A stakeholder win–win approach to software engineering education

Barry Boehm [a], Alexander Egyed [a], Dan Port [a], Archita Shah [a],
Julie Kwan [b] and Ray Madachy [c]

[a] *Center for Software Engineering, Henry Salvatori Computer Science 300,*
University of Southern California, Los Angeles, CA 90089-0781, USA
E-mail: {boehm,aegyed,dport,ashah}@sunset.usc.edu
[b] *USC University Libraries, University of Southern California, Los Angeles, CA 90089-1421, USA*
E-mail: jkwan@calvin.usc.edu
[c] *Litton Data Systems, Guidance & Control Systems, 5500 Canoga Avenue, Woodland Hills,*
CA 91367-6698, USA
E-mail: madachy@sunset.usc.edu

We have been applying the stakeholder win–win approach to software engineering education. The key stakeholders we are trying to simultaneously satisfy are the students; the industry recipients of our graduates; the software engineering community as parties interested in improved practices; and ourselves as instructors and teaching assistants. In order to satisfy the objectives or win conditions of these stakeholders, we have formed a strategic alliance with the USC Libraries to have software engineering student teams work with Library clients to define, develop, and transition USC digital library applications into operational use. This adds another set of key stakeholders: the Library clients of our class projects. This paper summarizes our experience in developing, conducting, and iterating the course. It concludes by evaluating the degree to which we have been able to meet the stakeholder-determined course objectives.

1. Key software engineering education stakeholders and their win conditions

Since establishing USC's MS-degree program in software engineering in 1993, we have been working with our various stakeholders to determine their primary win conditions for the MS program in general, and for our 2-semester Software Engineering core course in particular. The results are summarized in table 1.

Our students are a 70–30% mix of (1) fresh BA/BS computer science graduates and (2) experienced practitioners in industry, who mostly take the courses via our instructional television network. Both groups of students want a mix of long half-life fundamental knowledge and skills, and marketable near-term skills based on project experience. They want a reasonable workload level (averaging roughly 12 hours/week for a four-unit course), fair grading (a challenge with their mixed backgrounds), and a rewarding intellectual and social experience.

We determine industry (and government) win conditions from our 26 USC-CSE

Table 1
Software engineering education stakeholders and win conditions.

Stakeholders	Win conditions
Students	• Lifelong-value knowledge and skills • Marketable near-term skills, experience • Fair workload, grading • Rewarding intellectual, social experience
Industry	• Lifelong-value knowledge and skills • Relevant near-term skills, experience • Full-spectrum coverage of life-cycle, discipline (people, economics, domain) skills • High volume of graduates • Distance learning opportunities
Class-project clients	• Well-engineered products • Easy to transition and use • Efficient use of client time • Minimum operational disruption • Client knowledge and skills developed in information technology
Software engineering community	• Experience-based improvements in software engineering tools and techniques
Instructors, teaching assistants	• Create, sustain stakeholder win–win enterprise • Confront, resolve win–lose risks • Contribute to software engineering and SE education state-of-art, state-of-practice • Manageable workload • Rewarding intellectual, social experience

Affiliate organizations, including a mix of large and small commercial and aerospace organizations. Those organizations want to hire (or nurture) graduates with a mix of long-term and near-term knowledge and skills; including full-spectrum coverage of life-cycle skills (not only programming but also system engineering, requirements engineering, software architecting, testing, transition, and maintenance) and discipline skills (not only computer science by also applications-domain, economics, management, and people skills).

Given today's software skill shortages, they would like us to produce a large number of graduates. And they are increasingly willing to support TV downlinks and videoconference facilities to enable distance learning. Of course these win conditions give us significant challenges in satisfying some of the other win conditions, particularly in conducting team projects.

Our USC Libraries class-project clients have highly representative software client win conditions: well-engineered, easy-to-use products, with an emphasis on effective transition as the developers soon graduate and disappear. The USC CIO and Dean of the USC Libraries, Dr. Jerry Campbell, has been very supportive in stimulating librarians to serve as clients and enhance their information technology skills. But there is still a strong emphasis on students' not wasting librarians' time, and not disrupting Library Services as they develop their products.

The course provides us with an opportunity to experiment with improved software engineering processes. Normally, research in software engineering processes has a long gestation period before one can tell whether a new approach improves the process and avoids harmful side effects. With the digital library projects, we have happened onto the software engineering research equivalent of the fruit fly: a relatively large number of projects for which there is annual feedback on the efficiency of new approaches. Given our mix of experienced and inexperienced teams, we can also assess the degree to which the new methods are likely to apply to industrial practice.

Finally, as instructors and teaching assistants, we subscribe to Theory W [Boehm and Ross 1989]: that we will succeed to the extent to which we make winners of our critical stakeholders. As with the students we want to do this at a reasonable workload level and to have it be a rewarding intellectual and social experience.

These stakeholders' win conditions have thus become our objectives in developing, executing, evaluating, and improving the course. Section 2 summarizes our overall course development and evolution strategy. Section 3 describes our initial execution of the course in 1996–1997. Section 4 summarizes our evaluation results and improvements in giving the course in 1997–1998. Section 5 concludes with an evaluation of our current status and plans with respect to the course objectives.

2. Course development and evolution strategy

In developing and evolving the course, we have been using a combination of the WinWin Spiral Model [Boehm and Bose 1994] and the Experience Factory [Basili *et al.* 1986]. The approach is summarized in figure 1. We used the WinWin Spiral Model to develop the initial version of the course and its instrumentation. Each project uses the WinWin Spiral Model to define, develop, and transition its application product. We then analyze the course instrumentation results, student critiques, client evaluations, and grading information to determine improvements for the course in the following year, using the Experience Factory paradigm. Also, in developing and evolving the course, we have tried to anticipate and adapt to changes in applications technology (COTS, Web technology, etc.) and software engineering technology (Unified Modeling Language, Java, etc.).

2.1. The WinWin Spiral Model

The original spiral model [Boehm 1988] uses a cyclic approach to develop increasingly detailed elaborations of a software system's definition, culminating in incre-

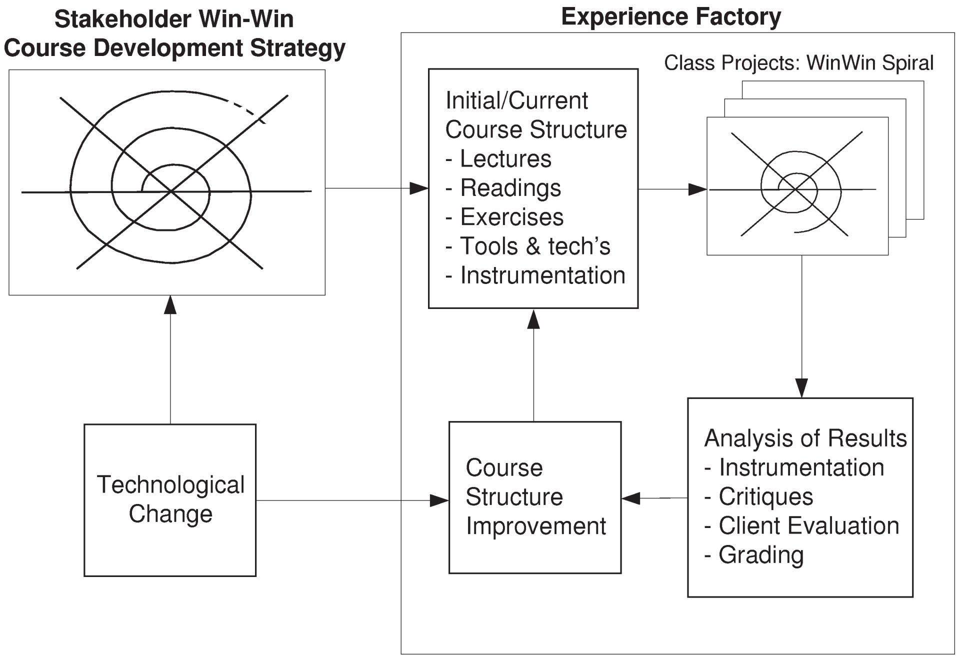


Figure 1. Course development and evolution strategy.

