University of Southern California
Center for Software Engineering

Model-Based System Architecting and Software Engineering (MBASE)

CS 510

Barry Boehm, USC

1999 Version

MBASE Outline

- Systems Engineering, Models, and Model Clashes
  - Model clash taxonomy and examples
- MBASE Framework
  - Relations to Win Win Spiral Model, Anchor Points
  - Relations to Objectory, AT&T/Lucent ARB's
  - Application to Digital Library projects, CCPDS-R
  - Detailed Guidelines
- Early Adopters
- Conclusions

“No scene from prehistory is quite so vivid as that of the mortal struggles of great beasts in the tar pits.

Large system programming has over the past decade been such a tar pit, and many great and powerful beasts have thrashed violently in it.”

Fred Brooks, 1975

“Everyone seems to have been surprised by the stickiness of the problem, and it is hard to discern the nature of it.

But we must try to understand it if we are to solve it.”

Fred Brooks, 1975

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
- Spec-based process and IKIWISI success model
  - I'll know it when I see it
- Golden Rule and stakeholder win-win

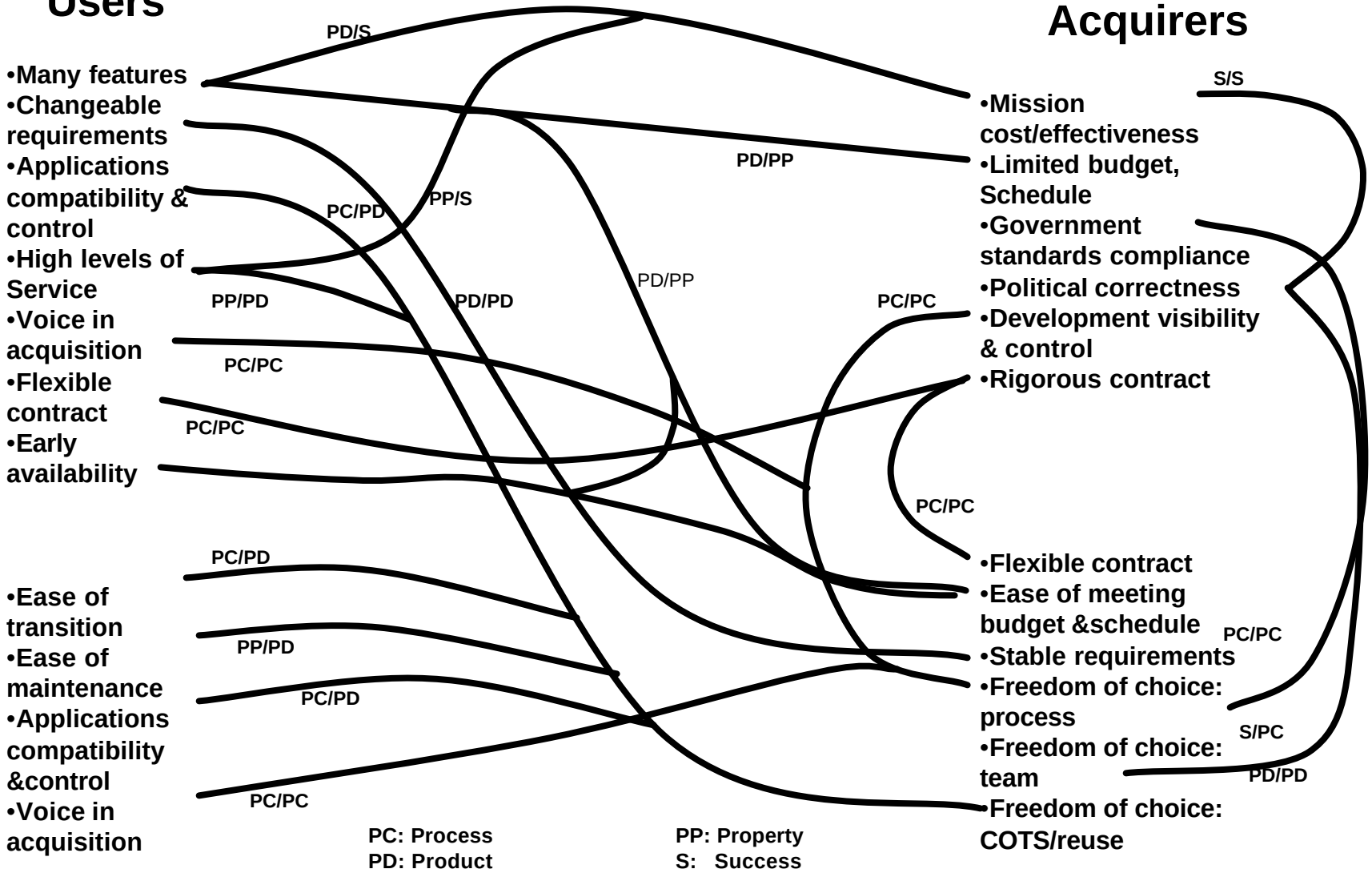
The Golden Rule as Software Success Model

- Do unto others
- •• As you would have others do unto you
- Build computer systems to serve users and operators
- •• Assuming users and operators like to write programs, and know computer science
- Computer science world (Compilers, OS, etc.)
  - Users love powerful, obscure, UNIX-like commands
- Applications world
  - Users are pilots, doctors, librarians: Keep it simple
- Better to use Platinum Rule
  - Do unto others as they would be done unto

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**
- **Multi-Stakeholder success model clashes**

Success Model-Clash Profiles: General, Master Net



Model Clashes and Tar Pits: Bank of America Master Net Example

- Master Net Stakeholders
- Master Net Contract
- Initial Progress
- The Tar Pit
- The Bottom Line
- Contributing Model Clashes

R.Glass, Software Runaways, Prentice Hall, 1998, PP. 152-182

Master Net Stakeholders- I

- Acquirer: BofA top management
  - Information technology leader in 1950-60's
  - IT laggard in 1970's
  - IT leadership push in 1980's
- Users: BofA, other banks' trust departments
  - Trust automation a key to bank client services
    - Corporate stock & bond holdings, pension plans
  - BofA operating with 1960's batch system
  - Consortium: build and use interactive trust system
    - Sell services to other banks

Master Net Stakeholders- II

- Maintainers: BofA IT organization
  - Large batch IBM-mainframe, COBOL shop
  - Unsuccessful $6M 1981 effort to rebuild trast system
- Developer: Premier Systems (Stephen Katz)
  - Successful with 1970's small-medium banking systems
  - Focused on Prime Computer equipment

Master Net Contract

- Consortium contract with Premier Systems, March 1984
- Develop core TrustPlus trust accounting system
  - Plus 8 user-services subsystems
  - Interactive access by remote users
  - Single 8MB, 1MIP Prime processor
- Complete by December 31, 1984

Initial Progress

- Good progress on user-service subsystem requirments
- Good conversion preparation and training program
- Vague definition of core TrustPlus system
  - Better BofA systems engineers reorganized elsewhere
- No system by 12/31/1984, but no major concern

The Tar Pit

- Slow progress into 1986
  - Growth to 3.5 million lines of code
- High-exposure demonstration event in May 1986
  - Inflated user expectations
- Unsuccessful conversion attempts in late 1986
- Several key BofA system engineers resign
- Katz and Premier increasingly uncooperative
  - “Don’t give us the solutions. Just tell us the problems.”
  - Still performance problems with 3 16 MB, 8 MIP Primes
- Major March 1987 conversion effort fails
  - Over a dozen Prime disk drives fail
- Increasing user frustration with workarounds

The Bottom Line

- July 1987: $23M additional budget
- October 1987: Trust and System Engineering department managers resign
- January 1988: Project canceled of cost of $80M
  - BofA divests its trust businesses
  - Major corporate image problems, business losses

