Published in Proceedings of the 5th European Software Engineering Conference held jointly with the 5th ACM SIGSOFT Symposium on the Foundations of Software Engineering (ESEC/FSE), Zurich, Switzerland, September 1997, pp. 20-39.

Developing Multimedia Applications with the WinWin Spiral Model

Barry Boehm, Alex Egyed, USC-Center for Software Engineering
Julie Kwan, USC University Libraries
Ray Madachy, USC-CSE and Litton Data Systems

Abstract

Fifteen teams recently used the WinWin Spiral Model to perform the system engineering and architecting of a set of multimedia applications for the USC Library Information Systems. Six of the applications were then developed into an Initial Operational Capability. The teams consisted of USC graduate students in computer science. The applications involved extensions of USC's UNIX-based, text-oriented, client-server Library Information System to provide access to various multimedia archives (films, videos, photos, maps, manuscripts, etc.).

Each of the teams produced results which were on schedule and (with one exception) satisfactory to their various Library clients. This paper summarizes the WinWin Spiral Model approach taken by the teams, the experiences of the teams in dealing with project challenges, and the major lessons learned in applying the Model. Overall, the WinWin Spiral Model provided sufficient flexibility and discipline to produce successful results, but several improvements were identified to increase its cost-effectiveness and range of applicability.

1. Introduction

At the last two International Conferences on Software Engineering, three of the six keynote addresses identified negotiation techniques as the most critical success factor in improving the outcome of software projects. Tom DeMarco stated that "how the requirements were negotiated is far more important than how the requirements were specified" [DeMarco, 1996]. In discussing "Death March" projects, Ed Yourdon stated that "Negotiation is the best way to avoid Death March projects," [Yourdon, 1997]. Mark Weiser concluded that "Problems with reaching agreement were more critical to his projects' success than such factors as tools, process maturity, and design methods" [Weiser, 1997].

At the USC Center for Software Engineering, we have been developing a negotiation-based approach to software system requirements engineering, architecting, development, and management. It is based on three primary foundations:

- Theory W, a management theory and approach. It is based on making winners of all of the system's key stakeholders as a necessary and sufficient condition for project success [Boehm-Ross, 1989].
- The WinWin Spiral Model, an extension to the Spiral Model of the software process. It is described further below.
- The WinWin groupware tool for facilitating distributed stakeholders' negotiation of mutually satisfactory (WinWin) system specifications [Boehm et al., 1995; Horowitz et al., 1997].

2. 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 incremental 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.

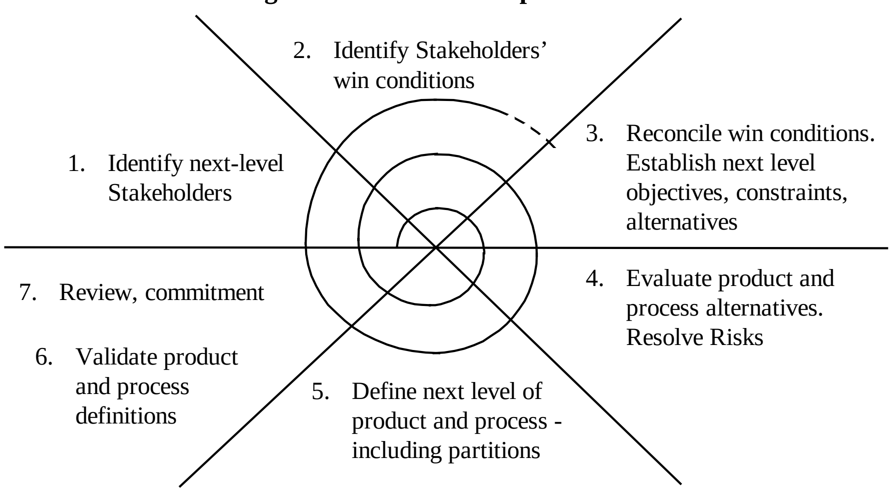
The Spiral Model has been extensively elaborated (e.g., SPC, 1994]), and successfully applied in numerous projects (e.g., [Royce, 1990], [Frazier-Bailey, 1996]). However, some common difficulties have led to some further extensions to the model.

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 1 [Boehm-Bose, 1994].

- Identify the system or subsystem's key stakeholders.
- Identify the stakeholders' win conditions for the system or subsystem.
- 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-Begeman, 1988], Viewpoints [Finkelstein, 1991], GRAIL [Dardenne et al., 1993], Tuiqiao [Potts-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-Ury, 1981].

