WinWin Requirements Negotiation Processes: A Multi-Project Analysis

Published in the Proceedings of the 5th International Conference on Software Processes, 1998

Barry Boehm
University of Southern California
Center for Software Engineering
Salvatori Computer Science Building
Los Angeles, CA 90089-0781 USA
+1 213 740 5703
boehm@sunset.usc.edu

Alexander Egyed
University of Southern California
Center for Software Engineering
Salvatori Computer Science Building
Los Angeles, CA 90089-0781 USA
+1 213 740 6504
aegyed@sunset.usc.edu

Abstract

Fifteen 6-member-teams were involved in negotiating requirements for multimedia software systems for the Library of the University of Southern California. The requirements negotiation used the Stakeholder WinWin success model and the USC WinWin negotiation model (Win Condition-Issue-Option-Agreement) and groupware system. The negotiated results were integrated into a Life Cycle Objectives (LCO) package for the project, including descriptions of the system's requirements, operational concept, architecture, life cycle plan, and feasibility rationale. These were subsequently elaborated into a Life Cycle Architecture package including a prototype; six of these were then implemented as products.

A number of hypotheses were formulated, tested, and evolved regarding the WinWin negotiation processes and their effectiveness in supporting the development of effective LCO packages, in satisfying Library clients, and in stimulating cooperation among stakeholders. Other hypotheses involved identification of WinWin improvements, relationships among negotiation strategies on LCO package and project outcomes.

Some of the more illuminating results were:

- *Most of the stakeholder Win Conditions were non-controversial (were not involved in Issues). Also, most Issues were decoupled from other Issues and were easy to resolve. This implies that requirements negotiation support systems should focus at least as much on handling simple relationships well as on handling complex relationships well.*
- *Similar applications projects did not follow similar processes, confirming our previous conclusion [11] that repeatability of software front-end processes is not a realistic goal.*
- *The strongest positive effects of using the WinWin approach were increasing cooperativeness, focusing participants on key issues, reducing friction, and facilitating distributed collaboration.*
- *The major improvements for the WinWin approach (now being implemented) were increasing WinWin training, reducing usage overhead, and concurrent negotiation and prototyping.*

1 Introduction

1.1 Motivation and Background

Negotiating requirements is one of the first steps in any software system life cycle, but its results have probably the most significant impact on the system's value.

Three of the six keynote addresses in the last two ICSE's identified negotiation techniques as the most critical success factor for software projects [10][22][23]. For example, Tom DeMarco [10] said that "how the requirements were negotiated is far more important than how the requirements were specified." However, the processes of requirements negotiation are not well understood.

We have had the opportunity to capture and analyze requirements negotiation behavior for groups of projects developing library multimedia archive systems, using an instrumented version of the USC WinWin groupware system for requirements negotiation.

The overall stakeholder WinWin negotiation approach is similar to other team approaches for software and system definition such as CORE [18], gIBIS [8], Viewpoints [12], REMAP [20], GRAIL [9], Tuiqiao [19], Participatory Design and JAD [7], NATURE [14], Requirements Meta-Model [21], and the Anderson-Rome Labs KBSA-ADM [1]. 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 [13].

An earlier paper [11] reported on a largely bottom-up analysis of data collected on 23 teams negotiating requirements for a single hypothetical library system. Based on this experience, we now report on testing a stronger set of hypotheses with more thoroughly-collected data on a group of 15 WinWin requirements negotiations for a set of actual USC Library multimedia applications [6].

1.2 Overview

Section 2 describes the context of the projects, elaborating on the people, tasks, and negotiation process involved. In Section 3 we describe our major hypotheses, their observational results, and implications of our findings. Section 4 summarizes our major conclusions.

Table 1: Library project topics

Team	Topic	Continued
1	Stereoscopic Slides	Yes *
2	Latin American Pamphlets	Yes
3	EDGAR Corporate Data	
4	EDGAR Corporate Data	Yes
5	Hancock Image Archive	
6	Interactive TV Material	
7	Technical Reports	
8	Cinema-TV Moving Images	Yes
9	Hancock Image Archive	Yes *
10	Technical Reports	Yes
11	Maps	
12	Searchable Archives for Images	Yes *
13	Korean-American Museum	
14	Planning Documents	
15	Medieval Manuscripts	Yes

* Continued into development as a combined product

2 Project Context

The projects were a diverse set of multimedia archive applications desired by USC Library clients (see Table 1). Their client-driven, evolving nature made the set of projects more of an observatory than an experimental laboratory.

2.1 Projects

Involved in the projects were a dozen librarians from the USC Library (University of Southern California), about 90 students from the computer science graduate-level course *Software Engineering*, as well as faculty and students from the USC Center for Software Engineering and a group of behavioral scientists from the USC Annenberg School of Communication (the latter were asked to independently observe and analyze the software requirements process). The projects were conducted over a period of two semesters, starting Fall 1996 and continuing through Spring 1997.

The projects focused on different forms of multimedia data like movies, books, manuscripts, pamphlets, pictures, papers, and so on. However, the project goals were diverse because requirements of an archive for technical reports for instance are quite different from those of an archive for medieval manuscripts. The

Table 2: Example Library Multimedia Problem Statements

Problem Set #5: Hancock Image Archive 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 #3: EDGAR Corporate Data 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 #8: CNTV 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.

students had to think about good ways of making this wide array of multimedia data accessible. One project even had to deal with multi-linguistic problems. For some teams, the user community was very specialized and for other teams, the users could include the general public.

