A Comparison Study in Software Requirements Negotiation

Published in the Proceedings of the 8th Annual International Symposium on Systems Engineering, 1998

Alexander Egyed and Barry Boehm
Center for Software Engineering
University of Southern California
Henry Salvatori Computer Science Building
Los Angeles, CA 90089-0781
{aegyed, boehm}@sunset.usc.edu

Abstract. In a period of two years, two rather independent experiments were conducted at the University of Southern California. In 1995, 23 three-person teams negotiated the requirements for a hypothetical library system. Then in 1996, 14 six-person teams negotiated the requirements for real multimedia related library systems.

A number of hypotheses were created to test how real software projects differ from hypothetical ones. Other hypotheses address differences in uniformity and repeatability.

The results indicate that repeatability in 1996 was even harder to achieve then in 1995 (Egyed-Boehm, 1996). Nevertheless, this paper presents some surprising commonalties between both years that indicate some areas of uniformity.

In both years, the same overall development process (spiral model) was followed, the same negotiation tools (WinWin System) were used, and the same people were doing the analysis of the findings. Thus, the comparison is less blurred by fundamental differences like terminology, process, etc.

INTRODUCTION

A year ago we presented an analysis of an experiment, which addressed the question of whether people-intensive activities like software requirements negotiation are repeatable. This was presented in the context of the SEI Capability Maturity Models (CMMs). The Software CMM identifies Repeatability as the goal of Level 2 processes (Paulk, et al, 1995), while the Systems Engineering CMM's Level 2 goal is planned and tracked processes (Bate, et al, 1995).

It was our conclusion that repeatability, if it can be achieved at all, would be very hard and would require strong proceduralization, which may lead to over-bureaucratization and which in turn may reduce the creativity of software designers.

The analysis was based on an experiment, in which we observed 23 three-person-teams in 1995 while they negotiated the requirements for a hypothetical project.

The main disadvantage of that analysis was that projects, conducted by students as part of their coursework, are only moderately representative of actual development practice. However, then we argued that such a setting could serve as a lower bound on the repeatability scale for software engineering processes and results.

In 1996 we addressed that issue in another way. We repeated a similar experiment in a real-client project environment. Here we observed 14 six-person-teams negotiating requirements for real customers. The following summarizes the key differences of the 1996 projects compared to 1995:

- Real customers and users
- Real vs. artificial conflicts to resolve
- Less negotiation pre-structuring, except for the use of a completeness checklist
- Complete development life-cycle performed (from inception to IOC) (Boehm, 1996)
- Double team size, however, same basic customer, developer and user roles

In the 1996 experiment we were even able to capture much more detailed information about the negotiation process and its results. However, due to limitation in space we cannot present all of that here. If interested please refer to (Boehm-Egyed, 1997) which is entirely devoted to the 1996 experiment.

In this paper we will analyze the commonalties and differences of both experiments and we will evaluate a number of hypotheses which have to do with the repeatability issue discussed above. The hypotheses and other observations are also put into context of the changed environment, which allows us to reason in what way hypothetical projects differ from real ones.

THE PROJECTS

All projects in both years were in the library domain. It was the students' task to negotiate the requirements of the proposed projects. The projects were either given to them (in 1995) or they selected them themselves (in 1996). During negotiation the team members assumed one of three stakeholder roles: developer, customer, or user. However, associated constraints with these roles differed between the years as described later. The following summarizes the projects.

1995 Projects. In 1995, 23 three-member student teams (actually 35 but some had incomplete data (Egyed-Boehm, 1996)) negotiated requirements of a hypothetical library system called SDI (Selective Dissemination of Information). The target environment was a hypothetical university with three campuses.

The goal was to integrate the libraries of these campuses and provide additional services like user interest profiling (compare new acquisitions with user profile and notify user if match is found). Additional monetary and schedule constraints were also given.

