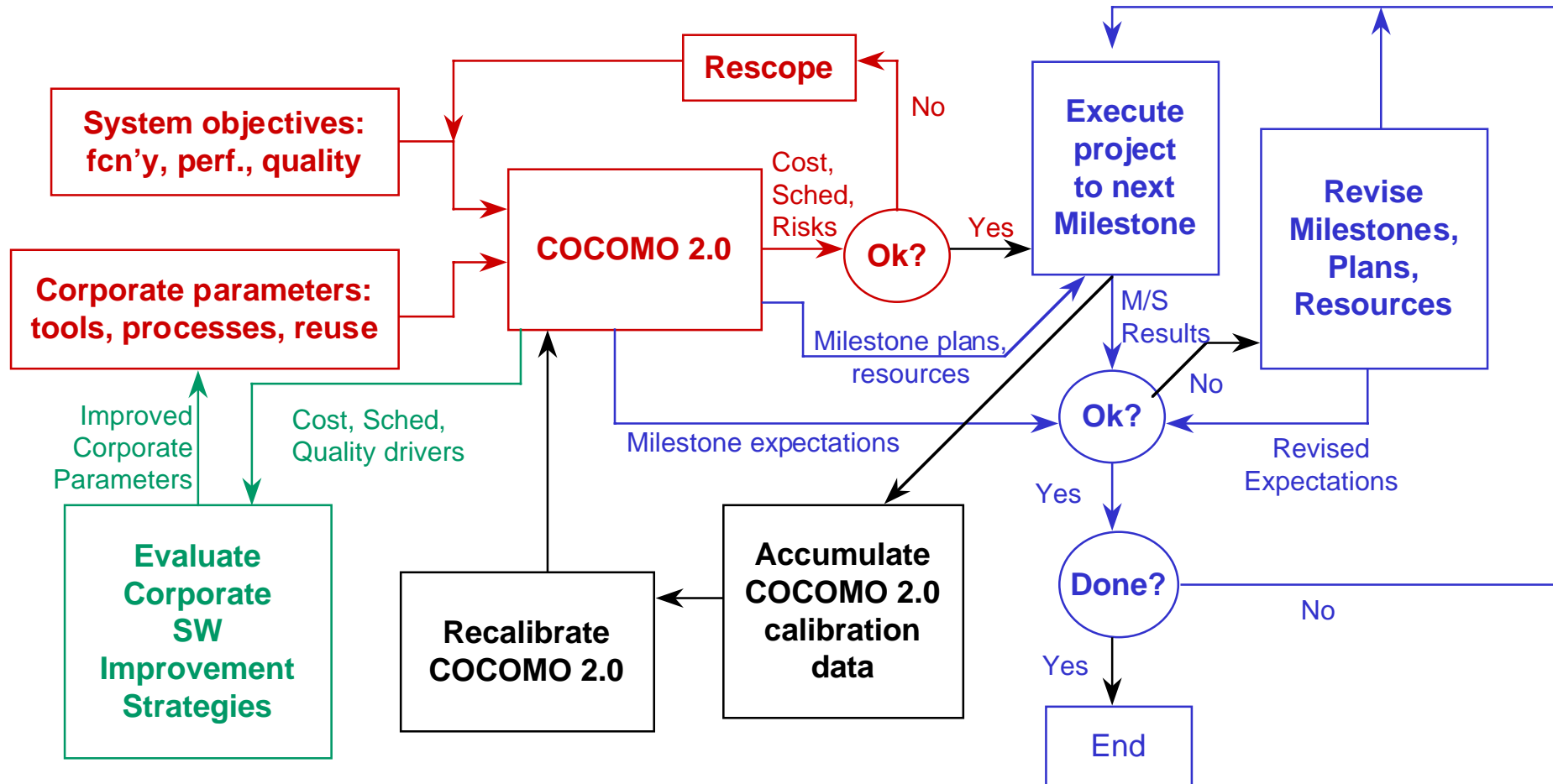


COCOMO II Long Term Vision



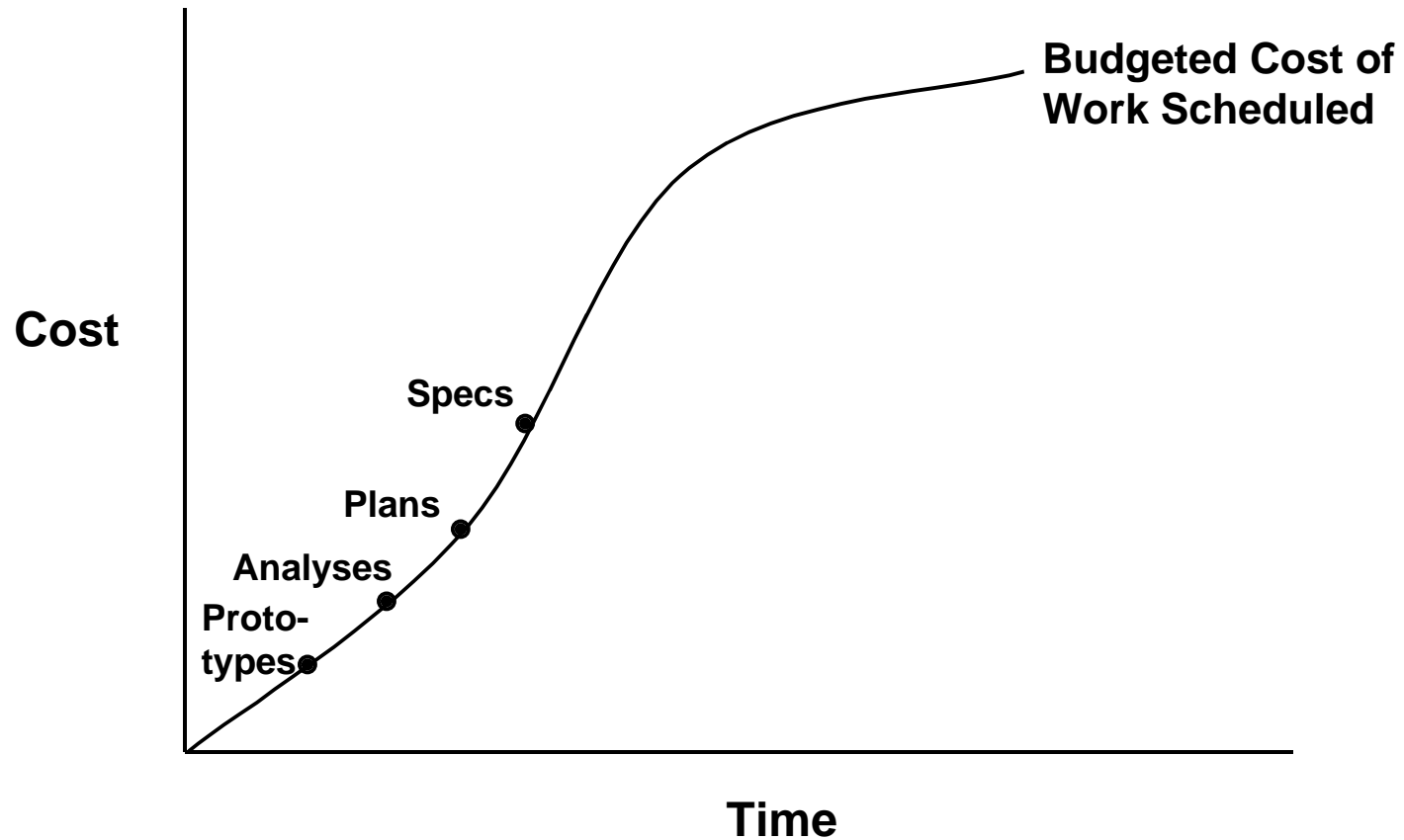


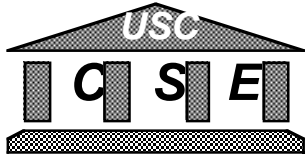
COCOMO II Long Term Vision



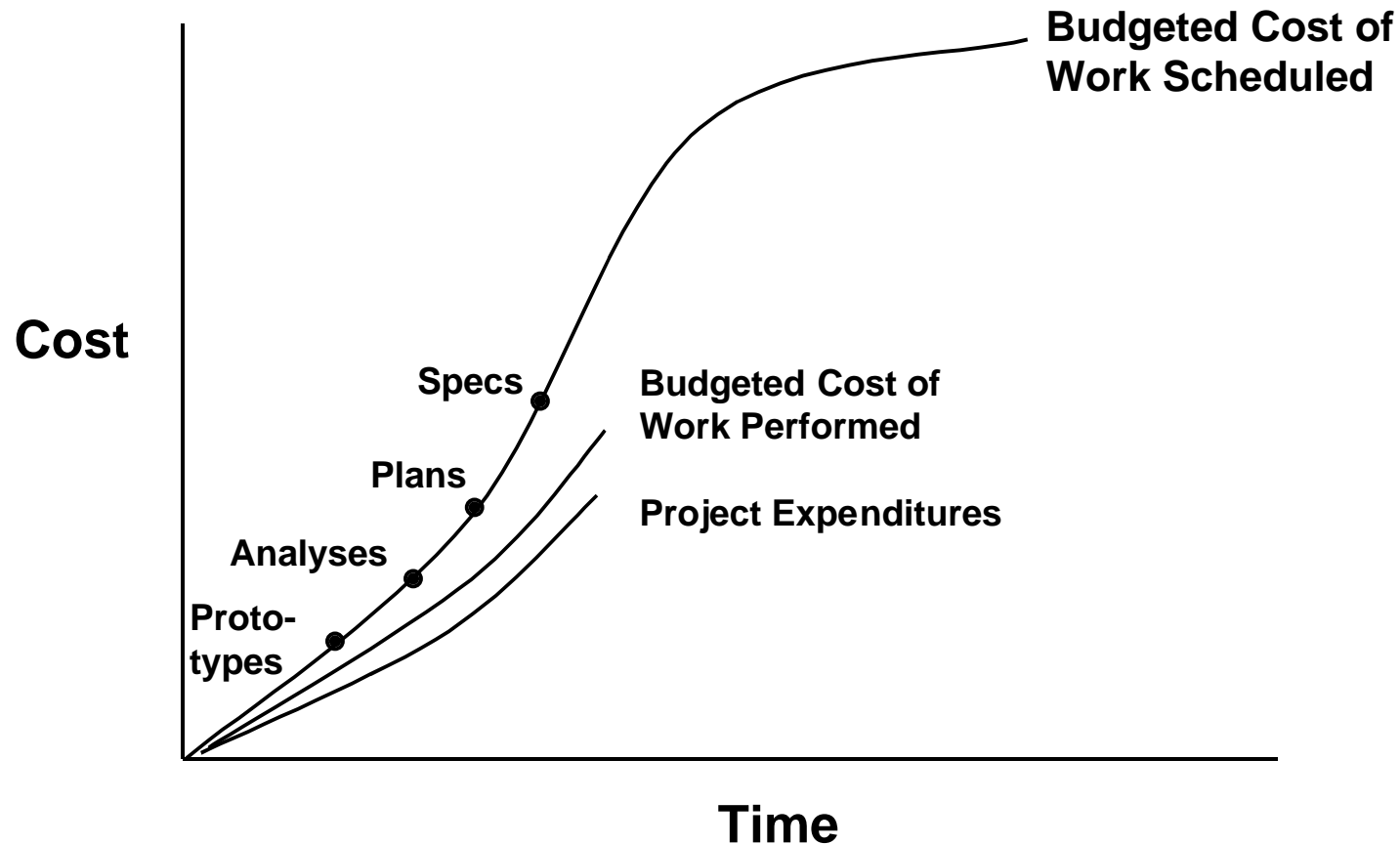


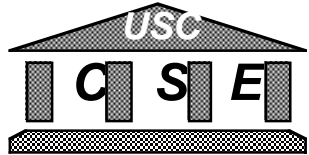
Earned Value System



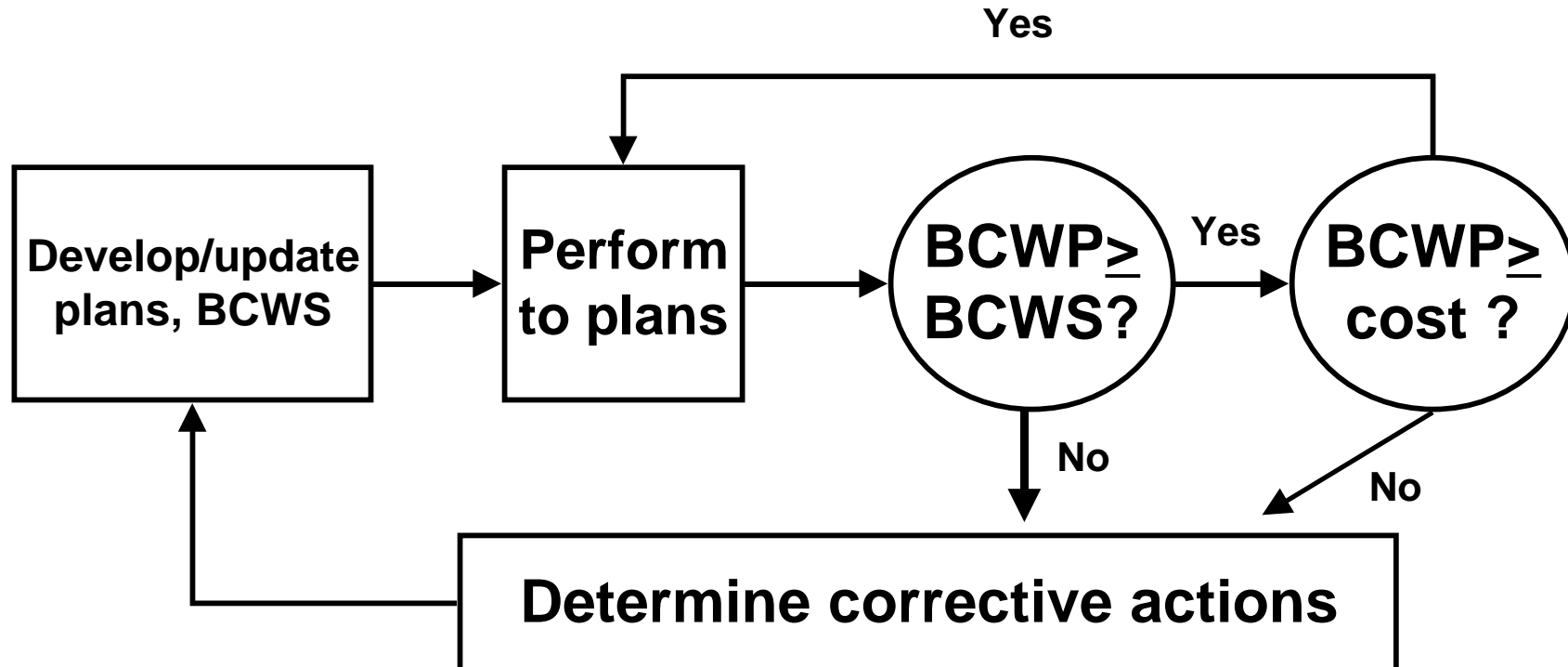


Earned Value System





Earned Value Feedback Process



- BCWS: Budgeted Cost of Work Scheduled
