USC CSE
USC - Center for Software Engineering

Experiences with the WinWin Groupware System

Alexander Egyed and Barry Boehm
{aegyed, boehm}@sunset.usc.edu
http://sunset.usc.edu

UCI Presentation

June 3rd, 1999

USC CSE **Outline**
USC - Center for Software Engineering

- **Motivation for WinWin Approach**
- **Theory W and WinWin Elements**
- **WinWin Concept of Operation**
- **WinWin Spiral Model**
- **Real-World Library Projects**
- **Integrating Ethics**

USC CSE **Motivation for WinWin**
USC - Center for Software Engineering

- Establishes objectives and procedures for integrated product teams (IPTs)
  - What should the IPT participants try to do?
  - How should they proceed?
  - How will they know when they're done?
- The fundamental success condition
- Some common counterexamples

USC CSE **Theory W**
USC - Center for Software Engineering

The Fundamental Success Condition:

your project will succeed *if and only if you make winners of all the critical stakeholders*

- Usually: Users, customers, developers, maintainers
- Sometimes: Interfacers, testers, reusers, general public

=> It is the Foundation of the WinWin Negotiation Model

USC CSE **Win-Lose Evolves into Lose-Lose**
USC - Center for Software Engineering

Proposed Solution	"Winner"	Loser
Cheap, Sloppy Product ("Buyer knows best")	Developer & Customer	User
Lots of bells and whistles ("Cost-plus")	Developer & User	Customer
Driving too hard a bargain ("Best and Final offers")	Customer & User	Developer

---

USC CSE **Outline**
USC - Center for Software Engineering

- **Motivation for WinWin Approach**
- ⇨ **Theory W and WinWin Elements**
- **WinWin Concept of Operation**
- **WinWin Spiral Model**
- **Real-World Library Projects**
- **Integrating Ethics**

Theory W Management Steps
USC - Center for Software Engineering

1. Identify success-critical stakeholders
2. Identify stakeholders' win conditions
3. Identify win condition conflicts as issues
4. Negotiate top-level win-win agreements
   – Invent options for mutual gain
   – Explore option tradeoffs
   – Manage expectations
5. Embody win-win agreements into specs and plans
6. Elaborate steps 1-5 until product is fully developed
   – Confront, resolve new win-lose, lose-lose risk items

Inventing Options for Mutual Gain
- The four basic steps: Fisher and Ury

USC - Center for Software Engineering

Win-Win, Win-Lose, and Lose-Lose
USC - Center for Software Engineering

Situations:

Getting to WinWin
USC - Center for Software Engineering

COCOMO F-16 Example:

Getting to WinWin
USC - Center for Software Engineering

COCOMO F-16 Example:

Outline
USC - Center for Software Engineering

- **Motivation for WinWin Approach**
- **Theory W and WinWin Elements**
- ➡ **WinWin Concept of Operation**
- **WinWin Spiral Model**
- **Real-World Library Projects**
- **Integrating Ethics**

Slide 1: WinWin Artifact Relationship

Win conditions · involve · Taxonomy · Issues · address · Options · adopt · Agreements · cover

Slide 2: 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

Slide 3: WinWin System

Customer · Developer · ... · User — **Distributed Stakeholders**

Internet / Intranet

WinWin System — **Rapid Requirements Negotiation Tool**

A4	Rapide	COCOMO II
• Performance	• Performance	• Cost
• Reliability	• Behavior	• Schedule

Tradeoff Analysis Tools

Slide 4: WinWin System

Slide 5: Outline

- **Motivation for WinWin Approach**
- **Theory W and WinWin Elements**
- **WinWin Concept of Operation**
- **WinWin Spiral Model** ⇒
- **Real-World Library Projects**
- **Integrating Ethics**

Slide 6: WinWin Spiral Model

WinWin Extension to the Spiral Model

1. Identify next-level Stakeholders
2. Identify Stakeholders' win conditions
3a. Reconcile win conditions
3b. Establish next level objectives, constraints, alternatives
4. Evaluate product and process alternatives. Resolve Risks
5. Define next level of product and process – including partitions
6. Validate product and process definitions
7. Review, commitment

Original Spiral Model

LCO/LCA Milestone Elements

Milestone Element	Life Cycle Objectives (LCO)	Life Cycle Architecture (LCA)
Definition of Operational Concept		
Definition of System Requirements		
Definition of System and Software Architecture		
Definition of Life-Cycle Plan		
Feasibility Rationale		