Altogether 12 project topics were chosen for this activity, six of which were continued for a second semester. See Table 2 for more detail on a few projects.

We were not able to evaluate the Medieval Manuscripts team because their electronic data was deleted before we were able to capture it. Teams 3 and 4, 5 and 9, 7 and 10 had a special role because they worked on the same problem sets.

Having done a similar activity the previous year as described in [11], we found that there was one important thing missing. The previous year we used students of the same class to independently develop the same component of an imaginary library system. With these results, we had an excellent opportunity to observe how approximately 25 independent teams would develop the same system using the same basic requirements to start with. This activity had the drawback that the projects were imaginary, which means that we did not have real customers. The students role-played the user, customer and developer roles.
This year's activity corrected this limitation by introducing real customers (USC library staff) and users. This introduced problems, which only happen in the real world such as fuzzy requirements, conflicts with resources and personnel, and so on. Further, we improved the activity in yet another way. Since we had real customers, they also wanted to see real results. Thus, a selected number of six projects were continued for a second semester so that a final product with sufficient initial capabilities could be released thereafter (see "Continued" column in Table 1).

The 14-project sample size compares well with other sources of data on real-client requirements negotiation processes. The single-site, single-domain context may bound the generality of the results. However, further proj-

Table 3: 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
System Prototype(s)	Exercise key usage scenarios Resolve critical risks	Exercise range of usage scenarios Resolve major outstanding risks
Definition of System Requirements	Top-level functions, interfaces, quality attribute levels, including: - Growth vectors - 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

ects are being (or are about to be) repeated on different sites (such as George Mason University and Johannes Kepler University) with the intent of replicating and extending our results in different domains and more complex applications.

2.2 Milestone Elements

The teams followed a WinWin refinement of the WinWin Spiral Model [2] described in *Anchoring the Software Process* [5], two generally applicable milestones were defined for the WinWin spiral model, called the *Life Cycle Objectives* (LCO) and the *Life Cycle Architecture* (LCA) (see Table 3). Each milestone corresponds to one spiral cycle and the LCA milestone is a refinement (a later cycle) of the LCO. Each milestone is divided into milestone elements, such as operational concept, system requirements, software architecture, plan and feasibility rationale. The table entries contain information of what is expected to be completed for a certain milestone and for a particular milestone element.

An initial milestone, completion of the WinWin requirements negotiation, was due at the end of Week 4. The LCO milestone was due in Week 6 and the LCA milestone was completed in Week 11 at the end of the first semester, including a prototype, which was mostly done as part of the second cycle. Thus, the net result of the first semester's activity was to go from the one-paragraph problem statements exemplified in Table 2 to LCA packages of roughly 200 pages plus a prototype.

The second semester started off by revisiting the LCA deliverables and continuing on to the IOC (Initial Operational Capabilities) milestone, which was due at the end of the second term.

The LCA and IOC milestones and their activities are subject of a separate paper [6]. However, we will discuss the LCO milestone since it was the first milestone after the successful completion of the requirements negotiation. Thus, the deliverables of the LCO milestone were most affected by the negotiation.

2.3 People

During the first semester (LCO and LCA), the student teams largely consisted of six people each. To emphasize the importance of the LCO/LCA milestones and elements, each team member was given primary responsibility for one of the six milestone elements shown in Table 3.

All students were graduate-level students in computer science, some of them working towards a Ph.D. Most of the 90 students were full-time students. A smaller group of about 20 people were remote students who were working full/part time at their company and who were also taking classes at the same time. Those remote students also deserve our special attention because they tend to be higher-experience people. Teams with mostly remote people in them were team 1 (half from JPL), team 2 (full XEROX), team 9 (full Rockwell, Allied Signal, and CATE), and team 15 (full Hughes).

2.4 WinWin System

The WinWin system is a requirements negotiation tool which supports the collaboration of a number of stakeholders with the goal of identifying, analyzing, and reconciling requirements. The WinWin System is based on the WinWin negotiation model [3][4][15]. It was used by all teams to negotiate the requirements of their systems and it is primarily based on four artifact types: *Win Conditions*, *Issues*, *Options*, and *Agreements*. Win Conditions capture the stakeholders' goals and concerns with respect to a new system. If a Win Condition is non-controversial, it is covered by an Agreement (see Figure 1). Otherwise, an Issue artifact is created to record the resulting conflict among Win Conditions. Options allow stakeholders to suggest alternative solutions, which address Issues. Finally, Agreements may be used to adopt an Option, which resolves the Issue.

The WinWin system also includes a tailorable Domain Taxonomy which enables Stakeholders to link artifacts to taxonomy items and to access those artifacts via the taxonomy. Table 4 shows a simplified version of the multimedia-archive domain taxonomy we gave the students before the negotiation started. The taxonomy structure was also used as the Table of Contents for the Requirements Description. The teams were asked to enter Win Conditions and negotiate Agreements covering each appropriate Domain Taxonomy element. This was a major factor conditioning the negotiation.