The basic components for the SDI system were defined in advance and estimation of their individual size in lines of code was provided as well. The students had the freedom to decide whether to incorporate those components and if yes in what level of detail. The latter allowed the students to choose different designs of the components which all provided basic capabilities and some additional 'nice-to-have' features. Components were User Interest Profiles, Access Control, Acquisition Handling, User Services, Usage Analysis, Trend Analysis, and Additional Network Access all edited to a client-server COTS package.

Besides the software components, the students had to make a few hardware choices as well. For instance, they had to choose the type of server, with different speed-risk tradeoffs.

The students were not asked to do the actual coding of the system. They only had to perform the initial stages of the development process, from requirements negotiation to high-level design. The projects ended after roughly 8 weeks.

To compensate for the lack of real customers, students were given special negotiation goals depending on the role they were playing. As described above, there were three stakeholder roles (customer, developer, and user) and each member of the team had to take one of these roles. The special goals just mentioned were provided with the intent to create some 'artificial conflicts', which the stakeholders had to master in order to complete the negotiation part of the development cycle.

Table 1: Project topics

1995	1996
Library SDI System Selective Dissemination of Information	Stereoscopic Slides
	Latin American Pamphlets
	EDGAR Corporate Data
	Hancock Image Archive
	Interactive TV Material
	Technical Reports
	Cinema-TV Moving Images
	Maps
	Searchable Archives for Images
	Korean-American Museum
	Planning Documents
	Medieval Manuscripts

For instance, the customer was burdened with a very tight cost constraint, the user got a list of his or her 'must have' items, and the developer was asked to minimize cost and schedule risks.

1996 Projects. In 1996, 14 six-member teams were evaluated (one other team had incomplete data (Boehm-Egyed, 1997)). Instead of being given a project, those teams selected their topic out of a wide range of multimedia related projects (see Table 1). Each project was conceived by a real customer from the USC Library and was derived out of a need in his/her community. Besides proposing the system, the library customers were also involved in negotiating the requirements with their student team(s) (a few customers initiated two topics or administered two teams working on the same topic). Further, the projects were planned and performed over a period of two semesters, with the ultimate goal of delivering a real product with sufficient initial capabilities at the end of the second semester.

Even though most projects were different from each other, many were built with a similar concept of providing some form of user and administrator interface on top of a multimedia database. Further, all projects were required to be developed using the World Wide Web as an interface (browser).

Since neither customer nor developer had detailed ideas on what the system should really look like, the 1996 projects were much less restrictive than the 1995 ones. In fact, there were a few cases where student teams actually came up with solutions which did not only solve the problem but also improved the entire business process much to the delight of the customers. Nevertheless, this new form of freedom brought also problems with it, like fuzzy requirements, personnel and resource conflicts, etc.

Figure 1: The WinWin Negotiation Model

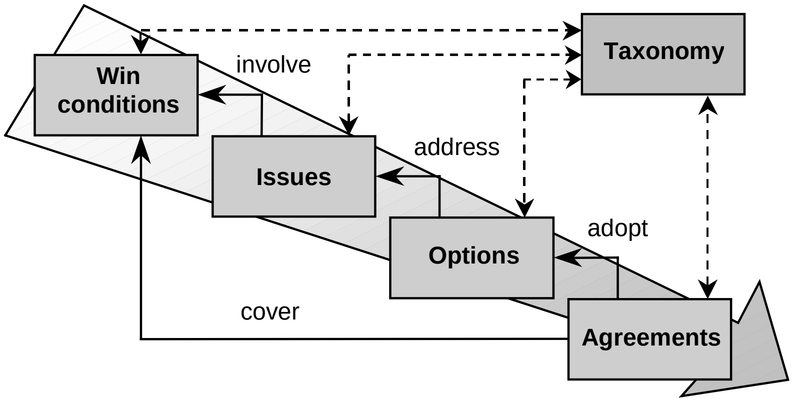


Table 2: Simplified Taxonomy

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	

Like the 1995 projects, the students had to find more than just a software solution (Boehm, 1994). They had to integrate new multimedia concepts into existing library hardware and COTS packages (e.g. Web browser). Further, they had to find procedures to dial with 'non-computer science-problems' like copyright issues or even how to scan fragile medieval manuscripts.