Clashes Among MBASE Models

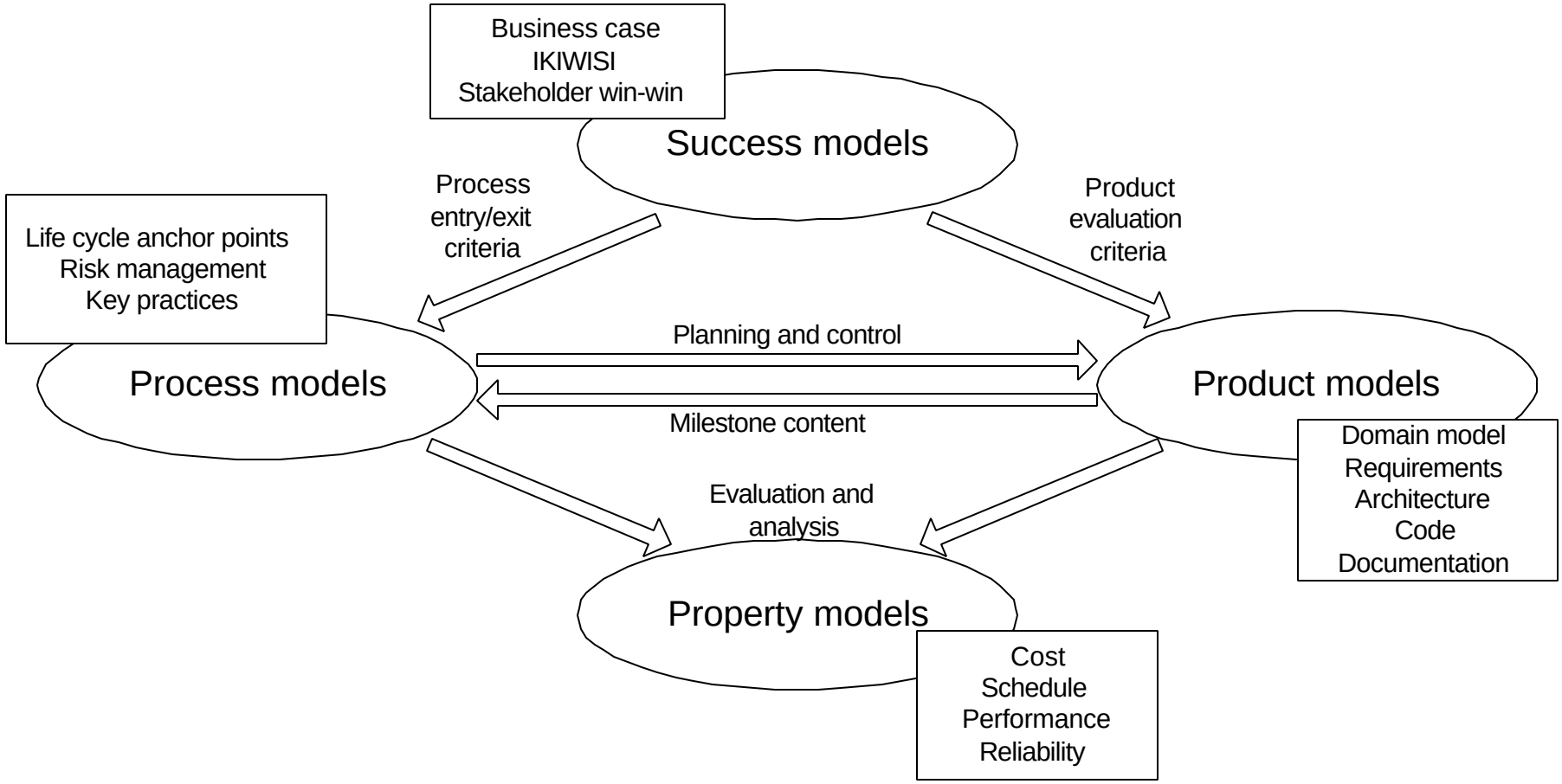
	Product Model	Process Model	Property Model	Success Model
Product Model	· Structure clash · Traceability clash · Architecture style clash	· COTS-driven product vs. Waterfall (requirements-driven) process	· Interdependent multiprocessor product vs. linear performance scalability model	· 4GL-based product vs. low development cost and performance scalability
Process Model		· Multi-increment development process vs. single-increment support tools	· Evolutionary development process vs. Rayleigh-curve cost model	· Waterfall process model vs. "I'll know it when I see it" (IKIWISI) prototyping success model
Property Model			· Minimize cost and schedule vs. maximize quality (Quality is free)	· Fixed-price contract vs. easy-to-change, volatile requirements
Success Model				· Golden Rule vs. stakeholder win-win

MBASE Outline

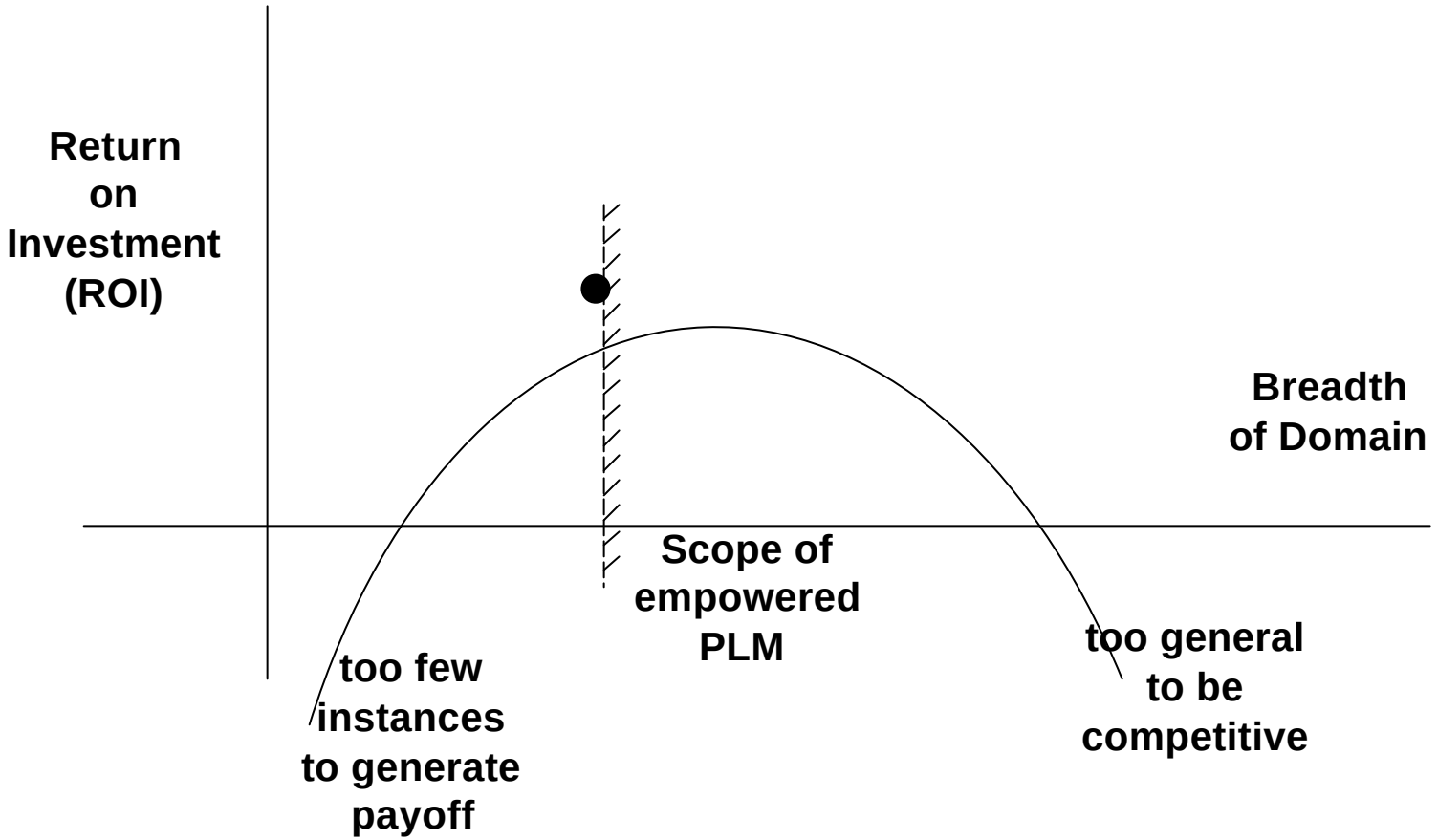
- Systems Engineering, Models, and Model Clashes
  - Model clash taxonomy and examples

➡ MBASE Framework
  - Relations to Win Win Spiral Model, Anchor Points
  - Relations to Objectory, AT&T/Lucent ARB's
  - Application to Digital Library projects
  - Detailed Guidelines
- Early Adopters
- Conclusions