- BCWP: Budgeted Cost of Work Performed

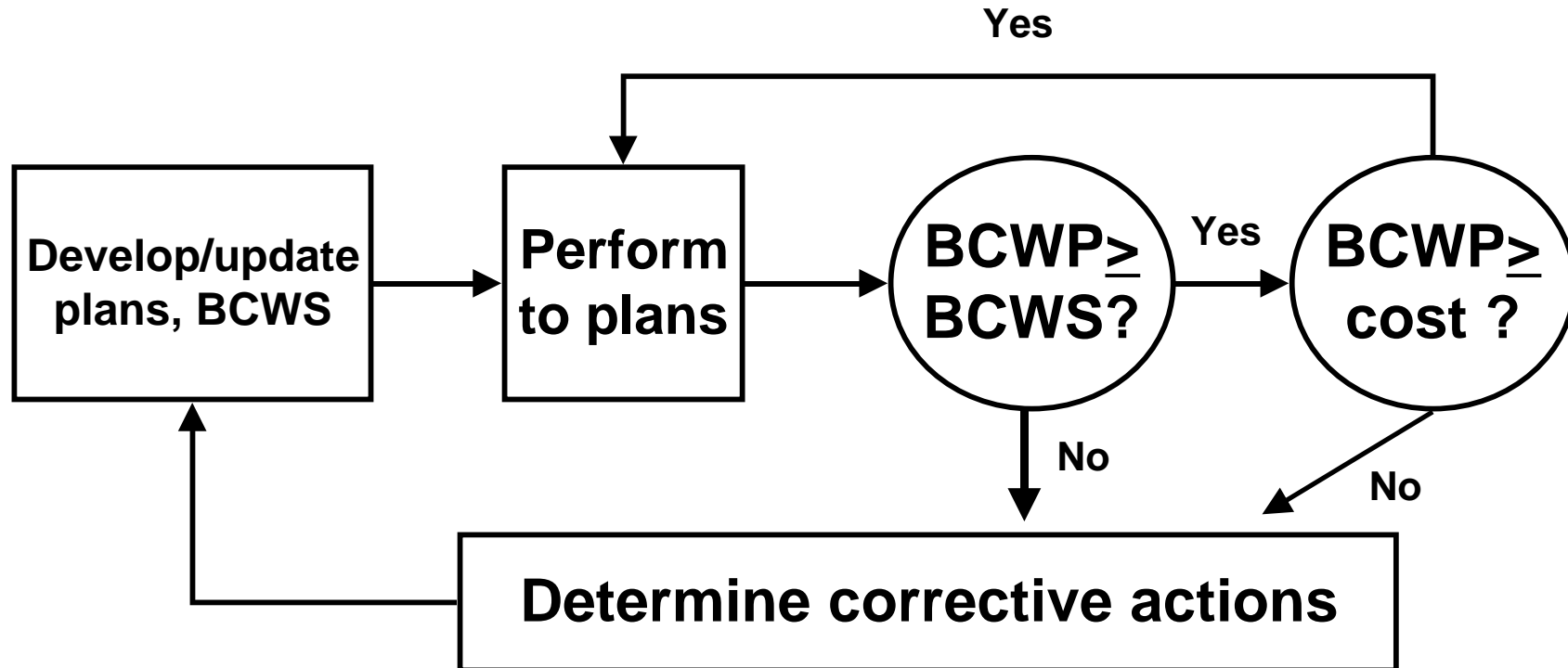


Outline

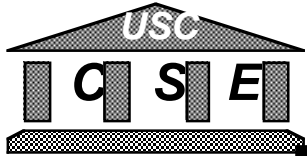
- **COCOMO II 4-Cycle feedback model**
- • **Value-based feedback approach**
 - **DMR Benefits Realization Approach (BRA)**
 - **Implementing DMR-BRA via MBASE**
 - **Model-Based (System) Architecting and Software Engineering**
- **Example application: Y2K**
 - **Relation to Laws of Evolution anomalies**
- **Conclusions**