mental releases of the system's operational capability. Each cycle involves four main activities:

- Elaborate the system or subsystem's product and process objectives, constraints, and alternatives.
- Evaluate the alternatives with respect to the objectives and constraints. Identify and resolve major sources of product and process risk.
- Elaborate the definition of the product and process.
- Plan the next cycle, and update the life-cycle plan, including partition of the system into subsystems to be addressed in parallel cycles. This can include a plan to terminate the project if it is too risky or infeasible. Secure the management's commitment to proceed as planned.

The Spiral Model has been extensively elaborated (e.g., [SPC 1994]), and successfully applied in numerous projects (e.g., [Royce 1990; Frazier and Bailey 1996]). However, some common difficulties have led to some further extensions to the model.

One difficulty involves answering the question, "Where do the elaborated objectives, constraints, and alternatives come from?" The WinWin Spiral Model resolves this difficulty by adding three activities to the front of each spiral cycle, as illustrated in figure 2 [Boehm and Bose 1994]:

- Identify the system or subsystem's key stakeholders.
- Identify the stakeholders' win conditions for the system or subsystem.

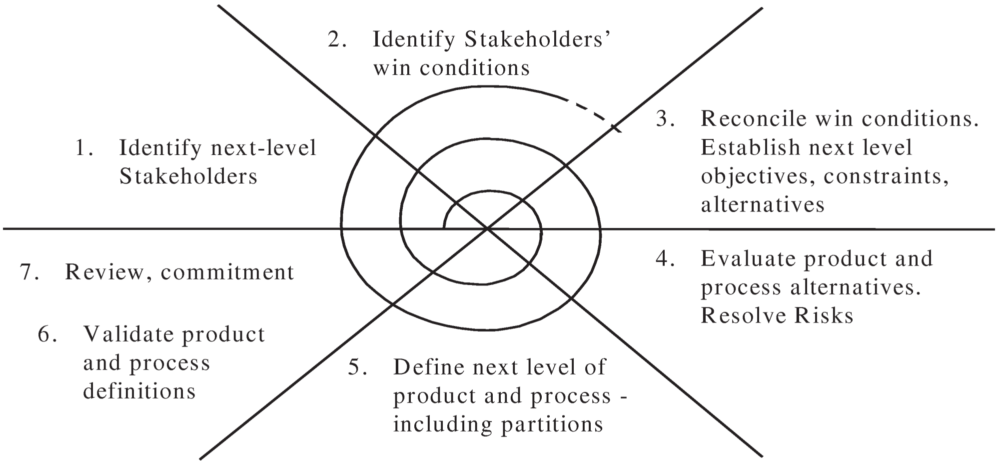


Figure 2. The WinWin Spiral Model.

- Negotiate win-win reconciliations of the stakeholders' win conditions.

In an experiment involving a bootstrap application of the WinWin groupware system to the definition of an improved version of itself, we found that these steps indeed produced the key product and process objectives, constraints, and alternatives for the next version [Boehm *et al.* 1994]. The overall stakeholder WinWin negotiation approach is similar to other team approaches for software and system definition such as CORE [Mullery 1979], gIBIS [Conklin and Begeman 1988], Viewpoints [Finkelstein *et al.* 1992], GRAIL [Dardenne *et al.* 1993], Tuiqiao [Potts and Takahashi 1993], Participatory Design, and JAD [Carmel *et al.* 1993]. Our primary distinguishing characteristic is the use of the stakeholder win–win relationship as the success criterion and organizing principle for the software and system definition process. Our negotiation guidelines are based on the Harvard Negotiation Project's techniques [Fisher and Ury 1981].

2.2. Process anchor points

Another difficulty in applying the Spiral Model across an organization's various projects is that the organization can be left with no common reference points around which to organize its management procedures, cost and schedule estimates, etc. In the process of working out this difficulty with our COCOMO II cost model industry and government Affiliates (see Acknowledgments), we found a set of three process anchor points which could be related both to the completion of spiral cycles and to the organization's major decision milestones. Two of these, the Life Cycle Objectives (LCO) and Life Cycle Architecture (LCA) milestones, are elaborated in table 2. The third, the Initial Operational Capability (IOC), is summarized in table 3. These anchor points are further elaborated and related to WinWin Spiral Model cycles in [Boehm 1996]. We also found that the LCO and LCA milestones are highly compatible with

Table 2
Contents of LCO and LCA milestone.

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 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 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 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

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

Table 3
Contents of the Initial Operational Capability (IOC) milestone.

The key elements of the IOC milestone are:

- *Software preparation*, including both operational and support software with appropriate commentary and documentation; data preparation or conversion; the necessary licenses and rights for COTS and reused software, and appropriate operational readiness testing.
- *Site preparation*, including facilities, equipment, supplies, and COTS vendor support arrangements.
- *User, operator and maintainer preparation*, including selection, teambuilding, training and other qualification for familiarization usage, operations, or maintenance.

the use of the successful Architecture Review Board practice pioneered by AT&T and Lucent Technologies [AT&T 1993].

2.2.1. Course initiation

Initiation of the project course involved a WinWin Spiral Model cycle with the USC Libraries personnel to achieve a Life Cycle Objectives (LCO) milestone for the family of digital libraries projects to be developed as class projects. The class projects then each involved three spiral cycles to achieve their LCO, LCA, and IOC milestones: The overall sequencing was as follows:

- Cycle 0 (Summer 1996): Determining feasibility of an appropriate family of multimedia applications (project family LCO milestone).
- Cycle 1 (Fall 1996): Determining feasibility of individual applications (project LCO).
- Cycle 2 (Fall 1996): Achieving a feasible LCA project milestone for each application.
- Cycle 3 (Spring 1997): Achieving a workable project IOC for each application.

2.3. Cycle 0: Project family life cycle objectives

During 1993–1996, the USC-CSE experimented with teaching the WinWin Spiral Model in its core 100-student MS-level software engineering course, using representative but hypothetical applications. In 1995–1996, the application was a hypothetical advanced library application: a selective dissemination of information system using a form of "push" technology. Some of the library staff, primarily Kwan (then Director of the Science and Engineering Library), and Denise Bedford (then ILS Project Manager), detected an unusually high level of student interest in library operations resulting from this assignment. They followed up with the instructor (Boehm) to determine whether all of this student energy and talent could be channeled toward developing useful USC Library applications.

Table 4
Primary stakeholder win conditions.

