Experimental Evaluation of Wiki Technology and the Shaper Role in Rapid Interdisciplinary Requirements Negotiation

Di Wu¹, Da Yang², Supannika Koolmanojwong¹, Barry W. Boehm¹
¹University of Southern California, 941 w. 37th Place Los Angeles, CA 90089-0781
²Laboratory for Internet Software Technologies, Institute of Software, Chinese Academy of Sciences, Beijing 100080, China
{diwu, koolman, boehm}@sunset.usc.edu, yangda@itechs.iscas.ac.cn

Abstract
The challenges driven by multi-culture, multi-discipline stakeholders collaborating in a rapidly changing global environment necessitates an easily approachable mechanism for negotiating WinWin outcomes. Wiki has become an enabling technology for interdisciplinary collaboration. Following our initial development of a wiki-based requirements negotiation support tool – WikiWinWin, we experimented with using WikiWinWin to negotiate requirements in 20 real-client projects at University of Southern California (USC). Our initial results indicated better project outcomes were achieved as the use of the tool increased, as compared with the previous EasyWinWin tool.

1. Introduction
1.1 The Challenge
In his seminal work on The Two Cultures [1], C. P. Snow found that science and technology policymaking was extremely difficult because it required the combined expertise of both scientists and politicians, whose two cultures had little understanding of each other’s principles and practices.

However, recent books such as The World is Flat [2] and Competing on Internet Time [3] indicate that far greater collaborative challenges await the designers of future software-intensive systems. These challenges are driven by opportunities for competitive success via rapid collaborative design and development of globally usable systems and services by multi-stakeholder, multi-culture, multi-discipline, globally distributed teams of teams. Empirical studies of communication shortfalls [4] and reported 50% failure rates of global outsourcing of software development projects [5] testify to the consequences of neglecting interdisciplinary collaboration critical success factors in such situations.

1.2 The Underlying Theory and Process
The Value-Based Systems and Software Engineering (VBSE) theory and process [6, 7] provides a framework for stakeholders to better understand each other’s value propositions, and to effectively collaborate to achieve mutually satisfactory enterprise outcomes. The VBSE theory rests on two primary theorems. The Enterprise Success Theorem, “Your enterprise will succeed if and only if it makes winners of its success-critical stakeholders,” provides necessary and sufficient conditions for a successful enterprise such as a software-intensive systems project [8]. The Enterprise Success Realization Theorem provides sufficient but not necessary conditions for a project to succeed:
• Identifying all of the success-critical stakeholders (primarily involves dependency theory)
• Understanding how the success-critical stakeholders want to win (primarily involves utility theory)
• Having the success-critical stakeholders negotiate win-win product and process plans (primarily involves negotiation theory and other aspects of decision theory)
• Adaptively controlling progress toward a success-critical stakeholder win-win outcome (primarily involves adaptive control theory).

The conditions are not necessary because some lucky projects violate all of the conditions but are saved in the end by the emergence of a new COTS product or the like.

The Enterprise Success Theorem and the four Enterprise Success Realization Theorem conditions establish not only a theory, but also a 4+1 process architecture and a 6-step process for achieving a
successful outcome, as shown in Figure 1. This process has been integrated with the spiral model of system and software development and evolution [6] and its next-generation successor, the Incremental Commitment Model [7].

![Diagram of VBSSE 6-step Process Framework]

**Figure 1: VBSSE 6-step Process Framework**

### 1.3 Examples of Rapid Interdisciplinary Collaboration Tools

A major remaining challenge has been to develop mechanisms for interdisciplinary stakeholders to more rapidly understand each other and to converge on mutually satisfactory outcomes. Over the past 10 years of evolving the VBSSE approach on very small to very large projects, we have developed several techniques for rapid interdisciplinary collaboration. These have included model clash analysis [8], stakeholder simplifier and complicator lists [9], software cost/schedule/quality tradeoff tools [10], and the WinWin family of requirements negotiation tools [11].

Our initial experience with wiki technology convinced us to experiment with creating and applying a WikiWinWin tool to see if it made it easier for clients and student teams to self-facilitate the WinWin negotiation process in comparison with our existing EasyWinWin system [15].