LCO/LCA Milestones in Detail

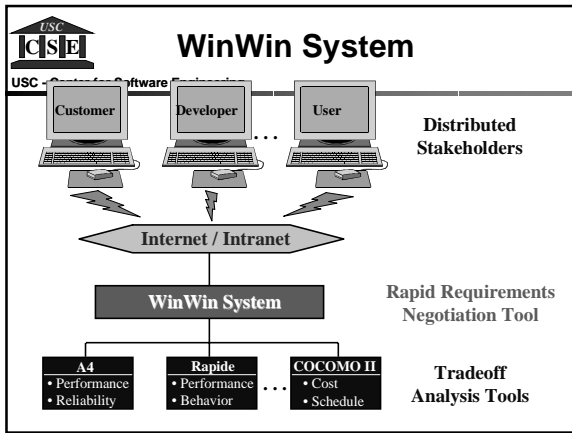| Milestone Element	Life Cycle Objectives (LCO)	Life Cycle Architecture (LCA)
Definition of Operational Concept	Top-level system objectives and scope System boundary - Environment parameters and assumptions - Evolution parameters Operational concept - Operations and maintenance scenarios and parameters - Organizational life-cycle responsibilities	Elaboration of system objectives and scope by increment Elaboration of operational concept by increment
Definition of System Requirements	Top-level functions, interfaces, quality attribute levels, including: - Growth vectors - Priorities Stakeholders' concurrence on essentials	Elaboration of functions, interfaces, quality attributes by increment - Identification of TBDs (to-be-determined) 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 TBDs 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 management plan

---

Outline

- **Motivation for WinWin Approach**
- **Theory W and WinWin Elements**
- **WinWin Concept of Operation**
- **WinWin Spiral Model**
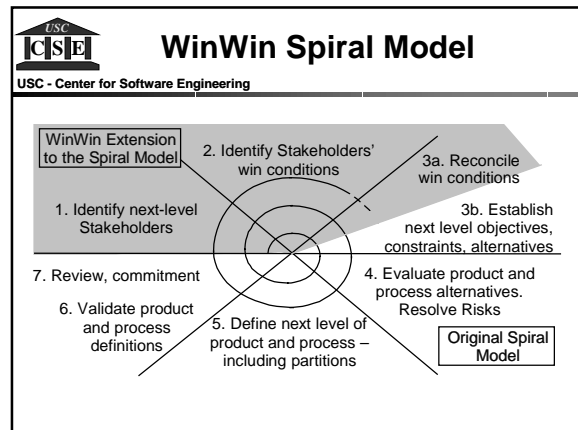- ➡ **Real-World Library Projects**
- **Integrating Ethics**

The Challenge

- **Roughly 15 Digital Library Applications a year**
  - 2 sentence problem statements
  - Librarian clients
- **Roughly 90 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

1996-98 Library Projects

1996-1997 Projects	1997-1998 Projects
Cinema-TV Moving Images[1]	Architecture & Fine Arts Databases
EDGAR Corporate Data	Bella Lewitsky Archives
Hancock Image Archive	Business School Working Papers[2]
Interactive TV Material	Inter-Library Loan[1]
Korean-American Museum	Engineering Technical Reports[2]
Latin American Pamphlets[1]	General Library FAQ's
Digital Maps	Hancock Museum Virtual Tour[1]
Medieval Manuscripts[1]	Lion Feuchtwanger Archive
Planning Documents[2]	Network Consultation Support
Searchable Archives for Images[2]	Serial Publication[1]
Stereoscopic Slides[2]	Statistical Charts[1]
Technical Reports[1]	Virtual Education Reference Assistant

(1) projects were continued for a second semester.

(2) projects were merged together and continued for a second semester.

