

WikiWinWin: A Wiki Based System for Collaborative Requirements Negotiation

Da Yang^{1,3}, Di Wu², Supannika Koolmanojwong², A. Winsor Brown², Barry W. Boehm²

¹ *Laboratory for Internet Software Technologies, Institute of Software, Chinese Academy of Sciences, Beijing 100080, China*

² *University of Southern California, 941 w. 37th Place Los Angeles, CA 90089-0781*

³ *Graduate University of Chinese Academy of Sciences, Beijing 100039, China*
 yangda@itechs.iscas.ac.cn, {diwu, koolmano, awbrown, boehm}@usc.edu

Abstract

Defining requirements is one of the most critical activities in the development of software intensive systems.

The EasyWinWin system has been very good in capturing initial requirements involving heterogeneous stakeholders in over 150 client-developer requirements negotiations. However, it has been less easy to use in updating requirements and related information as a project proceeds and adapting to the evolving nature of the requirements.

Because our clients are finding that wikis are easier to learn and use, and can organize information in a flexible and updatable manner, we have developed an initial version of a WikiWinWin system as a potential successor to EasyWinWin. We have conducted a case study of WikiWinWin, and the result shows that the initial WikiWinWin is basically good at facilitating stakeholder collaborative negotiation and learning, but has some limitations that we are now addressing.

1. Introduction

According to the 1995 Standish Group research report, 31% of its 8000 reported projects were canceled before completion, 52.7% were challenged, and only 16.2 % succeeded [1]. In the recent 2004 Standish Group research report, the percentage of failed projects drops to 18%, but the percentage of challenged projects is still as high as 53% [2]. The top 3 factors for the challenged projects, lack of user input, incomplete requirements & specifications, and changing requirements & specifications, are all related to requirements [1]. Defining requirements is one of the most critical activities in the development of software intensive systems, and also one of the most challenging.

Defining requirements is not a simple knowledge transfer process where requirement engineers elicit and document existing client knowledge [3]. Rather, it is a cognitive process, in which stakeholders collaboratively find out what has to be done [4] by understanding problems and domains, learning from other stakeholders, and by negotiating and discussing different viewpoints.

The major challenges of defining requirements for software intensive systems are:

- Different stakeholders – users, customers, managers, developers, and maintainers – come to a project with diverse perspectives, backgrounds, abilities and expectations. It is challenge to define feasible and mutually satisfactory requirements that accommodate all stakeholder goals and expectations [5][6][7].
- IKIWISI: some requirements are not specifiable in advance but emerge with system prototyping and usage. When users are asked to specify such requirements, they generally claim that “I don’t know how to tell you, but I’ll Know It When I See It” [8]. Stakeholders need to collaboratively and incrementally find out what has to be built [4][9].
- Rapid changes in COTS releases, competitive threats, stakeholders, reorganizations, and price structures make requirements volatile [8] [9].

The WinWin project management theory proposes that the project can be fully successful if and only if the stakeholders can collaborate and make everyone a winner [10]. Based on the WinWin approach, the EasyWinWin is a groupware system designed to involve heterogeneous stakeholders and help them to reach a mutually satisfactory win-win set of requirements [6]. At USC (University of Southern California), we had used EasyWinWin in our Software Engineering classes from 1999 to 2006. According to its usage in over 150 projects, EasyWinWin had been very good in capturing initial requirements. However, it has been less easy to adapt to the evolving nature of requirements, and to provide the flexibility for stakeholders, especially the client, to easily share knowledge and update the win conditions and agreements as a project proceeds.

In order to support knowledge sharing and information management, wiki technology plays an important role in various fields. As the well-known Wikipedia [11] shows, wikis provide a flexible platform for collaboration to create online content [12]. These days, more and more organizations are using wiki to support enterprise collaboration activities [13]. Organizations are also using wiki technology and the wiki way to facilitate higher levels of constructive customer engagement [14]. In the

requirement related area, the Fraunhofer Institute is developing a wiki-based system named SOP-wiki to support stakeholder participation in requirements engineering [15]. These developments indicate that the wiki technology may also be able to support the WinWin approach and facilitate stakeholder collaboration during requirements negotiation.

In this paper, we investigate how to adopt the wiki technology to support active stakeholder participation and collaborative requirements negotiation. We developed the WikiWinWin requirements negotiation support system by using the wiki technology, the WinWin approach, and lessons learned from using the EasyWinWin tool.

A case study will show how effective the initial WikiWinWin was in supporting WinWin requirements negotiation. We will also discuss the strengths and weaknesses of the initial WikiWinWin tool and identify future enhancements.

The remainder of the paper is organized as follows: Section 2 introduces the background theory and technologies of WikiWinWin. Section 3 explains the WikiWinWin requirements negotiation process and discusses the challenges of applying wiki technology to requirements negotiation. Section 4 describes a case study of using the WikiWinWin for requirements negotiation. Section 5 discusses the evaluations of WikiWinWin. Section 6 discusses our conclusions and future plans.

2. Background

2.1. WinWin approach and EasyWinWin tool

A software project involves multiple interdependent stakeholders, such as developer, customer, user, maintainer, manager, domain expert, architect, etc. In requirements negotiation, value propositions of all success-critical stakeholders need to be considered and reconciled. Experience shows that Win-Lose for some parties generally becomes Lose-Lose for everyone [10]. The WinWin approach is a set of principles, practices and tool, which enables a set of interdependent stakeholders to work out a mutually satisfactory (win-win) set of shared commitments [5][16].