### 1.4 WikiWinWin

WikiWinWin was designed as a successor to the previous EasyWinWin support system. The foundation of Wiki WikiWinWin is the WinWin requirements negotiation approach and the WinWin negotiation model, whose theoretical basis was developed in [14]. This negotiation model is used in WikiWinWin to guide the stakeholders working out mutually satisfactory requirements. During each step, the system displays one or more tools with which the team can generate, organize, and evaluate concepts and information. The detailed background information of the WinWin negotiation model and the WikiWinWin process can be found in our previous paper [12].

We had the initial evaluation experience with WikiWinWin using 20 real-client projects in the fall 2007 graduate software engineering course at University of Southern California (USC). The goal of this paper is to describe our experience and report what we have learned from the initial experiment with WikiWinWin.

The rest of the paper is organized as the following: Section 2 provides background information. Section 3 describes the experimental experience. Section 4 discusses lessons learned and possible extensions. Section 5 summarizes our conclusions with future work.
2. Background

2.1. Wiki

The utility of wiki technology for collaboration is widely recognized in corporate and open source communities [17, 18, 19]. Wiki has been identified as the enabling technology to change the way how people collaborate within and between firms [16]. Many organizations, such as SAP, use wiki internally for team collaboration. Novell was one of the early companies using wikis to enhance customer engagement (coolsolutions.com) [21]. The well-known open source community-Wikipedia [20] uses a wiki platform to support online collaboration among 50,000 contributors.

A wiki is a program that makes it exceptionally easy for a group of collaborating users to manage, manipulate, and publish web pages. It has several key characteristics [30]:

• It uses simple syntax and conventions to allow users easy editing or organizing of web pages.
• It enables web documents to be authored collectively without individual ownership of the document.
• The wiki can preserve the revision history, allow new information to update, and overwrite the old version.
• Wiki in general makes a basic assumption of the goodness of people. It allows content to be immediately published upon being saved. And it relies on the community of users to catch malicious content and correct it.
• The collaborative nature of wiki is established by providing a platform where everybody has read and write access to the web pages.

2.2. Roles

The users of wiki are encouraged to contribute content to the web pages and modify others' pages. Studies report wiki users do not exhibit the same level of participation [22, 23]. There are two types of contribution in using wikis: 1) contributing personal knowledge to the networks; 2) contributing primarily by repackaging the works of others, also known as shaping [25].

Shaping is defined as the act of integrating, distilling, refactoring, identifying areas of convergence and discrepancies, identifying topics receiving little attention in the community, and significantly rewriting the contributions of others.

Participants who primarily contribute their personal knowledge are called Personal Knowledge Contributors (PKC). Participants who perform the role of shaping are called shapers.

We adopted the above roles in our WikiWinWin experience.

2.3 Shaper

A recent research by Majchrzak [23] identified the shaper role as a critical success factor of many corporate wiki usages because shaper is more willing to focusing on identifying business opportunities for the group.

Adopting the shaper role is likely to bring a number of benefits to requirements negotiations.

• By focusing on similarity and discrepancy among win conditions, the shaper helps focus negotiations towards reaching agreements.
• Allows participants without software negotiation expertise to contribute since experienced wiki users/requirements engineers can take on shaper role.
• By identifying missing information, and inactive pages, the Shaper helps ensure stakeholders' inputs are recognized.
• Shapers coordinate collaboration tasks in asynchronous situations, such as monitoring changes and informing relevant stakeholders to participate.
• Reducing overhead by having the shaper take care of facilitation tasks instead of having every negotiator repeating the effort.

2.4. Software Engineering Team Project

The Software Engineering course (CSCI577ab) [35] is the keystone two-semester team project graduate software engineering course sequence at USC. Graduate students form teams of 6 on-campus students and 2 off-campus students to develop real-client software system projects. In each year, generally, we have about 150 students enrolled in the class, 20 clients and 20 project teams.

Off-campus students usually work full-time and have a lot of experience in the software engineering field, while the majority of on-campus students come directly from undergraduate programs and have very little work experience.

Most of the projects are web applications or database applications with the scope of implementable and deployable within 24 weeks.

These projects follow the Model-Based Architecting and Software Engineering (MBASE) approach [37] and produce artifacts according to the Lean-MBASE development guideline [38]. The model uses a set of common anchor point milestones at which a project
verifies that it has feasible objectives. The front-end anchor points are Life Cycle Objectives (LCO) milestone and Life Cycle Architecture (LCA) milestone.