Figure 1. The WinWin Spiral Model



2.1. 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 1. The third, the Initial Operational Capability (IOC), is summarized in Table 2. 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 the use of the successful Architecture Review Board practice pioneered by AT&T and Lucent Technologies [AT&T, 1993].

Table 1. Contents of LCO and LCA Milestones

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 2. 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.

3. Applying the WinWin Spiral Model

New software process models generally take years to validate. The Spiral Model was originated in 1978, first tried on a 15-person internal TRW project in 1980-82 [Boehm et al, 1982], and only in 1988-92 scaled up to a 100-person contract project [Royce, 1990] and fully-documented method [SPC, 1994]. For the WinWin Spiral Model, we were fortunate to find a family of multimedia applications upon which to test the model: a set of graduate student projects to develop candidate multimedia extensions for the USC Integrated Library System (ILS).

The ILS is a UNIX-based, client-server system based on the SIRSI commercial library information system package and the USC campus computing network. The ILS is primarily text-based, but the Library's management has been quite interested in providing multimedia services to the USC community. Exploratory discussions identified a number of USC multimedia archives--student films, photo and stereopticon archives, technical reports, medieval manuscripts, urban plans, etc.—which appeared to be attractive candidates for transformation into digitized, user-interactive archive management services.

The application of the WinWin Spiral Model to this potential family of multimedia applications involved four major spiral cycles:

- 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.

3.1. Cycle 0: Project Family Life Cycle Objectives

During 1993-96, 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 1995-96, 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.

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 1996-97 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 3 summarizes the win conditions for the three primary stakeholders: the Library information technology community,

Table 3. 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

including its users; the Library operational community, including its users; and the Center for Software Engineering.

As indicated in Table 3, 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 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

Figure 2. Example Library Multimedia Problem Statements

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 the1930s 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.

clients, and provided brief summaries of each. Examples are shown in Figure 2. 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).

3.2. Cycle 1: Individual Application Life Cycle Objectives

Figure 3 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 2 ½ 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 3, 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 4). 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.

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 offering 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.

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 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 4 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 (**).

Table 4. 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

Figure 3. Multimedia Archive Project Guidelines

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 1.

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.

Major Milestones

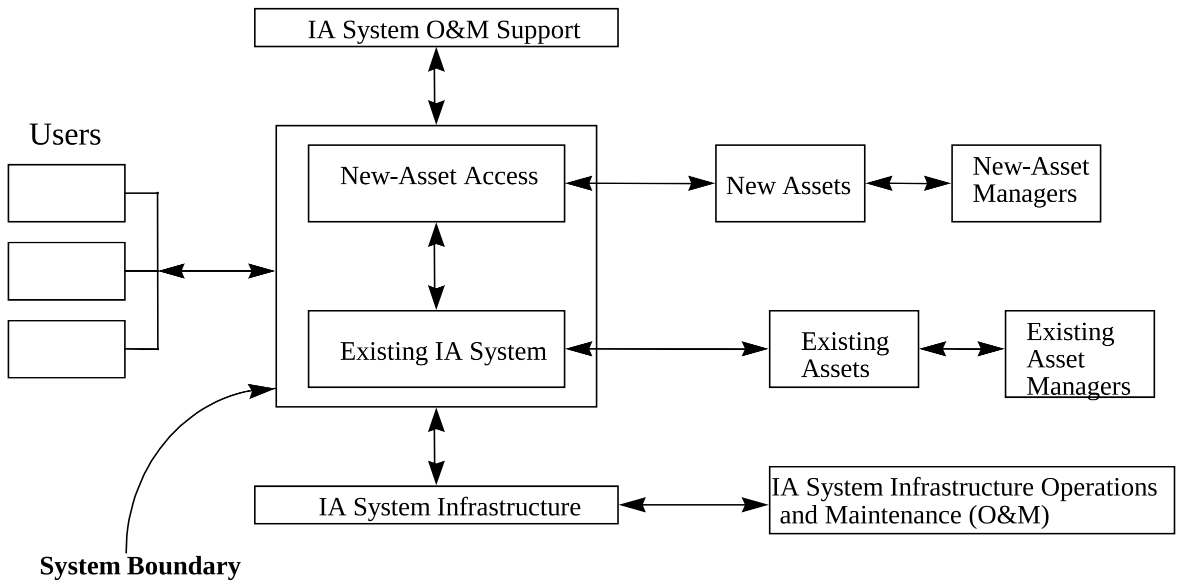
September 16	---	All teams formed
October 14	---	WinWin Negotiation Results
October 21,23	---	LCO Reviews
October 28	---	LCO Package Due
November 4	---	Feedback on LCO Package
December 6	---	LCA Package Due, Individual Critique Due

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. Information Archive Extension Domain Model

1. **System Block Diagram:**
   This diagram shows the usual block diagram for extensions providing access to new information archive assets from an existing information archive (IA) System:



The system boundary focuses on the automated applications portion of the operation, and includes 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 mangers.

