

WikiWinWin: A Wiki Based System Together with Win Win Method for Collaborative Requirements Negotiation

Ledan Huang¹, Xiaobo Wu², Yangu Zhang³

¹Wenzhou Vocational & Technical College, Wenzhou, China

²Concordia University, Montreal, Canada

³Wenzhou Vocational & Technical College, Wenzhou, China

hld_student@126.com, xiaobowu@gmail.com, jsj_zyg@wzu.edu.cn

Abstract—It is critically important to define the correct requirements in order to get the success when developing an intensive software system. Win approach is a good negotiation model which effectively enables the interdependent stakeholders to work out a set of satisfactory requirements. The current available tool, EasyWinWin, is a good practice of Win Win approach. Wiki technology, nowadays, is popular in being used on the collaborations thanks to its good features like flexibility and easy to access by internet browser. Being a potential successor to EasyWinWin, WikiWinWin establishes the Win Win approach on wiki basis. An experiment is illustrated in the paper to display that the WikiWinWin is a good choice for the collaboration activities of requirements negotiation though it still has quite a few limitations.

Keywords—Wiki; Win Win; EasyWinWin; requirement

I. INTRODUCTION

Defining correct requirements, according to the customer's needs and expectations is not an easy task.

All the development of the product or project depends on the correctness of requirements. If requirements are not captured correctly the product will be exceeded the expected time, will go over budget, will be unable to meet the customer expectation and will decrease customer's satisfaction.

As customers are not always IT oriented therefore they might do not know what is important for the software engineers. They usually say "I don't know how to tell you, but I'll know it when I see it." [1] So the requirements they provide are usually incomplete, vague and very fragile. The supplier has to use various requirement elicitation techniques of interview, brainstorming, storyboarding and prototyping in order to understand the business domain of the customer and capture the requirements.

Even if the supplier comes up with a requirement specification based on prototyping or other elicitation techniques there is constant change in the requirement during the product life cycle. The customer is likely to add a new function or he wants enhancement in a functionality already performed by the system. Therefore the supplier should create a flexible design which in turn will create a flexible and manageable implementation,

managing future changes or extensions without affecting the overall functionality of the product.

In order to address the above challenges, an environment should be provided where all the stakeholders of both the customer and supplier can get to common terms with each other on a set of requirements. The environment should provide

- Effective communication and coordination among the stakeholders on a set of requirements
- Stakeholders should develop a common understanding of the requirements; so that there is no more than one interpretation of the same requirement by different stakeholders.
- As the requirements evolve, the changes should be managed and communicated to the relevant stakeholders [2].

To provide collaboration among stakeholder to achieve their win conditions a tool EasyWinwin was developed by University of Southern California. However EasyWinWin is a desktop application which needs to be installed on the machine before using it. Therefore to transfer the functionality of EasyWinWin we noticed that the wiki technology is gaining popularity day by day due to its ease of use. In wiki user can create web pages very easily and the wiki automatically links to the pages that are already made by other users of the wiki when it finds a text as a title of some other wiki page. So the users can collaborate with each other to form a webpage, article etc. Therefore D.Yang [1] and his co-workers developed WikiWinWin which allows collaboration in the distributed environment much easily and also establishes the Win Win approach to form a shared understanding of the requirement among all the stakeholders.

II. BACKGROUND

A. Current resolutions for requirement's collaboration

Currently, EasyWinWin tool is available for requirements collaboration. It's an automation Group Support System realized on the basis of WinWin requirement negotiation methodology.

EasyWinWin is successful in the negotiation of software requirements among the stakeholders. But EasyWinWin still has its own problems. First, it's not easy to trace and get adapted to the changes on requirements as the project proceeds. Besides,

EasyWinWin is a groupware, which requires all the stakeholders who take part in the requirement negotiation have some knowledge of computer and software operations.

D.Yang is looking for an approach which is easy for all kinds of stakeholders to collaborate on the requirements negotiation both synchronously and asynchronously. Besides, the approach should overcome EasyWinWin's disadvantage on adapting to the requirements' evolution.

The Fraunhofer Institute develops a wiki-based system named as SOP-wiki (Software Organization Platform) which improves wiki by providing better structuring the contents [3]. This system is pretty good when used for editing requirement document, grouping and management. However it's still far away from satisfaction to support a requirements negotiation process.

It is a good idea to establish WinWin approach on wiki platform to get a tool which can ensure the satisfaction from all stakeholders and at the same time can keep this satisfaction all through the project in spite of the inevitable evolutions.

B. Foundation

Following we describe the three basis of WikiWinWin, which are Win Win model, the object boundary and Wiki technology.

1) Win Win Model

The win win spiral model is based on the principle of "making everyone a winner". Following first three stages of win win spiral model are used to form a win win system which is the basis of WikiWinWin together with the Wiki technology.

- Identify stakeholders
- Identify Stakeholders' win conditions
- Reconcile win conditions [4].

First of all, the stakeholders are necessary to be identified. These stakeholders then define their win conditions. Stakeholder then review their own win conditions in conjunction with other stakeholder win conditions and work collaboratively to reconcile on the set of win conditions.

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

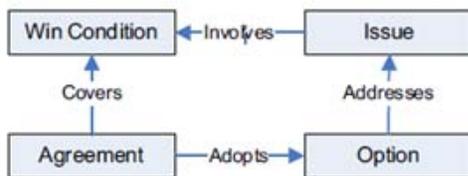


Figure 1. WinWin WIOA Negotiation Model [1]

- 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 [1].