3. Experience

In this section, the experiences the teams had in using WikiWinWin are discussed. Section 3.1 describes the subjects. Section 3.2 explains activities. Section 3.3 describes collaboration scenarios. Section 3.4 illustrates tool usages and roles with examples. Data collection is discussed in section 3.5.

3.1. Subjects

Students in the fall 2007 software engineering course (CSCI577a) formed 20 project teams. All 20 project teams used the WikiWinWin tool for their WinWin negotiation. The negotiation participants were students, client representatives, and other key stakeholder representatives, e.g. users.

The WikiWinWin tool was configured to support multi-project concurrent negotiation activities. To avoid interference between project teams, each project team was given its own negotiation space in the WikiWinWin tool. We used the group access control to grant access privilege to participants. Participants can view and edit topics in their own negotiation space.

We defined two types of roles in WikiWinWin – Personal Knowledge Contributor (PKC) and Shaper. All participants played the PKC role. In addition, two students on each team performed the shaper role: usually one full-time, on-campus student and one part-time, off-campus working-professional student. The off-campus shaper was generally the lead shaper.

All students received some training on the general WinWin approach and the WikiWinWin tool functions. In addition, the shapers had additional one hour training on the shaper tasks. Clients did not receive the same level of training due to schedule and availability factors.

3.2. Negotiation Activities

The project teams conducted two rounds of WinWin negotiations. The initial negotiation took place during the LCO phase, an updated negotiation happened during the LCA phase.

Before the project teams formally started the initial WinWin negotiation, stakeholders participated in mutual learning activities including a classroom workshop engaging collaborative learning and early team meetings which focus on understanding project backgrounds and benefits. The purpose of these activities is to help stakeholders with different backgrounds quickly build shared understanding among each others and set a collaborative context for the up-coming negotiations.

The Initial WinWin negotiation consists of the following activities in a two weeks period.

- Brainstorm/ identify stakeholders’ interests;
- Converge on win conditions from brainstorming results;
- Prioritize these win conditions;
- Identify issues
- Provide options;
- Reach agreements

The WinWin agreements then served as the basis for generating the initial System and Software Requirements Document (SSRD).

During the updated WinWin negotiation, the project team used the previous WinWin agreements as a baseline, evaluated changes and identified options for remaining issue closure.

3.3. Collaboration Situations

The above WinWin negotiation activities were carried out in two different collaboration situations: negotiation meeting and off-line negotiation.

Each project team had two negotiation meetings. The first meeting focused on inventing win conditions. The second meeting focused on inventing options. By having participants brainstorm and stimulate each other’s ideas in a group setting, the teams were likely to produce more and better ideas than brainstorming individually [36]. The course provided facilities, computers, telephones and network support for these negotiation meetings. Co-located participants came to campus to participate in the meeting. Remote participants joined the meeting through web conference and telephone.

During the off-line negotiation, participants reviewed and refined ideas. Activities that express individual’s preference, such as prioritizing on win conditions, also happened in the off-line negotiation. Participants were not required to meet together. Typically participants used the WikiWinWin tool at their convenience with some degree of coordination among them. The WikiWinWin tool has some features for supporting coordination, such as displaying recently changed topics, sending email notification about daily changes. Some teams used them, but some did not. Most teams actually used a combination of telephone, email, messaging, and face to face communication.

3.4. Usage Scenarios and Roles
In this section, we discuss four negotiation activities to illustrate how the WikiWinWin tool was used and what were performed by the PKC role and the shaper role.

### 3.4.1. Brainstorm

The purpose of the brainstorming meeting is to produce as many win condition ideas as possible. To keep the brainstorming focused, a shaper acted as facilitator. The tool provided a list of pre-defined negotiation topic taxonomy that corresponds to categories in the requirements documentation. With the shaper guidance, PKCs brainstormed their win conditions one topic at a time. Originally we planned to have all PKCs type their own ideas in the tool, but we ran into the wiki page editing conflict problem when we conducted the training. So instead of having all PKCs using the tool to submit ideas, co-shaper recorded every idea into the tool as PKCs spoke out their ideas. Figure 2 shows an example of initial ideas surfaced at the brainstorming meeting. As circled in the figure, several of these ideas point to a prospect agreement: using a COTs candidate – Joomla.