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

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 3 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, navigation, and terminology control of these artifacts. Table 5 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.

Table 5. 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
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.

Figure 5 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

Figure 5. Example WinWin Artifacts

- 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

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.

3.3. 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 crated 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 way" 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. 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 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] and [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. We plan on using a more concise and integrated set of views next year, based on the Rational Unified Modeling Language and toolset [Booch-Jacobson-Rumbaugh, 1997].

3.4. 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 6. Risks were prioritized by assessments of their risk exposure (probability-of-loss times magnitude-of-loss), and reassessed weekly with respect to changes in criticality and progress in risk resolution. As indicated in Table 6, a key strategy was design-to-schedule: identifying a feasible core capability and optional features to be implemented as schedule permitted.

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 next year's projects.

From the client standpoint, all of the librarian participants had been very pleased with the prototype demonstration and LCA packages, and were fully supportive of continuing work with their student teams during 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.

The librarians' involvement with the student teams during the second semester was, for the most part, qualitatively and quantitatively different than during the preceding semester. Major system requirements had already been negotiated, but there were typically a few new requirements when the project was taken on by newly reconstituted project teams whose views added subtle differences to the original concepts. Nonetheless, the time required for the librarians' participation was not as extensive as during the preceding semester with the exception of one team. The LAPIS project faced another challenge, having technical problems with scanning and OCR of sample documents until just before the final IOC demonstration. Consultation with a faculty member who uses these technologies for his research in machine translation indicated that the types of documents used, given their historic type fonts, represented

Table 6. 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 2: Adopt an incremental approach to the development by building a prototype first.	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: 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.

a separate OCR research problem in itself; the faculty member was then able to help the project implement a fallback solution.

With one exception, the librarians were delighted with the final IOC presentations. The skillful integration of the requirements and functionality of finished products was evident to all. 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." Barbara Robinson, who proposed LAPIS (Latin American Pamphlet Information System), saw the project as a means for the international community of Latin Americanists to preserve fragile material, a difficult conservation issue for the community; after the IOC delivery, she prepared a proposal to expand the project for full-scale implementation.

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 other five application are either being adopted or extended for possible adoption by the Library elements.

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.

Further details on the project processes and artifacts can be found in their USC-CSE Web pages: 'http://sunset.usc.edu/classes/cs577a' and 'http://sunset.usc.edu/classes/cs577b.'

4. Conclusions

We had a number of hypotheses we wished to test with respect to the use of the WinWin Spiral Model for multimedia applications or other similar applications. Unfortunately, considerations of stakeholder satisfaction (successful applications for the library clients; fairness of grading for the students) conflict with the most rigorous forms of experimental design. For example, having some teams operate in a contract-oriented, adversarial, waterfall-model mode would have been a better test of the relative benefits of using the WinWin Spiral Model. However, given available experience, it did not seem feasible or fair to consign some projects to use such a mode.

Modulo these caveats, here are the main hypotheses we wished to test, and a summary of the best evidence we can provide about them.

Hypothesis 1. Teams can use the WinWin Spiral Model to simultaneously develop the components of a consistent and feasible LCA package for a new multimedia application in 11 weeks. Each of the 15 LCA teams delivered their packages on time, and satisfied an extensive set of grading criteria covering each LCA component the conceptual integrity of the integrated package, and client evaluations of the prototypes.

Hypothesis 2. Using two (LCO and LCA) spiral cycles to develop the LCA package was feasible and value-adding. Feasibility of two cycles is covered under Hypothesis 1. Based on the results of the LCO reviews, using a single spiral cycle would have produced less satisfactory results in about half of the projects. Several projects produced unbalanced detail in either the archiving or the query/browsing part of their LCO packages; the LCA cycle enabled than to balance their architecture packages. On the other hand, using three cycles would have left insufficient time to both produce and coordinate three sets of artifacts.

Hypothesis 3. The Library clients will see enough prospective value in the LCA packages to decide to continue as many as possible into full-scale development. There were more than enough clients for the six project teams available in Spring 1997. Perhaps erroneously, we tried to have one project team address all three image-archive applications. Some additional LCA packages (historical maps, urban plans) had considerable client interest but could not be pursued.