The WinWin negotiation model defines 4 types of artifacts – Win Condition, Issue, Option, and Agreement (WIOA) [17]. Figure 1 illustrates their relationships.

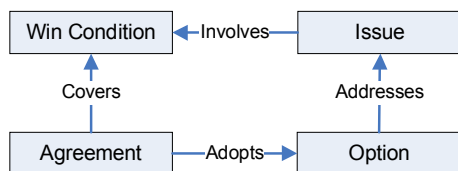


Figure 1: WinWin WIOA Negotiation Model

- Win Condition – captures individual stakeholders’ desired objectives.

- Issue – captures conflicts between win conditions and their associated risks and uncertainties.
- Option – Identifies candidate solutions to resolve an issue.
- Agreement – captures shared commitment of stakeholders with regard to accepted win conditions or adopted options.

In the WinWin negotiation process, stakeholders express their goals as win conditions, and explore others’ win conditions. Win conditions become agreements if everyone agrees. If not, issues are identified and lead to inventing options and negotiated changes in win conditions. Stakeholders’ win conditions are converged at a WinWin equilibrium condition when all the win conditions are covered by agreements and there are no outstanding issues [17].

EasyWinWin Tool

The first generation of WinWin requirements negotiation support system was developed at USC in 1994, and the EasyWinWin is the fourth generation implementation for the WinWin requirements negotiation approach based on a Group Support System [6][18]. It uses the groupware supported facilitation techniques and defines a set of activities guiding stakeholders through a process of gathering, elaborating, prioritizing, and negotiating requirements. Compared with the earlier three generations of WinWin requirements negotiation support system, EasyWinWin is more robust, easier to use, and has more friendly user interface.

With the USC real-client software engineering project course, EasyWinWin has been used in more than 150 real-world projects of various domains, e.g., digital libraries, e-marketplaces, and video data processing. It has been very good in capturing initial requirements in an informal way [19]. But it is not easy for clients with moderate computer skills to update content, preserve revision history, share various requirements related information, or support higher level of constructive customer engagement. In addition, Briggs [20] proposed a future work to extend the EasyWinWin and integrate it with a requirements management system so that the requirements can be easily traced back to win conditions, and Kitapci has developed such a system at USC [21].

2.2. Boundary objects theory

Boundary objects refer to a type of team mental model or physical object that allows people to bridge different areas of expertise in order to collaborate more effectively and efficiently [22]. Star [23] identified four categories of boundary objects: (a) repositories, such as cost databases, parts libraries, that supply common reference points of data, measures, and labels to provide shared definition and values for solving problems; (b) standardized form and methods, such as problem-solving methodologies and standards for reporting findings, that define and categorize

differences and potential consequences to make them more shareable; (c) visual representations including sketches, prototypes, and simulations that help to demonstrate differences and dependencies; and (d) boundary maps, such as Gantt charts or process maps that also help to clarify dependencies.

Over the last ten years, USC-CSSE (USC Center for Systems and Software Engineering) has developed several boundary objects to facilitate interdisciplinary collaborative design and development. For example, the Model Clash Spiderweb [24], cost and schedule estimation models [25], risk lists [26], project Benefit Chain [27], etc.

The requirements negotiation will involve different stakeholders with diverse backgrounds, expertise, and perspectives. We suggest that teams can use boundary objects such as cost and schedule estimation models, simulations, and prototypes during requirements negotiation to bridge the expertise difference and facilitate the collaborative learning of stakeholders. Accordingly, a set of boundary objects is built into the WikiWinWin system.

2.3. Wikis and their application

A wiki is actually two things: a program that makes it exceptionally easy for people to edit web pages and a philosophy regarding how users should go about the editing of web pages [12].

The Wall Street Journal and Business Week have recently identified wiki technology as an up-and-coming technology to engage customers [28][29]. A recent survey of 168 corporate wiki users indicates that corporate wikis appear to be sustainable. The corporate wiki users also stated three main types of benefits from wikis: enhancing reputation, making work easier, and helping the organization to improve its processes [30]. The wiki appears to be a viable new collaborative technology to facilitate higher levels of constructive customer engagement [14].

A wiki has several key characteristics, which make it different from the other widely used collaborative technologies, e.g., discussion forums, weblogs, and GSS (group support systems) [31].

- 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.

Many wiki engines are good in supporting collaboration activities e.g. the popular MediaWiki [32]

and TWiki [13]. The Wikipedia, built upon the MediaWiki wiki engine, is one of the most well known cases. It's an online encyclopedia, collaboratively written by a group of about 50,000 volunteers.

Among many success stories about the use of TWiki [13], here we will discuss several well known ones such as: Yahoo uses TWiki internally to manage documentation and project planning; Disney uses TWiki as the central resource for ideas, notes, how to's, specs, and brainstorming; and SAP uses TWiki for multi-team collaboration.

The Fraunhofer Institute uses wiki technology to facilitate stakeholder participation [15]. The developed collaboration support system provides a useful document structure to guide the stakeholder in editing, grouping, and structuring requirements. The system is good at general requirement document editing and management, but it does not support a requirements negotiation process like the WinWin requirements negotiation.