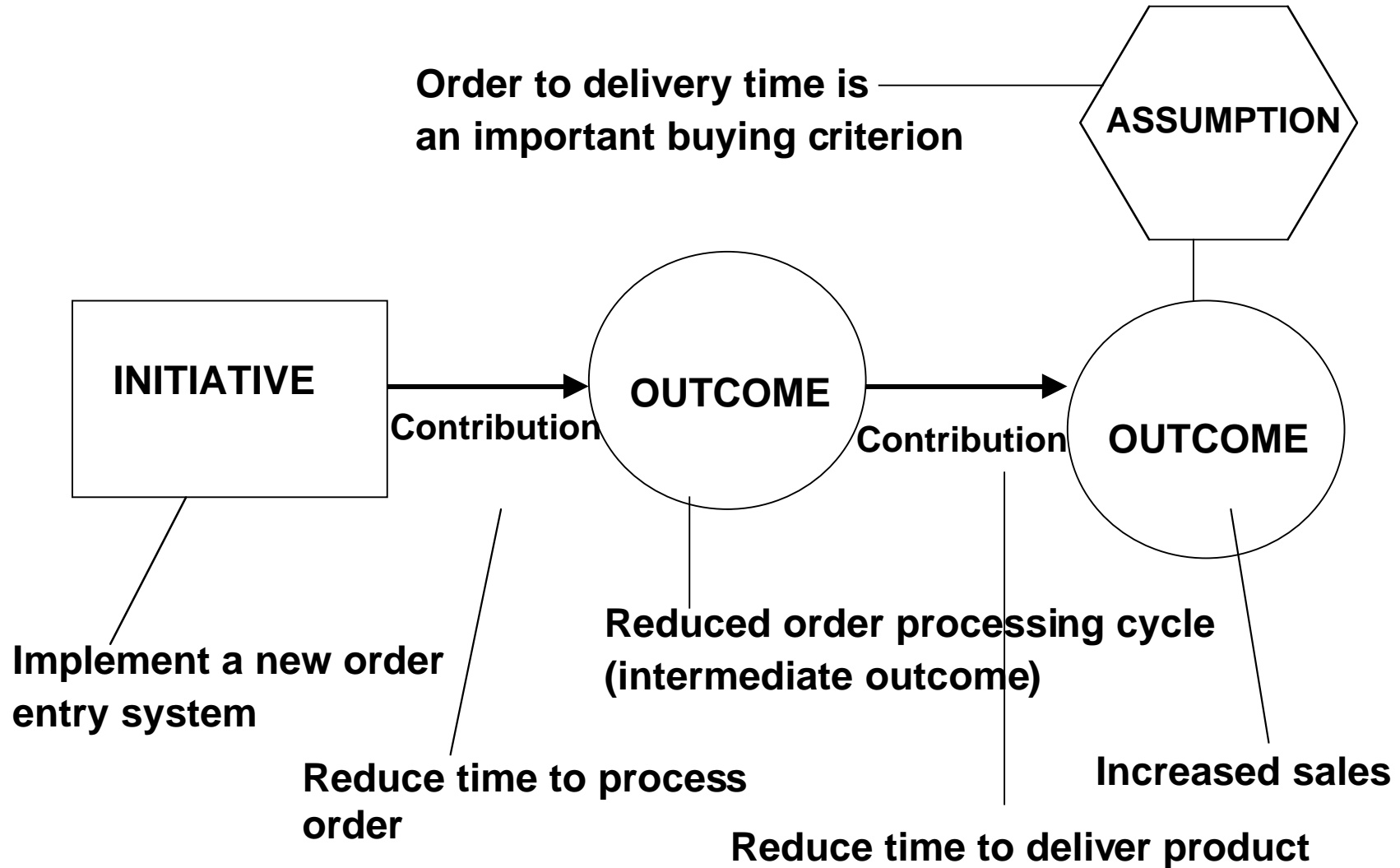
“Earned Value” Tracks Cost, Not Value



- BCWS: Budgeted Cost of Work Scheduled
- BCWP: Budgeted Cost of Work Performed

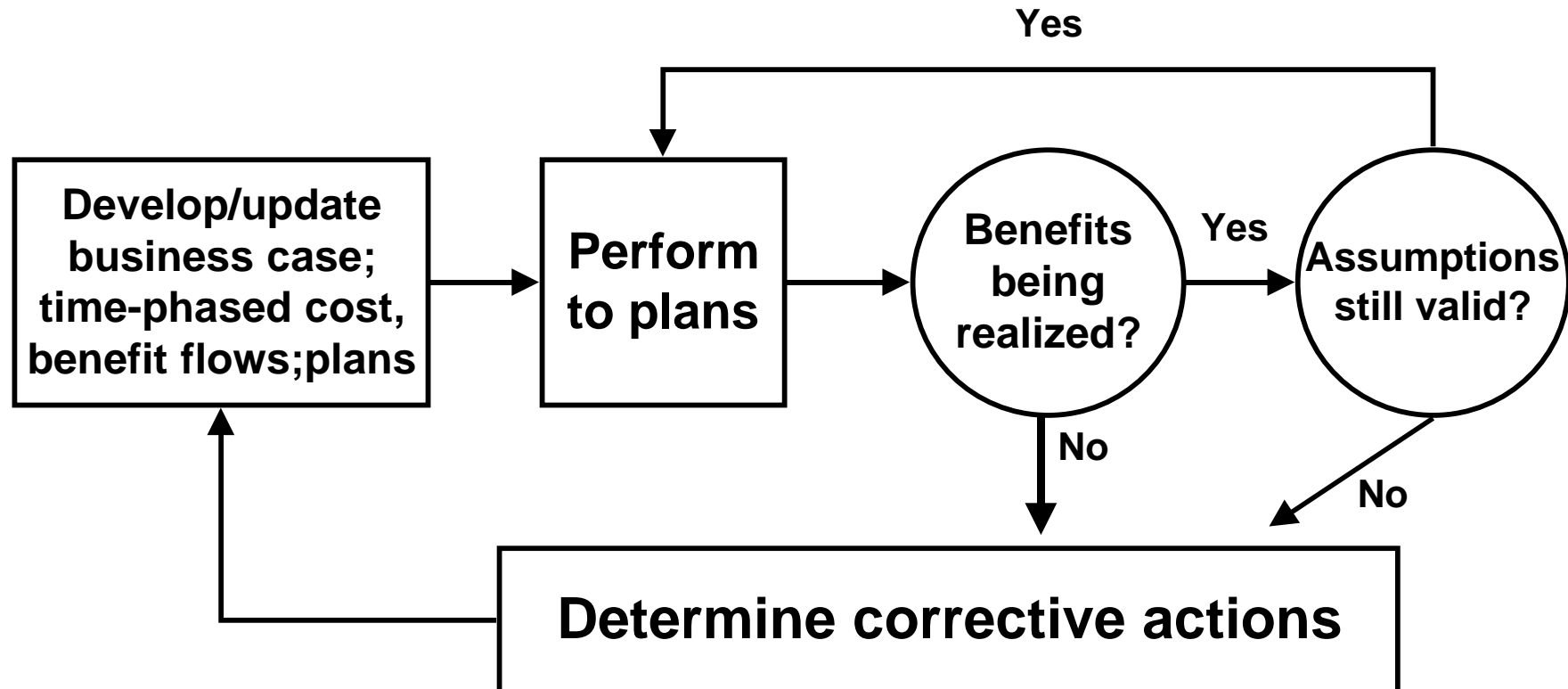


DMR/BRA Results Chain





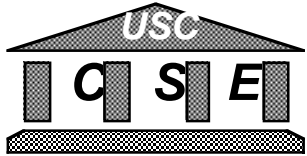
Benefits Realization Feedback Process





Implementing BRA via MBASE

- **Use WinWin Spiral Model**
 - Negotiated win conditions become goals and plans
 - Situation- and risk-driven process and product plans
 - Risk-, benefit-, and plan-driven feedback control
- **Avoid model clashes among success, product, process, property models**
 - Via MBASE integration framework, process framework
- **Use life cycle commitment anchor points**
 - Feasibility validation and stakeholder commitment
- **Use electronic process and product guides and templates**
 - Web-based, hyperlinked, tailorable



Spiral Model Refinements

- Where do objectives, constraints, alternatives come from?