Digital Manuscript Project

Moving Image Archive Project
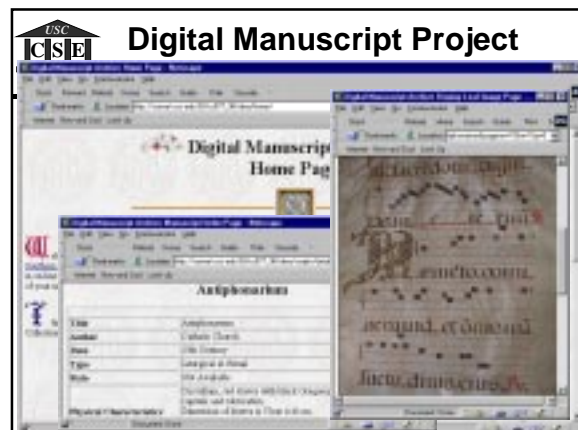
Milestones

USC - Center for Software Engineering

- WinWin Requirements Negotiation on October 21
- Life Cycle Objectives (LCO) on November 4
- Life Cycle Architecture (LCA) on December 4

- Revised Requirements, Plan, etc. on Feb 11
- Design Plan, Test Plan, Inspection Plan on March 18
- Test Report, Inspection Report on April 8
- Initial Operational Capabilities (IOC), etc. on April 22

- Further increments throughout the summer
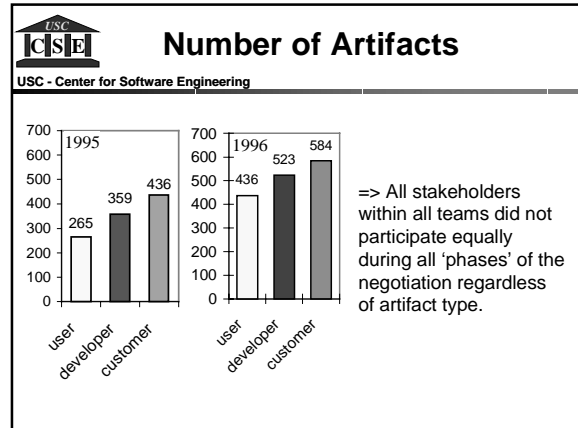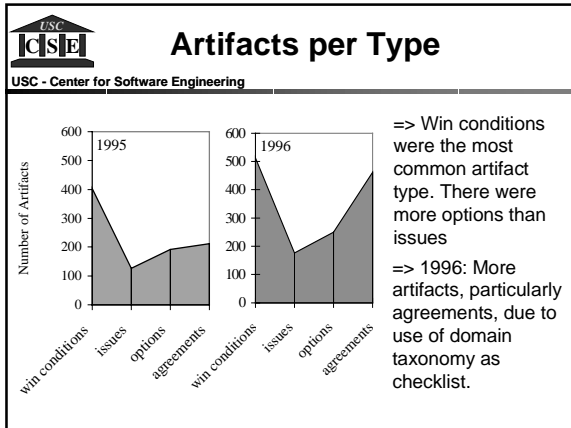- All deliverables completed on time; Library clients highly satisfied with results

Stakeholder Win Conditions

USC - Center for Software Engineering

Stakeholders	Win conditions
Developers (Students)	• Full range of software engineering skills • Real-client project experience • Advanced software technology experience
Customers (Librarians)	• Useful applications • Advanced software technology understanding • Moderate time requirements
Faculty and Staff	• Educate future software engineering leaders • Better software engineering technology • Applied on real-client projects

Some Real-World Problems

USC - Center for Software Engineering

- **Availability of Equipment (server), Tools, and COTS (SIRSI) packages.**
- **Fuzzy and Unstable Requirements.**
- **Librarians were not available all the time.**
- **Personnel turnover: The second non-core-CS course is always much smaller.**
- **Personnel conflicts.**
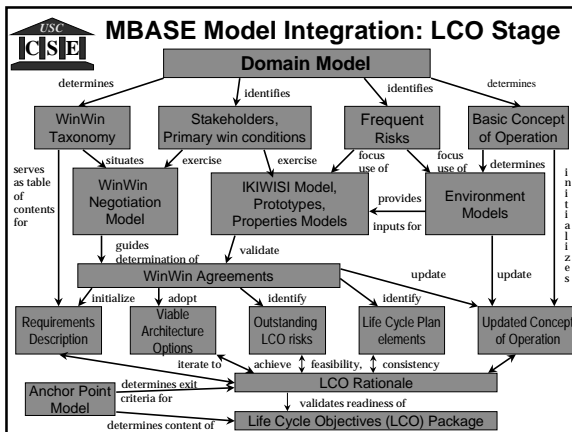
MBASE Model Integration: LCO Stage

Outline

USC - Center for Software Engineering

- **Motivation for WinWin Approach**
- **Theory W and WinWin Elements**
- **WinWin Concept of Operation**
- **WinWin Spiral Model**
- **Real-World Library Projects**
  - **Metrics**
  - **Summary**
- **Integrating Ethics**

Artifacts per Type

Number of Artifacts — 1995, 1996 — win conditions, issues, options, agreements

=> Win conditions were the most common artifact type. There were more options than issues

=> 1996: More artifacts, particularly agreements, due to use of domain taxonomy as checklist.