Library information technology and users	Library operations and users	Center for software engineering
• Accelerated transition to digital library capabilities; Dean's vision • Evaluation of emerging multimedia archiving and access tools • Empowering library multimedia users • Enhancing library staff capabilities in high-performance online library services • Leveraging limited budget for advanced applications	• Continuity of service • No disruption of ongoing transition to SIRSI-based Library Information System • Operator career growth opportunities • No disruption of USC Network operations and services • More efficient operations via technology	• Similarity of projects (for fairness, project management) • Reasonable match to WinWin Spiral Model • 15–20 projects at 5–6 students per team • Meaningful LCA achievable in 1 semester • Meaningful IOC achievable in 2 semesters • Adequate network, computer, infrastructure resources

CSE had been looking for such a source of new applications, so in Summer 1996, Kwan, Bedford, Boehm, and Egyed (the prospective teaching assistant for the 1996–1997 software engineering course), explored each other's win conditions to determine whether a feasible set of life-cycle objectives for a family of USC Library applications could be identified. The most feasible applications area turned out to be the exploratory multimedia applications. Table 4 summarizes the win conditions for the three primary stakeholders: the Library information technology community, including its users; the Library operational community, including its users; and the Center for Software Engineering.

As indicated in table 4, the *Library information technology community* was energized by the vision of the new Dean of the University Libraries, Dr. Jerry Campbell, to accelerate the Libraries' transition to digital capabilities. A new dedicated computer-interactive facility, the Leavey Library, and the transition to the SIRSI client-server library information system were whetting users' appetites for advanced applications. However, there was little budget for evaluating emerging multimedia technology and developing exploratory applications.

The *Library operations community* and its users were already undergoing a complex transition to the new SIRSI system. They were continually on the lookout for new technology to enhance their operations, but also highly sensitive to the risks of disrupting continuity of service, and limited in their resources to experiment in new areas.

The *Center for Software Engineering* had a large pool of talent to develop exploratory applications, if the applications could fit within the constraints of student courses. These included not only schedule and computer resource constraints (e.g., 10 megabytes of disk storage per student), but also constraints on fairness of grading and

Problem Set #2: Photographic Materials in Archives
Jean Crampon, Hancock Library of Biology and Oceanography
There is a substantial collection of photographs, slides, and films in some of the Library's archival collections. As an example of the type of materials available, I would like to suggest using the archival collections of the Hancock Library of Biology and Oceanography to see if better access could be designed. Material from this collection is used by both scholars on campus and worldwide. Most of the Hancock materials are still under copyright, but the copyright is owned by USC in most cases.

Problem Set #8: 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.

Problem Set #9: Formatting Information
Caroline Sisneros, Crocker Business Library
Increasingly the government is using the WWW as a tool for dissemination of information. Two much-used sites are the Edgar Database of Corporate Information (http://www.sec.gov/edgarhp.htm) and the Bureau of the Census (http://www.census.gov). Part of the problem is that some of the information (particularly that at the EDGAR site) in only available as ASCII files. For information that is textual in nature, while the files can be cleaned up, formatting of statistical tables is often lost in downloading, e-mailing, or transferring to statistical programs. And while this information is useful for the typical library researcher, who usually have a very distinct information need, the investment in what it would take to put this information is a usable format is often too much trouble.

Problem Set #13: Moving Image Archive
Sandra Joy Lee, Moving Image Archive, School of Cinema/TV
The USC Moving Image Archive houses USC student film and video productions dating from the 1930s to current productions in the School of Cinema–Television. Moving image materials in multiple formats, specialized viewing equipment, limited storage space, and complex access needs create challenges that may be solved with new computer technologies. Fifteen movie clips (.mov format), each approximately 45 minutes in length, over 100 digital film stills (.gif format), and textual descriptions of the films will be made available to students wishing to explore this project.

Figure 3. Example library multimedia problem statements.

available instructor and teaching assistant time, which translated into the need for a family of highly similar (but not identical) projects.

During Summer 1996, Kwan and Bedford identified a set of candidate Library multimedia projects and clients, and provided brief summaries of each. Examples are shown in figure 3. Successful convergence on the project-family LCO milestone was achieved by an exchange of memoranda between the Library and the CSE. A memo from Boehm to Charlotte Crockett, Director of the Leavey Library, summarized the proposed set of projects, the potential Library costs and risks and how they would be addressed, and the envisioned Library benefits in terms of their win conditions. A memo to Boehm from Lucy Wegner, the Library's interim Assistant Dean for Information Technology, provided specific constraints under which the Library would participate (e.g., no disruption of Library services; no interference with other librarian responsibilities; use of only the Library's test LIS host, only after LIS testing was complete; no advance commitments to use the results or to continue into product development in Spring 1997).

2.4. Resulting initial course structure

Figure 4 shows the multimedia archive project guidelines as provided to the Library staff during Cycle 0 and provided to the students on the first day of class, August 28, 1996. The guidelines provided about 2½ weeks for the students to organize into teams, and 11½ weeks to complete the LCO and LCA milestones.

In addition, the projects were provided with guidelines for developing each of the five documents indicated in the Product Objectives of figure 4, including approximate page budgets for the LCO and LCA version of the documents. They were also provided with guidelines and an example of a multimedia archive prototype, and a domain model for a typical information archive extension (figure 5). The domain model identifies the key stakeholders involved in such systems, and such key concepts as the system boundary: the boundary between the system being developed and its environment.

Figure 6 shows the course schedule: for each course day, the lecturers other than Boehm, the course assignments, and the readings. Readings are either chapters from the [Sommerville 1996] textbook (SE) or papers from a set of supplementary course notes (CN) covering such topics as the WinWin Spiral Model, anchor points, architecture review boards, library information systems, the WinWin and COCOMO tools used in the course, and annotated outlines for the documents required in the LCO and LCA packages (table 2).

The course assignments included individual homework for familiarization with the WinWin and COCOMO tools, group assignments for the WinWin negotiation results, LCO package, and LCA package, and an individual project critique. These are elaborated in figure 4.

The course lectures followed the WinWin Spiral Model in beginning with overviews of the project artifacts and how they fit together, and with key planning and organizing guidelines. The project teams were self-selected; a key risk management emphasis was on the risk of forming teams with incompatible people and philosophies. As a result, there were relatively few personnel problems during this phase, compared with previous offerings of the course. Later lectures provided more detail on the artifacts, plus guest lectures from Kwan and others on Library operations and the SIRSI system, and from experts in such areas as user interface design and multimedia system architecting.

Sessions were also used to conduct tailored Architecture Review Board reviews for some of the LCO and LCA packages. An extra day was also scheduled in late November to demonstrate all of the teams' prototypes.

3. Course execution

3.1. Cycle 1: Individual application life cycle objectives

The Fall 1996 course ended up with 86 students. Most were in 6-person teams. To accommodate special cases, including roughly 25 off-campus students, there were

Project Objectives
Create the artifacts necessary to establish a successful life cycle architecture and plan for adding a multimedia access capability to the USC Library Information System. These artifacts are:

1. An Operational Concept Definition.
2. A System Requirements Definition.
3. A System and Software Architecture Definition.
4. A Prototype of Key System Features.
5. A Life Cycle Plan.
6. A Feasibility Rationale, assuring the consistency and feasibility of items 1–5.

Team Structure
Each of the six team members will be responsible for developing the LCO and LCA versions of one of the six project artifacts. In addition, the team member responsible for the Feasibility Rationale will serve as Project Manager with the following primary responsibilities:

1. Ensuring consistency among the team members' artifacts (and documenting this in the Rationale).
2. Leading the team's development of plans for achieving the project results, and ensuring that project performance tracks the plans.