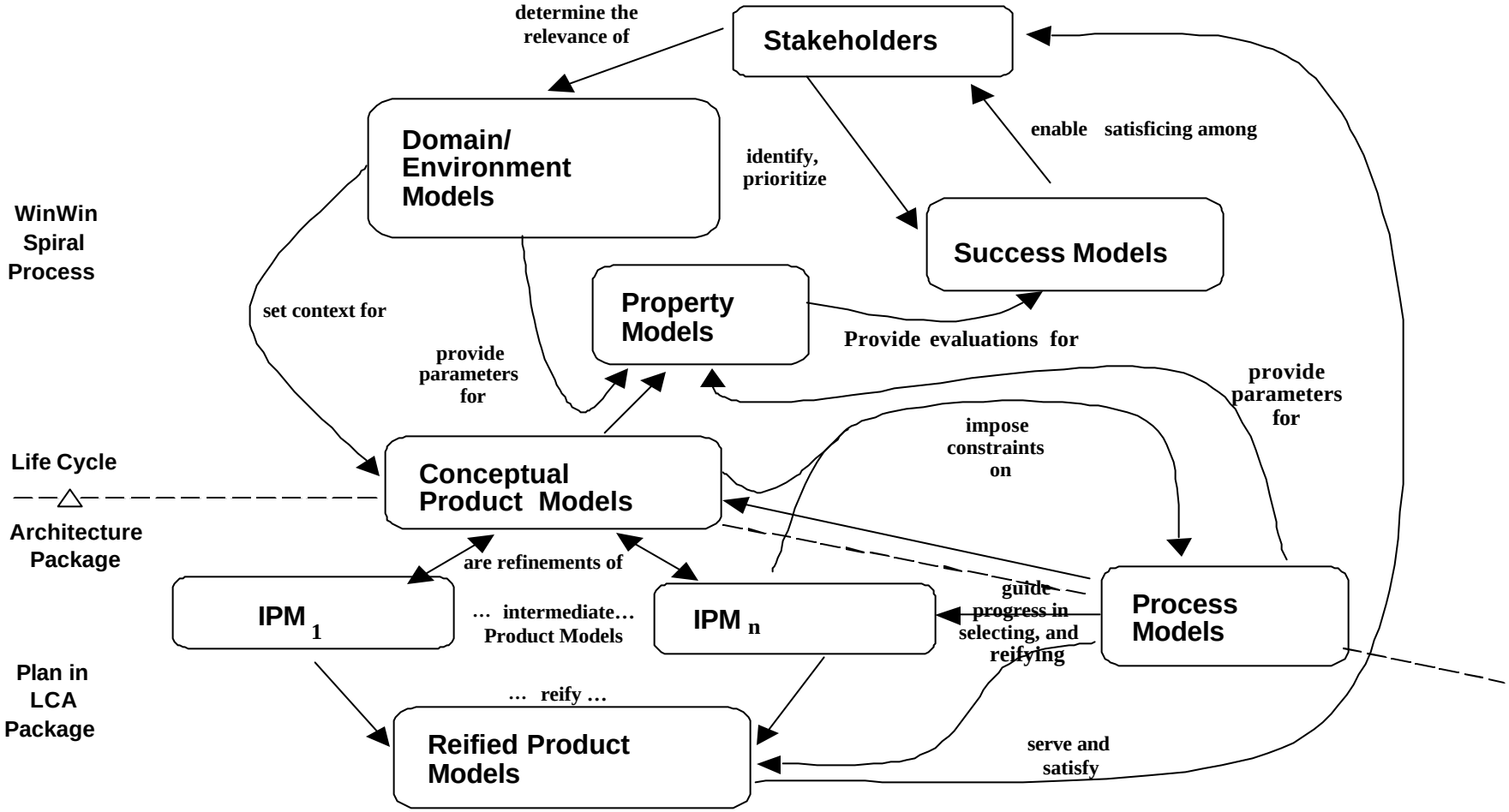
MBASE Integration Framework



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



MBASE Conceptual 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 Stake-holders	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

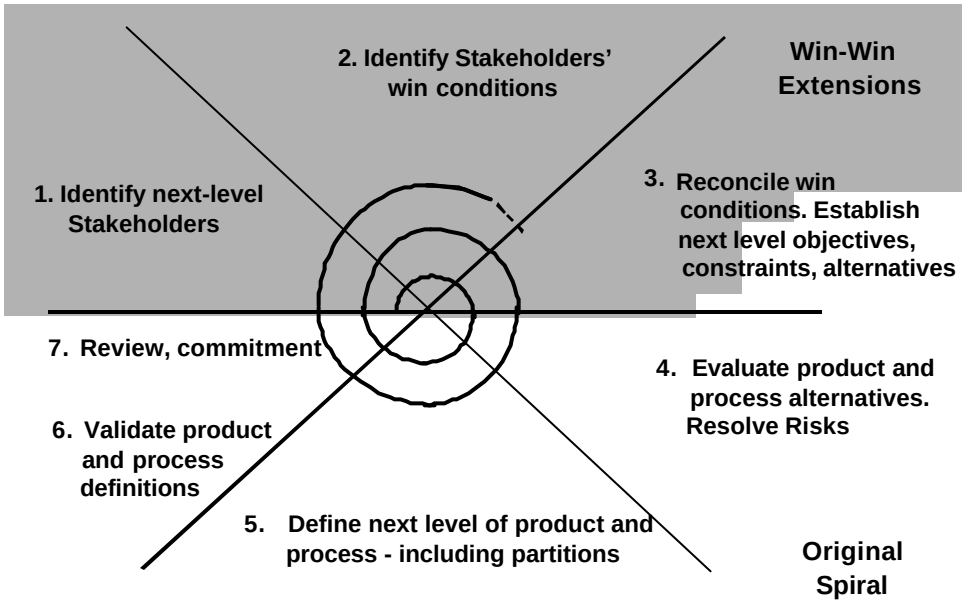
Process Model Decision Table

Objectives, Constraints			Alternatives		Model	Example
Growth Envelope	Understanding of Rqts.	Robustness	Available Technology	Architecture Understanding		
Limited			COTS		Buy COTS	Simple Inventory Control
Limited			4GL, Transform		Transform or Evol. Devel.	Small Business - DP Application
Limited	Low	Low		Low	Evol. Prototype	Advanced Pattern Recognition
Limited to Large	High	High		High	Waterfall	Rebuild of old system
	Low	High			Risk Reduction followed by Waterfall	Complex Situation Assessment
		High		Low		High-performance Avionics
Limited to Medium	Low	Low-Medium		High	Evolutionary Development	Data Exploitation
Limited to Large			Large Reusable Components	Medium to High	Capabilities-to-Requirements	Electronic Publishing
Very Large		High			Risk Reduction & Waterfall	Air Traffic Control
Medium to Large	Low	Medium	Partial COTS	Low to Medium	Spiral	Software Support Environment

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)
Definition of Operational Concept	• Top-level system objectives and scope - System boundary - Environment parameters and assumptions - Evolution parameters • Operational concept - Operations and maintenance scenarios and parameters - Organizational life-cycle responsibilities (stakeholders)	• Elaboration of system objectives and scope of increme • Elaboration of operational concept by increment
System Prototype(s)	• Exercise key usage scenarios • Resolve critical risks	• Exercise range of usage scenarios • Resolve major outstanding risks
Definition of System Requirements	• Top-level functions, interfaces, quality attribute levels, including: - Growth vectors and priorities - Prototypes • Stakeholders' concurrence on essentials	• 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	• Top-level definition of at least one feasible architecture - Physical and logical elements and relationships - Choices of COTS and reusable software elements • Identification of infeasible architecture options	• Choice of architecture and elaboration by increment - Physical and logical components, connectors, configurations, constraints - COTS, reuse choices - Domain-architecture and architectural style choices • Architecture evolution parameters
Definition of Life-Cycle Plan	• Identification of life-cycle stakeholders - Users, customers, developers, maintainers, interoperators, general public, others • Identification of life-cycle process model - Top-level stages, increments • Top-level WWWWWHH* by stage	• Elaboration of WWWWWHH* for Initial Operational Capability (IOC) - Partial elaboration, identification of key TBD's for later increments
Feasibility Rationale	• Assurance of consistency among elements above - via analysis, measurement, prototyping, simulation, etc. - Business case analysis for requirements, feasible architectures	• Assurance of consistency among elements above - All major risks resolved or covered by risk manageme plan