Table 1: Challenges of requirements negotiation and Strengths of wiki

Challenges	Wiki's strengths
Need active involvement of stakeholders, especially the clients and users [4][5].	Wiki can facilitate high level of constructive customer engagement [14].
Stakeholders need to collaboratively and incrementally find out what has to be built [4][9].	It is easy to exchange ideas and share knowledge in wiki [12].
Need to adapt to the changes of requirements [8][9].	It is easy to update content and preserve the revision history in wiki.
Need to integrate requirements negotiation with other project activities, e.g. requirements management [20].	It is easy to integrate new information or new collaboration-activity support into the wiki website [15][13].
Using various boundary objects to facilitate interdisciplinary collaborative design and development [22]	It is easy to integrate new boundary object information into the wiki.
Overlapping, obscure, and ambiguous inputs cause confusion [21].	If some users have reorganization (Shaper) skills, wiki flexibility can accommodate them.

There are several reasons that have led us to explore the wiki technology and develop a collaborative requirements negotiation system based on wiki. Table 1 shows the summary of the challenges of requirements negotiation and the strengths of wiki that support the corresponding challenges.

3. The WikiWinWin

The WikiWinWin approach includes a stakeholder collaboration process and corresponding support tools. To

create predictable and repeatable patterns of group interaction, it is important to define the sequence of events, instruction, tools used, and the configuration of tools. The WikiWinWin creates a sequence of steps and instructions 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. This section will explain the WikiWinWin requirements negotiation support system, identify the challenges of applying a general wiki for requirements negotiation, and discuss our solutions to these problems. Section 4 will discuss a case study on using WikiWinWin for requirements negotiation.

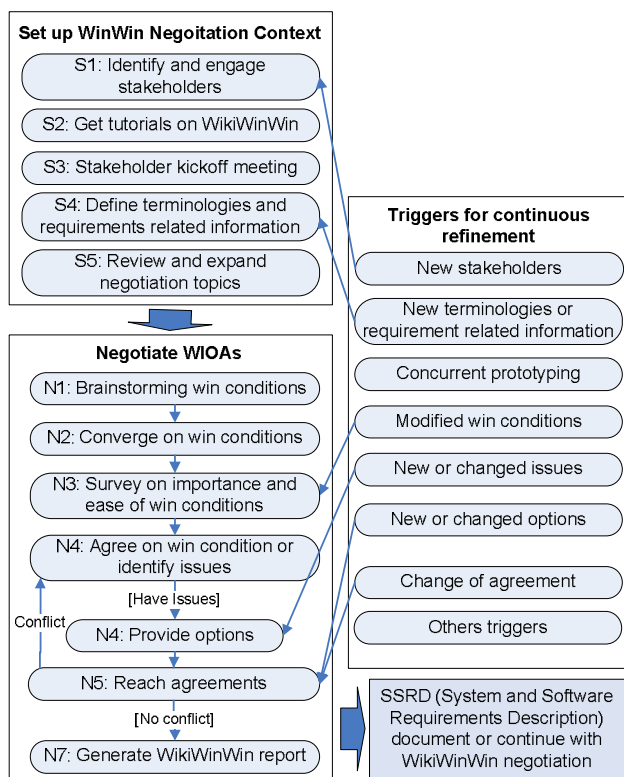


Figure 2: Overview of WikiWinWin process

According to the research works on wiki [33], there are two main kinds of users; a) persons who act as the “Shapers” and b) other participants who act as “Personal Knowledge Contributors (PKCs)”. The roles of Shaper and PKC also represent different level of participation in wiki-based collaboration. PKCs primarily contribute their personal knowledge to wiki while shapers, who may also be PKCs, contribute by integrating, distilling, organizing & rewriting contributions of others. In WikiWinWin, all stakeholders will play as PKCs and optionally as Shapers in expressing their ideas, win conditions, and shared knowledge. A facilitator, who could be an appropriately skilled stakeholder, will serve to initialize the WinWin

process and participate as a shaper, who will encourage PKCs to express their ideas, moderate the negotiation process, and help to shape stakeholders’ inputs.

3.1. WikiWinWin requirements negotiation process

As Figure 2 shows, the WikiWinWin negotiation process begins with setting up the negotiation context, proceeding to negotiate the WIOAs (win conditions, issues, options, and agreements), and continuously refining the negotiation as a project proceeds. The negotiation results will be used to generate the SSRD (System and Software Requirements Description) [34]. The current SSRD is created in the form of a MS-Word file based on a template. Therefore, it is not easy to trace the requirements specification back to the win conditions, which are stored in the requirements negotiation support system. As a future work, we plan to integrate the SSRD requirements management activity into the WikiWinWin. So that the requirements specification can be easily traced back to related win conditions and agreements.

3.1.1. Set up the WinWin negotiation context

According to our experiences of EasyWinWin requirements negotiation in over 150 projects, we find that it is very important to set up a good context for effective WinWin requirements negotiation. Effective negotiation requires developers to learn more about the client’s and user’s world, while clients and users learn more about what is technically feasible. When the developers are unfamiliar with the project background knowledge, they should complete more learning work before the negotiation, e.g. student developers can fail to raise key issues and questions during the first WinWin requirements negotiation because they have little knowledge about the project background. Furthermore, the information from prototyping and COTS assessment reports that emerge as projects proceed should also be easily incorporated into the context to facilitate the negotiation.

Figure 2 shows the 5 elements of setting up the WikiWinWin negotiation context.