Figure 1: WinWin Artifact Relationships and Taxonomy

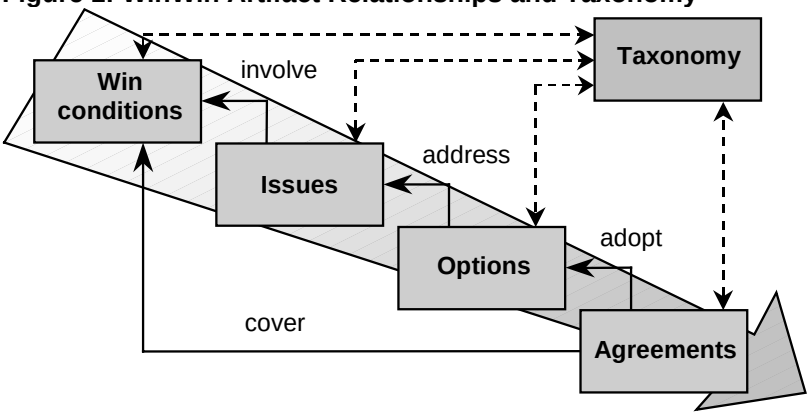


Table 4: Simplified Taxonomy [6]

1 Media operations	4 Quality
1.1 Query/Search/Browse	4.1 Response Time
1.2 Access Control	4.2 Reliability
1.3 Audio/Video Operation	4.3 Security
1.4 Update/Input	4.4 Usability
1.9 Others	4.5 Interoperability
2 Interface	4.6 Workload
2.1 COTS (SIRSI, etc.)	4.7 Cost
2.2 Database (File Access)	4.8 Schedule
2.3 User/Admin. Interface	4.9 Others
2.9 Others	5 Operations/Environment
3 Administration	6 Development/Process
3.1 User Management	7 Evolution/Maintenance
3.2 Usage Monitoring	9 Others
3.9 Others	

The first three items in this taxonomy have to do with functional capabilities for user, administrator and interfaces to other products. The fourth item contains non-functional quality attributes. Item 5 describes the operational environment and the remaining items cover development and maintenance.

The Library customers did not use the WinWin tool directly during the negotiation due to time constraints. Instead, it was decided that two of the team members would serve as scribes for their respective customers. These scribes were generally the people who were in charge of the operational concept and the requirements descriptions.

For the WinWin negotiation activity, the team members assumed different roles. The architecture and prototype team members functioned as developers, the plan and rationale members were the customers. The scribes to the librarians became the users. The team leader was always the rationale stakeholder.

For more information about the tool please see [5], [15], http://sunset.usc.edu/Ongoing_Research/topics.html.

2.5 Constraints on Analysis

There are a number of questions that one would like to answer by analyzing the results of the WinWin negotiation, such as:

- *How are people and negotiation results affected by using a negotiation tool such as WinWin?*
- *How do people factors, like work experience or age, effect the process and the outcome of the negotiation?*
- *Do people use the tool as it was anticipated by the model?*

However, various constraints preclude being able to provide definitive answers on all such questions. As discussed above, considerations of real-world client satisfaction, fairness of grading, and changes in operating conditions (e.g. availability of Library support software) caused this to be more of an observatory than an experimental laboratory. The type of application varied from a pure archive to access facilitation for dynamic information sources on the Web. Students self-selected their teammates and applications; thus, we could not cleanly compare teams with and without application experience. Many factors are difficult to quantify. Some quantified data are hard to fully interpret. For example, we captured total WinWin system login and logoff times, and individual timestamped transactions, but cannot say how much of the login time was spent doing real negotiation work.

2.6 Data Collection

To deal with the constraints, we obtained data from multiple sources:

- WinWin login times and timestamped transactions
- Structured questionnaires formulated and analyzed by behavioral scientists [17]
- Partly-structured evaluation memoranda by the Library clients
- Student critiques structured only by the question, "If we were to do the project over again, how would we do it better?"
- Questionnaire data submitted by participants on their years of practical experience, facility with English, etc.
- Grades on the students' LCO packages, based primarily on thoroughness of artifact and application coverage, and on conceptual integrity.

The hypotheses and results summarized below are conditioned by the constraints and data available.

3 Key Hypotheses and Results

Hypotheses were formulated, tested, and iterated in the following major areas:

A. Relations among WinWin artifacts;
B. Effects of negotiator characteristics;
C. Effects of negotiation process;
D. WinWin contributions;
E. Needed WinWin improvements.

Most of the hypotheses in Area A, B, and C could be investigated quantitatively, based on WinWin negotiation artifact counts and time histories, participant attributes, structured questionnaire results, and outcome attributes (e.g., LCO grade, customer satisfaction level). Since we had no projects not using the WinWin approach, the best we could do in areas D and E was to compare frequencies of students' comments and librarians' satisfaction levels on aspects of WinWin contributions and needed improvements. Key hypotheses and results in each area are summarized below.

A. Relations Among WinWin Artifacts

We wished to determine whether requirements negotiation issues tend to be simple or complex, as this affects the need for relatively simple or powerful negotiation support tools and frameworks. We also wished to see the extent to which our guidance to teams (consider each domain taxonomy element as a source of Win Conditions and Agreements) influenced team behavior, and to look for patterns of regularity among artifact counts.

Hypothesis A1. Most win conditions will be non-controversial: As indicated in Table 6, only 45% (232 of 513) of the win conditions across all teams were involved in issues. "Projects had twice as many win conditions as issues" was true in 13 out of 14 cases.

Table 6: Complexity of win cond. and issues

Condition	Number of Artifacts
Win conditions involved in issues	232 / 513
One option per issue	123 / 179
More than one option per issue	56 / 179

Hypothesis A2. Most issues will be straightforward to resolve: As indicated in Table 6, 69% (123 of 179) issues required only one option to resolve.

Hypothesis A3. Most relationships among artifacts are simple 1:1 relationships: As indicated in Table 5, 1:1 relationships exceed 1:m relationships by facts of 3 to 8 (the table reflects the relationships in both directions, e.g., Agreements: Options relationships and Options: Agreements relationships).