Like the 1995 projects these teams were also subdivided into the three basic stakeholder roles for the purpose of negotiating requirements (two students per role). The students who played the customer and user roles served as mediators to the real library customers and users since these people were not always available.

Furthermore, no additional artificial constraints were defined for the stakeholder roles. The real constraints were difficult enough. However, since the projects were required to be carried out in much more detail, we defined rules on who should become a developer, customer or user based on the development roles they were choosing. So for instance, the architect and prototype people represented the developer, the operational concept and requirements people represented users, and the team leader and life-cycle plan people represented the customers in the requirements negotiation.

At the end of the negotiation, the stakeholders assumed new roles as developers; each person with distinct primary responsibilities as described above (e.g. requirements, architecture, etc.). For more detailed information about the project environment and the involved people refer to (Egyed-Boehm, 1996), (Boehm-Egyed, 1997), and (Boehm, et al, 1997).

OUTLINE

In the following, this paper will cover the following items:

- Present the WinWin Negotiation Model which all teams used to conduct their negotiation.
- Present the relevant results of both 1995 and 1996 experiments. Due to limitation in space we defer some of the details to the individual experiment papers of 1995 (Egyed-Boehm, 1996) and 1996 (Boehm-Egyed, 1997).
- Compare both experiments in order to point out their commonalties and differences with respect to the 'key characteristics' described above.

WINWIN DEVELOPMENT MODEL

The WinWin development model incorporates a number of basic models; the *WinWin Spiral Model (Boehm, 1988)(Boehm,1996)*, the *WinWin Negotiation Model (Boehm, et al, 1995)(Lee, 1996)*, *COCOMO (Boehm, 1981)*, and others. It is out of the scope of this paper to address all of them. In the following, we will therefore only describe the WinWin negotiation model, which is based on the Theory W (Boehm-Ross, 1989)(Fisher-Ury, 1981), since many of the results presented in this paper are based on this model.

All teams in both years used the WinWin negotiation model and its supporting tool the *WinWin System* (Boehm, et al, 1995)(Horowitz, 1996) to negotiate the requirements for their system. It is based on four artifact types: Win Conditions, Issues, Options and Agreements (see Figure 1). Win conditions capture the stakeholder goals and concerns with respect to the new system. If a Win condition is non-controversial, it is adopted by an Agreement. 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. In the 1996 projects the taxonomy structure followed closely the table of contents of requirements documents tailored towards multimedia library applications (see Table 2). Thus, the negotiators used the taxonomy as a checklist for sufficient coverage.

The 1995 teams used the taxonomy in a similar context. However, their negotiation guidance was not so much to achieve sufficient coverage of taxonomy items but to resolve the built-in conflicts in the hypothetical SDI application.

LIMITATIONS

This paper addresses (and resolves) some of the limitations of the 1995 study because some of its limitations had to do with the fact that the project environment was more hypothetical than real. Nevertheless, a few limitations remain: Even though, the 1996 data collection was quite comprehensive, the 1995 study lacked some of the data. Thus, we can only compare data that was captured by both studies.

PROJECT ANALYSIS

A number of hypotheses were formulated about the individual experiments and the comparison. The experiment hypotheses are for the most part about uniformity, repeatability and negotiation results across the 23/14 teams. The comparison hypotheses are mostly generalizing those experiment hypotheses in order to allow meaningful comparisons. Hypotheses were investigated in the following categories:

- Negotiation Cardinality
- Effects of People
- Negotiation Schedule and Process

Some of the hypotheses in these categories were assessed quantitatively based on the information captured by the WinWin tool through its instrumentation. Other hypotheses were investigated using more qualitative means of evaluation like questionnaires, reviews, etc.

Negotiation Cardinality. A series of mostly quantitative hypotheses about the negotiation results can be investigated by looking at the cardinality of negotiation elements like artifacts, comments, relations, etc. For instance, a uniform solution path would imply a uniform number of artifacts for each type and a similar way of connecting them.