S1: Identify and engage stakeholders

This step includes identifying stakeholders, recording their contact information, and mapping negotiation roles to stakeholders. Developers should begin by visiting the clients and getting a top-level understanding of their operational context. Each stakeholder can have a wiki role and a project role, which together will determine the stakeholder’s responsibilities during WikiWinWin negotiation. The wiki roles include Shaper and Personal Knowledge Contributor (PKC). The project roles include client, user, maintainer, developer, etc.

In this step, the WikiWinWin posts the stakeholders' contact information and roles to a shared webpage. The Shapers or other stakeholders can easily edit this page.

S2: Get tutorials on WikiWinWin

The stakeholders will get tutorials on the WikiWinWin negotiation process and tools before negotiation.

S3: Stakeholders kick-off meeting

The stakeholder kick-off meeting initiates a collaborative learning process by adopting the following practices:

- i) Appoint a “Learning” facilitator
- ii) Start meetings with learning objectives

Before the meeting, the project team should prepare a list of role-based learning objectives. The list should cover and extend the example in Table 2.

iii) Stakeholders should get familiar with WinWin WIOA structure (win conditions, issues, options, and agreements).

iv) The meeting will be ended with assessment of learning by going back to the stakeholders’ learning objectives, checking whether these objectives are all satisfied, summarizing the knowledge learned concerning each learning objective, and identifying future needs.

Table 2: Kickoff Meeting Objectives

Role	Learning objectives
Developers	1. Understand the current system 2. Understand the type of system to be built 3. The target user(s) for the system 4. Critical domain knowledge for the system
Client	1. Technology feasibility 2. Project risks

In this step, WikiWinWin presents the kickoff meeting steps and instructions, and posts templates for recording the learning objectives, assessment of learning, and related information.

S4: Define terminologies and requirements related information

This step captures the terminologies that are important for the communication among stakeholders. WikiWinWin presents the stakeholders with a shared list of terminologies. Any stakeholder can contribute to the terminology list, and the revising history will be preserved. It is the typical way that wikis manage information, e.g. the case of Wikipedia [32].

WikiWinWin already includes some general terminologies, and stakeholders can work out the first set of project specific terminologies after the “Stakeholder Kickoff Meeting”. Shapers or other personal knowledge contributors can record these terminologies, expand and refine them as the project proceeds.

As it is easy to introduce, update, or organize information in the wiki, the stakeholders are encouraged to contribute requirements related information like project background information, COTS assessments, prototypes, etc. to facilitate the requirements negotiation. These types of information are usually boundary objects, which can contribute to the collaborative learning of stakeholders.

S5: Review and expand negotiation topics

The negotiation topics are organized according to the ICM (Incremental Commitment Model) [9][35] win condition taxonomy. Five main win condition categories are project win condition, capability win condition, System Interface win condition, Level of Service win condition, and Evolutionary win condition. In this step, the system presents a shared outline of the taxonomy. Stakeholders will review this outline and expand the taxonomy according to project needs. The resulting taxonomy is another useful boundary object that can facilitate stakeholders’ collaboration and learning. It is used to guide the stakeholder negotiation, make the negotiation focused, organize the WIOAs, and check the completeness of negotiation.

3.1.2. Negotiate the WIOAs

The WikiWinWin defines a set of activities guiding stakeholders to identify goals and preferences, identify and resolve conflicts, prioritize requirements, and achieve mutually satisfactory agreements.

N1: Brainstorming win conditions

Stakeholders share their goals, perspectives, view, background, and expectations by gathering statements about their win conditions. In this step, the WikiWinWin provides an electronic discussion sheets for each win condition category. Stakeholders will brainstorm win condition ideas for each category, taking the current category description as a prompt. By dividing a compound project-brainstorming prompt into separate sub-prompts of categories, the team may produce more ideas of higher quality and creativity [36].

Brainstorm Win Conditions (ComputerSoftwareWinConditions)

The server-side system should be able to run on Windows, Unix, Linux, and Macintosh platforms.

Developers use software installed at USC computing lab.

Developers are familiar with Windows.

ITS prefer MySQL as a Database Management Software

Developers prefer to use my MySQL too.

Converge on Win Conditions (Win Condition List)

State	Name	Timestamp	Creator	Role	Issues,Options, and
Agreed	Database software	06 Jun 2007 15:30	DiWu	Shaper	enter
Agreed	System support 2 platforms	06 Jun 2007 15:27	DiWu	Shaper	enter
Agreed	Development environment	06 Jun 2007 15:29	DiWu	Shaper	enter

Name:

Description:

Figure 3: Brainstorm and converge on win conditions

N2: Converge on win conditions

Stakeholders jointly create a non-redundant list of clearly stated, unambiguous win conditions by considering

all ideas contributed in the brainstorming session. It is easy to identify the redundancies, when the brainstormed ideas are organized according to the win condition taxonomy. WikiWinWin provides in each win condition category a shared win condition list, which every stakeholder can edit. The Shapers will capture initial win condition lists based on the brainstormed ideas. Then other stakeholders can review and make suggestions on how to revise the lists. Figure 3 shows the discussion sheet and the corresponding converged win condition list in the category Computer-Software Win Conditions.

N3: Survey on win conditions

In this step, stakeholders will use a multi-criteria polling tool to rate each win condition along two criteria: business importance and ease of realization. The results are not used to eliminate any win conditions. Rather they are used to identify the discrepancies of opinions. Then stakeholders can discuss and explore the reasons for these discrepancies. Key information cues about the project may emerge from these discussions, e.g. project constraints, assumptions, unshared information, and hidden agendas.