*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

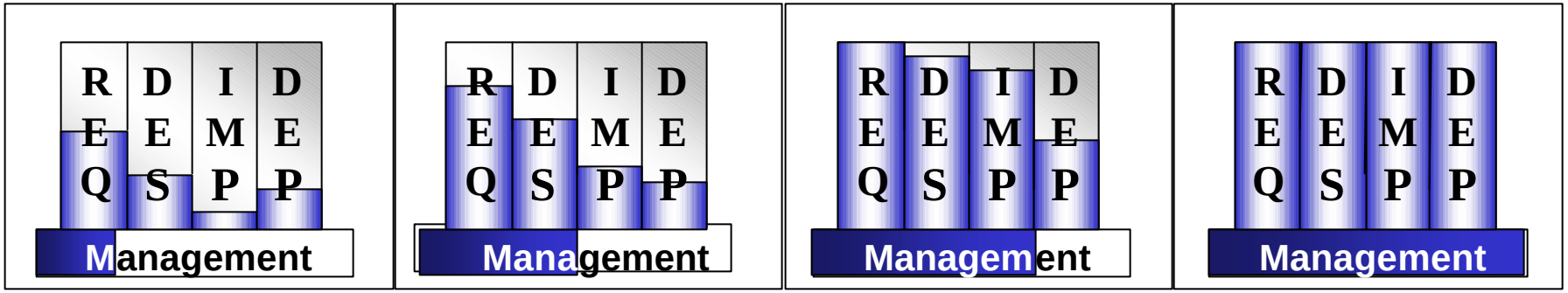
Anchor Points and Rational USDP Phases

Engineering Stage	Manufacturing Stage

Inception	Elaboration	Construction	Transition

LCO LCA IOC

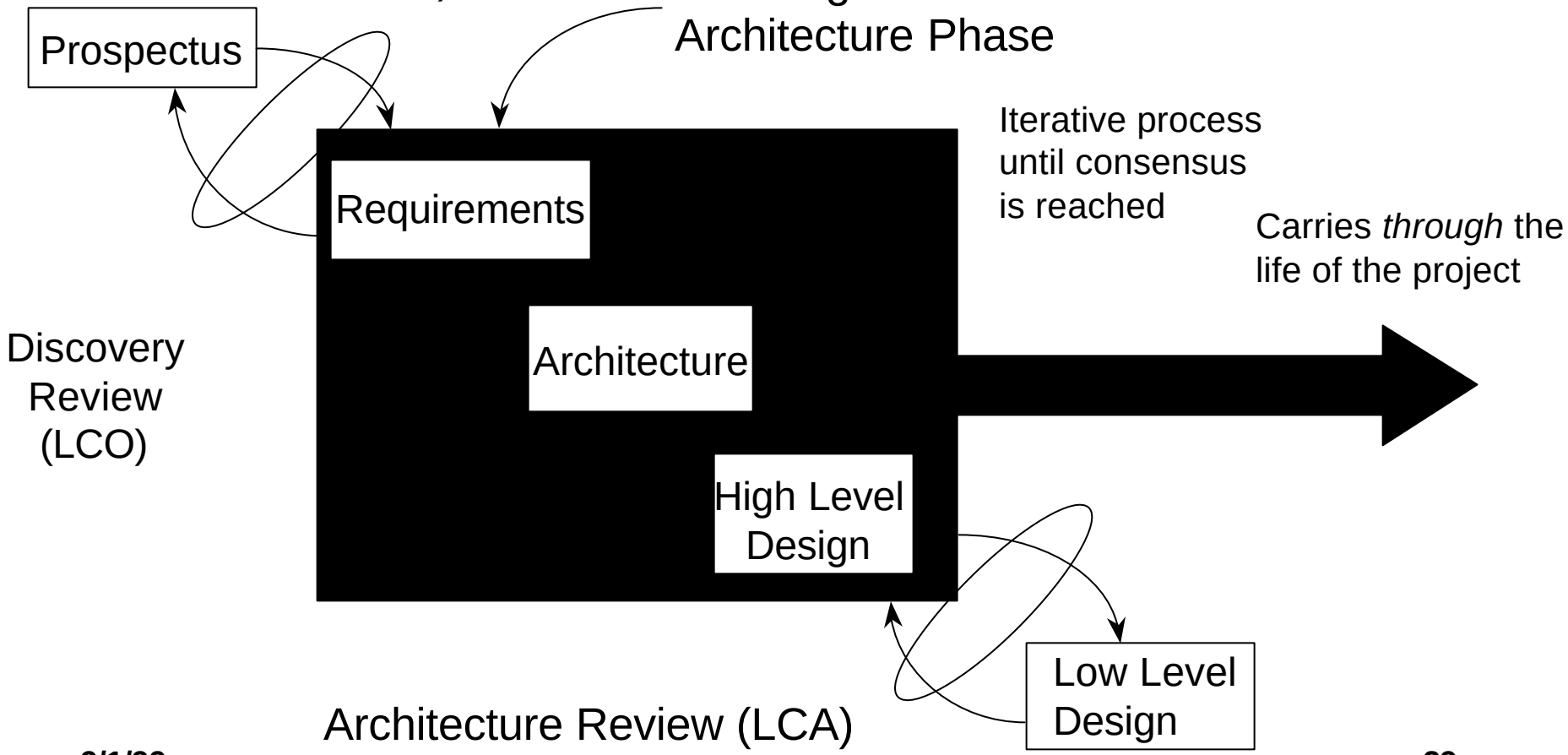
Feasibility Iterations	Architecture Iterations	Usable Iterations	Product Releases



RATIONAL
Software Corporation

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).



Lucent/AT&T Architectural Review Boards:

How a Review Is Conducted

- Chairperson meets with the project to determine technical focus and required expertise for review
- Chairperson assembles review team of stakeholders and subject matter architecture experts; project sends out review material
- A 2 or 3 day review is conducted. Detailed talks are presented on key technical areas. Issues raised during discussions are recorded on cards
- Immediate readout is given to the team at the end of the review. Cards are grouped by- Things Done Right, Issues, and Recommendations
- Chairperson follow up with a written report and presentation to the project's management if requested
- Used regularly since 1988, with over 10% project savings

MBASE Outline

- MBASE Overview and Motivation
  - Model clash taxonomy and examples
- MBASE Framework
  - Relations to Win Win Spiral Model, Anchor Points
  - Relations to Objectory, AT&T/Lucent ARB's
  - Application to Digital Library projects, CCPDS-R
  - Detailed Guidelines
- Early Adopters
- Conclusions

MBASE Example I - Digital Library Applications

- **The Challenge**
- **MBASE Approach**
- **1996-97 Results**
- **1997-98 Results**

The Challenge

- 15-20 Digital Library Applications
  - 2 sentence problem statements
  - Librarian clients
- 80-100 Graduate Students
  - 30% with industry experience
  - Largely unfamiliar with each other, Library ops.
* Develop LCA packages in 11 weeks
- Re-form teams from 30-40 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.