- Win Win extensions

- Lack of intermediate milestones

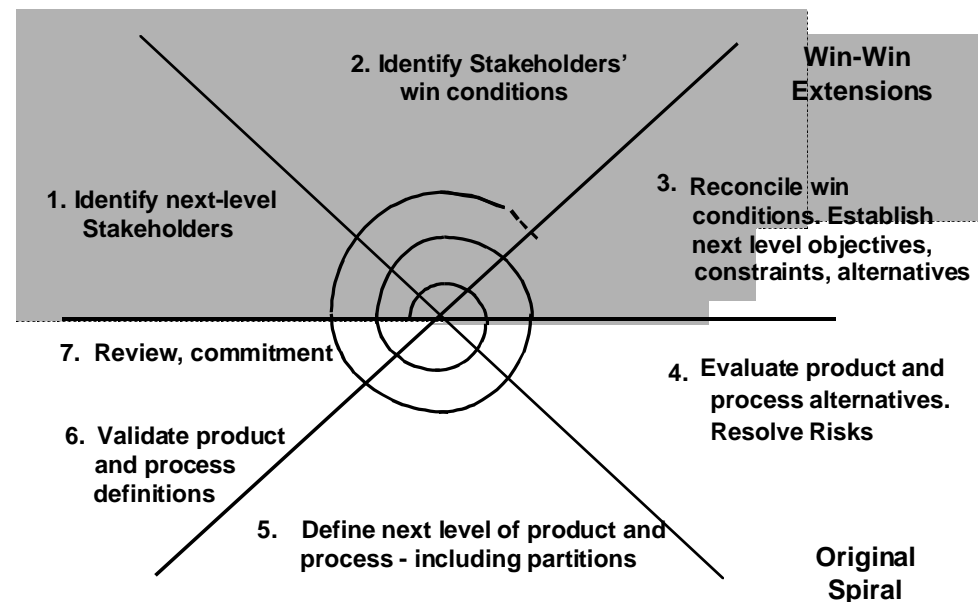
- Anchor Points: LCO, LCA, IOC

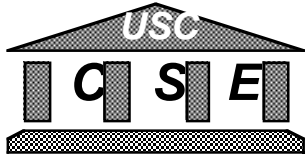
- Concurrent-engineering spirals between anchor points

- Need to avoid model clashes, provide more specific guidance

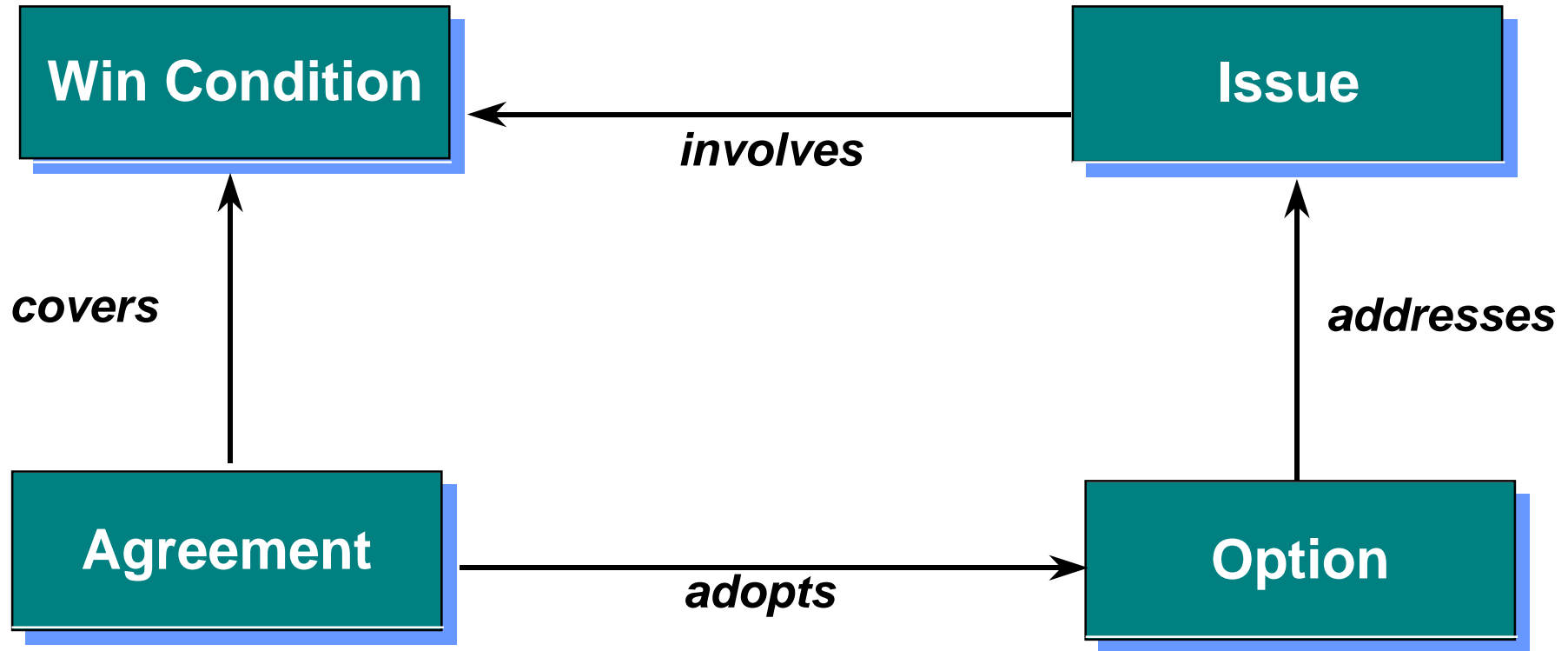
- MBASE

The WinWin Spiral Model





WinWin Negotiation Model



WinWin Equilibrium State

- All Win Conditions covered by Agreements
- No outstanding Issues



Easy Win Win Groupware System

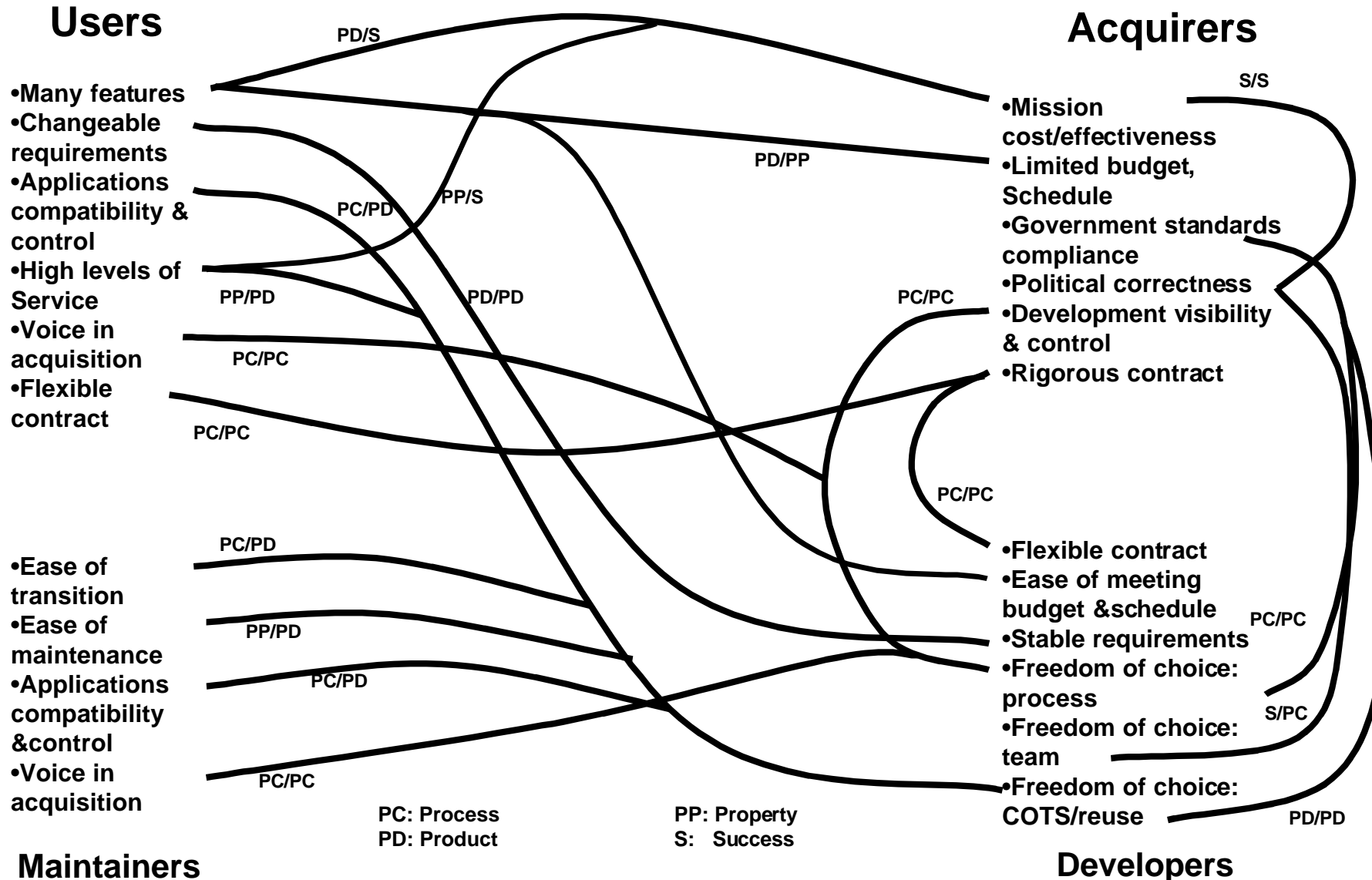
Progress	Win Condition
0 / 0	1. W01 Site management adds bookstore layout information (floor, location, type of material).
0 / 0	2. W03 The banner will provide a link to the university bookstore
0 / 0	3. W04 Interface for advertisers to select their schedule
2 / 0	4. W05 Default banner of bookstore if no other events
	...st have a website
	...sing, including sales,
	... of the bookstore, a map of it