N4: Agree on win condition or identify issues

Stakeholders either agree on a win condition or raise related issues, which may be conflicts among win condition and related risks and uncertainties.

N5: Provide options

Stakeholders provide candidate options to resolve issues.

N6: Reach agreements

Stakeholders reach agreements on which options to adopt or reach agreements on win conditions.

In step N4, N5 and N6, the WikiWinWin posts an outline of the Win conditions, Issues, Options, and Agreements in a tree structure, named as WIOA tree. The team will make three passes through the WIOA tree. On the first pass, each person reviews each win condition. If an issue is raised for a win condition, the issue name will become a sub-heading to the win condition in the WIOA tree. On the second pass, each participant reviews each issue and think of options for resolving issues. The name of the option will appear as sub-headings to the target issue.

Then the group addresses each issue in turn with oral negotiation. They will reach agreements on what options can be adopted to resolve the issues. When every issue is resolved and every win condition has an agreement, the state of WinWin Equilibrium has been achieved.

3.1.3. Continuous refinement

To adapt to the evolving nature of requirements, stakeholders can refine the negotiation as the project proceeds. The triggers for continuous refinement can be new stakeholders, new terminologies, concurrent prototyping, modified win conditions, new or changed issues, new or changed options, etc.

Any change in the WikiWinWin system will be automatically summarized in a list of recent changes. Shapers will check the recent changes to ensure the WinWin equilibrium that all the win conditions are covered by agreement and there is no outstanding issue.

Table 3: Collaboration situations of stakeholders

	Co-located	Dislocated
Synchronous communication	Same time & Same place	Same time & different place
Asynchronous communication	Different time & Same place	Different time & Different place

Table 3 shows four different scenarios of negotiation. For synchronous requirements negotiation, stakeholders can use the WikiWinWin tools and hold face-to-face meetings or telecon. When there is not a lot content for negotiation, Shaper can use WikiWinWin and Email to organize an asynchronous negotiation, which may save the stakeholders a lot time by avoiding the face-to-face meeting or telecon.

In asynchronous collaborations, Shapers can configure the WikiWinWin system and let it automatically notify every participant with the recent changes in WikiWinWin. Then participants will use WikiWinWin to review new win conditions, identify issues, and jointly work out options for resolving issues. At last, Shapers will set a negotiation deadline and send out emails asking participants to confirm with the options and agreements. The negotiations will continue until all issues are resolved and WinWin Equilibrium has been achieved.

3.2. Wiki problems and how we resolve them

There are several challenges of applying wiki technology to WinWin requirements negotiation. We have provided our solutions to these problems, and developed the WikiWinWin tools based on TWiki [13].

Ease of use

Although the wiki has new style of syntax and editing conventions, these syntax and editing conventions are easy to learn. Furthermore, we developed some specific user interfaces to save learning time of WikiWinWin users.

In WikiWinWin, we developed a set of form-based user interfaces for stakeholders to create or change the win conditions, issues, options, and comments.

Maintain the WIOAs

The relationship between WIOA artifacts that are created in WinWin negotiation must be correctly maintained throughout the system development process. Examples are the relationships between issues and their win conditions or options. In WikiWinWin, we organize the WIOAs as topics and define metadata to express the relational semantics among them.

Guidance for negotiation process

It is difficult for stakeholders to collaborate and conduct WinWin negotiation without clear guidance of the negotiation process. The WikiWinWin process defines clear steps to guide the WinWin requirements negotiation, and provides a clear structure to organize requirements and related information. WikiWinWin also includes boundary objects to facilitate the collaborative learning of stakeholders, e.g. the general terminologies in WinWin negotiation and the win condition taxonomy.

Support of synchronous collaboration

Information in the wikis is organized by topics. When more than two people edit the same topic at the same time, there will be a conflict. The strategy in TWiki to resolve such conflict is to lock the page until the first person finishes editing. It is not easy for more than one person to edit the same page to exchange ideas. And wiki is generally only used for asynchronous collaboration [15]. However, initial requirements negotiation often involves synchronous collaboration.

We found several ways to resolve this issue. First, there is hardly any chance of conflict when more than one person uses the “comment” function to append information to a page during synchronous collaboration. And we can organize the information into different topics to reduce the chance of conflict. The information from different topics can also be automatically collected into the same page. To some extent, people can exchange ideas upon the same web page. A shaper will be a person who plays a major role in ensuring the integrity and consistency. With the various forms, WikiWinWin will automatically and correctly organize the various inputs from users into the corresponding WIOA topics and maintain the relationships among them.

Because of the well-organized requirements structure, the form based user interfaces, the clear negotiation steps, and the use of shapers in facilitating, the WikiWinWin can effectively reduce the chance of wiki editing conflict and support both asynchronous and synchronous collaboration.

Versioning across several pages

Since wikis only provide versioning on the basis of a single page, defining releases based on a certain status of several pages (for example, a baseline) can become laborious. WikiWinWin provides features to support cross-page versioning and baselining. It can collect the links to the current version of all the requirements related topics into a certain page and thus represent a release and baseline of the requirements negotiation.