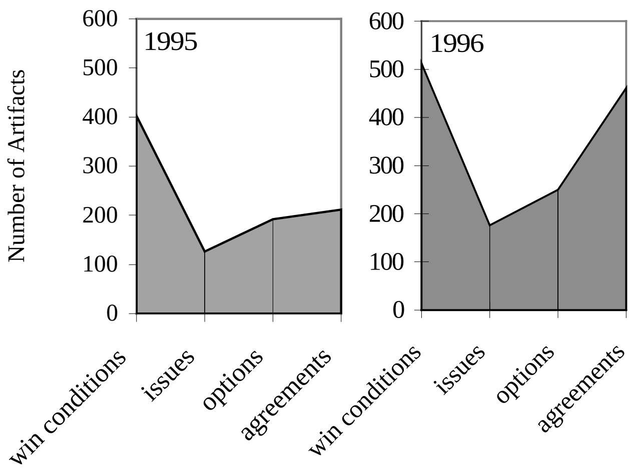
Hypothesis A1: Number of Artifacts, Comments, and Connections. Like in 1995 the hypothesis that *the number of artifacts, connections, comments, etc. created by the stakeholders would be similar for all teams (±10%)* was rejected in 1996 as well. The average, minimum, and maximum number of artifacts, comments, and connections were even more diverse in 1996. To get more insight, artifact types were evaluated in more detail (Figure 2). Here we can find strong similarities. We were able to accept the hypothesis that *Win Conditions would be the most common artifact type because they represent the knowledge base and that there would be more Options than Issues* for both years. Agreements, however, turned out to be exceptional. In 1996 the Agreement vs. Win Condition ratio was much higher than the corresponding one in 1995. This may be attributed to the fact that the 1996 projects were asked to use the Library Multimedia domain taxonomy as a checklist for formulating win conditions and agreements. The number of win conditions per team was considerably higher in 1996 (513/14=37 vs. 402/23=17).

Hypothesis A2: Most Win Conditions will be non-controversial. As indicated in Table 3 less than half of the win conditions (242 of 513) across all 1996 teams were involved in issues. In 1995 the number is higher with about 60% of all win conditions being involved by issues (246 of 402). Thus, although the hypothesis is accepted for the real 1996 projects, it is rejected for the 1995 projects, which were structured to emphasize conflicts.

Table 3: Connectivity Complexity

Conditions	1995	1996
Win conditions covered by issues	246/402	242/513
One option per issue	55/121	123/179
More than one option per issue	66/121	56/179

Figure 2: Number of Artifacts per Type



Hypothesis A3: Most issues will be straightforward to resolve. Like in the hypothesis above, 1995 projects were a bit more complex in that they had more complex options than the 1996 projects. Again we accept this hypothesis for the real 1996 projects but reject it for the artificial 1995 projects.

More hypotheses about this category, defined in (Egyed-Boehm, 1996) and (Boehm-Egyed, 1997), show further commonalties and discrepancies in cardinality of the same nature as presented above. Generally, the 1995 projects exhibited more complex artifact interrelationships. The higher complexity in 1995 may have been caused by a few strong constraints placed on the projects. For instance, they had to deal with a very tight cost constraint, which had side effects throughout the entire negotiation and on many artifacts. Overall, the 1995 projects were focused on conflict resolution. The 1996 projects were focused more on completeness of requirements, via the use if the domain taxonomy and the checklist for win conditions and agreements. The real 1996 projects, the main conclusions were that most requirements were non-controversial and that most issues were simple to resolve.

Effects of People. Of high interest was also whether the stakeholders showed similar unusual patterns in the way they interacted and collaborated and how this was affected by people factors like experience. We further wished to verify whether the stakeholders would again exhibit similar 'role behaviors' (Bullen-Bennett, 1990).