Project Approach
Each team will develop the project artifacts concurrently, using the WinWin Spiral approach defined in the paper "Anchoring the Software Process." There will be two critical project milestones: the Life Cycle Objectives (LCO) and Life Cycle Architecture (LCA) milestones summarized in table 2.
The LCA package should be sufficiently complete to support development of an Initial Operational Capability (IOC) version of the planned multimedia access capability by a CS577b student team during the Spring 1997 semester. The Life Cycle Plan should establish the appropriate size and structure of such a team.

WinWin User Negotiations
Each team will work with a representative of a community of potential users of the multimedia capability (art, cinema, engineering, business, etc.) to determine that community's most significant multimedia access needs, and to reconcile these needs with a feasible implementation architecture and plan. The teams will accomplish this reconciliation by using the USC WinWin groupware support system for requirements negotiation. This system provides facilities for stakeholders to express their Win Conditions for the system; to define Issues dealing with conflicts among Win Conditions; to support Options for resolving the Issues; and to consummate Agreements to adopt mutually satisfactory (win–win) Options.

There will be three stakeholder roles:

- Developer: The Architecture and Prototype team members will represent developer concerns, such as use of familiar packages, stability of requirements, availability of support tools, and technically challenging approaches.
- Customer: The Plan and Rationale team members will represent customer concerns, such as the need to develop an IOC in one semester, limited budgets for support tools, and low-risk technical approaches.
- User: The Operational Concept and Requirements team members will work with their designated user-community representative to represent user concerns, such as particular multimedia access features, fast response time, friendly user interface, high reliability, and flexibility of requirements.

Figure 4. Multimedia archive project guidelines.

Major Milestones

September 16	– All teams formed
October 14	– WinWin Negotiation Results
October 21,23	– LCO Reviews
October 28	– LCO Package Due
November 4	– Feedback on LCO Package
December 6	– LCA Package Due, Individual Critique Due

Individual Project Critique

The project critique is to be done by each individual student. It should be about 3–5 pages, and should answer the question, "If we were to do the project over again, how would we do it better – and how does that relate to the software engineering principles in the course?"

Figure 4. (Continued.)

Table 5
Library multimedia applications.

Team	Application	Client
1.*	Stereoscopic slides	John Ahouse
2.**	Latin american pamphlets	Barbara Robinson
3, 5.**	EDGAR Corporate data	Caroline Cisneros
4.**	Medieval manuscripts	Ruth Wallach
6, 10.*	Hancock photo archive	Jean Crampon
7.	ITV courseware delivery	Julie Kwan
8, 11.**	Technical reports archives	Charles Phelps
9.**	CNTV moving image archive	Sandra Joy Lee
12.	Student access to digital maps	Julie Kwan
13.*	LA regional history photos	Dace Taube
14.	Korean–American museum	Ken Klein
15.	Urban planning documents	Robert Labaree

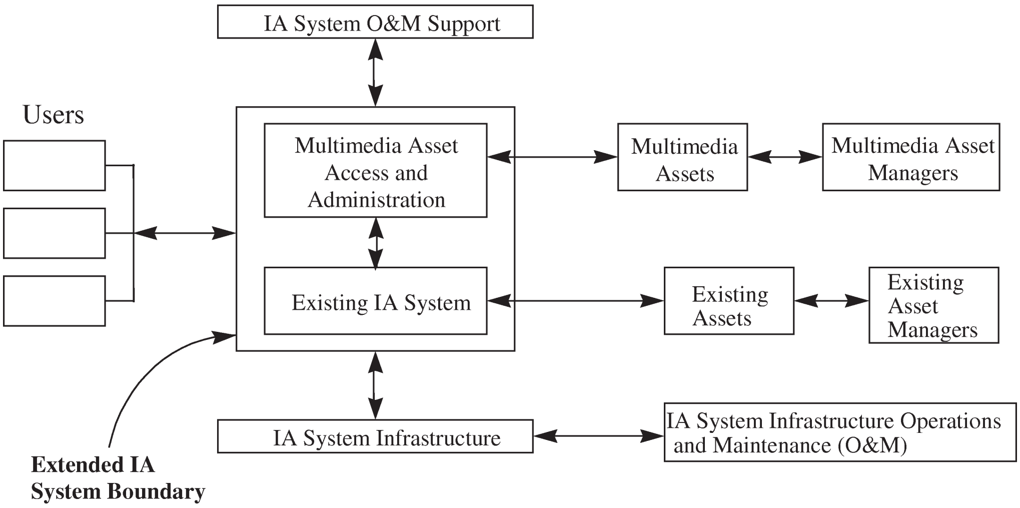
*Combined in Spring 1997.
**Implemented in Spring 1997.

2 teams with four students, one with five, and one with seven, for a total of 15 teams. The course ended up with 12 Library multimedia applications to be architected. Table 5 lists these, and indicates which three applications were done by two teams, and also which were implemented directly ($*$) by five of the six teams in Spring 1997, and which were combined into a single implementation by the sixth team ($**$).

Each project's LCO cycle was focused by the use of the USC-CSE WinWin groupware system for requirements negotiation [Boehm *et al.* 1995; Horowitz *et al.* 1997]. "The WinWin User Negotiations" section of figure 4 summarizes the WinWin artifacts and the stakeholder (developer, customer, and user) roles to be played by the various project team members. To minimize the impact on Library operations, the user artifacts were entered by the student Operational Concept and Requirements team members, rather than the librarians themselves.

Besides support for entering, refining, and negotiating Win Conditions, Issues, Options, and Agreements, WinWin includes a Domain Taxonomy to aid in organization,

1. System Block Diagram:
This diagram shows the usual block diagram for extensions providing access to and administration of multimedia information archive assets from an existing text-based information archive (IA) System:



The system boundary focuses on the automated applications portion of the operation, and excludes such entities as users, operators, maintainers, assets, and infrastructure (campus networks, etc.) as part of the system environment. The diagram abstracts out such capabilities as asset catalogues and direct user access to O&M support and asset managers.

2. Some Stakeholder Roles and Responsibilities

2.1 Asset Managers. Furnish and update asset content and catalogue descriptors. Ensure access to assets. Provide accessibility status information. Ensure asset-base recoverability. Support problem analysis, explanation, training, instrumentation, operations analysis.

2.2 Operators. Maintain high level of system performance and availability. Accommodate asset and services growth and change. Protect stakeholder privacy and intellectual property rights. Support problem analysis, explanation, training, instrumentation, operations analysis.

2.3 Users. Obtain training. Access system. Query and browse assets. Import and operate on assets. Establish, populate, update, and access asset-related user files. Comply with system policies. Provide feedback on usage.

2.4 Application Software Maintainer. Perform corrective, adaptive and perfective (tuning, restructuring) maintenance on software. Analyze and support prioritization of proposed changes. Plan design, develop, and verify selected changes. Support problem analysis, explanation, training, instrumentation, operations analysis.

2.5 Infrastructure Service providers (e.g., network, database, or facilities management services). Similar roles and responsibilities to Asset Managers under 2.1.

Figure 5. Multimedia archive extension domain model.