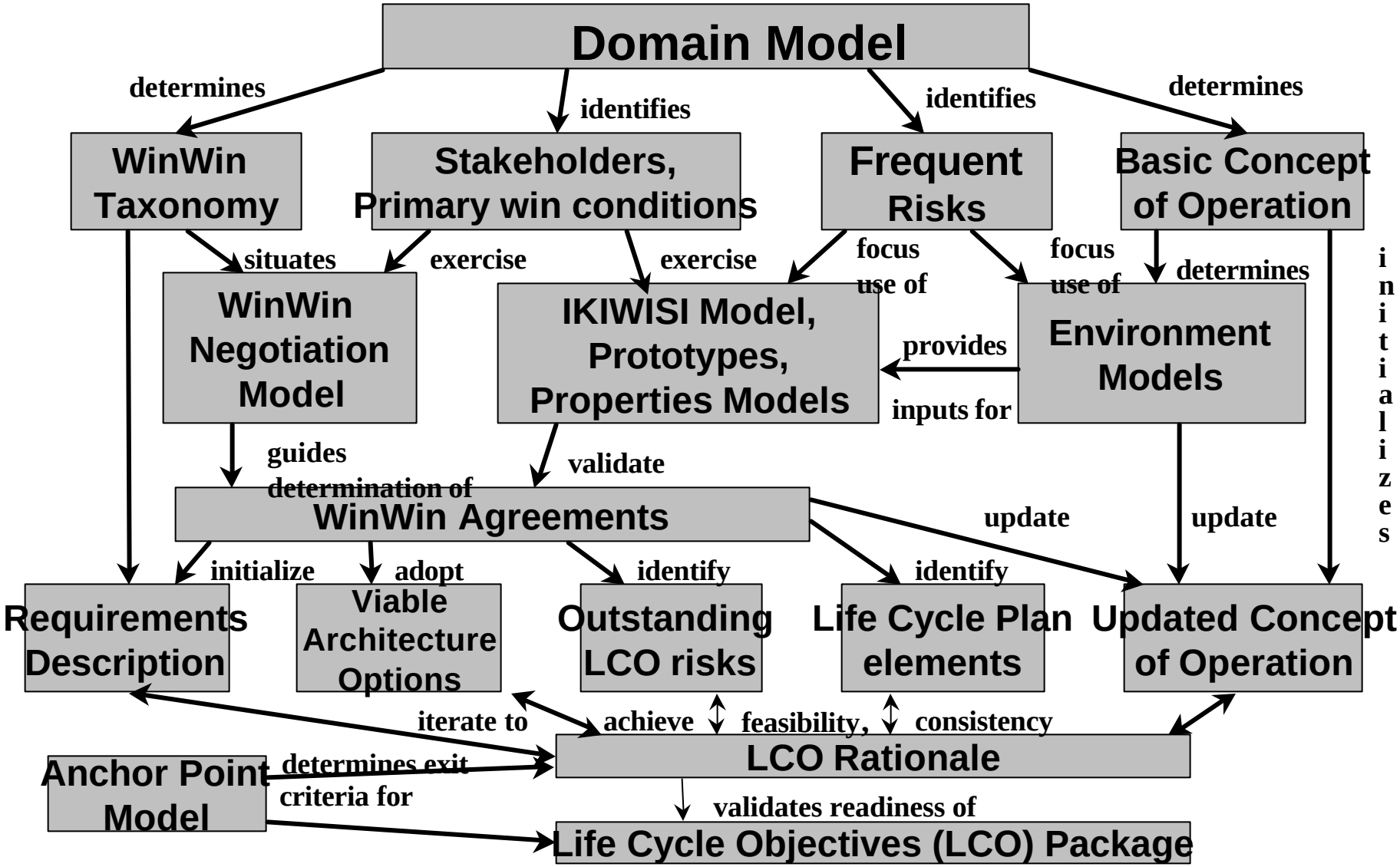
Elements of Critical Front End Milestones

(Risk-driven level of detail for each element)

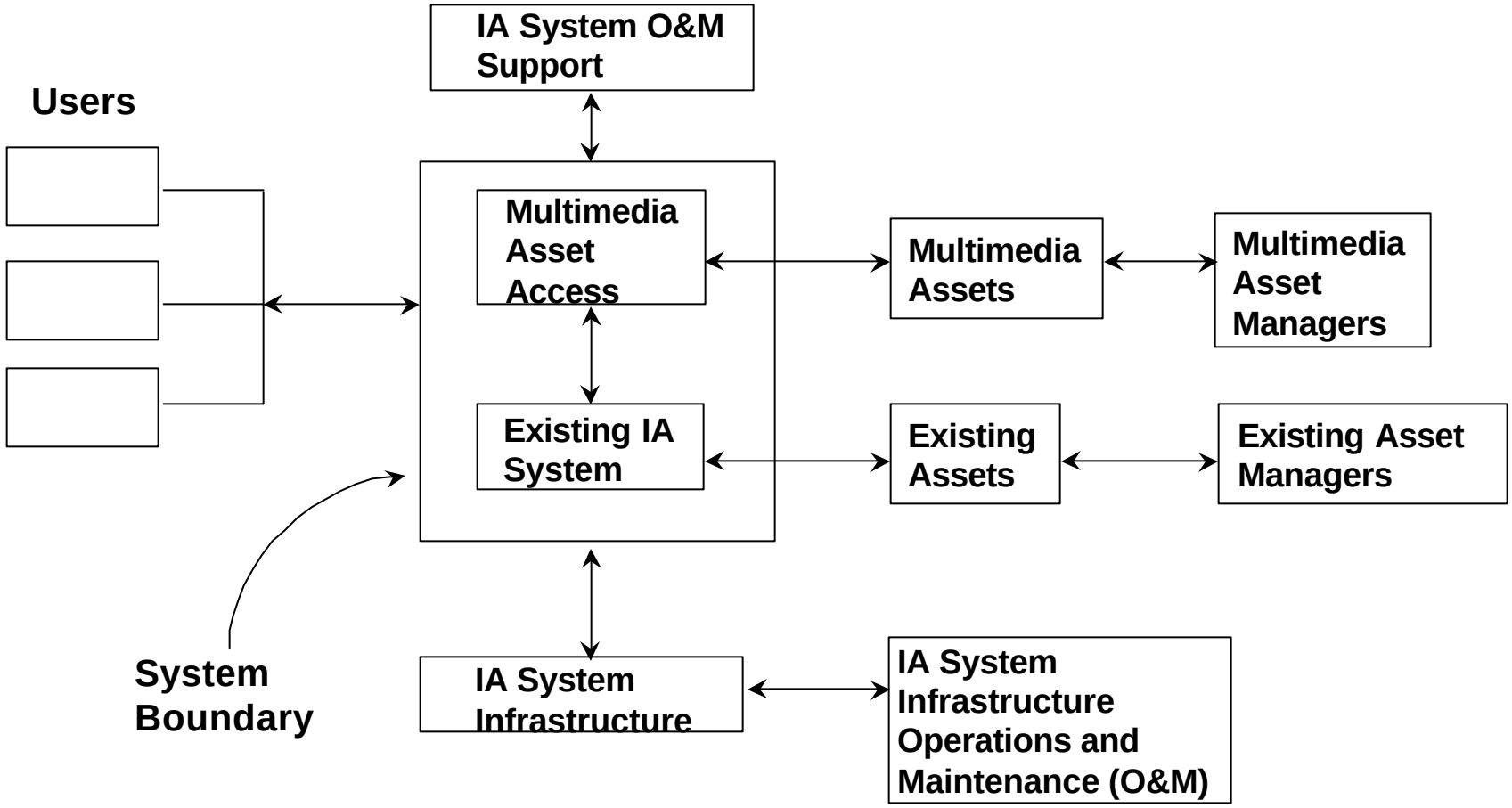
Milestone Element	Life Cycle Objectives (LCO)	Life Cycle Architecture (LCA)
Definition of Operational Concept	• Top-level system objectives and scope - System boundary - Environment parameters and assumptions - Evolution parameters • Operational concept - Operations and maintenance scenarios and parameters - Organizational life-cycle responsibilities (stakeholders)	• Elaboration of system objectives and scope of increme • Elaboration of operational concept by increment
System Prototype(s)	• Exercise key usage scenarios • Resolve critical risks	• Exercise range of usage scenarios • Resolve major outstanding risks
Definition of System Requirements	• Top-level functions, interfaces, quality attribute levels, including: - Growth vectors and priorities - Prototypes • Stakeholders' concurrence on essentials	• 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	• Top-level definition of at least one feasible architecture - Physical and logical elements and relationships - Choices of COTS and reusable software elements • Identification of infeasible architecture options	• Choice of architecture and elaboration by increment - Physical and logical components, connectors, configurations, constraints - COTS, reuse choices - Domain-architecture and architectural style choices • Architecture evolution parameters
Definition of Life-Cycle Plan	• Identification of life-cycle stakeholders - Users, customers, developers, maintainers, interoperators, general public, others • Identification of life-cycle process model - Top-level stages, increments • Top-level WWWWWHH* by stage	• Elaboration of WWWWWHH* for Initial Operational Capability (IOC) - Partial elaboration, identification of key TBD's for later increments
Feasibility Rationale	• Assurance of consistency among elements above - via analysis, measurement, prototyping, simulation, etc. - Business case analysis for requirements, feasible architectures	• Assurance of consistency among elements above - All major risks resolved or covered by risk manageme plan