Taken together, the results of these three hypotheses indicate that requirements negotiation support systems should focus at least as much on handling simple relationships well as on handling complex relationships well. The current trend tends to be in the other direction, e.g., extending gIBIS [8] towards more complex relationships [20], [1]. This result is in the spirit of Knuth [16], which found that 68% of a sample of 250,000 Fortran statements were of the form A=B, and formed a similar conclusion about compilers.

Table 5: Ratio of 1:1 vs. 1:m relationships

Artifact types	Ratio
Agreements : Options (and vice versa)	8.3
Agreements : Win conditions (and vice versa)	5.2
Options : Issues (and vice versa)	6.2
Issues : Win conditions (and vice versa)	3.0

Hypothesis A4. Negotiation activity will be uniformly distributed across the taxonomy elements: As indicated in Figure 3 the number of artifacts were not distributed uniformly across taxonomy elements. Quality, operational, media, and development artifacts were among the most important. Administration, maintenance, and interface artifacts were among the least important. Figure 2 shows that not all teams were concerned with all taxonomy items. A few of the quality items (e.g., interoperability) and the administration items were not covered at all by many teams. Thus, the course guidance to consider each taxonomy element as a source of win conditions and agreements was not considered as a mandate to cover each taxonomy element.

Figure 2: Number of teams having at least one artifact by taxonomy element

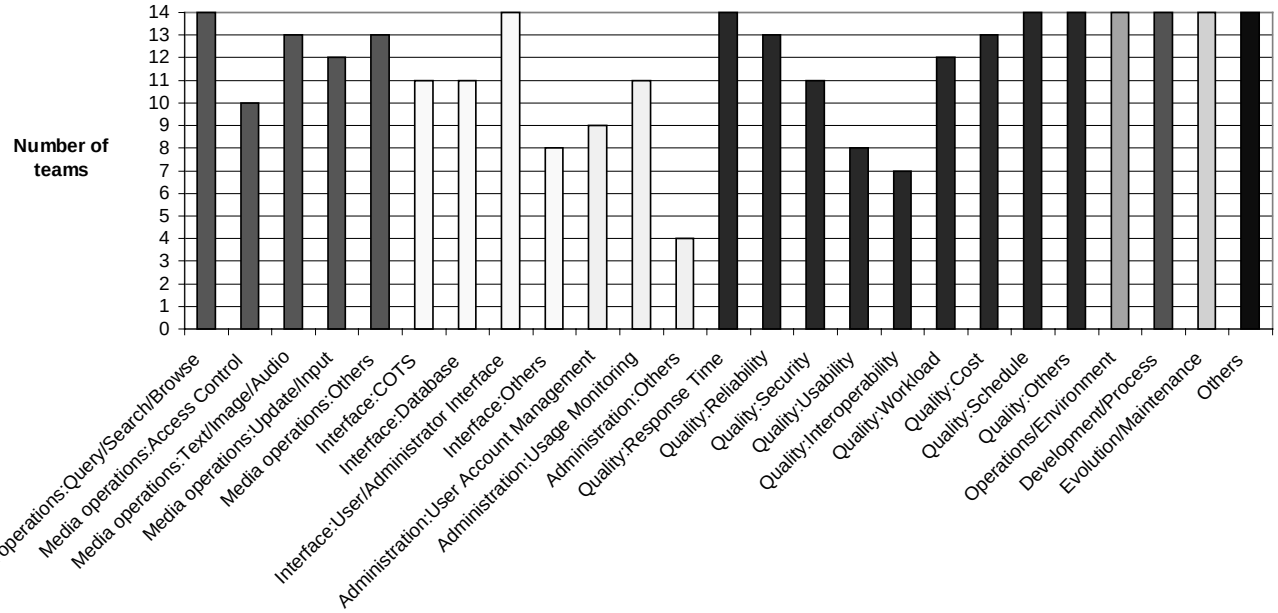


Figure 3: Number of Artifacts related to taxonomy

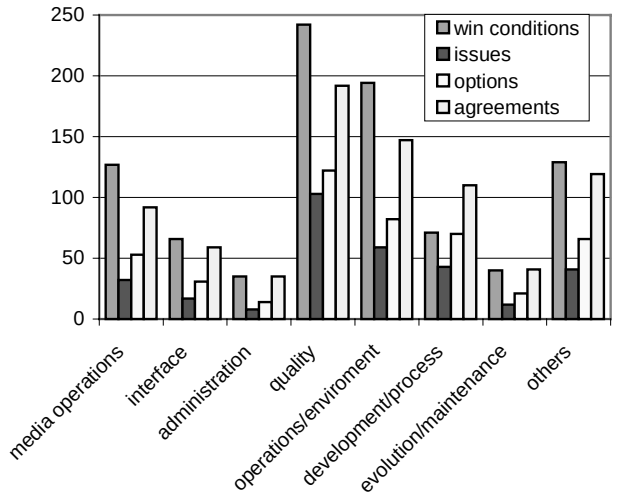
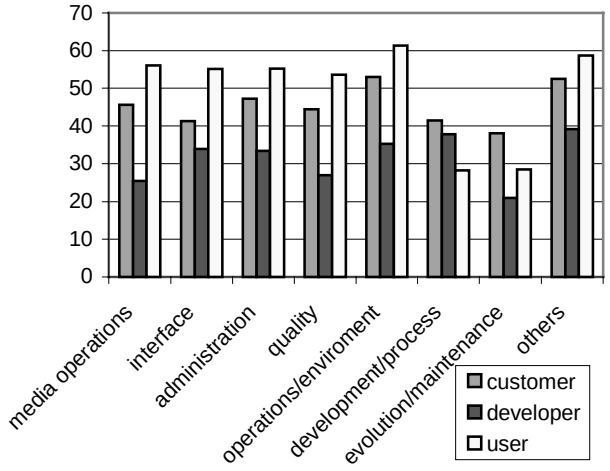


Figure 4: Time to complete artifacts by taxonomy classes and role



Hypothesis A5. Some relations among artifacts will be relatively invariant across teams: It was found that the number of messages produced correlated highly with the number of artifacts produced (except agreements which involve additional voting messages). The logon hours (which can be viewed as a proxy for effort) correlated well with the number of artifacts. However, for these cases and others, a few occasional outliers can be found.