8/28	W	Course Organization: Library Multimedia Project	Boehm, Lee		CN-1,2
9/4	W	Software Engineering Trends; Software Process			SE-1; CN-3
9/9	M	Library I: SIRSI	Humphrey		CN-4,5
9/11	W	Theory W Software Management			CN-21
9/16	M	Library II: ILS Overview	Taylor	Teams formed	
9/18	W	WinWin Spiral Model Case Study		WinWin Homework (25)	
9/23	M	USC COCOMO and Cost Estimation	Clark		SE-29; CN-6,7
9/25	W	Distributed Multimedia Architecting	Port		CN-22,23
9/30	M	WinWin System I	Lee	COCOMO Homework (25)	CN-8
10/2	W	WinWin System II	Lee		CN-24
10/7	M	Library III: Example Prototype	Scheding		SE-8
10/9	W	Ops. Concept and Requirements Formulation			SE-2,4; CN-9
10/14	M	Architecture, Plans, and Rationale			SE-6,13; CN-10,11,12
10/16	W	Software Architecture Reviews			CN-13
10/21	M	System Engineering & Ops. Concept Definition		WinWin Negotiation Results (100)	
10/23	W	Software Requirements I			
10/28	M	Project Life Cycle Objectives Reviews		Project LCO Reviews	
10/30	W	Project LCO Reviews		Project LCO Reviews	SE-4,5,7
11/4	M	Software Requirements II		LCO Package Due (250)	
11/6	W	Prototyping and User Interface Design	Jacobs		SE-17; CN-33
11/11	M	Software Life Cycle Plans		LCO Packages Graded	SE-3; CN-11
11/13	W	Software Architecture I			SE-12,13
11/18	M	Software Architecture II	Gacek		SE-14,15; CN-24
11/20	W	Software Architecture III	Abd-Allah		CN-25
11/25	M	Project Life Cycle Architecture Reviews		Project LCA Reviews	
11/27	W	Project LCA Reviews		Project LCA Reviews	
12/2	M	Project LCA Reviews			
12/4	W	No Class: Complete LCA Package		LCA Packages Due (450)	
12/6	F			Individual Critiques Due (150)	

Figure 6. Schedule for CS577a: Software Engineering I, Fall 1996.

navigation, and terminology control of these artifacts. Table 6 shows the domain taxonomy for multimedia archive systems furnished to the teams, along with guidelines for relating the taxonomy elements to the requirements specification elements needed for the LCO package.

Figure 7 shows two examples of Win Condition artifacts from the Moving Image Archive (student films) team. It shows how the artifacts are related to each other (the Referenced By entries) and to the domain taxonomy elements (the Taxonomy Element entries), plus additional information on the artifact's owner, priority, status, etc. It also shows how the Comments field is used by the team members in clarifying concepts, removing inconsistencies, and informally exploring negotiated agreements.

The WinWin negotiation period took longer than expected. Complexities in scaling up the tool to 15 on-campus/off-campus teams caused difficulties, and the teams needed to simultaneously learn enough about WinWin, team operations, and the library multimedia applications domain to succeed. As a result, the deadlines for completing the WinWin package and the LCO package were moved back a week. Fortunately, the LCO packages were good enough that the LCA cycle could be compressed by a week.

All 15 of the project LCO packages were delivered on time with respect to the revised schedule. Their degree of completeness was generally appropriate for an LCO package, but the components often had serious inconsistencies in assumptions, relationships, and terminology. Most teams had planned time for members to review each others' artifacts, but this time was generally spent finishing up one's own artifacts. Some concepts caused problems for many teams: the nature of the system boundary; organizational relationships; and the primary focus of the life-cycle plan (development of the Initial Operational Capability). These were then discussed further in the course lectures.

3.2. Cycle 2: Individual application life cycle architectures

All 15 of the project LCA packages were delivered on time, including the prototypes, which were demonstrated to the instructors and librarian clients in two special half-day sessions. The documentation packages had effectively fixed the problems surfaced in the LCO packages but had additional challenges in accommodating the new user insights stimulated by the prototypes.

Although the librarians created the problem statement and participated in the requirements negotiation with the student teams and with various stages of the prototype, the final prototype presentations yielded insightful surprises. Caroline Sisneros, the librarian who proposed the Edgar corporate data problem was "blown away" with the resultant product which built upon the seemingly simple text formatting problem and delivered a one-stop Java site which synthesized several kinds of business information. She commented in her evaluation memo "[The team] obviously looked beyond the parameters of the problem and researched the type of information need the set of data meets. My interactions with the team were minimal, not because of any difficulty, but because as a group they had a synergy and grasped the concepts presented to them.

Table 6
Multimedia archive domain taxonomy.

1. Operational modes
   1.1 Classes of service (research, education, general public)
   1.2 Training
   1.3 Graceful degradation and recovery
2. Capabilities
   2.1 Media handled
      2.1.1 Static (text, images, graphics, etc.)
      2.1.2 Dynamic (audio, video, animation, etc.)
   2.2 Media operations
      2.2.1 Query, browse
      2.2.2 Access
      2.2.3 Text operations (find, reformat, etc.)
      2.2.4 Image operations (zoom in/out, translate/rotate, etc.)
      2.2.5 Audio operations (volume, balance, forward/reverse, etc.)
      2.2.6 Video/animation operations (speedup/slowdown, forward/reverse, etc.)
      2.2.7 Adaptation (cut, copy, paste, superimpose, etc.)
      2.2.8 File Operations (save, recall, print, record, etc.)
      2.2.9 User controls
   2.3 Help
   2.4 Administration
      2.4.1 User account management
      2.4.2 Usage monitoring and analysis
3. Interfaces
   3.1 Infrastructure (SIRSI, UCS, etc.)
   3.2 Media providers
   3.3 Operators
4. Quality attributes
   4.1 Assurance
      4.1.1 Reliability/availability
      4.1.2 Privacy/access control
   4.2 Interoperability
   4.3 Usability
   4.4 Performance
   4.5 Evolvability/portability
   4.6 Cost/schedule
   4.7 Reusability

Table 6
(Continued.)

5. Environment and data 5.1 Workload characterization 6. Evolution 6.1 Capability evolution 6.2 Interface and technology evolution 6.3 Environment and workload evolution	The taxonomy serves as a requirements checklist and navigation aid: • The taxonomy elements map onto the Requirements Description table of contents in the Course Notes. • Every WinWin stakeholder artifact should point to at least one taxonomy element (modify elements if appropriate). • Every taxonomy element should be considered as a source of potential stakeholder win conditions and agreements.

The solution the team came up with was innovative, with the potential to be applied to other, similar problems."

The library clients were generally very satisfied with the value added relative to their time invested. Sandra Joy Lee, the proposer for the Digital Moving Image Archive, commented "They were very instrumental in the discovery of solutions that did not demand too much staff time from my office. In short order, they solved all the problems with creativity and technical sophistication."

The projects also surmounted a number of challenges characteristic of real-world projects. The Library Information System test server continued to be needed for the LIS cutover, and was therefore unavailable to the project prototypes. There were delays in arranging for a suitable alternative Web server for developing prototypes. At times librarians were unavailable to provide inputs on critical decisions, leading to extra rework. Inevitable personnel conflicts arose among the 15 teams. However, the WinWin Spiral Process provided an appropriate mix of flexibility and discipline to enable the projects to adapt to these challenges while staying on schedule. In particular, the use of risk management and a continuously-evolving Top 10 Risk Item list for prioritizing team effort [Boehm 1991] helped the teams focus their effort on the most critical success factors for their projects.

With respect to the LCO-LCA process, the student critiques provided a number of areas for future improvement. The WinWin groupware tool helped with team building and feature prioritization, but people needed more preliminary training and experience in its use. It was also cumbersome to modify groups of WinWin artifacts. Several items, particularly the prototyping capabilities, should have been provided and employed earlier. The prototypes helped a great deal in clarifying and stabilizing the librarians' requirements; they could have helped even more if available during the initial WinWin requirements negotiation process.