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

MBASE Model Integration: LCO Stage



Domain Model: Block Diagram



IA: Information Archive

WinWin Taxonomy Mapping to Requirements Description Outline

DOMAIN TAXONOMY

1 Interfaces

1.1 Infrastructure (SIRSI, UCS, etc.)

1.2 Media providers

2 Operational Modes

2.1 Classes of Service (research, public)

2.2 Training

2.3 Graceful Degradation and Recovery

3 Capabilities

3.1 Media Handled

3.2 Media Operations

3.3 Help

3.4 Administration

REQUIREMENTS

5 Interface Requirements

3 Required States and Modes

4 Capability Requirements

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

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

Document: Done

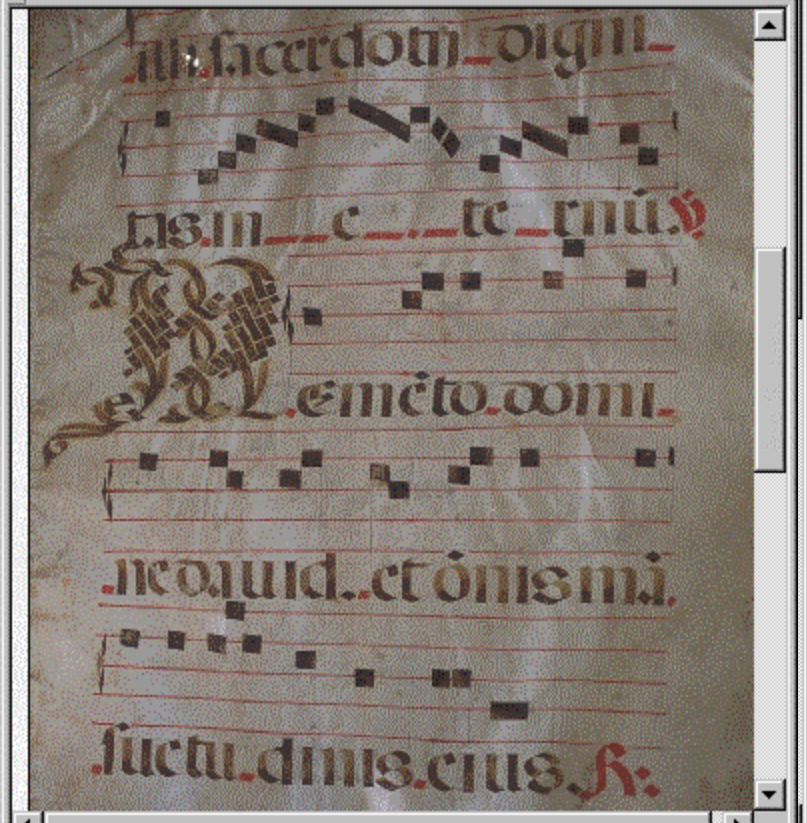
Digital Manuscript Archive Display Leaf Image Page -...

File Edit View Go Communicator Help

Back Forward Reload Home Search Guide Print Se

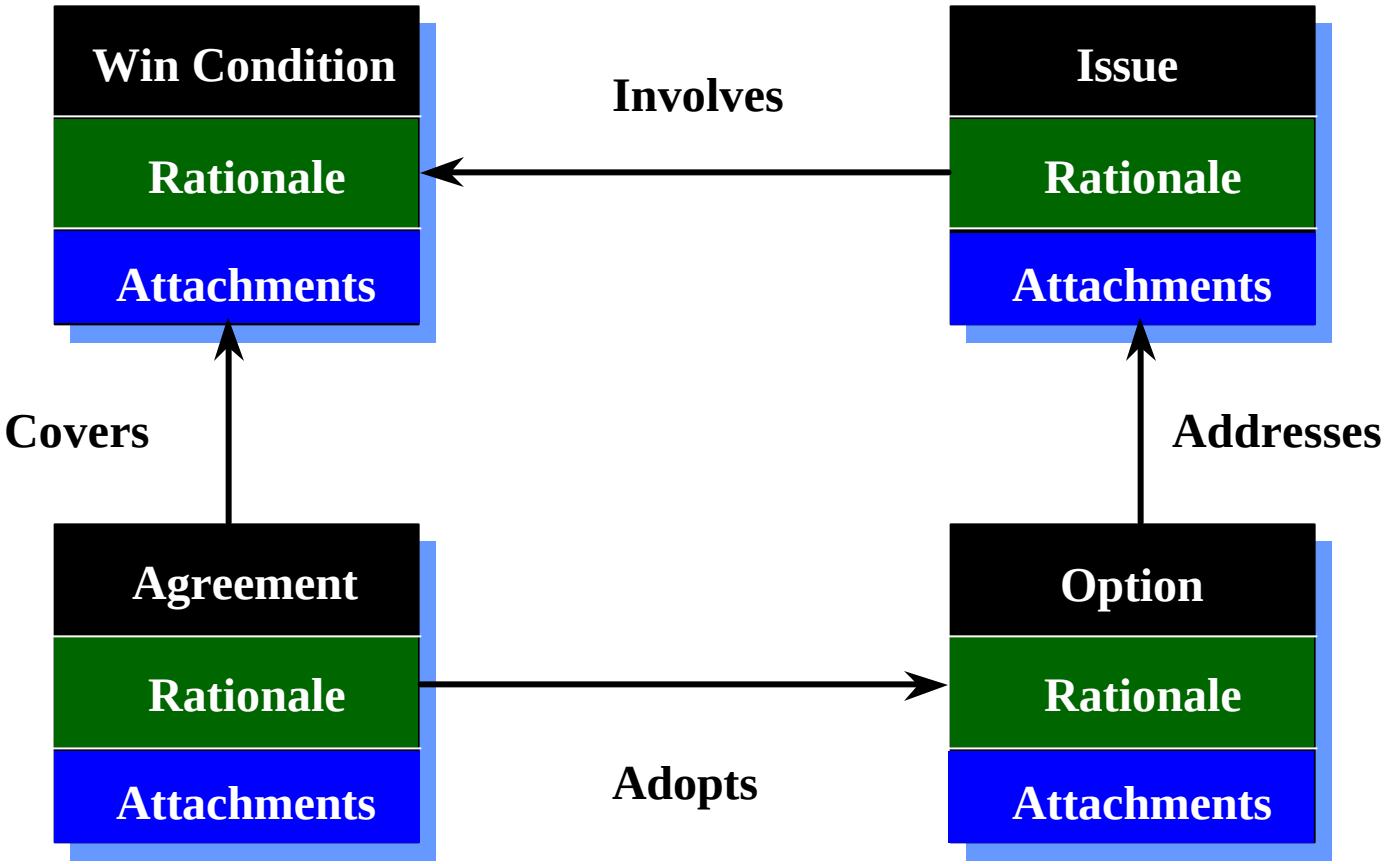
Bookmarks Location: tiphonarium&pageno=10&m=1&p=0