Hypothesis A6. It will take a similar amount of time to resolve artifacts for each taxonomy element: As it can be seen in Figure 4, the average time duration to resolve artifacts tends to be somewhat lower for development and maintenance, areas where the domain expertise of the students was the best.

B. Effects of Negotiator Characteristics

One difficulty with analyses of student projects is the applicability of the results to industrial practice. We had an opportunity to test hypotheses about the differences, as half of our teams had average experience levels of *Medium* or *High*, and half had average experience levels of *Low* or *Very Low*.

As indicated in some of the A5 hypotheses about invariant artifact relationships, some negotiation features such as relative frequency of artifact relationships held about equally for more and less experienced teams. However, we wished to test further hypotheses about the relationships of team experience level to outcomes, and to investigate possible differences in negotiation behavior as a function of stakeholder roles.

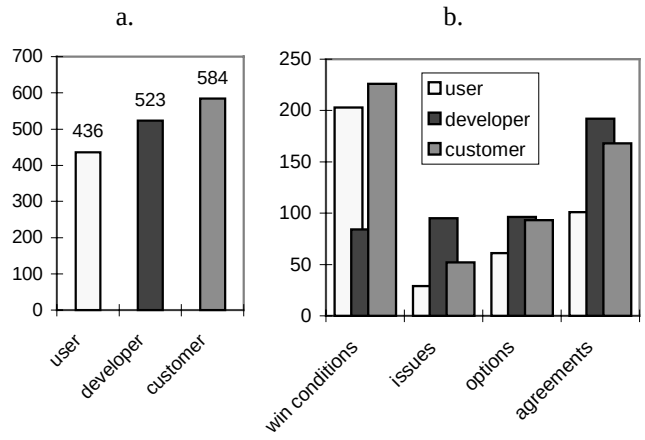
Hypothesis B1. More experienced teams will have some different negotiation characteristics: The primary difference was that the more experienced teams completed their negotiation with less effort (in 5 of 7 cases) and in a shorter amount of time (compressed schedule).

Hypothesis B2. More experienced teams will produce more effective negotiation results: This was measured by the teams' LCO package grades. The *medium* to *high* experience teams achieved an average LCO package grade of *medium* to *high.* The less experienced teams got on average *low* grades (see Table 7). A few low experience teams achieved higher LCO scores, but these teams invested more effort and excelled in artifact iterations and participation. Thus, it was possible for teams to compensate for lack of experience by good teamwork and extra effort.

Hypothesis B3. Some negotiator characteristics will correlate with higher success: The correlation of LCO grade with factors like experience in Hypothesis B2 is only one example. It was also found that customer satisfaction (evaluated by a survey and summarized in Table 9) correlated positively with team experience and team success. Further, "students who reported higher levels of satisfaction were also more likely to report that their groups worked well together and that their group meetings were more enjoyable and productive [17].

Hypothesis B4. Artifact contributions will vary by role: As indicated in Figure 5, stakeholders did not participate equally throughout the negotiation process. Users and customers originated more win conditions whereas developers and customers originated more agreements (Figure 5b). Users and Customers tended to start negotiation before the developers did. Further, users tended to use less

Figure 5: Number of artifacts per role, stakeholder, and artifact type per role



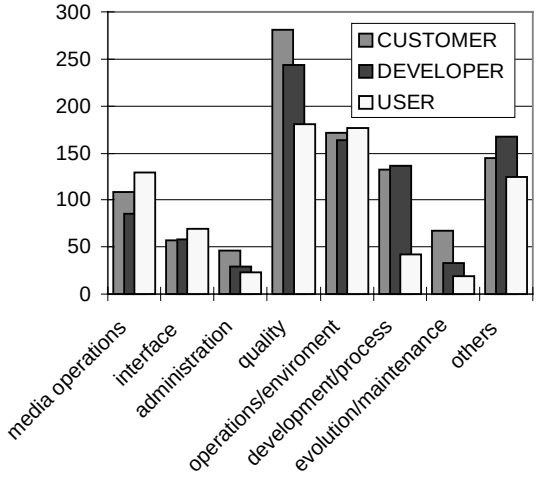
artifact types and created fewer artifacts.

This indicates that developers were more important during the issue resolution phase and later whereas the users were more important during the goal identification phase. Customers were mostly represented strongly throughout the negotiation (Figure 5b).

Further, it was found that stakeholders' interest in taxonomy items were not uniform either. Figure 7 indicates that users were stronger in functional and operational matters. Quality and development artifacts, on the other hand, were more important to customers and developers.

Another observation was that in most teams (12 out of 14) there were primary negotiators (who produced more artifacts) and secondary negotiators (with less artifacts).