Hypothesis B1: Artifact Contribution will vary by Roles. In 1995 it was assumed that *all stakeholders within all teams would participate equally during all 'phases' of the negotiation […] regardless of artifact type*. Figure 3 shows the absolute number of artifacts for both 1995 (left side) and 1996 (right side). In 1995 it was found that users had almost only half of the number of artifacts the customers had. More detailed investigation revealed that this was also true for most teams individually. It was very surprising to see that the 1996 experiment resulted in a very similar stakeholder ranking. Again, the user has much fewer artifacts than the customer does; however, the ratio is not as extreme as in 1995. Thus we accept above hypothesis because uniformity of artifacts is not given for both years. Either customer or developer had more artifacts than the user with high significance (exceptions were 2 out of 23 in 1995 and 1 out of 14 in 1996).

Figure 3: Number of Artifacts

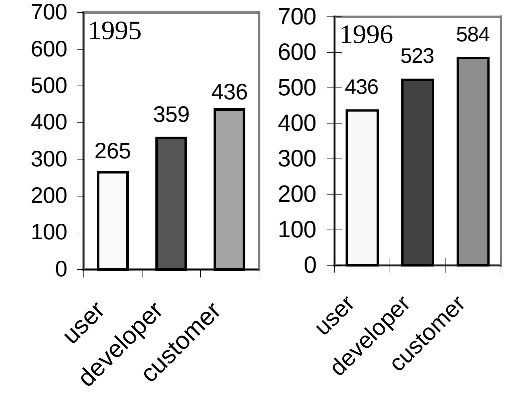


Table 4: Number of Artifacts per Roles and Artifact Type

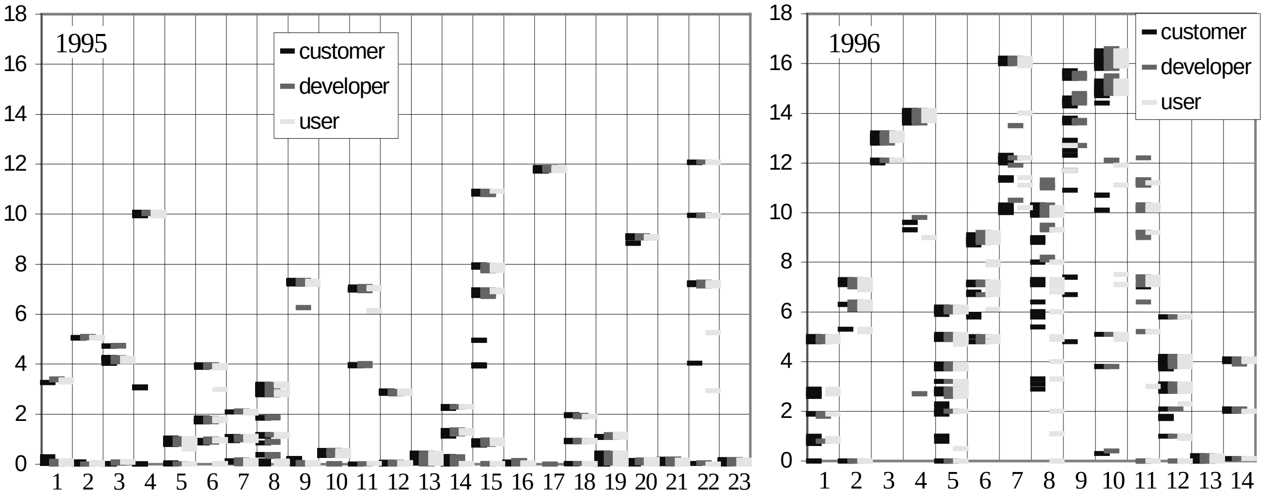
Artifacts 1995/1996	Win conditions	Issues	Options	Agreements
Customer	18/10	9/4	10/7	15/5
Developer	6/0	15/9	12/5	10/8
User	4/4	4/1	6/2	7/1

Hypothesis B2: Artifact Contribution will vary by Types. It was found in 1996 that users and customers originated more win conditions, developers originated the most issues, and developers and customers originated more options and agreements. This indicated that customers and users were more important during goal identification and developers were more important during risk (issue) identification and resolution. Comparing these figures with 1995 we identified a similar pattern. Table 4 shows the number artifacts per role and type. The number on the left is 1995 and the number on the right is 1996. Note the similar patterns in both years.