Although it was strongly emphasized during the initial lectures, students felt that an even stronger emphasis was needed on the risks of forming teams with personality conflicts and critical-skill shortfalls. The strong focus on the six specific team member

- ID: arucker-WINC-6
- Owner: arucker
- Role: user
- Creation_Date: 10/15/96 12:25
- Revision_Date: 10/15/96 12:25
- Name: View holdings
- Body: The system should be capable of showing the different types of media holdings (production notebook, vhs, 16mm film, etc.) that are available for a particular movie.
- Priority: High
- Status: Active
- State: Covered
- Taxonomy Elements: 3.2.1 Query
- Taxonomy Elements: 3.2.2 Browse
- ReferencedBy: arucker-AGRE-2, LinkFromAgre,Passed
- Comments:

  firouzta 10/16/96 07:52

I am not clear on this win condition. Does this mean that for the material that is not digitized, the system should only present information on the type of the media on which the material is stored? Or, is it that all material, digitized or not, has information on other types of media that the material is stored on, and the system will provide the user with this information?

arucker 10/16/96 12:51

It means that for each movie, the system will provide information about the various types of media that the movie is stored on.

- ID: arucker-WINC-7
- Owner: arucker
- Role: user
- Creation_Date: 10/16/96 13:00
- Revision_Date: 10/17/96 13:13
- Name: Online Request
- Body: The system should allow online requests of movies from the Moving Image Archive.
- Priority: Medium
- Status: Active
- State: Covered
- Taxonomy Elements: 3.2.1 Query
- Taxonomy Elements: 3.2.2 Browse
- ReferencedBy: arucker-AGRE-1, LinkFromAgre,Passed
- Comments:
  arucker 10/16/96 16:30
  I'm not sure which item of the taxonomy this should refer to.
  firouzta 10/16/96 21:05
  2.2.1 and 2.2.2

Figure 7. Example WinWin artifacts.

roles was good in ensuring that each product component was successfully generated, but it caused difficulties in keeping all the team members apprised of issues and developments with the other components. Consistency management of partially redundant components (operational concept, requirements, architecture) became particularly difficult, especially in adapting to change. There was strong consensus that smaller teams and fewer, better-integrated components would have been more effective.

Another difficulty involved consistency maintenance among the multiple views. The various product views required were synthesized from multiple sources: the [Sommerville 1996] course textbook, evolving commercial standards [IEEE-EIA 1995], and object-oriented methods, particularly [Booch 1994; Rumbaugh *et al.* 1991]. The views included system block diagrams, requirements templates, usage scenarios, physical architecture diagrams, class hierarchies, object interaction diagrams, data flow diagrams, state transition diagrams, data descriptions, and requirements traceability relations. Each had its value, but the overall set was both an overkill and was weakly supported by integrated tools. As will be discussed in section 5, we went to a more concise and integrated set of views in the next year, based on the Unified Modeling Language and Rational Rose toolset [Booch *et al.* 1997].

3.3. Cycle 3: Development of initial operational capabilities

The transition from an LCO/LCA phase with 86 students, 15 teams, and 12 applications to an IOC phase with 28 students, 6 teams, and 8 applications caused a number of challenges. Only one team retained the majority of their LCO/LCA participants for their IOC phase. The other teams had to work with a mix of participants with varying project backgrounds.

Even more challenging was the integrating of teams who had produced different LCA artifacts for the same application: the two EDGAR Corporate Data teams and the two Technical Reports teams. In two cases, the instructors had to persuade students to join different teams rather than continuing to fight about whose architecture was best. Other conflicts developed within teams where some team members had extensive LCA experience on the application and others had none (in one case, the experienced members exploited the less experienced members; in another case, vice versa).

Other challenges included a change of instructor (Boehm to Madachy), a change of process model (spiral to risk-driven waterfall), and documentation approach (laissez-faire to everything-on-the-Web). Also, there were infrastructure surprises: the SIRSI server and the SIRSI-related search engine were expected to be available for Cycle 3, but were not.

Nonetheless, each of the projects successfully delivered their IOC packages of code, life cycle documentation, and demonstrations on time. A major reason was the strong emphasis on risk management, which enabled teams to depart from a pure waterfall approach to resolve whatever critical risk items surfaced. An example of one of the teams' initial Top-N risk item lists is shown as table 7. Risks were prioritized by assessments of their risk exposure (probability-of-loss times magnitude-of-loss), and

Table 7
Example top-N risk item list.

	Risk	Risk aversion options	Risk monitoring
1.	Changes of requirements from previous semester	Option 1: Propose a solution for the system (describing the requirements in details) to the users and having them commit to the requirements.	Option 1: Once committed, the requirements must be closely monitored. Changes to requirements must be thoroughly assessed and if excessive, they should be defer till later.
		Option 2: Adopt an incremental approach to the development by building a prototype first.	Option 2: This has an impact on the schedule and hence close monitoring on progress and effort are required.
2.	Tight schedule	Study the requirements carefully so as not to overcommit. Descope good-to-have features if possible. Concentrate on core capabilities.	Close monitoring of all activities is necessary to ensure that schedule are met.
3.	Size of project	If requirements are too excessive, descope good-to-have features and capabilities out of the project. Identify the core capabilities to be built.	
4.	Finding a search engine	Conduct a software evaluation of search engine. Have team members actively source for free search engines and evaluate them. Determine the best for the project.	Have team members submit evaluation report and conduct demos so that an informed decision can be made.
5.	Required technical expertise lacking	Identify the critical and most difficult technical areas of the project and have team members look into them as soon as possible.	Monitor the progress of these critical problems closely. If need be, seek external help.

reassessed weekly with respect to changes in criticality and progress in risk resolution. As indicated in table 7, a key strategy was design-to-schedule: identifying a feasible core capability and optional features to be implemented as schedule permitted.

4. Course evaluation and improvement

4.1. Cycles 1 and 2: LCO and LCA

The basic evaluation results were positive:

- It was feasible for the teams to concurrently develop the 6 components of both the LCO and LCA packages in 11 weeks.

Table 8
Student critiques summary.

Positive comments about WinWin	Count	Negative comments about WinWin	Count
Promoted more cooperativeness and mutual understanding	9	Need more pre-WinWin homework	14
Should continue use of WinWin	9	Too much overhead in WinWinmechanics, bugs decreased negotiability	10
Focused team on key issues	8	Prototype concurrently with WinWin conflict identification	10
Objective artifacts reduced frictions, equalized loud and quiet participants	6	Should have direct Librarian involvement	6
Helped in distributed collaboration	5	Complement WinWin with e-mail, whiteboards, video conferencing, etc.	5
Helped create better requirements	4		

- The Library clients were highly satisfied and wished to complete more projects than we had teams available.
- The WinWin approach built trust that the various stakeholders were looking out for each others' interests. As a result, the teams were able to adapt successfully to a number of complex unforeseen circumstances.

However, there were a number of suggestions for improvement. For example, table 8 summarizes the results of the student critiques of using the WinWin system. The trust and effectiveness results are noted, but also a number of suggested improvements we implemented in the Fall 1997 course:

- Provide more WinWin training and relevant homework prior to project use.
- Fix a number of scalability and multiple-team use problems with the WinWin tool.
- Perform prototyping concurrently with the WinWin negotiation. A number of Library client win conditions changed after they saw their prototypes.