Figure 7: Number of artifacts by taxonomy and role

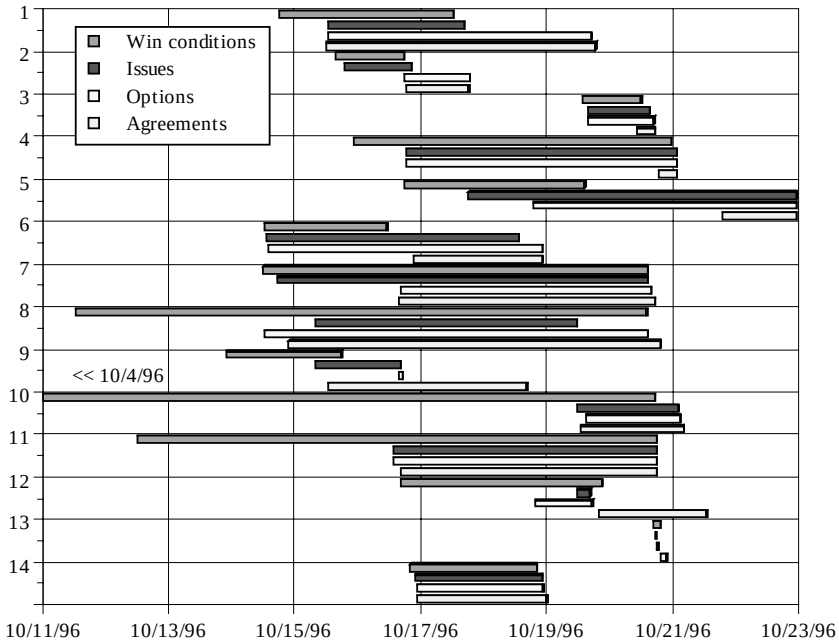


A possible limiting factor on the generality of these results was that the teams were self-selected and self-organized.

C. Effects of Negotiation Process

We wished to investigate whether the teams would follow the WinWin negotiation model as given in Figure 1 and its explanation. We also wished to investigate whether some negotiation patterns correlated better with successful results than others, and the extent to which WinWin usage would displace other modes of team interaction.

Figure 6: Creation of first vs. last artifact by team, type and time



Hypothesis C1. WinWin system usage will follow the WinWin negotiation model: It was found that the model conditioned the real usage in many cases. For instance, the number of win conditions, issues, and options entered did indeed correlate strongly with each other. It was also found that the amount of overhead (messages, etc.) created by each of these three artifacts was similar.

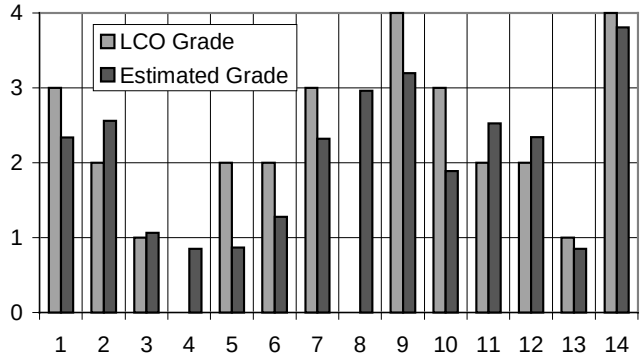
Agreements were an exception. It was expected that they would begin to be created as soon as a few win conditions were entered. This was found to be wrong. While win conditions, issues, and options were created and modified over an extended period of time, many teams started voting only towards the end (see Figure 6).

Nevertheless, we found that those teams who started passing agreements somewhat earlier did also end up with higher grades. Thus, it appears that the model's guidance is useful. This is also supported by the fact that

Table 7: LCO grade and negotation approach and results

Predictor	1	2	3	4	5	6	7	8	9	10	11	12	13	14
LCO grade	H	M	L	VL	M	M	H	VL	VH	H	M	M	L	VH
Experience	L	M	VL	VL	VL	M	L	M	H	L	M	M	VL	M
Iterations	M	L	L	L	VL	VL	H	H	L	H	H	L	L	VH
Productivity	VH	VH	H	M	VH	VL	M	M	VH	VL	VL	H	M	VH

Figure 8: Actual vs. Estimated LCO Grade



the more experienced teams intuitively followed this approach of voting as early on as possible, even though this meant revisiting passed agreements and re-voting them later on.

Hypothesis C2. Some negotiation patterns will correlate with more effective negotiation results (again measured by LCO package grades): Two features of the negotiation approach (iterations and productivity) and team experience were found to be a strong predictor for the LCO grade (the latter was already discussed in Hypothesis B2). The LCO package marked the end of the first life cycle iteration roughly two weeks after the negotiation had ended. Table 7 shows the grades of the LCO packages as well as the predictor variables that were found to be significant (Team 8 is the only exceptional case). The resulting correlation between estimated grade and actual grade is 0.82, which can be considered highly significant (see Figure 8).

The predictor variable *Iterations* rates whether new win conditions were introduced after voting had started or not (the overlap between win conditions and agreements can also be observed in Figure 6). A number of teams chose to follow a rather waterfall like negotiation approach starting with win conditions, issues, and options, and completing them before starting agreements. Other teams followed a more iterative process. The predictor variable *Productivity* measures the number of artifacts created divided by total login hours.

The differences in the results were strong: If the process was more iterative then the teams ended up with a *medium* to *very high* LCO grade. If a more waterfall like approach was followed then only a *very low* to *medium* LCO grade was achieved (with two exceptions: Team 8 and 9).

Hypothesis C3. Negotiation process patterns will be uniform across teams: Figure 9 indicates that most teams got together to use the WinWin tool at the same time and location. Many teams also interrupted negotiation for some time. Overall, however, the process patterns exhibit considerable diversity.