Team builds a list of win conditions and organizes win conditions into buckets

- Application (3)
- Interfaces (7)
- Level of Service (3)
- Project & Process (0)
- Evolution (3)
- <TBD> (7)

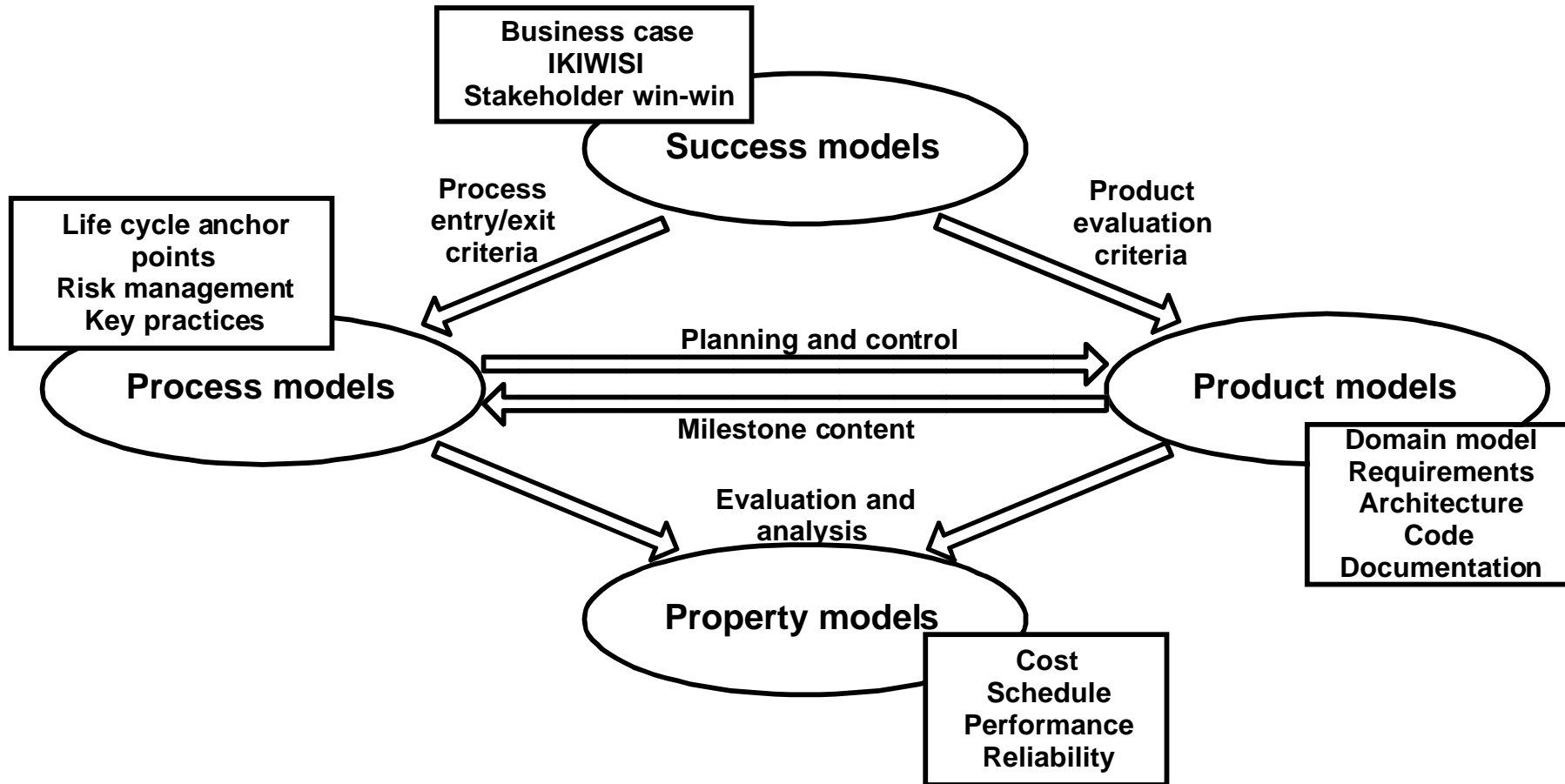


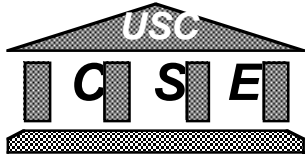
Success Model-Clash Profiles: General



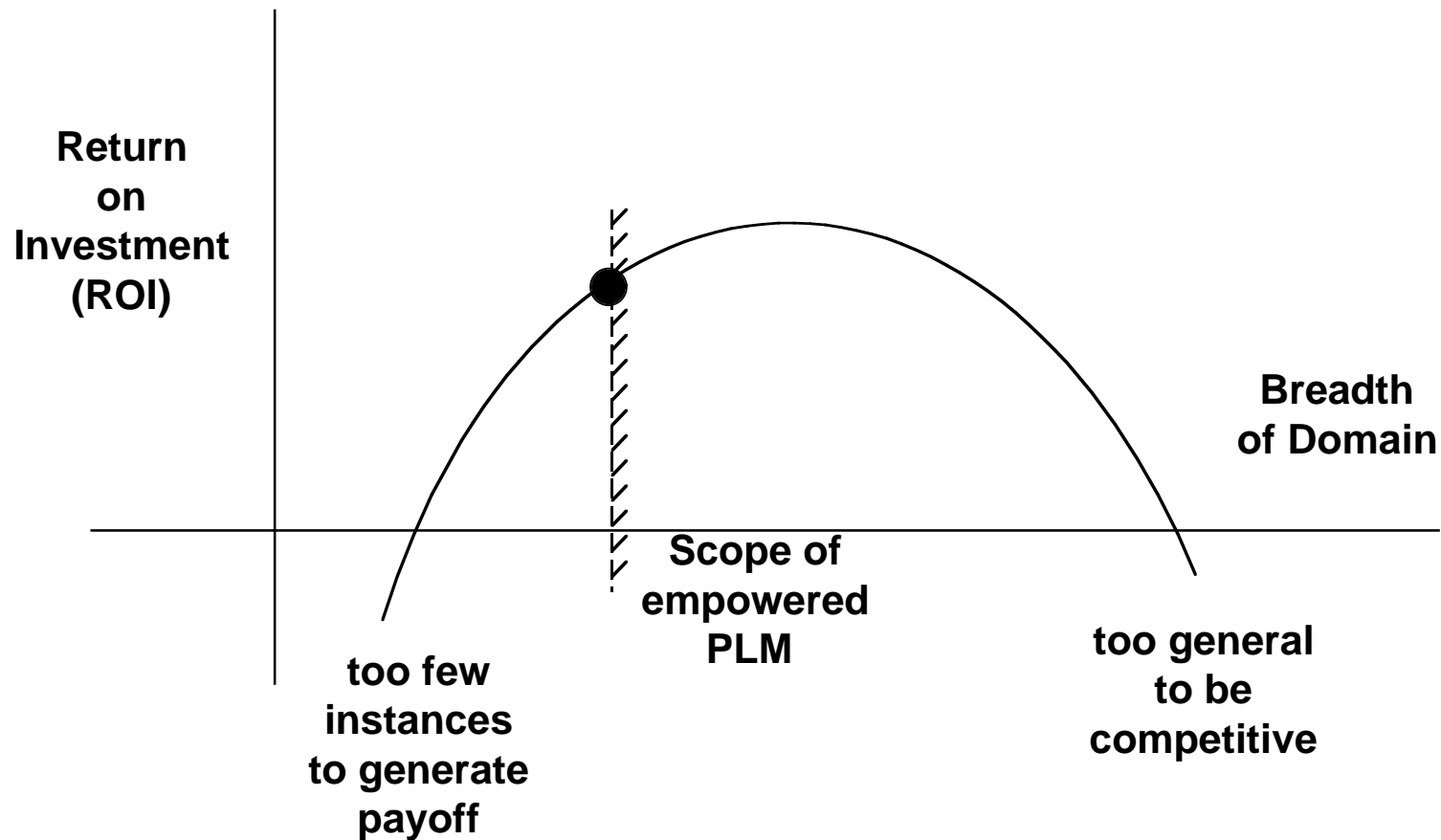


MBASE Integration Framework



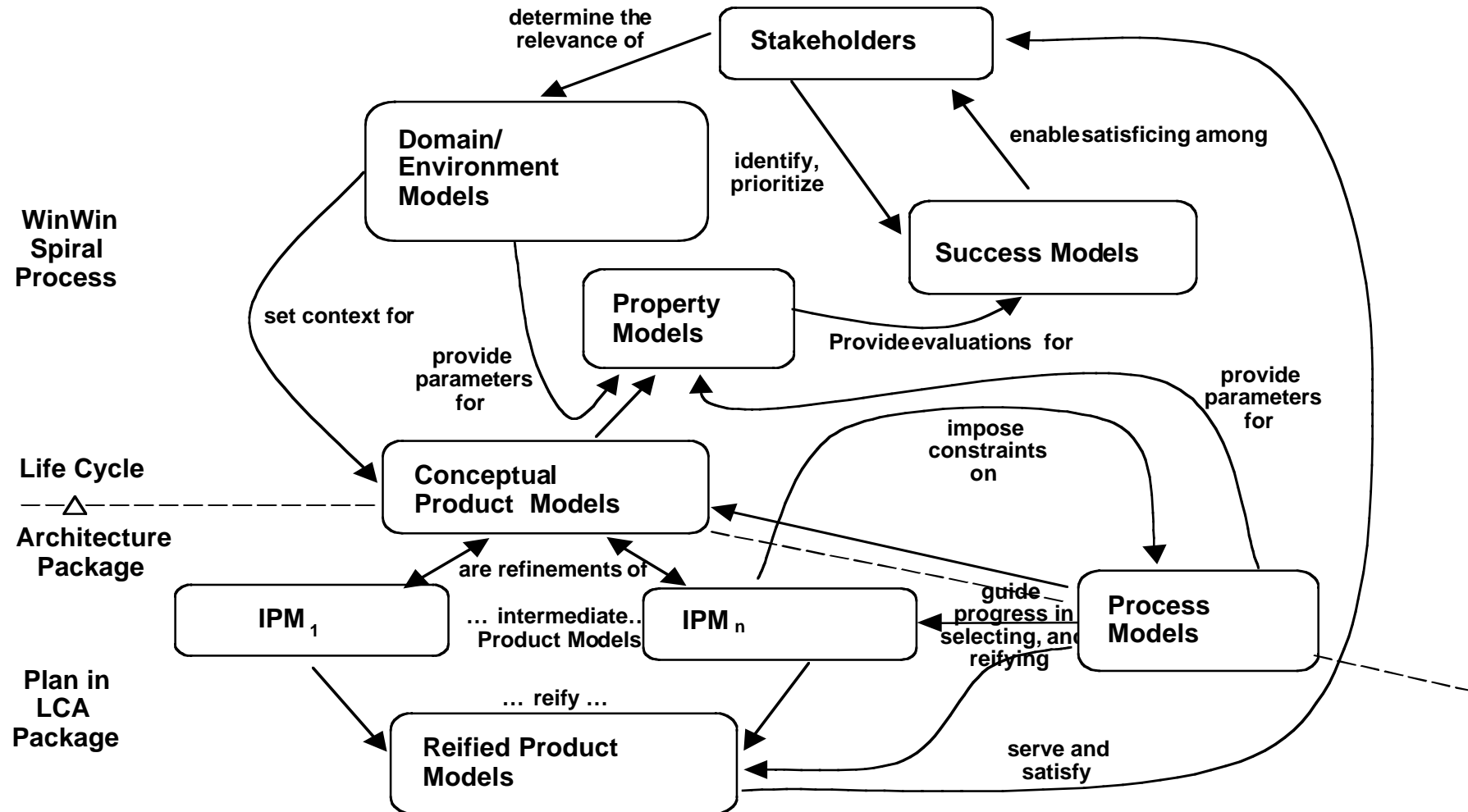


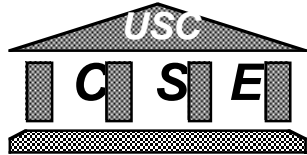
Product Line Domain Scope a Function of ROI, Scope of Empowered PL Manager





MBASE Process Framework





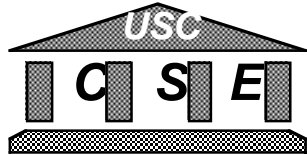
Success Models Drive Other Model Choices

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



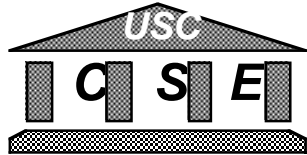
RAD Analysis via Process Simulation

- **Cost: task savings basically map 1:1 into project savings**
- **Schedule: task savings map 1:1 into project savings only while task is on critical path**
 - **Complicating factors: scale, dynamism, interdependent tasks**