Hypothesis B3: More experienced teams will have some different negotiation characteristics. In 1996 the more experienced teams usually completed their project negotiations in a shorter amount of time and also applied a more iterative negotiation approach by introducing artifacts long after voting had started (many other teams did not do that).

We found that 1995 projects did not show these trends. This may be due to strong differences between both years in negotiation schedule and process, as it will be discussed in the next section. Unfortunately, we cannot elaborate on other people factors here since we only started capturing sufficient information in 1996. For instance, in 1996 (Boehm-Egyed, 1997) we found some very high correlation between experience, iteration, and LCO package grade (series of documents). Further, we were able to capture people 'satisfaction' of both customer (user) and developer, which showed some interesting patterns as well (Boehm-Egyed, 1997). For example, WinWin was found to promote cooperativeness, focus teams on key

Figure 4: Artifact Creation and Revision Table



issues, but would have been more effective with better preparation and concurrent prototyping.

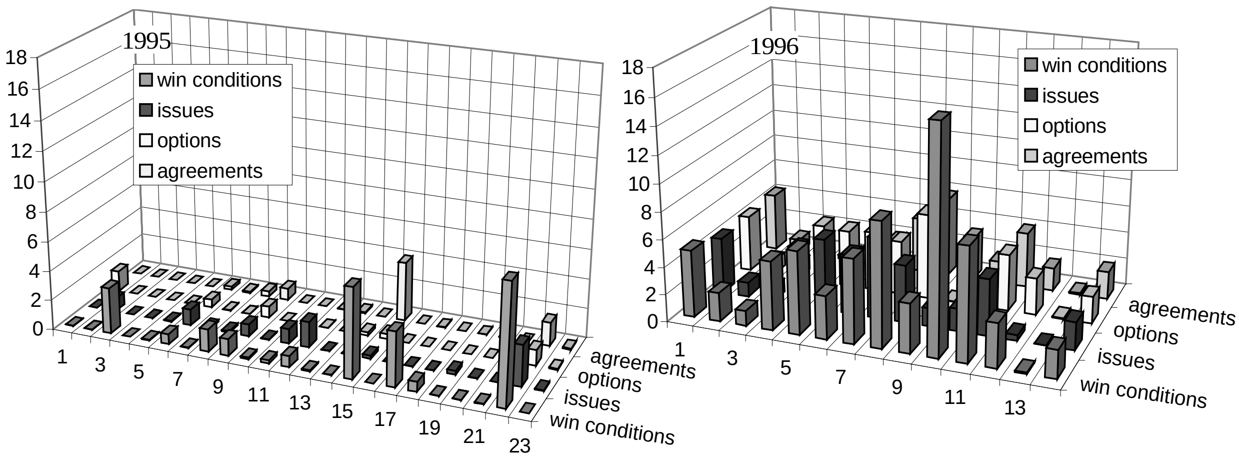
Negotiation Schedule and Process. This section will investigate the negotiation process and schedule of teams in both years. For instance, we wished to investigate whether both years would exhibit some similar negotiation patterns.

Hypothesis C1: Creation time of artifacts per role will be similar. Based on the role behavior patterns we described in section B, we wished to identify whether different stakeholders participated at different times during the negotiation. This hypothesis goes hand in hand with the hypothesis whether *the tool was used synchronously (at the same time) or asynchronously* (see Figure 4). It was found that stakeholders used the tool synchronously for the most time. This observation is also related to the finding that most stakeholders were involved (created or revised artifacts) from the beginning until the end.

Hypothesis C2: Creation time of artifact per type will be similar. It was expected that *Agreements would begin as soon as a few win conditions were entered.* This was found to be wrong for a large number of teams in both years. Many teams started to create agreements only after most if not all win conditions were entered. Further it was also found that both years the teams made rather poor use of time. Most teams used the WinWin tool only for a rather brief period in this multiple week activity, even less so in 1995.