4. Retrospective Case Study

As Yin suggests in [37], case study research is most likely to be appropriate for “how” and “why” questions. To answer the questions like “how stakeholders will use WikiWinWin during requirements negotiation”, “How

stakeholder can use WikiWinWin and better adapt to the evolving nature of requirements”, and “What’s the strengths and weaknesses of WikiWinWin, and why”, so, we conducted a retrospective case study of requirements negotiation using the initial WikiWinWin tool.

It is based on a real-client e-service project, which is one of the student projects in the USC Software Engineering Class. The project is to develop the Early Medieval East Asian Tombs Database system [38] for a client from the East Asian Library at USC. The system will provide capabilities such as a web-based interface, searching the database that holds the tombs information from East Asia, online database management, show location of tombs in a map, etc.

Six graduate students played as 1 client, 3 developers, 1 shaper/developer, and 1 shaper/quality focal point. All students are familiar with the typical scenarios and difficulties during WinWin requirements negotiation. We performed requirements negotiation for this e-service project, and went through the typical negotiation scenarios as well as some of the difficult negotiation scenarios, e.g. collaborative learning, concurrent prototyping to resolve issues, and asynchronous negotiation.

Table 4: WinWin negotiation effort and results

Wiki Role (number of person)	Contributing effort (total man-hours)	Facilitating effort (total man-hours)
PKC (4)	5 hours * 4 people = 20	0
Shaper (2)	5 hours * 2 people = 10	8
Total (6)	30	8
WinWin negotiation results		
Win conditions ideas: 62		Win conditions: 30 (All agreed)
Issues: 5 (All resolved)		Options: 8 (6 were adopted)

Table 5: Effort of facilitating tasks

Facilitating task	Effort (number of hours)
Tool preparation	2
Pre-negotiation preparation	1.5
Reorganizing and shaping information	3
Survey table preparation	1
Report generation	0.5
Total Effort	8

We had 1 session of 1-hour kickoff meeting and 2 sessions of 2-hours in-negotiation meeting. Table 4 shows the total effort distribution in number of man-hours. Each stakeholder spent 5 hours in brainstorming ideas, expressing their win conditions, identifying issues, providing options, and reaching agreements. As shown in table 5, shapers spent an extra 8 hours in facilitating the negotiation. Tool preparation tasks include creating login for stakeholders, setting up group users, etc. Pre-negotiation preparation tasks include defining learning objectives, entering initial win conditions derived from general meeting or kick off meeting, etc. Reorganizing and shaping information tasks include collecting and

correcting win conditions, checking win condition for integrity, defining terminologies, etc. The survey table preparation task refers to preparing the multi-criteria polling tools for stakeholders to rate on the importance and ease of win conditions. The report generation task will prepare all information for the requirements specification document.

In the two negotiation sessions, stakeholders collaborated to work out 62 win condition ideas, 30 converged win conditions, 5 issues, and 6 adopted options. The number of resulting artifacts is more than that in the original EasyWinWin negotiation report. The WikiWinWin can effectively facilitate stakeholders to identify win conditions and resolve issues in this case.

Table 5 shows that the Shapers spent a lot of effort on facilitating tasks. The Shaper/facilitator is one of the most important success factors of the WikiWinWin collaboration process. We identified ways in which the WikiWinWin could improve to package more facilitation skills and automate some of those facilitating tasks.

In this retrospective case study, we have also experimented using WikiWinWin to adapt to the evolving nature of requirements. Here we will discuss two examples of how we use WikiWinWin to perform the major negotiation activities.

Concurrent prototyping

Sometimes it is hard to provide an applicable solution to resolve issues right away during the requirements negotiation. And stakeholders have to collaboratively acquire new knowledge to come to reasonable decisions, e.g. learning through concurrent prototyping.

In this case study, the client identified the win condition “Display tomb location on the maps”. She thought this win condition was important and easy to implement. However, the developers were not familiar with the map technology, and thought this capability was not important and probably difficult to implement. Therefore, a developer raised the issue “Developers are not familiar with the map displaying techniques, neither Google Map or others.” As an option, the stakeholders agreed on “Prototype first, and find a solution”.

After the negotiation meeting, the developers prototyped the map display capability with the Google map API, and found that it is not so hard to implement the function. The information about prototyping was shared on WikiWinWin, and the discrepancy of opinions was resolved.

Adapt to changes

After the negotiation session, the client raised a new win condition that a lot of tomb data needs to be migrated into the new system. The Shaper organized an asynchronous negotiation for this new win condition, and sent out emails asking stakeholders to participate. The WikiWinWin system was configured to automatically notify participants with the recent changes.

The students raised two issues: "We need to know the format of the data" and "It may be difficult to migrate the data within the schedule". The client replied that the data was stored in an Access database and resolved the first issue. Getting to know the data format, one student searched and found an solution for the second issue, that "There is a COTS product can be used to migrate data from Access into MySQL database". When Shaper sent out emails asking participants to confirm with the negotiation results, no one raised new issues or found problem with the options. So, the options were adopted, issues were resolved, and the WinWin Equilibrium was achieved. In this case, the WikiWinWin provided a structured collaboration forum, preserved the negotiation history, and automatically notified the participants with new changes.

5. Evaluation of WikiWinWin

During the retrospective case study, we found that the initial WikiWinWin could effectively facilitate the WinWin requirements negotiation. We also identified some limitations of current WikiWinWin, and planned future elaboration activities.

Table 6 shows a comparison of the EasyWinWin, the WikiWinWin, the requirement engineering support system SOP-wiki [15], and the general-purpose wikis.