 - **System dynamics an attractive analysis approach**



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

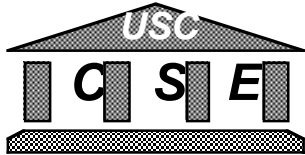


Win Win Spiral Anchor Points

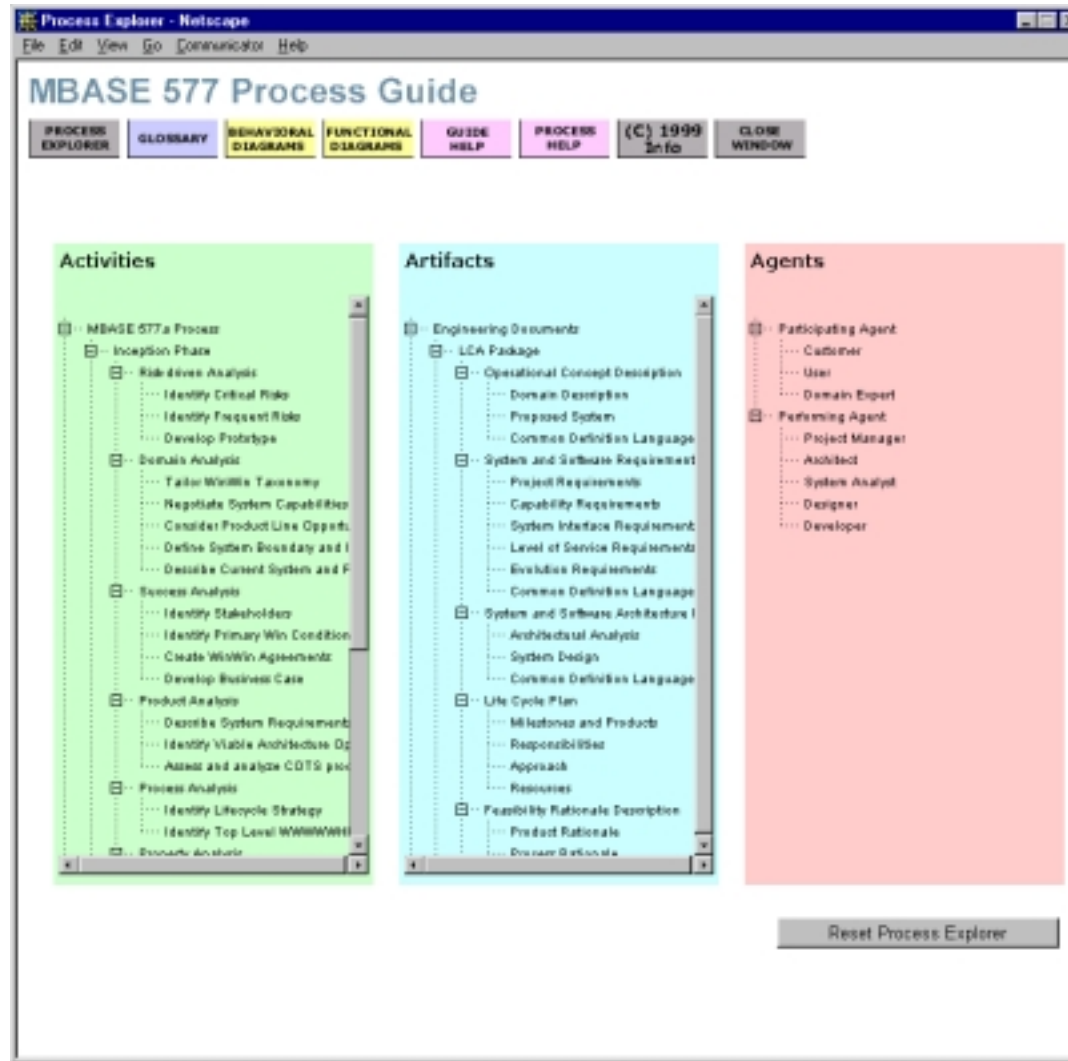
(Risk-driven level of detail for each element)

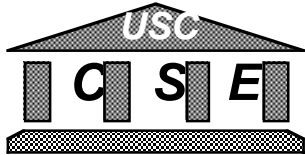
Milestone Element	Life Cycle Objectives (LCO)	Life Cycle Architecture (LCA)
Definition of Operational Concept	<ul style="list-style-type: none"> • Top-level system objectives and scope <ul style="list-style-type: none"> - System boundary - Environment parameters and assumptions - Evolution parameters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

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



MBASE Electronic Process Guide (1)





MBASE Electronic Process Guide (2)

Activity: MBASE 577a Process

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 exactors 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 studies can be recognized and recognized 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