Hypothesis 4. The LCA packages would be adequate to ensure satisfactory IOC development in 11 weeks. Again, all six teams completed full IOC packages on time. The projects having the most difficulties were the ones which started with two LCA packages for the same application (startup difficulties) or with LCA packages for three separate image archive applications (conceptual integrity difficulties).

Hypothesis 5. The WinWin Spiral approach will produce wins for the stakeholders. Five of the six completed projects had highly enthusiastic clients who are continuing with the applications developed. The sixth IOC product's clients did not wish to continue with the product developed, but were receptive to another try. The preponderance of the student critiques indicated that the experience had been valuable and career-enhancing. Even the documentation overkill was considered by some students as good preparation for many industrial projects with similar overkill.

Hypothesis 6. The WinWin Spiral approach will be flexible enough to adapt to real-world conditions. Section 3 summarized many real-world conditions (pleasant and unpleasant surprises with COTS packages; unavailability of expected infrastructure packages and library information system expertise; personnel complications and changes) to which the projects were successfully able to adapt. More formal or contract-oriented approaches would not have been able to accommodate the necessary change processing in the short times available for architecting and development.

Hypothesis 7. The WinWin Spiral approach will efficiently use the developers' time. As indicated under Hypothesis 6, the approach avoided some inefficiencies. However, as implemented, it had some significant inefficiencies in document overkill and multiple-view coordination. Next year's projects will have less redundant and voluminous documentation, an integrated toolset (the Rational ROSE system and its associated packages), and smaller development teams.

Hypothesis 8. The WinWin tool outputs can transition smoothly into requirements specifications. This had been a problem in previous uses of WinWin. Mapping the WinWin domain taxonomy onto the table of contents of the requirements specification, and requiring the use of the domain taxonomy as a checklist for developing WinWin Agreements, effectively focused stakeholder negotiations and facilitated transitioning WinWin Agreements into requirements specifications. The manual transition engendered some inefficiencies; we are exploring automated aids for the transition.

Hypothesis 9. The WinWin approach will improve developer-client relations. In terms of the fear, uncertainty, and doubt often exhibited by clients toward new applications, the Library clients exhibited virtually no fear, considerable uncertainty, and some doubt about going forward with the projects, as indicated by the project conditions stipulated by the Library memo (Section 3.1). By the LCA milestone, as indicated by a meeting between the computer science principals and Dean Campbell and the Library principals, the uncertainty and doubt about working with the student teams had been replaced by enthusiasm and considerable trust (although a good deal of uncertainty remained about the applications' technical parameters). This growth of enthusiasm and trust continued through the development period, and has led to a mutual commitment to pursue further projects in 1997-98. The ability of the WinWin approach to foster trust was consistent with earlier experiences [Boehm-Bose,1994].

Bottom Line:

Overall, the projects' results indicate that the WinWin Spiral Model is a good match for multimedia applications, and likely for other applications with similar characteristics (rapidly moving technology; many candidate approaches; little user or developer experience with similar systems; premium on rapid completion). It provides sufficient flexibility to adapt to the accompanying risks and uncertainties, and the discipline to maintain focus on achieving its anchor-point milestones. Finally, it provides the means for growing trust among stakeholders, enabling them to evolve away from adversarial contract-oriented system development approaches toward mutually supportive and cooperative approaches.

5. Acknowledgments

This research is sponsored by DARPA through Rome Laboratory under contract F30602-94-C-0195 and by the Affiliates of the USC Center for Software Engineering: Aerospace Corp., Air Force Cost Analysis Agency, Allied Signal, Bellcore, Boing, Electronic Data Systems, E-Systems, GDE Systems, Hughes Aircraft, Interactive Development Environments, Institute for Defense Analysis, Jet Propulsion Laboratory, Litton Data Systems, Lockheed Martin, Loral Federal Systems, MCC, Motorola, Network Programs, Northrop Grumman, Rational Software, 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, Ellis Horowitz, Ming June Lee, Bill Scheding, Archita Shah, and Nirat Shah for support in key areas.

6. References

[AT&T, 1993]. "Best Current Practices: Software Architecture Validation," AT&T, Murray Hill, NJ 1993.

[Boehm, 1988]. B. W. Boehm, "A Spiral Model of Software Development and Enhancement," Computer, May 1988, v. 21 no. 5, pp. 61-72.

[Boehm, 1991]. B. W. Boehm, "Software Risk Management: Principles and Practices," IEEE Software, January 1991, pp. 32-41.

[Boehm, 1996] B.W. Boehm, "Anchoring the Software Process," IEEE Software, July 1996, v.13 no.4, pp.73-82.