Table 6: Evaluation of WikiWinWin

Rating Scale		EasyWinWin	WikiWinWin	SOP-wiki	wikis
*	Unacceptable				
**	Marginal				
***	Acceptable				
****	Strong				
Support of WinWin approach		***	***	*	*
Easy to exchange ideas and share knowledge		**	***	***	***
Easy to update content and preserve revision history		*	***	***	***
Easy to extend		**	***	***	***
Easy to incorporate boundary object		**	***	***	***
Support synchronous collaboration		***	**	*	*
Release & baseline requirements		***	***	***	*
Automation of content management		***	**	**	*
Exporting the content [15]		***	**	***	*
Multiple webpage (re)classification [15]		---	**	***	*

The main strengths of WikiWinWin are:

- It can effectively support the WinWin requirements negotiation approach.
- Easy to exchange ideas and share knowledge.
- Easy to update content and preserve the revision history.
- Can be extended to incorporate and interact with other project management activities, e.g., SSRD management [34] and project planning.

- Incorporates various boundary objects to facilitate the collaborative negotiation and learning of stakeholders, e.g., the WikiWinWin negotiation process, win condition taxonomy, general terminologies, cost model results, and prototype results.

As discussed in Table 1, these strengths of WikiWinWin are helpful for resolving several challenges of requirements negotiation.

Another contribution of WikiWinWin is that it identified the problems of applying wikis for requirements negotiation and provided solutions for these problems.

- Develop specific user interface and improve the ease of use
- Use metadata to express the relational semantics among the artifacts during negotiation, and design structured topics to organize these artifacts
- Provides solutions to support synchronous collaboration
- Provides solutions to release and baseline the requirements negotiation

The main weaknesses of the initial WikiWinWin are:

- The automation level is not as high as the EasyWinWin. For example, a resolved issue should have at least one adopted option, and such consistency checking is not automated in the current WikiWinWin. The Shaper spends a lot of effort on facilitating tasks.
- During synchronous negotiations, there is still some chance of topic editing conflict.

Briggs [18] proposed several guidelines for creating effective group support systems, and some of these guidelines are also useful for evaluating a requirements negotiation support system like WikiWinWin. The GSS success guidelines and related WikiWinWin features are shown in Table 7.

Table 7: GSS success guidelines and WikiWinWin

GSS success guidelines	WikiWinWin features
Focus on creating solutions for mission-critical collaborative tasks.	Pro: Focus on gathering, elaborating, prioritizing, and negotiating initial requirements.
Design effective collaborative processes.	Pro: Based on the WinWin requirements negotiation processes. Con: The Shapers have to do a lot of facilitating tasks.
Package facilitation skills to reduce usage conceptual load and get predictable results.	Pro: Provides clear negotiation process, easy-to-use user interfaces, and boundary object like win condition taxonomy. Con: Need more automatic tool support. Need to make the asynchronous negotiation more predictable.

6. Conclusions and Future Work

Defining stakeholder mutually satisfactory requirements is one of the top critical success factors for developing software intensive systems. It is also one of the top factors that make a software project challenged. Successfully defining requirements requires stakeholder collaborative negotiation as well as collaborative learning.

Wikis are becoming more and more effective in supporting collaborative activities, e.g. the case of wikipedia, wiki based collaboration platform in yahoo, SOP-wiki, and related publications. It suggests that wiki can also be used to facilitate the collaborations during requirements negotiation.

We developed the initial WikiWinWin requirements negotiation system based on the wiki technology and the WinWin approach. The retrospective case study shows that initial WikiWinWin can effectively facilitate stakeholders to share information, identify win conditions, resolve issues, adapt to change, and reach mutually satisfactory requirements. The WikiWinWin has the strengths such as easy to exchange ideas and share information, easy to update content and preserve history, and joint ownership of content. These features are also good for encouraging higher level of constructive client engagement.

WikiWinWin also tries to incorporate various boundary objects to facilitate collaborations based on the boundary objects theory. For example, the win condition taxonomy and general terminologies can well facilitate the communication and learning among stakeholders.

We got some encouraging results during the development and the experiment of the initial WikiWinWin. However, we also identified some limitations of current WikiWinWin. We have planned several future works to make the WikiWinWin more mature, and better cope with the challenges of requirements engineering:

- Use WikiWinWin in over 20 real-world e-services projects during the 2007 Fall semester and 2008 Spring semester.
- Extend WikiWinWin to incorporate and interact with other project management activities like requirement management and project planning.
- Package more facilitation skills and improve the automation level of negotiation support.
- Analyze the negotiation patterns in wiki based WinWin requirements negotiation, especially the asynchronous negotiation.
- Improve WikiWinWin for synchronous collaboration, and also explore its strengths in asynchronous collaboration.