Internet New and Cool Look Up

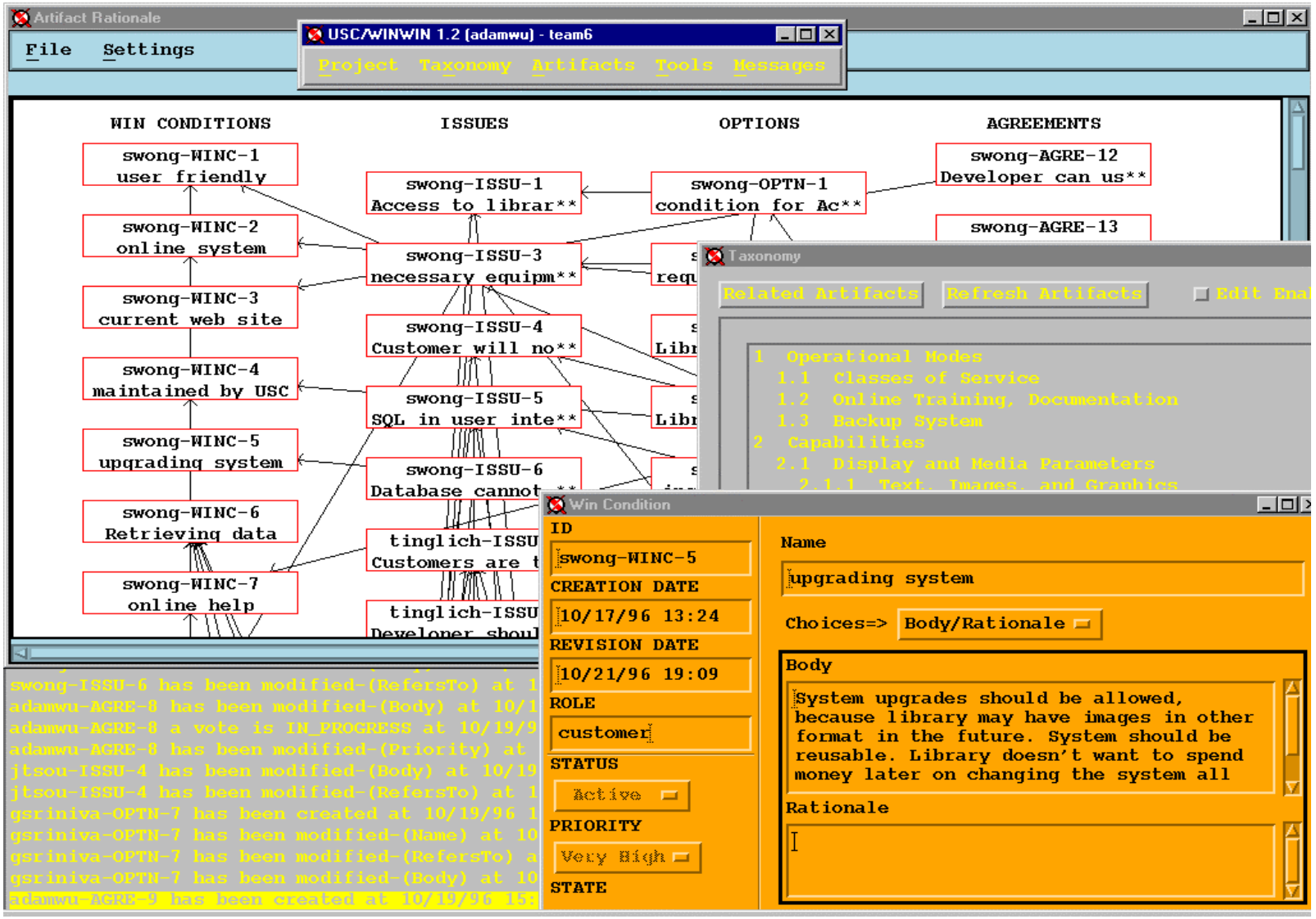


Document: Done

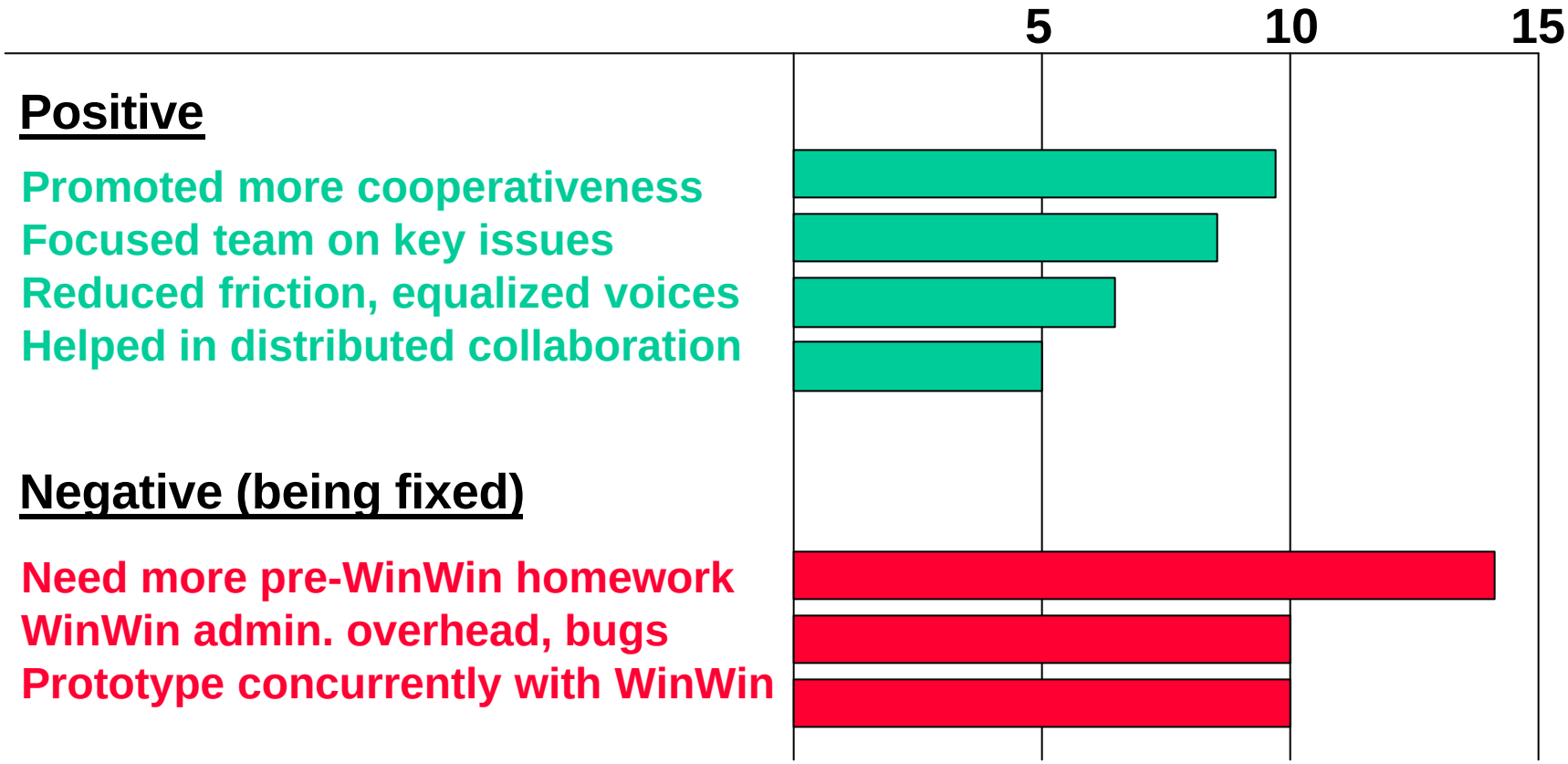
WinWin Negotiation Model



WinWin Look and Feel



Primary WinWin Critique Comments



Transition Between Cycle 2 and 3

- Smaller class (30 vs. 86 students)
  - Not a MS-CS core course
- Mixed team backgrounds from Cycle 1 and 2
  - Biggest problem: reconciling architectures from candidate LCA packages
- Some serious personality conflicts
  - Made workable via reassignments
- Several Library asset assumptions invalid
  - Server, SIRSI package, search engine
- New application priorities
  - Performed, risk-driven rebaselining of LCA packages

Project Results: 1996-97

- All products completed on schedule
- Librarians generally enthusiastic about products

  3-Committed to implementation
    - Cinema-TV, Business School, Tech. Reports

  2-Investing in further effort
    - Latin American Pamphlets, Medieval Manuscripts

  1-Awkward synthesis of 3 applications
    - 3 photo archives not equivalent
- Continuing in 1997-98
  - 20 candidate Library projects; mostly new

1997-98 Results

- Better integrated LCO, LCA packages
  - LCO: 103 vs. 160 pages
  - LCA: 154 vs. 230 pages
  - Stronger ARB reviews
- Higher client satisfaction ratings
  - 4.7 vs. 4.4 on scale of 5
- More effective transition into practice
- Improvements still needed for 1998-99
  - Rational Rose and OO training
  - Better integrated LCO, LCA packages
  - Better integrated architecture, product elements

MBASE Project Experience at USC/Columbia

Metric	USC 1996-97	USC 1997-98	USC 1998-99	Columbia U-grad. 99	Columbia Grad. 99
Fall Semester: LCA Package					
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	-	-
Spring Semester: IOC Package				Remained the same	
Teams	6	5	6	since projects were only	
Students	28	23	28	one semester long	
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	4	-	-
Applications used	1	3	TBD	10	5
Client evaluation	-	4.15	4.3	4.44	4.21

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**

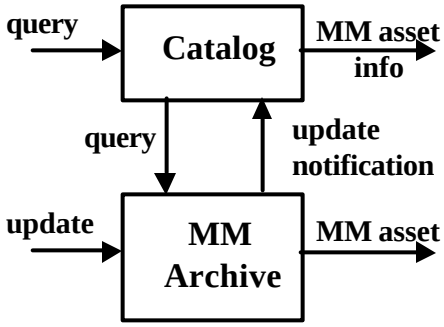
  "It was nice that you could add this access feature, but it overly (centralizes, decentralizes) control of our intellectual property rights."