Some additional improvements identified from the 1996 experiences and implemented in 1997 were:

- Reducing team size from 6 to 5 students.
- Using the Web for all course communications and documentation (plus e-mail for time-critical communications).
- Removing redundancy from and tightening linkages between the LCO and LCA package elements. For example, this reduced the average size of an LCO package from 160 to 103 pages.
- Conducting Architecture Review Board meetings for both LCO and LCA packages for all teams.
- Integrating an object-oriented language and toolset (UML and Rational Rose) and methodology (Integrated System Development Methodology [Port 1998]).

In addition, we identified and removed a number of model clashes among the projects' success models, product models, process models, and property models, in concert with our development of a more general Model Based Architecting and Software Engineering (MBASE) approach [Boehm and Port 1998]. For example, the client interview-based WinWin success model conflicts with the client prototype-based IKIWISI (I'll Know It When I See It) success model. Performing client prototyping and WinWin negotiation concurrently removed the model conflict.

4.2. Cycle 3: IOC

From the client standpoint, all of the librarian participants had been very pleased with the Cycle 2 prototype demonstration and LCA packages, and were fully supportive of continuing work with their student teams during Cycle 3 in the second semester. However, the second semester had a smaller enrollment since it was not a required course as during the first semester. Consequently, only six projects were continued during the IOC phase due to the reduction in the number of teams. The LCA projects performed by the continuing students then directed the choice of continuing projects rather than any priority views of the librarians.

With one exception, the librarians were delighted with the final IOC presentations in Cycle 3. Kwan noted in her evaluation memo "The interaction between the student teams and the librarians produced obvious differences in products designed for different users. For example, the technical reports interface mirrored the technical nature of the type of material included and expected future users of the system while the moving image archive interface reflected the needs and interests of a very different clientele."

The one exception project was the attempt to integrate the three photographic-image application (stereoscopic slides, Hancock photo archive, LA regional history photos) into a single application. The short schedule required the team to patch together pieces of the three architectures and user interfaces. Some features of the result were good (e.g., a colored-glasses stereo capability with good resolution), but none of the clients were enthusiastic about implementing the results.

The librarians expressed in their evaluations that working with Theory W and WinWin philosophy made it easy for them to "think big" about their projects. The negotiation process, however, made it possible for the teams and librarians to agree mutually on a feasible set of deliverables for the final IOC products during the academic session. And, although the time commitment was not great, participation in this project allowed the librarians to focus a part of their time and thinking on multimedia applications and software engineering. One of the greatest advantages for the librarians involved was to become more familiar with digital library issues and the software engineering techniques which are involved in their implementation.

In the student critiques for Cycle 3, the most common suggestion for *course improvement* was to provide a solid DBMS and search engine (13 of 28 critiques). The next highest was again to reduce the quantity and redundancy of the documentation (9 of 28 critiques). Project timesheets indicated that total documentation-related effort (requirements, plans, design, product documentation) during Cycle 3 was 47% of the total, with two projects as high as 54% and 60%.

Other common suggestions (appearing in 6 to 8 critiques) were for better documentation guidelines, better match of course notes and lectures to project activities, more timely feedback on intermediate products, more disk space, better tools (scanning, HTML conversion, CM) and more training on key Web skills. The most common suggestions for *project improvement* were improved intra-team communication (8 critiques), early error elimination (7), improved client communication (5), and improved on/off-campus team coordination (5). We are using these insights to improve the organization of both semesters of the 1997–1998 projects.

The major change in Cycle 3 for 1997–1998 has been a much stronger emphasis on transition of the applications to Library use. In 1996–1997, although the Library clients were happy with the Cycle 3 IOC results, we subsequently found that they were not fully empowered to allocate the resources necessary to sustain the applications.

In 1997–1998, continuing Cycle 3 projects were selected by Library personnel on the basis of the Library's ability to sustain the applications. This led to the identification of additional operational stakeholders, and additional win conditions in such areas as training, installation, and beta-testing prior to course completion. These have been accommodated by extensions to the 1997–1998 Cycle 3 Software Development Plan, (SDP) guidelines, and closer links between the SDP and the transition considerations in the LCA Operational Concept Description.

5. Conclusions: evaluation versus course objectives

Table 9 summarizes our evaluation of the course's current status with respect to the objectives determined from the stakeholder win conditions in table 1.

1. With respect to integrating software engineering principles and practice, things appear to be coming together fairly well. The practice enables us to validate that the newer principles, such as the WinWin Spiral Model and anchor points, are basically workable across a number of modern applications. The principles are evolving based on the project experiences. There are still some gaps we are working on, such as the development of a COCOMO II cost estimation model tailored to this class of project.

2. The range of life cycle skills covered is good for the very early stages through acceptance test. As discussed in section 5.2, we are fully addressing transition this year. As the Library applications begin to build on each other, we will increasingly confront and address maintenance considerations.

3. The range of discipline skills is covered across the teams, but we find that each student learns more about some disciplines than others. This is a shortfall versus our objectives which we try to address via individual homework assignments. The course grading criteria (product conceptual integrity, informed critiques) also encourage cross-learning. However, specialization does reflect reality with respect to project practice, and provides team experience in dealing with skills coverage and balance. The biggest challenge is at the beginning of the first semester, when the students need to learn about many things right away. Library information systems, software life cycle processes and

Table 9
Software Engineering Core Course evaluation vs. objectives.

Objectives	Evaluation
1. Integrate principles and practice	• Coming together • Still some gaps and sequencing problems
2. Cover full range of life cycle skills: system engineering, requirements, architecture, programming, test, transition, maintenance	• Well covered through test • Transition strengthened in 1998 • Some coverage of maintenance in future
3. Cover full range of discipline skills: computer science, application domain, economics, people management skills	• Some student specification inevitable • Emphasis on conceptual integrity, course critiques helps cross-fertilization • Some just-in-time sequencing problems • Developing course-tailored COCOMO II cost model
4. Involve real, non-computer science clients	• Basically covered • Transition strengthened in 1998 • Broader, deeper client participation in future
5. Add value to client community (USC Libraries)	• Absorbing transition lessons learned • Librarian evaluations strongly positive • Library committed to continue
6. Experiment with emerging application technology (Web, Java, COTS, multimedia)	• Digital library applications a good match • Mixed student skill level challenges • Hard to stabilize products' infrastructure
7. Experiment with emerging SW Engineering tools & techniques (WinWin Spiral Model, OO, Architecture Review Boards, distributed collaboration)	• Digital library applications a good match • Mixed student skill-level challenges • Some tools & techniques kinks, instabilities • Tool/model integration still evolving and converging
8. Develop experience-based improvements to general SW Engineering tools & techniques	• Sustainable MBASE-based experience factory • Accelerating adoption of WinWin Spiral Model, WinWin tools, anchor points, Integrated System Development Methodology
9. Scale up to large classes, distance learning	• Successful 80-student classes • Some difficulties with distant projects (client interaction, tool licenses/access)
10. Provide rewarding intellectual, social experience	• Successful team and client experiences (some exceptions) • Difficulties with high student/instructor ratio

products, stakeholder win–win negotiation, requirements, plans, cost estimation, risk management, project staffing and organizing.

4,5. The Librarians' involvement and the value added to the Library has worked out quite well, as indicated by their project evaluations and commitment to continue. Again, we have found that successful transition often requires participation by additional operations and maintenance stakeholders.

6,7. The digital library applications have been a good match both to emerging applications technology (multimedia, Web technology, Java, COTS-based systems) and to emerging software engineering technology (WinWin, collaboration technology, UML). Here also, there are both opportunities and challenges involved with the different skill sets the students bring to the course. There are also significant challenges in keeping the course material up with technological change. An example of kinks and instabilities is the challenge of reconciling aspects of OO/UML technology with evolving Web application paradigms.