![Figure 2: Initial brainstormed ideas](image)

3.4.2. Converge/ Identify Issue

The resulting ideas from brainstorming were captured as comment under each topic category in the tool. Shaper then crafted the initial win conditions based on those ideas. The tool provided a form-based interface for submitting a win condition. When going through the ideas, shapers combined ideas indicating similar meanings. If an idea covered several aspects, shapers split it into separate win conditions. Shapers also re-categorized ideas and rewrote descriptions to make the win condition more understandable. Shapers also collected terms defined in the project context and added them to the glossary. Figure 3 shows shaper proposed a content management win condition by combining the similar ideas.

![Figure 3: Win condition organized by shaper](image)

3.4.3. Invent Options

Most of the teams had a second negotiation meeting to discuss issues and options within a few days after the first meeting. Some teams did not finish converging win conditions, so they discussed the remaining win conditions at the meeting. The meeting was carried out in the same fashion as the brainstorming meeting, except a shaper used the issues logged in the tool as the discussion thread. PKCs discussed possible options for each issue and the co-shaper recorded them in the tool. During the discussion, participants often agreed on which options to adopt. Figure 5 shows a PKC proposed an option after further investigating the proposed COTs candidate.

![Figure 5: Option proposed by PKC](image)
3.4.4 Reach Agreement

Because project teams’ WinWin negotiation took place concurrently with prototyping, exploring COTs, and analyzing alternatives [39], we did not expect the teams to address all the issues and options by the end of the second meeting. After the meeting, PKCs continued identifying options to close remaining issues. In this step, shapers periodically checked the discussion progress in the tool and tagged the adopted option as agreement. Figure 6 shows the tree view of the agreements.

3.5. Data Collection

The contributions that each team submitted into the tool during the initial round of negotiation were collected. All new entries, comments, updates and proposed replacements were counted as contributions. The team grade of LCO package reflects the quality of project artifacts produced based on the WinWin negotiation results. The loss of grade indicates shortfalls in thoroughness, clarity, and degree of issue closure.

4. Lessons Learned

4.1. Non-uniformity of Usages and Outcomes

Although we provided an equal amount of training and tool support to all of the project teams, some teams’ PKCs and shapers used the tool more frequently than others, as shown in Table 1. Overall, shapers were more active in contributing to topics in the WikiWinWin tool than were PKCs.

<table>
<thead>
<tr>
<th>Table 1: Contribution by team and by role</th>
</tr>
</thead>
<tbody>
<tr>
<td>Team total</td>
</tr>
<tr>
<td>Shaper</td>
</tr>
<tr>
<td>Mean</td>
</tr>
<tr>
<td>Std. Deviation</td>
</tr>
</tbody>
</table>

Two hypotheses were developed to find out the degree to which using WikiWinWin tool and the shaper role affected projects’ outcomes, as shown in Table 2. The regression was performed to estimate the correlation between the contributions and the log () value of loss of grade. The log () was used because the data reflected an exponential distribution. The significance value of 0.05 was used to reject the hypotheses.

The results in figure 7 showed teams who were more active in using the WikiWinWin tool tended to achieve better results in their LCO package.

A similar trend was observed with the shaper’s contribution. It showed teams with active shaper contribution in the WikiWinWin tool tended to have better results in their LCO package, as indicated in Figure 8.

<table>
<thead>
<tr>
<th>Table 2: Correlation of contribution vs. outcome</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hypotheses</td>
</tr>
<tr>
<td>H1: Degree of WikiWinWin tool usage by team has no correlation with LCO package loss.</td>
</tr>
<tr>
<td>H2: Degree of WikiWinWin tool usage by shaper has no correlation with LCO package loss.</td>
</tr>
</tbody>
</table>

![Figure 7: Contribution by team vs. outcome](image)

![Figure 8: Contribution by shaper vs. outcome](image)
4.2. Comparison to previous year’s EasyWinWin

A significant change of the software engineering course from fall 2006 to fall 2007 was switching from using the EasyWinWin tool to the WikiWinWin tool. Table 3 compares team performance between fall 2006 and fall 2007 in terms of loss of grade of LCO package. We included all the projects from fall 2006 and fall 2007 in our results, and there is no source of bias from choosing projects. The grading criteria were the same for both years.

The t-test was performed to estimate the probability that the two groups’ means will be equal. Based on the t-test result, we were able to reject the following hypothesis:

H3: Using WikiWinWin tool has no correlation with LCO package loss.