Artifact: Operational Concept Description

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

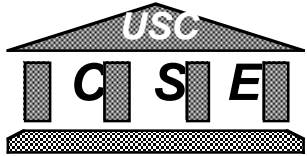
Description

DCD Audience and Participants

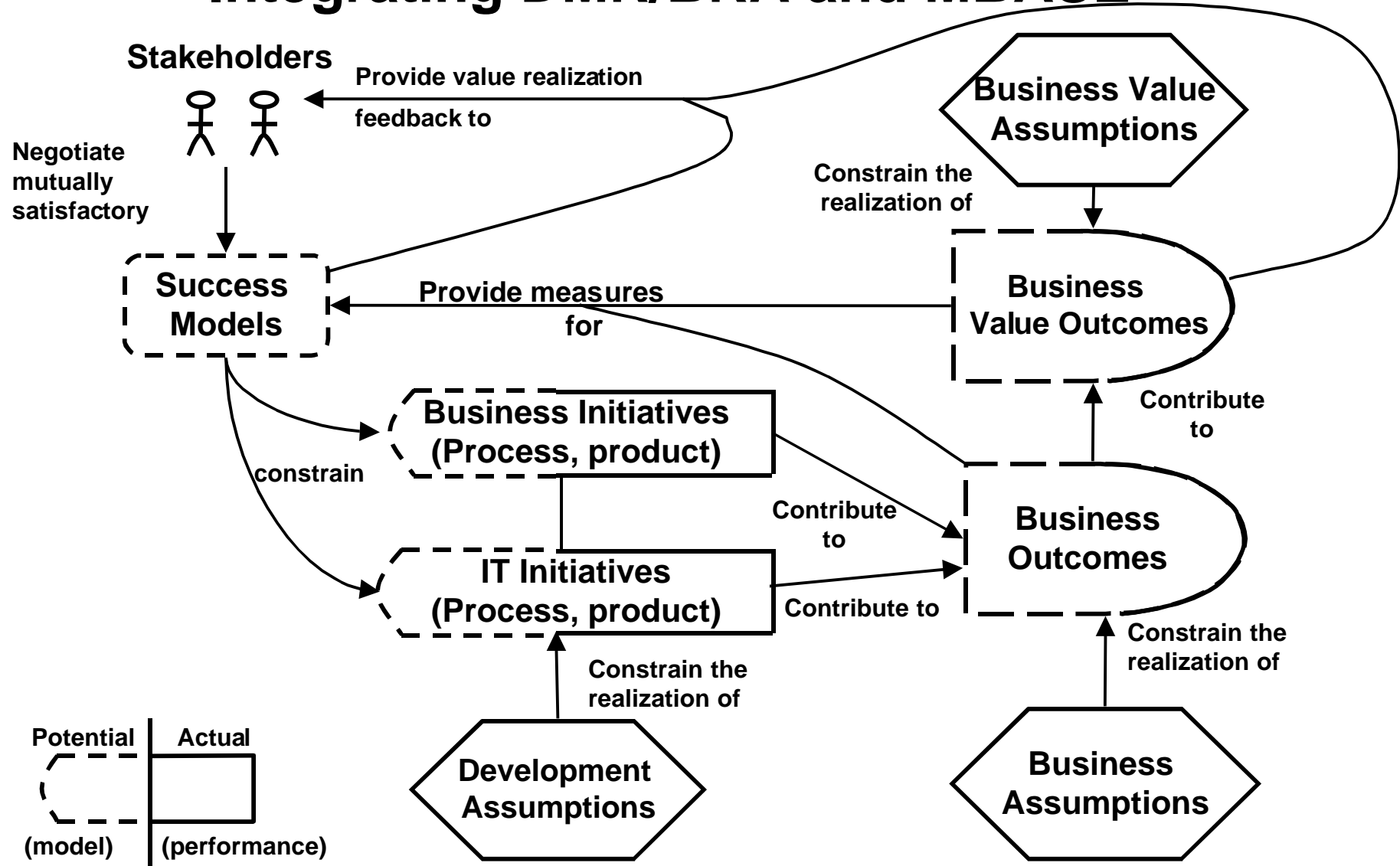
- Audience
 - Customer for Domain Description Domain
 - Domain Expert for System Analyst
- Use language and define CDL appropriately for intended audience
- Participants
 - Same stakeholders as WinWin negotiation
 - Establish concept of operation agreed on by all stakeholders

DCD High-Level Dependencies

- WinWin Negotiations Give
 - System Responsibility
 - Changes Considered But Not Included
 - Domain Description Terms
 - Project Goals, Quality Goals
 - DCD Yields
 - Project, System and Quality Regt for SSOH
 - Domain Description and Initial Analysis for SSAD
 - Stakeholder and Organizational Responsibilities



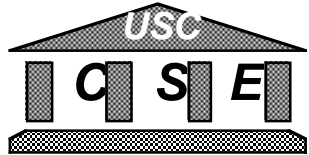
Integrating DMR/BRA and MBASE





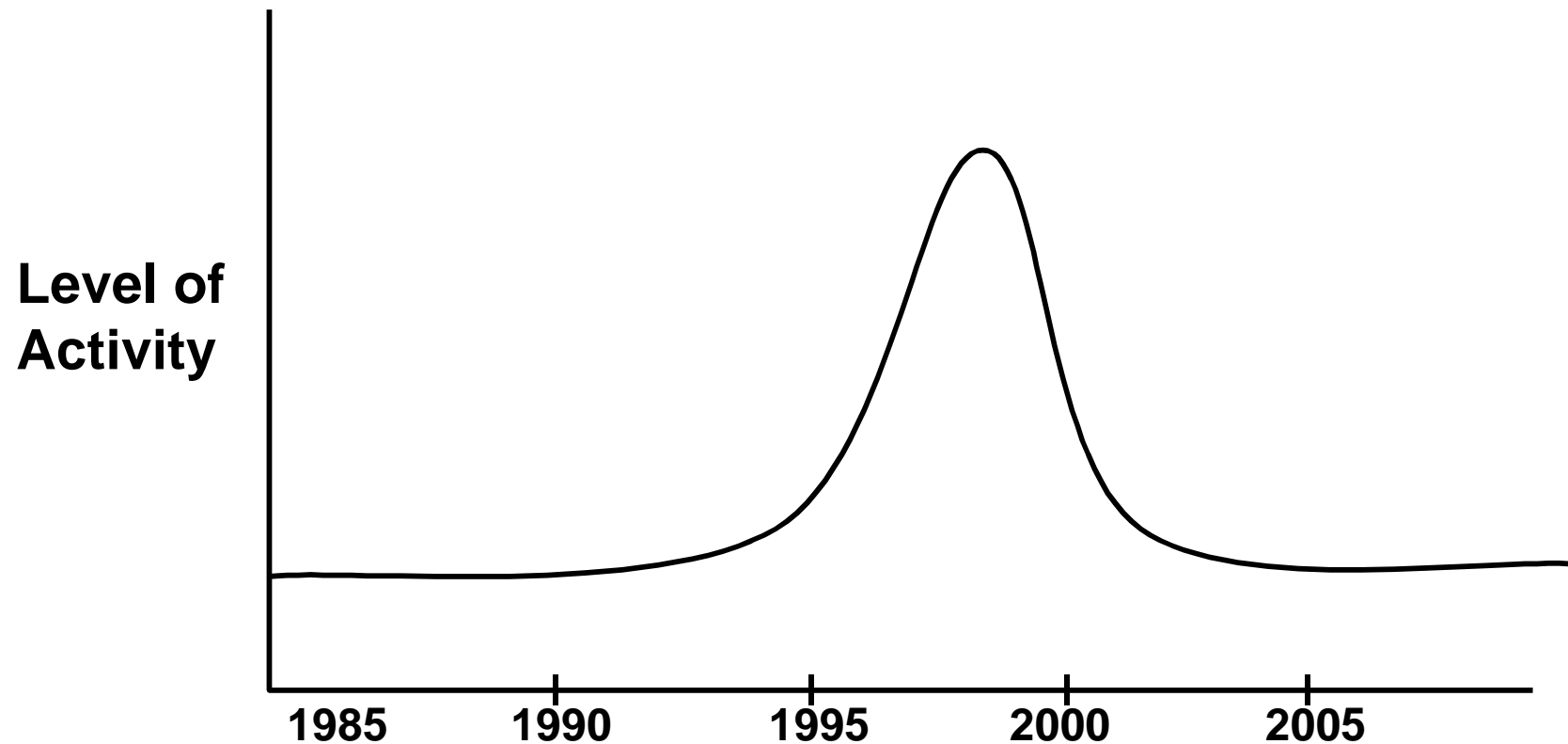
Outline

- **COCOMO II 4-Cycle feedback model**
- **Value-based feedback approach**
 - **DMR Benefits Realization Approach (BRA)**
 - **Implementing DMR-BRA via MBASE**
 - **Model-Based (System) Architecting and Software Engineering**
- ➔ • **Example application: Y2K**
 - **Relation to Laws of Evolution anomalies**
- **Conclusions**

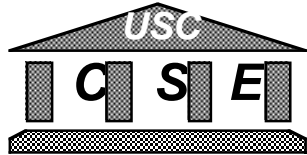


BRA-Based Feedback Process: Y2K Example

– Explaining Laws-of-Evolution Anomalies

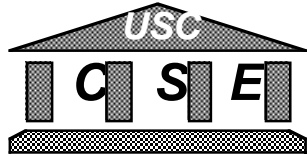


- **Law IV: Level of activity stays relatively constant**



Y2K Example Chronology

- **1994-95: Business case analysis: LCO spiral cycles**
 - Initial triage: some phaseouts, outsourcing, upstaffing
 - Resulting BRA plans, LCO milestone artifacts
- **1996: Post-Y2K architecture: LCA spiral cycles**
 - Pilot projects in riskier areas
 - Feedback control/update to BRA plans
- **1997-99: Construction and transition spiral cycles**
 - Incremental development by system
 - Feedback control/update to BRA plans