8. The ability to instrument, analyze, and iterate the tools and techniques supporting a sizeable number of annual applications has accelerated our ability to verify that the techniques and/or their improvements work on real projects. The project experience patterns have given us insights on how to better integrate project's success models, process models, product models, and property models, as elaborated in the MBASE approach [Boehm and Port 1998].

9. The multiple-project approach scales up reasonably well to large classes of the projects are relatively similar. However, there are serious peak-load problems: a set of 15–16 LCO or LCA reviews requires a virtually dedicated week of activity. It is difficult to find TA's with enough experience to serve as sole reviewers or graders for the LCO and LCA packages. Project courses are difficult with remote students, especially of they are in different locations. If they are at the same remote location, it is better for them to find a comparable client project at their location, unless they are willing and able to make regular visits to on-campus clients. Remote sites with firewalls prohibiting student access to on-campus computer and tool resources are another source of difficulties.

10. Overall, students have considered the team projects to be rewarding intellectual and social experiences, as have the instructors. However, large classes make it very difficult to know all the students well, and to provide extensive monitoring. The smaller Spring class can be done with an instructor and TA for up to 30 students. The large Fall class requires an additional part-time associate instructor and another part-time TA. Overall, however, the projects have been exciting, highly satisfying in providing value for real clients, and fascinating as opportunities to better understand and improve software project practice.

Other Universities are beginning to adopt the approach and course materials, particularly at George Mason University and the Johannes Kepler University in Linz, Austria. Most of the course materials are available via the USC-CSE Web site, at `http://sunset.usc.edu/classes`.

Acknowledgements

This research is sponsored by DARPA through Rome Laboratory under contract F30602-1994-C-0195 and by the Affiliates of the USC Center for Software Engineering: Allied Signal, Bellcore, Boeing, Electronic Data Systems, Federal Aviation Administration, GDE Systems, Hughes Electronics, Institute for Defense Analysis, Litton Data Systems, Lockheed Martin, MCC, Motorola, Network Programs, Northrop Grumman, Rational Software, Raytheon, Science Applications International, Software Engineering Institute, Software Productivity Consortium, Sun Microsystems, TI, TRW, USAF Rome Laboratory, US Army Research Laboratory, and Xerox. We also thank Denise Bedford, Anne Curran, Simei Du, Ellis Horowitz, Ming June Lee, Phil Reese, Bill Scheding, and Nirat Shah for support in key areas.

References

AT&T (1993), *Best Current Practices: Software Architecture Validation*, AT&T, Murray Hill, NJ.

Basili, V.R., R.W. Selby and D.H. Hutchens (1986), "Experimentation in Software Engineering," *IEEE Transactions on Software Engineering*, July, 733–743.

Boehm, B.W. (1988), "A Spiral Model of Software Development and Enhancement," *Computer 21*, 5, 61–72.

Boehm, B.W. (1991), "Software Risk Management: Principles and Practices," *IEEE Software*, 32–41.

Boehm, B.W. (1996), "Anchoring the Software Process," *IEEE Software 13*, 4, 73–82.

Boehm, B.W. and P. Bose (1994), "A Collaborative Spiral Software Process Model Based on Theory W," In *Proceedings of 3rd International Conference on the Software Process, Applying the Software Process*, IEEE, Reston, VA.

Boehm, B.W., P. Bose, E. Horowitz and M.J. Lee (1994), "Software Requirements As Negotiated Win Conditions," In *Proceedings of ICRE*, pp. 74–83.

Boehm, B.W., B.K. Clark, E. Horowitz, R. Madachy, R.W. Selby and C. Westland (1995), "Cost Models for Future Software Processes: COCOMO 2.0," *Annals of Software Engineering 1*, 57–94.

Boehm, B.W., A.F. Egyed, J. Kwan and R. Madachy (1997), "Developing Multimedia Applications with the WinWin Spiral Model," In *Proceedings of ESEC/FSE 97*, Springer, pp. 20–39.

Boehm, B.W., J.F. Elwell, A.B. Pyster, E.D. Stuckle and R.D. Williams (1982), "The TRW Software Productivity System," In *Proceedings of 6th International Conference on Software Engineering*, ACM/IEEE, pp. 148–156.

Boehm, B.W. and D. Port (1998), "Conceptual Modeling Challenges for Model Based Architecting and Software Engineering (MBASE)," In *Proceedings of Symposium on Conceptual Modeling*, Springer, to appear.

Boehm, B.W. and R. Ross (1989), "Theory W Software Project Management: Principles and Examples," *IEEE Transactions on Software Engineering*, 902–916.

Booch, G. (1994), *Object-Oriented Analysis and Design*, 2nd ed., Benjamin/Cummings.

Booch, G., I. Jacobson and J. Rumbaugh (1997), "The Unified Modeling Language for Object-Oriented Development," Documentation set, version 1.0, Rational Software Corporation.

Carmel, E., R. Whitaker and J. George (1993), "PD and Joint Application Design: A Transatlantic Comparison," *Communications of the ACM*, June, 40–48.

Conklin, J. and M. Begeman (1988), "gIBIS: A Hypertext Tool for Exploratory Policy Discussion," *ACM Transactions OIS*, October, 303–331.

Dardenne, A., S. Fickas and A. Lamsweerde (1993), "Goal-Directed Concept Acquisition in Requirement Elicitation," In *Proceedings of 6th IWSSD*, IEEE, pp. 14–21.

Finkelstein, A., J. Kramer, B. Nusibeh, L. Finkelstein and M. Goedicke (1992), "Viewpoints: A Framework for Integrating Multiple Perspectives in System Development," *International Journal of Software Engineering and Knowledge Engineering*, 31–58.

Fisher, R. and W. Ury (1981), *Getting to Yes*, Penguin Books.

Frazier, T.P. and J.W. Bailey (1996), "The Costs and Benefits of Domain-oriented Software Reuse: Evidence from the STARS Demonstration Projects," IDA paper P-3191, Institute for Defense Analysis.

Horowitz, E. *et al.* (1997), "WinWin Reference Manual: A System for Collaboration and Negotiation of Requirements," Technical Report, Center for Software Engineering, University of Southern California, Los Angeles, CA.

IEEE-EIA (1995), "Software Development," Trial Use Standard J-STD-016-1995, formerly known as IEEE 1498/EIA 640.

Mullery, G. (1979), "CORE: A Method for Controlled Requirements Specification," In *Proceedings of 4th ICSE*, IEEE, pp. 126–135.

Port, D. (1998), *Integrated Systems Development Methodology*, Telos Press, to appear.

Potts, C. and K. Takahashi (1993), "An Active Hypertext for System Requirements," In *Proceedings of 7th IWSSD*, IEEE, pp. 62–68.

Royce, W.E. (1990), "TRW's Ada Process Model for Incremental Development of Large Software Systems," In *Proceedings of 12th ICSE*, IEEE/ACM, pp. 2–11.

Rumbaugh, J., M. Blaha, W. Premerlani, F. Eddy and W. Lorensen (1991), *Object-Oriented Modeling and Design*, Prentice-Hall, Englewood Cliffs, NJ.

Sommerville, I. (1996), *Software Engineering*, 5th ed. Addison-Wesley, Reading, MA.

SPC (1994), "Process Engineering with the Evolutionary Spiral Process Model," SPC-93098-CMC, version 01.00.06, Software Productivity Consortium, Herndon, VA.