Hypothesis C4. Teams addressing the same problems will exhibit similar process characteristics: As indicated in Figure 6 and Figure 9, the three pairs of teams negotiating the same applications with the same clients (Team 3 and 4, 5 and 9, 7 and 10) exhibited widely varying requirements negotiation processes. This correlates with our previous finding that repeatable requirements processes are an unrealistic goal [11]. Nevertheless, there are negotiation patterns (e.g., starting agreements early, continuing to introduce win conditions) that are correlated to good results. We plan to incorporate these success factors into the negotiation guidelines for next year's projects, and hopefully converge on a set of best practices for repeatable successful negotiations.

Figure 9 shows the times during an 18-day period when stakeholders used the WinWin tool. E.g. in team 1 all stakeholders used the tool on day 1, 2, and 5. On day 3 it was used by the customer and user only and on day 4 it was not used at all.

Figure 9: Times people used WinWin (18 day period) by team number

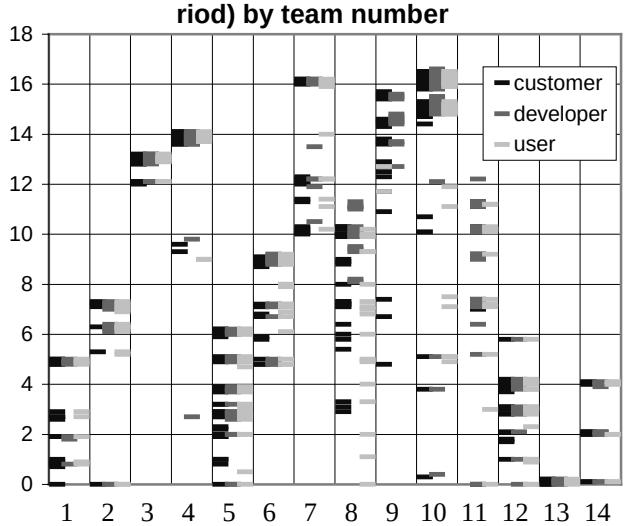
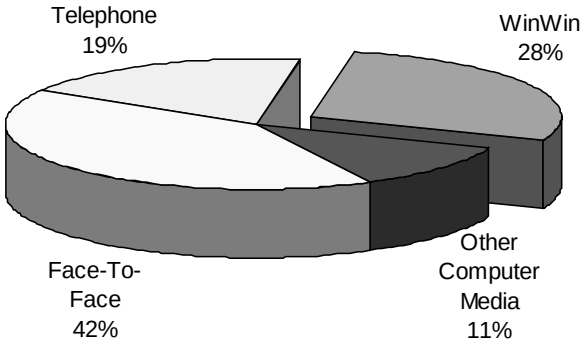


Figure 10: Usage of Negotiation Media



Hypothesis C5. The WinWin tools will not completely replace other modes of interaction: Figure 10 indicates that the WinWin tool was used for about one fourth of the negotiation time. This was the result of the independent behavioral study [17].

D. WinWin Contributions

Given that we had no teams not using the WinWin approach, the strongest type of hypothesis we could test was of the form, "Participants will identify some positive WinWin contributions to their requirements negotiation process and results." This was tested by the various structured questionnaire and non-directed critique results we analyzed. To some extent, therefore, the participants' previous experiences with non-WinWin approaches to requirements determination serve as a partial form of control case for the results.

For instance, the student critiques revealed among the most positive influences of the WinWin approach that it promoted more cooperativeness and mutual understanding, it focused on key issues, reduced friction and equalized participants (Table 8). Similar findings were made by the independent survey [17]. There it was concluded that if a team's group norm involved the understanding that giving feedback, having collective responsiveness, and having flexible approach; then the group would also be more satisfied with both project topic and negotiation approach.

Therefore, in supporting cooperation and mutual understanding, team satisfaction and success is increased. Further, good results in the eyes of the library customers was also correlated to customer responsiveness, as indicated in Table 9. The "willingness to do it again" has been verified in practice: not only individual librarians but also the Library's administration committed to forming further projects this year. For 1997-98 16 more digital library applications were prototyped and architected, and 5 are being developed and transitioned.

Table 8: Student Critiques Summary

Positive Comments about WinWin	Count
Promoted more cooperativeness and mutual understanding	9
Should continue use of WinWin	9
Focused team on key issues	8
Objective artifacts reduced friction, equalized loud and quiet participants	6
Helped in distributed collaboration	5
Helped create better requirements	4
Helped in understanding the requirements process	2
Helped in adapting to changes	2

Negative Comments about WinWin	Count
Need more pre-WinWin homework	14
Too much overhead in WinWin mechanics, bugs decreased negotiability	10
Prototype concurrently with WinWin conflict identificiation	10
Should have direct Librarian involvement	6
Complement WinWin with email, whiteboards, video conferencing, etc.	5
Need more time to do thoroughly	4
UNIX platform limitations	2
Need easier discussion support	2
Objective artifacts often hard to interpret	2

E. Needed WinWin Improvements

Similarly to Area D, our primary hypotheses in this area was, "Participants will identify some aspects of the WinWin process needing improvement." Our main source of data here was the student critiques (Table 8). Major improvements we implemented for this year's set of 16 projects included more WinWin training and WinWin homework, and improvement of the WinWin system. Further, we required the teams to develop a simple prototype (reflecting the user interface) concurrently while the WinWin negotiation was going on.