Table 3: Fall 2007 vs. fall 2006 projects’ outcome

<table>
<thead>
<tr>
<th>LCO package grade loss</th>
<th>Fall 2007</th>
<th>Fall 2006</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>13.78</td>
<td>18.83</td>
</tr>
<tr>
<td>Std. Deviation</td>
<td>7</td>
<td>6.03</td>
</tr>
<tr>
<td>T-Test</td>
<td>0.02</td>
<td></td>
</tr>
</tbody>
</table>

4.3. Other Lessons

We also learned the following three types of lessons from this experiment: training, tool, and process.

4.3.1 Training

We provided training to student participants before the negotiation. The feedbacks indicated we did well in delivering the WinWin concept, but the hands-on tutorial sessions needed to be more specific on explaining how to carry out negotiation tasks using the tool. Shapers preferred to receive more guidance on facilitating negotiations. Our future training for PKCs and shapers needs to ensure they have sufficient skills to do their tasks.

Because the client training was a bare minimum, most clients couldn’t use the tool on their own. Shapers tried to help in some cases, but they weren’t successful. It told us next time we need to include more context-setting and training for the clients. This has been incorporated in the client support activity.

4.3.2 Tool

Participants felt the current WikiWinWin tool had some strength and weakness at supporting negotiation tasks. (+) indicate strong point; (-) indicate weak point.

- (+) Platform support - the tool was easy to access at any time and from anywhere with internet connection.
- (+) Tool robustness - there was no tool crash during negotiations.
- (+) Maintain negotiation results - the tool captured different types of artifacts and it was easy to see the relationships of the negotiation results (Win Condition, Issue, Option, Agreement).
- (+) Terms and definitions - the glossary captured in WikiWinWin was useful.
- (+) Automated notification aid - users who tried this feature felt it was useful.
- (+/-) Negotiation taxonomy - the pre-defined win condition taxonomy helped to keep the brainstorming focused, but too many sub-categories were confusing.
- (-) Email configuration - in a number of cases when users requested reset passwords, the tool failed to email passwords.
- (-) User interface - the interface was not easy for clients to use. The amount of information per page need to be reduced.
- (-) Overview – lack of overview of outstanding issues and win conditions contribution.
- (-) Task simplicity - Shaper specific tasks such as renaming and re-categorizing were not straightforward. It was time consuming for shapers to manually prepare voting page for all the stakeholders.

The needed tool improvements will focus on making it simple to use and resolving usability issues.

4.3.3 Process

We will also improve on the current WikiWinWin negotiation process based on the experimental experience. Identified improvements include:

- Use critical success factors to identify persons taking on shaper role.
- Incorporate collaboration techniques, such as ThinkLets [34], to the process guideline.
- Provide a baseline of shaper and co-shaper responsibilities.
- Mandate enough gaps between the brainstorming meeting and the identifying option meeting.
- Use concurrent prototyping to expose conflict/constrains.
- Establish deadline and stakeholders’ accountability for the off-line tasks.
4.3.4 Possible Extension

Future iterations will continue advancing the wiki-based platform and the shaper role in supporting stakeholders’ collaborative negotiations.

The next version of the tool will implement the above lessons learned. We will evaluate the improved tool in the fall 2008 software engineering course. Our future research plan also includes: 1) exploring whether we can go from WikiWinWin negotiation results directly to test cases, avoiding the need for requirements documentation; 2) exploring the combination of wiki with other social networking tools.

5. Conclusions

Overall, the wiki technology enabled stakeholders to collaborate more rapidly, create and deal with more contributions, and achieve higher success rates than were achieved by the previous year’s EasyWinWin teams.

The initial results from the experiment were promising. Although the teams did not achieve uniform success in using the tool, it did show that teams who used the tool more extensively with active shaper involvement tended to achieve better project outcomes.

From this experiment, we also learned the strengths and shortfalls in the current WikiWinWin tool, process and training preparation. A number of needed improvements were identified and have been incorporated. We plan to evaluate the improved system in the fall 2008 software engineering team projects and report again our results.

6. References

[16] R.D. Hof, Something wiki this way comes: They’re Web sites anyone can edit – And they could transform corporate America. Business Week, June 7, 2004


[38] B. Boehm, et al, Guidelines for Lean Model-Based Architecting and Software Engineering (Lean MBASE), http://greenbay.usc.edu/csci577/fall2007/site/guidelines