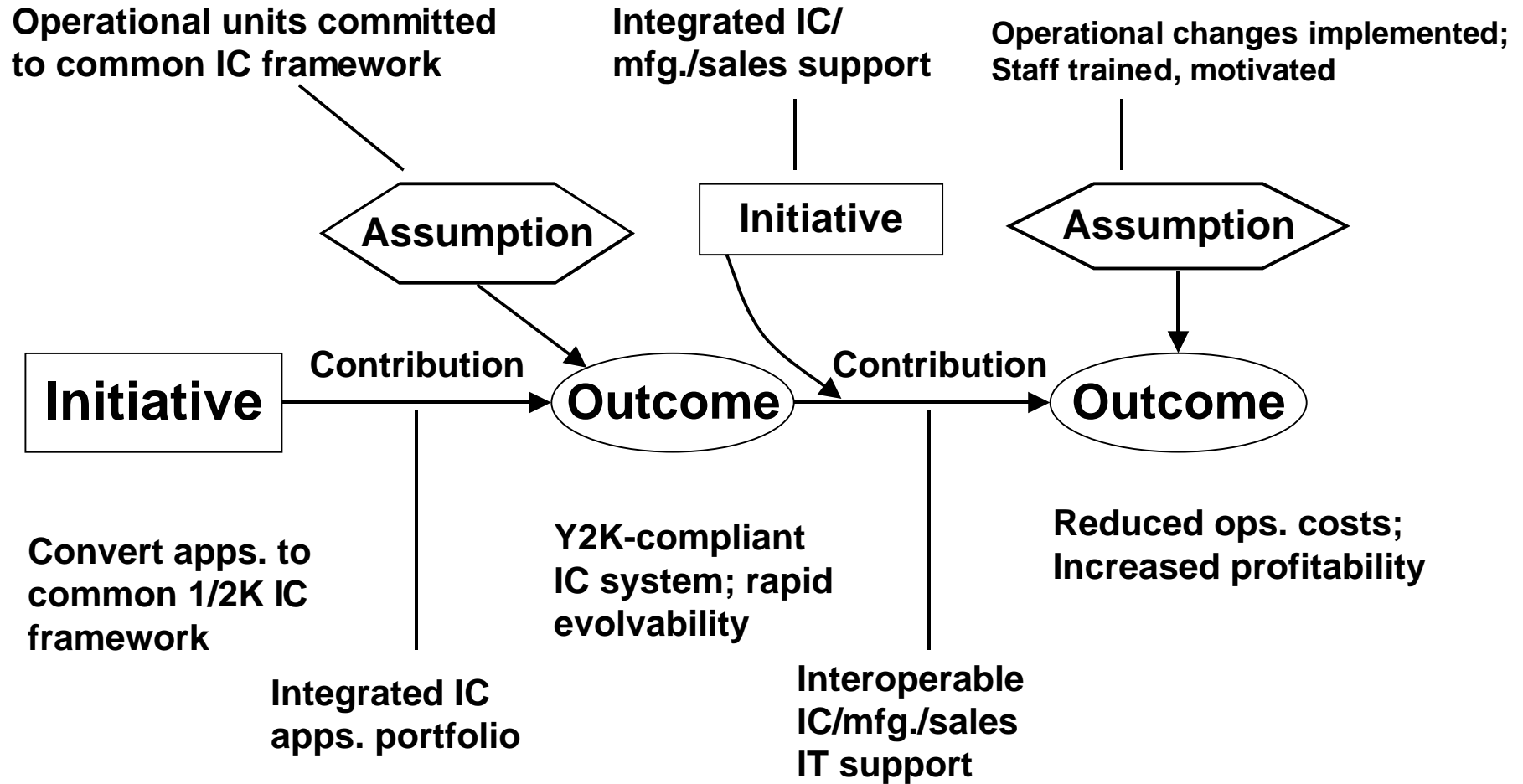


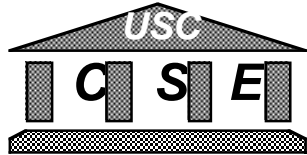
1994 Business Case Analysis

- **Objectives:** No serious Y2K problems;
portfolio integration; rapid evolvability;
cost-effective support of business strategy
 - **Business strategy:** Focus on core E-box product line;
shorten time-to-market;
closed loop sales/service/marketing/development
- **Alternatives:** New/modified/same portfolio architecture;
applications phaseout/migration/modification;
fix in-house/upstaff/outsource
- **Constraints:** Y2K aspects transitioned by mid-1999
- **Risks:** Scope too ambitious for timeframe;
obsolescence due to technology/market trends;
underestimate Y2K transition complexity



1995 Example Results Chain: Inventory Control (IC)



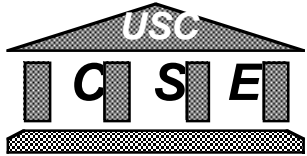


Individual Business Case Analysis: Control of manufacturing tools inventory

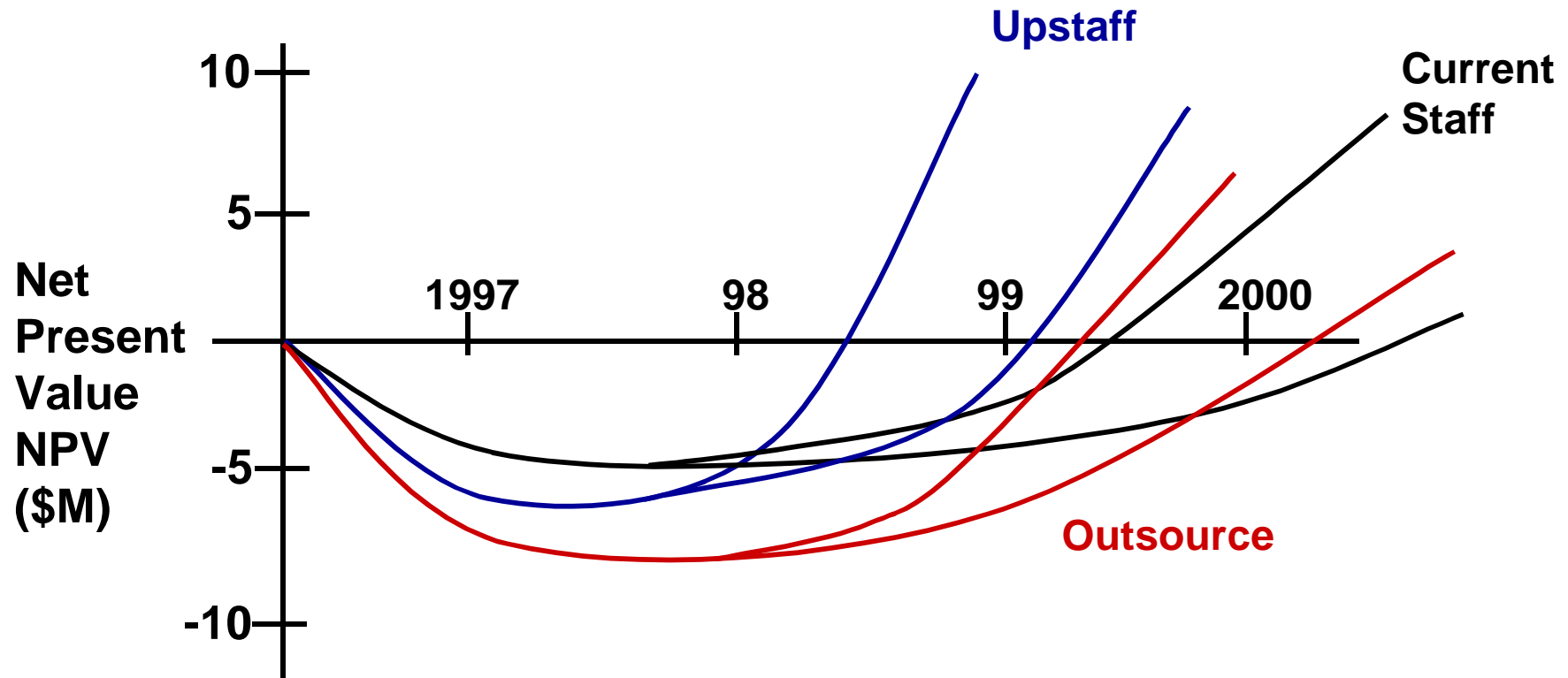
- Relational DBMS and 250 KSLOC of COBOL
- Decoupled from product inventory control, sales
- Analyze Net Present Value (NPV) for main alternatives

$$\text{NPV} = \sum \text{PV}(\text{benefit flows}) - \sum \text{PV}(\text{cost flows})$$

- Alternatives: current staff/upstaff/outsource
- Costs: COCOMO II maintenance model; outsource bids; Y2K noncompliance costs; training costs
- Benefits: Reduced clerical costs; inventory carrying costs



Manufacturing Tools IC Business Case Summary



- **Upstaff has best payoff, least Y2K risk**
- **Implies non-constant activity levels**



Conclusions

- **Need to embed IT feedback processes within enterprise feedback processes**
 - DMR/BRA and MBASE a good way to do this
 - Organizational evolution and process simulation provide valuable analysis frameworks
- **Value based feedback process can help explain anomalies in Laws of Evolution**
 - Y2K phenomenon representative of other large isolated events
 - Mergers, acquisitions, downsizing, e-commerce