  "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		1, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 20, 31, 32, 35, 36, 37, 39	· Use standard query languages · Use standard or COTS search engine · Uniform media formats	· Natural language processing · Automated cataloging or indexing · Digitizing large archives · Digitizing complex or fragile artifacts · Rapid access to large Archives · Access to heterogeneous media collections · Automated annotation/description/ or meanings to digital assets · Integration of legacy systems

The Results

- **Projects That Failed LCO Criteria**
  - **1996: 4 out of 16 (25%)**
  - **1997: 4 out of 15 (27%)**
  - **1998: 1 out of 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**

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
  - Now using SEI Electronic Process Guide tool
- Results transitioning to early adopters
- Ultimate goal: Model-integrated SW Engr. agents

Initial Cognitive Demands Analysis for MBASE Inception Phase

Technology / Task	Artifact Guidelines	Domain Models	WinWin Tool	UML/Rose Tool	COCOMO Tool	Risk, SC Checklist	Other
Team Formation	Understand required artifact skills LRC	Understand required domain skills CH				Understanding assess team personnel risks LR	Collaborate among team; team formation negot. team rules LR
Select, tailor domain model	Understand OpCon guidelines LRC	Select, adapt domain model LRCH		Understand tool op. Develop top-level use cases LRCTH			Collaborate w/ client on domain model tailoring LR
Negotiate stakeholder win-win agreements	Understand OpCon, Rqts., Plan G/L's LRC	Apply domain taxonomy as checklist LRC	Understand tool oper., Devel., negot., WW artifacts LRCTH		Understand tool oper. Devel. negot. COCOMO est's. LRCTH	Manage stakeholder expectations LRH	Collab. among team members & client on winwin agreements LRCT
Prototype key product features						Formulate risks needing proto-Types LR	Understand, select proto. tools Develop proto's. Iterate with clients LRCT
Develop LCO package artifacts	Apply artifact guidelines LRC	Iterate adaptation of domain model LRCH	Use, iterate WinWin agreements LRC	Develop top-level OO artifacts LRCTH	Perform cost/sched. tradeoff analyses LRC	Develop, resolve list of top risks LRC	Collab. among team, client to balance LCO package LRC

V – videos, L – lectures, G/L – guideline, WW – winwin, S – simulations, R – readings, H – homework, SH – stakeholder, D – decision aids,
C – case studies, T – tutorial, S/C – simplifier/comp.

Prospective SEDA Model Clash Aids

Success Model – Property Model Clash

You have a model clash between your

Success Model: Complete development in 12 weeks

and your

Property Model: COCOMO II schedule estimate of 20 weeks

You have two primary options to remove the model clash. They are complementary and can be pursued in parallel.

1. Adjust your project's cost/schedule driver decisions to produce a shorter estimated schedule (e.g., via reuse, better tools, more stable infrastructure). Click on the S-COST tool to get help in assessing your options.
2. Use a schedule-as-independent-variable (SAIV) process model. Click on the SAIV section of the MBASE Guidelines for how-to information.

Process Model – Product Model Clash

You have a model clash between your

Process Model: Schedule as Independent Variable (SAIV)

and your

Product Model: Unprioritized requirements

If you proceed as-is and need to drop requirements to meet schedule, you will lose valuable time working out their relative priorities. It's best to do it now.

Conclusions: Digital Library Projects

- WinWin Spiral process well matched to multimedia applications
- Model-integrated (MBASE) approach enabled rapid architecting and development
- WinWin approach built trust
  - Enabled cooperative adaptation to change
- Successful 1997-98 improvement cycle
  - Concurrent WinWin negotiation and prototyping
  - Smaller teams; less artifact overlap; integrated automation (Rational ROSE; Web)
  - Higher client satisfaction; tighter specs
- Further projects, improvements identified for 1998-99

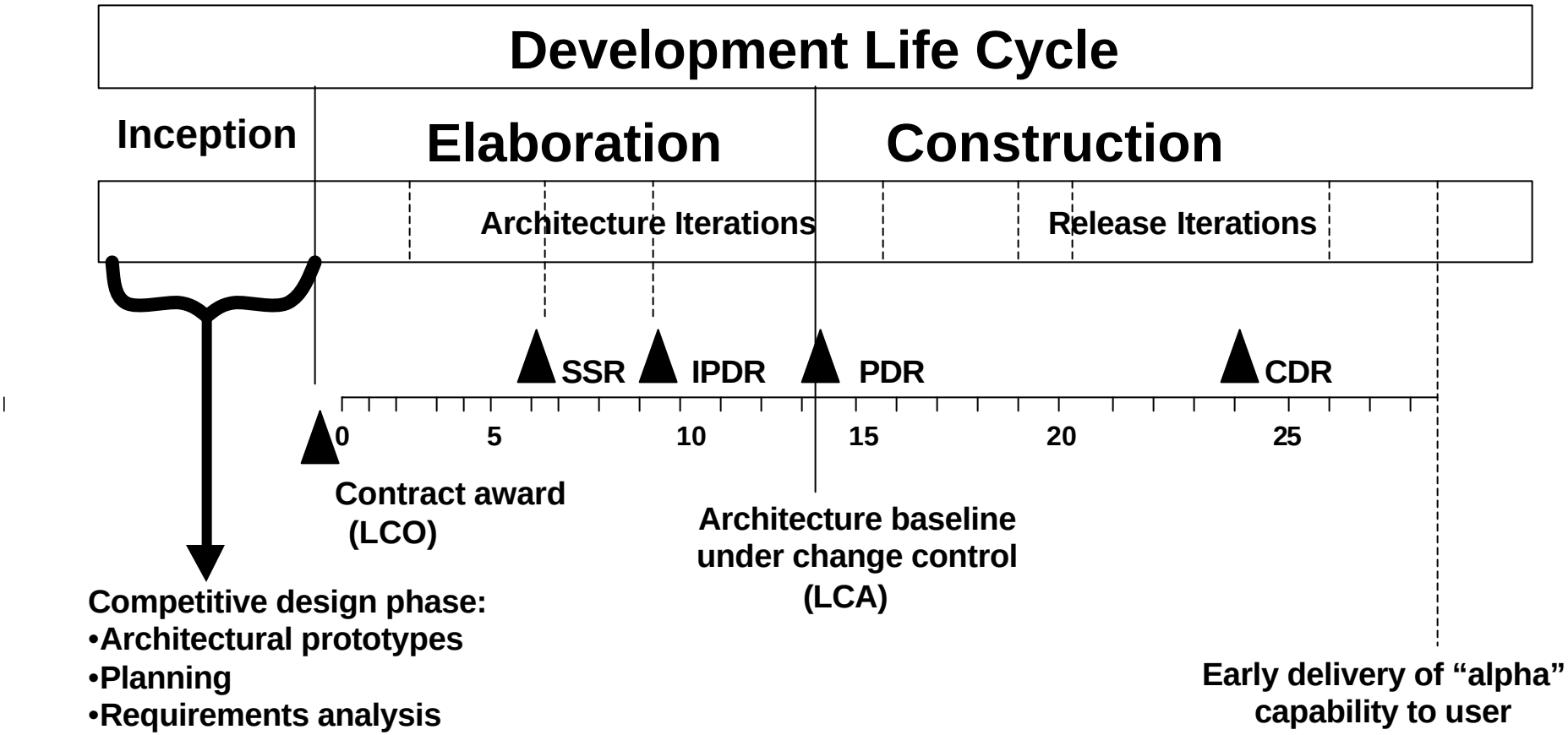
Case Study: CCPDS-R Project Overview

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



Detailed Guidelines

- **LCO/LCA Deliverables**
  - Operational Concept Description (18 pp.)
  - System and Software Requirements Description (15 pp.)
  - System and Software Architecture Description (16 pp.)
  - Life Cycle Plan (19 pp.)
  - Feasibility Rationale Description (13 pp.)
  - Appendices (12 pp.)
- **IOC Deliverables**
  - Detailed Construction Plan
  - Iteration Plans, Assessments
  - Test Plans, Procedures, Results
  - Inspection Plans, Reports
  - Detailed Design, Code, Release Notes
  - Transition Plan and Results
  - User's Manual, Training

Early Adopters

- Rational, Xerox, FAA, Litton
- 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"

MBASE Conclusions

- Successfully used on Digital Library projects
  - And CCPDS-R MBASE precursor
- Key to reducing cycle time (USC RAD Workshop)
  - Top people and teambuilding
  - Prepositioning assets (people, tools, architectures, components, models)
- Key to mastering increasingly complex systems
- Complementary to, integrates existing partial models
  - CMM's, J-STD-016, ISO/IEC 12207, Architecture-based models
- Avoids many current model clashes
  - Due to uncoordinated model-element choices (mandated, legacy, default, arbitrary)

References

(Boehm et al. papers available at http://sunset.usc.edu)

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.

Web Site Information

- CS 577a Software Engineering I Home Page
  http://sunset.usc.edu/classes/cs577a_98/index.html
- CS 577b Software Engineering II Home Page
  http://sunset.usc.edu/classes/cs577b_99/index.html
  (contains links to current project teams' home pages)
- USC Chronicle Article 'Library Gives Real-World Challenges to Student Software Designers' by Eric Mankin
  http://sunset/news/Library_Gives_Real-World_.html