Hypothesis C3: Artifacts will be revised with similar frequency. The 1995 teams produced in average only half as many artifacts but as it can also be seen in Figure 5 they spent less time (schedule) with the artifacts. It was therefore assumed that *the average time from artifact creation to artifact revision would be similar between both years*. This hypothesis turned out to be very wrong and had to be rejected. Figure 5 shows that 1995 projects rarely ever modified the same artifact for more than 1 or 2 days in average for all types. On

Figure 5: Average Duration to resolve Artifacts (first creation vs. last revision)



the other hand, 1996 project artifacts were usually modified over a much longer time frame.

This may be an indication that the higher degree of freedom in 1996 made it also very hard to find and resolve problems. Increases in schedule may also have been caused by the higher amount of artifacts produced, which in turn would cause more complex artifact dependencies. The next hypotheses seems to reflect this as well.

Hypothesis C4: Average number of iterations to resolve artifacts will be similar. Similarly, the number of iterations turned out to be much higher in 1996 than the ones in 1995. This issue was addressed in a previous hypothesis (C1) as well and is further supported by Figure 4. Figure 4 shows the creation and revision of artifacts for each team over a period of 18 days. We assumed that *the negotiation problems were rather simple and would require only one or two sessions to solve them*. This was mostly true for 1995 but we found that the hypothesis does not apply for 1996. Clearly, most teams in 1996 were not able to come up with a solution immediately (in the first session) but had to conduct multiple iterations over a period of almost 18 days.

Hypothesis C5: Average time to resolve artifacts will be similar (duration from problem to solution). In C3 we wished to evaluate the time duration an artifact got modified. On the other hand, here we wished to know how much time it would take to actually resolve an artifact. Thus, starting from its creation, how much time would be spend until it is *resolved* by an *passed* agreement (from problem to solution). Again, both years exhibited strong differences except for win conditions, which were. Most obvious are the differences in issues and options. It took less than a third of the time to resolve them in 1995 compared to 1996. On the other hand, it took somewhat more time to resolve 1995 agreements than the 1996 ones. Thus, we have to reject similarities between both years except for win conditions. Altogether, the average elapsed time to resolve artifacts of all types in 1996 was more than three times as much as in 1995 (4300 elapsed hours per team vs. 1300 elapsed hours).

KEY RESULTS DIFFERENCES

Both experiments exhibited a considerable number of commonalties and differences. In the following we will summarize them with special attention to the differences because they are a good source to reason about the effect of the differences in the project outline discussed above:

1995 Projects. The 1995 project boundaries were well understood because of the limitation in choices. It was easier to make more complex dependencies in such an artificially circumscribed domain. Therefore, it is not surprising that it took less time to find issues and options. On the other hand, the 1995 projects were primarily constrained from a monetary point of view. This constraint however may have been the cause for longer resolution times of agreements since it affects the entirety (configuration) of the system, which is primarily captured through the agreements. This would also justify why its quantitative artifact interdependencies were much more complex.

In 1995 the project setting did not require the students to go below high-level design issues. No implementation was required. Thus, it is not necessary to come up with a great number of negotiation items except for the ones, which are needed to solve the few built-in conflicts. Having less artifacts in a well understood domain would therefore reduce the time to resolve artifacts. Solutions in form of agreements were only needed to support high-level design issues and even there a number of design issues were already predefined and the students task was a matter of choosing between them

1996 Projects. The 1996 projects were only little understood by all participants (even the customers). Also, the 1996 projects emphasized requirements completeness via the use of the domain taxonomy as a checklist. The need to understand the problem in more detail implies a higher demand in time. The extra investment of time is reflected in the higher number of artifacts (which is not caused by the higher number of participants as it was shown in (Boehm-Egyed, 1997)).