[Boehm et. al., 1982]. B. W. Boehm, J. F. Elwell, A. B. Pyster, E.D. Stuckle, and R. D. Williams, "The TRW Software Productivity System," Proceedings, 6th International Conference on Software Engineering, ACM/IEEE, September 1982, pp. 148-156.

[Boehm et al., 1994]. B.W. Boehm, P Bose, E. Horowitz,, M.J. Lee, "Software Requirements As Negotiated Win Conditions", Proceedings of ICRE, April 1994, pp.74-83.

[Boehm et al., 1995]. B.W. Boehm, B. K. Clark, E. Horowitz, R. Madachy, R.W. Selby, and C. Westland, "Cost Models for Future Software Processes: COCOMO 2.0," Annals of Software Engineering, 1995, v.1, pp. 57-94.

[Boehm-Bose, 1994]. B.W. Boehm and P. Bose, "A Collaborative Spiral Software Process Model Based on Theory W," Proceedings, 3rd International Conference on the Software Process, Applying the Software Process, IEEE, Reston, Va. October 1994.

[Boehm-Ross, 1989]. B.W. Boehm and R. Ross "Theory W Software Project Management: Principles and Examples," IEEE Transactions on Software Engineering, July 1989, pp.902-916.

[Booch, 1994]. G. Booch , Object-Oriented Analysis and Design, 2nd Edition, Benjamin/Cummings Publishing, 1994.

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

[Carmel et al., 1983]. E. Carmel, R. Whitaker, and J. George, " PD and Joint Application Design: A Transatlantic Comparison," Comm. ACM, June 1993, pp. 40-48.

[Conklin-Begeman, 1988]. J. Conklin and M. Begeman, "gIBIS: A Hypertext Tool for Exploratory Policy Discussion," ACM Trans. OIS, October 1988, pp.303-331.

[Dardenne et al., 1993] A. Dardenne, S. Fickas, and A. van Lamsweerde, "Goal-Directed Concept Acquisition in Requirement Elicitation," Proceedings, IWSSD 6, IEEE, October 1991, pp. 14-21.

[DeMarco, 1996]. T. DeMarco, "The Role of Software Development Methodologies: Past, Current and Future," Keynote Address, ICSE 18, IEEE/ACM, March 1996, pp. 2-4.

[Finkelstein et al., 1991]. A Finkelstein, J. Kramer, B. Nusibeh, L Finkelstein, and M. Goedicke, "Viewpoints: A Framework for Integrating Multiple Perspectives in System Development," International J. Software Engineering and Knowledge Engineering, March 1992, pp. 31-58.

[Fisher-Ury, 1981]. R. Fisher, W. Ury, Getting to Yes, Penguin Books, 1981.

[Frazier-Bailey, 1996]. T. P. Frazier, J.W. Bailey, "The costs and benefits of domain-oriented software reuse: Evidence from the STARS demonstration projects," IDA Paper P-3191, Institute for Defense Analysis, 1996.

[Horowitz et al., 1997]. Horowitz, E. "WinWin Reference Manual: A System for Collaboration and Negotiation of Requirements", Center for Software Engineering, University of Southern California Technical Report, Los Angeles, CA 90089-0781, 1997.

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

[Mullery, 1979]. G. Mullery, "CORE: A Method for Controlled Requirements Specification," Proceedings, ICSE 4, IEEE, Septmember 1979, pp. 126-135.

[Potts-Takahashi, 1993]. C. Potts and K. Takahashi, "An Active hypertext for System Requirements," Proceedings, IWSSD 7, IEEE, December 1993, pp. 62-68.

[Royce, 1990]. W.E. Royce, "TRW's Ada Process Model for Incremental Development of Large Software Systems," Proceedings, ICSE 12, IEEE/ACM, March 1990, pp. 2-11.

[Rumbaugh et. al., 1991]. J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy, and W. Lorensen, Object-Oriented Modeling and Design, Prentice Hall, 1991.

[Sommerville, 1996]. I. Sommerville, Software Engineering, Addison-Wesley, 5th Edition, 1996.

[SPC, 1994]. Software Productivity Consortium, "Process Engineering with the Evolutionary Spiral Process Model," SPC-93098-CMC, version 01.00.06, Herndon, Virginia, 1994.

[Weiser, 1997]. M. Weiser, "Software Engineering that Matters to People," Keynote Address, ICSE 97, IEEE/ACM, May 1997.

[Yourdon, 1997]. E. Yourdon, "'Death March' Projects," Keynote Address, ICSE 97, IEEE/ACM, May 1997.