Number of Artifacts

1995: user 265, developer 359, customer 436
1996: user 436, developer 523, customer 584

=> All stakeholders within all teams did not participate equally during all 'phases' of the negotiation regardless of artifact type.

Most Artifacts per Team/Role

Artifacts 1995/1996	Win conditions	Issues	Options	Agreements
Customer	18/10	9/4	10/7	15/5
Developer	6/0	15/9	12/5	10/8
User	4/4	4/1	6/2	7/1

=> customers and users were more important during goal identification. Developers were more important during risk (issue) identification and resolution

Creation/Revision Table

1995 — customer, developer, user; 1996 — customer, developer, user

=> Stakeholders participated mostly at the same time. However sessions were much more frequent in 1996, likely due to real-client involvement

Duration to resolve Artifacts

1995, 1996 — win conditions, issues, options, agreements

=> It took much more time to resolve artifacts in 1996

Outline

- **Motivation for WinWin Approach**
- **Theory W and WinWin Elements**
- **WinWin Concept of Operation**
- **WinWin Spiral Model**
- **Real-World Library Projects**
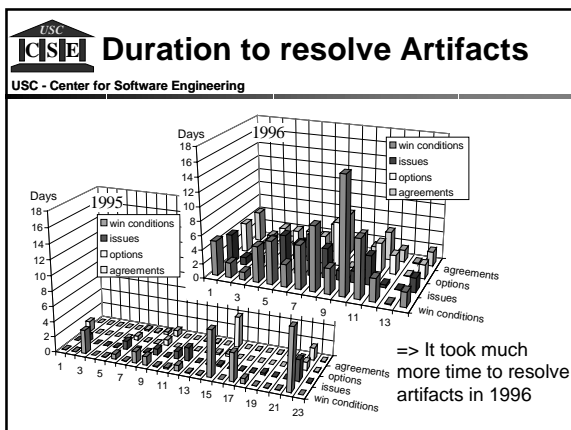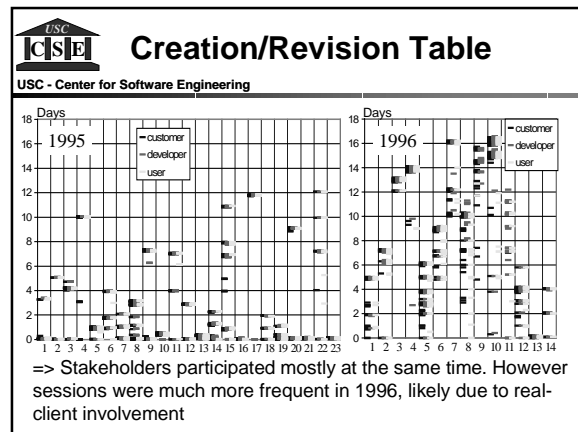 - **Metrics**
  - **Summary**
- **Integrating Ethics**

Project Critiques

Positive
Promoted more cooperativeness
Focused team on key issues
Reduced friction, equalized voices
Helped in distributed collaboration

Negative (being fixed)
Need more pre-WinWin homework
WinWin admin. overhead, bugs
Prototype concurrently with WinWin

(scale: 5, 10, 15)

WinWin Spiral Model Results

- **Used to architect 31 digital library products**
  - **For USC Library**
  - **Using 6-person student teams**
- **Two spiral cycles using LCO and LCA milestones**
  - **Developed Ops Concept, Requirements, Architecture, Development Plan, Prototype, Rationale**
  - **Used WinWin tool, Arch. Review Boards**
- **Librarians excited by results**
  - **Committed to implementing top products**
  - **Convinced that Win-Win approach works**

WinWin Benefits

- **Gets key stakeholders involved**
- **Provides collaborative operational guidelines**
- **Provides criteria for evaluating success**
- **Reduces cycle time**
  - **Especially for distributed collaboration**
- **Complements other key front-end methods**