7. References

- [1] The Standish Group Chaos Report (1995), <http://www.standish-group.com>, 1995.
- [2] The Standish Group Chaos Report (2004), <http://www.standish-group.com>, 2004.
- [3] J.O. Grady, System requirements analysis, McGraw-Hill, 1993
- [4] A.G. Sutcliffe, User-Centred Requirements Engineering, Springer, London
- [5] B. Boehm, P. Bose, E. Horowitz, and M.J. Lee, "Software Requirements Negotiation and Renegotiation Aids: A Theory-W Based Spiral Approach", Proceedings of the 17th international conference on software engineering, April 1995
- [6] B. Boehm, P. Grünbacher, and R.O. Briggs, "Developing Groupware for Requirements Negotiation: Lessons Learned", IEEE Software, May 2001, pp. 46–55
- [7] E. Gottesdiener, Requirements by Collaboration: Workshops for Defining Needs, Addison-Wesley Professional, 2002
- [8] B. Boehm, "Requirements that Handle IKIWISI, COTS, and Rapid Change", IEEE Computer, July 2000, pp. 99-102
- [9] R. Pew and A. Mavor, Human-Systems Integration in the System Development Process: A New Look, National Academy Press, 2007
- [10] B. Boehm and R. Ross, "Theory-W Software Project Management Principles and Examples", IEEE Transactions on Software Engineering, Vol. 15, No. 7, July 1989, pp. 902 - 916
- [11] Wikipedia: <http://wikipedia.org/>
- [12] P. Louridas, "Using Wikis in Software Development", IEEE Software, vol. 23, no. 2, 2006, pp. 88-91
- [13] TWiki success stories, <http://twiki.org/>
- [14] C. Wagner, and A. Majchrzak, "Enabling Customer-Centricity Using Wikis and the Wiki Way", Journal of Management Information Systems,
- [15] B. Decker, E. Ras, J. Rech, P. Jaubert, and M. Rieth, "Wiki-Based Stakeholder Participation in Requirements Engineering", IEEE Software, Vol. 24, No. 2, March/April 2007, pp. 28-35
Vol. 23 No. 3, Winter 2007, pp. 17 – 43
- [16] P. Grunbacher, EasyWinWin OnLine: Moderator's Guidebook, A Methodology for Negotiating Software Requirements. GroupSystems.com and USC-CSE 2000.
- [17] M. J. Lee, "Foundations of the WinWin Requirements Negotiation System," Ph.D. thesis, Computer Science Department, University of Southern California, Los Angeles, CA 90089, August 1996.
- [18] R.O. Briggs, G. D. Vreede, and J. F. Nunamaker, "Collaboration Engineering with ThinkLets to Pursue Sustained Success with Group Support Systems", Journal of Management Information Systems
Vol. 19 No. 4, Spring 2003 pp. 31 - 64
- [19] H. Kitapci, B. Boehm, "Formalizing Informal Stakeholder Decisions – A Hybrid Method Approach", The 40th Annual Hawaii International Conference on System Sciences, 2007
- [20] R.O. Briggs and P. Gruenbacher, "EasyWinWin: Managing Complexity in Requirements Negotiation with GSS", The 35th Annual Hawaii International Conference on System Sciences, 2002
- [21] H. Kitapci, "Formalizing Informal Stakeholder Inputs Using Gap-bridging Methods", Ph.D. thesis, Computer Science Department, University of Southern California, Los Angeles, CA 90089, August 1996.
- [22] P.R. Carlile, "A Pragmatic View of Knowledge and Boundaries: Boundary Objects in New Product Development", Organization Science, Vol. 13, No. 4, July 2002, pp. 442 – 455
- [23] S. L. Star, "The structure of ill-structured solutions: Boundary objects and heterogeneous distributed problem solving", Readings in Distributed Artificial Intelligence, M. Huhns and L. Gasser, Eds. Menolo Park, CA: Morgan Kaufman, 1989
- [24] B. Boehm, D. Port, and M. Al-Said, "Avoiding the software model-clash spiderweb," Computer, vol. 33, pp. 120-122, 2000
- [25] B. Boehm and et al. Software Cost Estimation with COCOMO II, Prentice Hall (2000)
- [26] B. Boehm and D. Port, "Educating Software Engineering Students to Manage Risk," in Proceedings, ICSE, ACM/IEEE, 2001, pp. 591-600
- [27] J. Thorp, The Information Paradox: Realizing the Benefits of Information Technology: McGraw-Hill, 1998
- [28] 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, 128
- [29] K. Swisher, "Wiki may alter how employees work together", Wall Street Journal, July 29, 2004, B1
- [30] A. Majchrzak, C. Wagner, and D. Yates, "Corporate Wiki Users: results of a Survey", WikiSym '06: Proceedings of the 2006 international symposium on Wikis , 2006, pp. 99-104
- [31] C. Wagner and P. Prasarnphanich, "Innovating Collaborative Content Creation: The Role of Altruism and Wiki Technology", The 40th Annual Hawaii International Conference on System Sciences, 2007
- [32] Wikipedia: MediaWiki : www.mediawiki.org
- [33] A. Majchrzak, C. Wagner, and D. Yates, "The role of shapers in virtual firm-based practice networks," under review at Management Science.
- [34] B. Boehm, et al., "Guidelines for Lean Model-Based (System) Architecting and Software Engineering (LeanMBASE)", USC-CSSE, (<http://greenbay.usc.edu/csci577/spring2007/site/guidelines/index.html>), 2005-2007
- [35] B. Boehm and J. Lane, "Using the Incremental Commitment Model to Integrate System Acquisition, Systems Engineering, and Software Engineering", CrossTalk, Oct. 2007, to appear
- [36] A.R. Dennis, et al., "Research Report: The effectiveness of multiple dialogs in electronic brainstorming", Information Systems Research 9(2), 1997, 203-211
- [37] R.K. Yin, Case study research: Design and methods. Thousand Oaks, CA: Sage, 2003
- [38] Early Medieval East Asian Tombs Project, <http://greenbay.usc.edu/csci577/spring2007/projects/team14/>