Table 9: Comments of Library Customers (1-disagree; 3-neutral; 5-agree)

Question	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Good responsiveness	4	4+	2	5	4	5	3	5	5	5	5	5	4	5	5
Good results	5	4+	3	5+	4	5	4	5	4+	4+	5	5	4	5	5
Frequent communication	3	3	2	3	4	5	1	4	4	5	5	-	2	5	4
Better understanding of software considerations	4	5	4	4	5	5	-	-	5	-	5	-	4	-	5
Increased confidence, trust, willing to do it again	4	5	3	5	4	5	3	5	4	4	5	5	4	5	4

4 Conclusions

Among the various results presented above, here are some of the most significant conclusions and implications.

- Most of the stakeholder Win Conditions were non-controversial (were not involved in Issues). Also, most Issues were decoupled from other Issues and were easy to resolve. This implies that requirements negotiation support systems should focus at least as much on handling simple relationships well as on handling complex relationships well.
- Negotiation activity varied by stakeholder role. Users and customers were more active in the early stages; developers and customers in the late stages.
- LCO package strength and consistency (measured by LCO grading criteria) could be predicted from three attributes (team experience, non-sequential negotiation, and efficiency in producing negotiation artifacts). Overall, more experienced teams performed better and somewhat differently, indicating that requirements negotiation is an area in which inexperienced-student experiment results may not be predictive of industrial practice.
- Similar applications projects did not follow similar processes, confirming our previous conclusion [11] that repeatability of software front-end processes is not a realistic goal.
- The strongest positive effects of using the WinWin approach were increasing cooperativeness, focusing participants on key issues, reducing friction, and facilitating distributed collaboration. The positive librarian evaluation have been followed up by both individual-librarian and institutional USC Library commitments to form further projects this year.
- The major improvements for the WinWin approach (now being implemented) were increasing WinWin training, reducing usage overhead, and concurrent negotiation and prototyping.

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: 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.

6 REFERENCES

1. Andersen Consulting, KBSA/ADM Tutorial, June 1997.
2. Boehm, B. W., "A Spiral Model of Software Development and Enhancement," *Computer*, May 1988, v. 21 no. 5, pp. 61-72.
3. Boehm, B.W. and Ross, R. "Theory W Software Project Management: Principles and Examples," *IEEE Transactions on Software Engineering*, July 1989, pp.902-916.
4. Boehm, B.W., Bose, P., Horowitz, E., Lee, M.J. "Software Requirements Negotiation and Renegotiation Aids: A Theory-W Based Spiral Approach", *Proceedings of ICSE-17*, April 1995, pp.243-253.
5. Boehm, B.W., "Anchoring the Software Process," *IEEE Software*, July 1996, pp.73-82
6. Boehm, B.W., Egyed, A., Kwan, J., Madachy, R., "Developing Multimedia Applications with the WinWin Spiral Model," *Proceedings, ESEC/FSE 97*, 1997.
7. E. Carmel, R. Whitaker, and J. George, " PD and Joint Application Design: A Transatlantic Comparison," *Communications ACM*, June 1993, pp. 40-48.
8. Conklin, J., Begeman, M., "gIBIS: A Hypertext Tool for Exploratory Policy Discussion," *ACM Trans. Office Info. Systems*, October 1988, pp. 303-331.
9. A. Dardenne, S. Fickas, and A. van Lamsweerde, "Goal-Directed Concept Acquisition in Requirement Elicitation," *Proceedings, IWSSD 6*, IEEE, October 1991, pp. 14-21.
10. DeMarco, T., "The Role of Software Development Methodologies: Past, Current and Future," Keynote Address, *ICSE 18*, IEEE/ACM, March 1996, pp. 2-4.
11. Egyed, A., Boehm, B., "Analysis of System Requirement Negotiation Behavior Patterns," *Proceedings of INCOSE-7*, August 1997, pp. 269-276.
12. 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.
13. R. Fisher, W. Ury, "Getting to Yes," *Penguin Books*, 1981.
14. Jarke M., Pohl, K., "Requirements Engineering in the Year 2001: On (Virtually) Managing a Changing Reality," Nature Report 95-03, ESPIRIT Project, Informatik V. RWTH Aachen, 1995.
15. Horowitz, E. "WinWin Reference Manual: A System

for Collaboration and Negotiation of Requirements", Center for Software Engineering, University of Southern California Technical Report, March 1996.
16. Knuth, D., "An Empirical Study of Fortran Programs," *Software Practice and Experience*, Vol. 1, 1971, p. 105.
17. Monge, P., Baskin, A., Fulk, J., Contractor, N., Signer, J., Gibbs, J., Lee, B., Wilson, M., "Report on WinWin Study Findings: Executive Summary", Annenberg School Report, May, 1997.
18. G. Mullery, "CORE: A Method for Controlled Requirements Specification," *Proceedings, ICSE 4*, IEEE, Septmember 1979, pp. 126-135.
19. C. Potts and K. Takahashi, "An Active Hypertext for System Requirements," *Proceedings, IWSSD 7*, IEEE, December 1993, pp. 62-68.
20. Ramesh B., Dhar, V., "Supporting Systems Development by Capturing Deliberations During Requirements Engineering," *IEEE Trans. Software Engineering*, June 1992, pp. 498-510.
21. Robinson, W. N., Volkov, S., "A Meta-Model for Restructuring Stakeholder Requirements," *Proceedings ICSE 1997*
22. M. Weiser, "Software Engineering that Matters to People," Keynote Address, ICSE 97, IEEE/ACM, May 1997.
23. E. Yourdon, "'Death March' Projects," Keynote Address, *ICSE 97*, IEEE/ACM, May 1997.