Due to some extent to the domain taxonomy stimulus, the 1996 project exhibited less complex inter-artifact connectivity. Problems were divided into sub-problems and often resolved individually (as it was shown in (Boehm-Egyed, 1997)). This and the broader range of real library client desires and real COTS package constraints may have been responsible for increased time to resolve issues and options. It was harder to find options which would resolve issues than to finally vote on an agreement once something acceptable was found.

Since the 1996 negotiation guideline was defined through its taxonomy, the negotiation was much more solution oriented by cross referencing the negotiation items to the table of contents of requirements documents. Thus a great number of agreements had to be created to cover all taxonomy items sufficiently.

Both projects also exhibited a number of commonalties. Sometimes these commonalties were weaker in one year than in the other, but they were still recognizable.

CONCLUSIONS

The primary conclusions from the 1995 projects – that requirements negotiation patterns did not exhibit repeatability – was confirmed by the 1996 projects. The diversity in patterns of artifact creation and revision in Figure 4 and Figure 5 is roughly equivalent for the 1995 and 1996 projects (the greater uniformity for 1996 projects is due primarily to the shorter deadline for completing the negotiation). Thus, the goal of achieving repeatability in requirements engineering processes continues to appear unrealistic.

ACKNOWLEDGEMENT

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.

REFERENCES

Bate, R., et al, “A Systems Engineering Capability Maturity Model,” Technical Report CMU/SEI-95-MM-003, Software Engineering Institute, Pittsburgh, PA 15213, November 1995

Boehm, B.W. *Software Engineering Economics*, Prentice Hall, 1981

Boehm, B.W. “A Spiral Model of Software Development and Enhancement,” *IEEE Computer*, May 1988, pp. 61-72

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

Boehm, B.W., Bose, P., Horowitz, E., Lee, M.J. “Software Requirements As Negotiated Win Conditions”, *Proceedings of ICRE*, April 1994, pp.74-83

Boehm, B.W., “Integrated Software Engineering and System Engineering,” *The Journal of NCOSE*, Vol. I, No. I, July/September 1994, pp. 61-67

Boehm, B.W., “Anchoring the Software Process,“ *IEEE Software*, July 1996, pp.73-82

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

Boehm, B., Egyed, A., “WinWin Requirements Negotiation: A Multi-Project Analysis,” USC-CSE Technical Report, January 1997, at http://sunset.usc.edu/TechRpts/ usc-cse-97-508.html (submitted for ICSE’98)

Boehm, B.W., Egyed, A., Kwan, J., Madachy, R., “Developing Multimedia Applications with the WinWin Spiral Model,” *Proceedings, ESEC/FSE 97*, 1997.

Bullen, C.V., Bennet, J.L., “Learning from User Experience with Groupware”, Conference on Computer-Supported Cooperative Work, October 1990, pp.291-302

Egyed, A., Boehm, B., “Analysis of System Requirement Negotiation Behavior Patterns,” *Proceedings of INCOSE-7*, August 1997, pp. 269-276.

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

Lee, M.J., “Foundations of the WinWin Requirements Negotiation System,” Ph.D. Dissertation, Center for Software Engineering, University of Southern California Technical Report, May 1996

Paulk, M.C., Weber, C.V., Curtis, B., Chrissis, M.B., *The Capability Maturity Model - Guidelines for Improving the Software Process*, Addison-Wesley, 1995

Fisher, R., Ury W., “Getting to Yes,” *Penguin Books*, 1981.

BIOGRAPHIES

Alexander Egyed is a PhD student at the Center for Software Engineering at the University of Southern California. His research interests are in software architecture, and requirements negotiation. He received a Dipl.-Ing. in Informatics from the Johannes Kepler University in Linz, Austria and a MS in Computer Science from USC.

Barry Boehm is the TRW Professor of Software Engineering and Director of the Center for Software Engineering at the University of Southern California. His current research involves the WinWin groupware system for software requirements negotiation, architecture-based models of software quality attributes, and the COCOMO II cost-estimation model. Boehm received a BA in mathematics from Harvard University and an MS and PhD in mathematics from the University of California at Los Angeles. He is a fellow of the IEEE and the AIAA.