2) Boundary Objects

Stakeholders from different background, who play different roles in software development, gather together to solve the common problems. A concept that is understood differently by different stakeholders due to their varied interest in the product is called Boundary Object. Different stakeholder can be talking about the same concept but they might have different interpretation of the same concept. Thus stakeholders might not have a shared understanding of the concept, which can be a software requirement.

Star 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 [1].

3) Wiki Technology

The word Wiki comes from Hawaii language which means "quick" [5]. In wiki anyone can create a webpage which can be viewed by other users. Similarly any other user can also edit and update any page created by any other user. So wikis provide an excellent way for different people distributed around the world to collaboratively work to create webpages. This collaborative nature of wiki can be used in requirement engineering to allow people to define a set of requirements and form a shared understanding over them using wiki. Wikis have a number of advantages. That's the reason why Wikis are prevailing in the corporate culture now. Some of the advantages of wiki over other technologies like "discussion forums, weblogs, and Groupware" [1] are

- It is extremely easy even for novice users to create and edit web pages
- Reduces redundancy of content by linking already available pages in wiki as the user writes the contents.
- The web pages are not scrutinized, therefore the content written by any user is immediately available for other users to view or edit
- There is not a single author for a webpage. However different people collaborate to create a webpage.
- Wiki maintains web pages revision history automatically.

Wiki supports effective collaboration among stakeholder over requirements however it does not support

win win requirement negotiation model to handle conflicts and suggesting alternate options to reach a consensus.

TABLE I. Challenges of requirements negotiation and Strengths of wiki [1]

Challenges	Wiki's strengths
Need active involvement of stakeholders, especially the clients and users	Wiki can facilitate high level of constructive customer engagement
Stakeholders need to collaboratively and incrementally find out what has to be built	It is easy to exchange ideas and share knowledge in wiki
Need to adapt to the changes of requirements	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	It is easy to integrate new information or new collaboration-activity support into the wiki website
Using various boundary objects to facilitate interdisciplinary collaborative design and development	It is easy to integrate new boundary object information into the wiki.
Overlapping, obscure, and ambiguous inputs cause confusion	If some users have reorganization (Shaper) skills, wiki flexibility can accommodate them.

III. PROPOSED SOLUTION- THE WIKIWINWIN

The proposed solution to improve the features of EasyWinWin and provide collaboration among stakeholders is WikiWinWin which implements the Win win requirements negotiation model and provides collaboration among stakeholder disregard of their location in an easy to use Wiki.

There are two kinds of users in Wiki. "Personal Knowledge Contributors (PKC) and Shapers" [1][6]. The PKC are the people who contribute their knowledge to the Wiki so that other people can view and comment on their knowledge. The shapers are the people who can be said at an upper level from PKCs, may be more knowledgeable people. Shapers basically organize and review the contributions of PKCs, see if the information provided by the PKC is correct or not. In case the information contributed by the PKC is not correct the shapers identify the issues or update the information when the information is known. All shapers can also act as PKCs. One of the Shapers can be said as facilitator or moderator who will initiate the requirement negotiation process and help in having a smooth requirement negotiation process.

The WikiWinWin requirement negotiation model has three phases:

- Setup a win win negotiation environment
- Negotiate WIOAs
- Continuous refinement of win requirements

At the end of the requirement negotiation a System and software Requirement Description is created in a word file [1]. In the current version of WikiWinWin, the traceability of the requirement to the win conditions is not possible.

As the project proceeds, various things change in the project, there is a need to continuously update these changes so that the latest information is available to all the stakeholders. The changes can be adding of new stakeholder, adding new terminology or requirement related information, mentioning about concurrent prototype, modifying win conditions, adding or updating

issues/ conflicts, adding or updating the options, changing of agreement 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." [1]

In this section we described about the activities involved in Win Win requirement negotiation. In the next section, we will look into an experiment done on WikiWinWin by D.Yang and his workmates in order to illustrate how WikiWinWin behaves on the true requirements negotiation activities.

IV. EXPERIMENT AND EVALUATION

A. D.Yang's Experiment

D.Yang [1] applied WikiWinWin approach on a real-client e-service student project, which is to develop the Early Medieval East Asian Tombs Database system for a client from the East Asian Library at USC [7]. The project provides a web-based interface and shows location of tombs in a map after searching the tomb information in the database.

There are altogether six graduate students working on this project, who are familiar with the typical scenarios and difficulties during Win Win requirements negotiation. They played as 1 client, 3 developers, 1 shaper/developer and 1 shaper/quality assurance. They went through the Win Win requirements negotiation process for this project by means of WikiWinWin. At the same time, D.Yang concentrated on simulating the case of requirements evolutions and described two methods they took with WikiWinWin to improve its being adaptive to changes and asynchronous negotiation.

TABLE II. 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
WinWin negotiation results		
Win conditions ideas: 62	Win conditions: 30 (All agreed)	
Issues: 5 (All resolved)	Options: 8 (6 were adopted)	

TABLE III. Effort of facilitating tasks [1]

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

The above two tables (Table II and Table III) summarize the working efforts the stakeholders spent on the experiment [1]. Each stakeholder spent 5 hours on requirements negotiation including brainstorming, giving

out win conditions, identifying issues, providing options and reaching agreements. The shapers spent extra 8 hours to do the facilitation tasks for the negotiation. As a result, they worked out 62 win condition ideas, resolved 5 issues, proposed 8 options, 6 among which were adopted and achieved 30 agreements.

The other important artifacts the experiment boasts are two examples of how WikiWinWin is used to perform asynchronous negotiation and to meet requirements evolutions as project goes on.

B. Evaluation on WikiWinWin according to D.Yang's experiment

D.Yang gives the comparison between EasyWinWin, WikiWinWin, SOP-wiki and regular wikis after doing the experiment of WikiWinWin described in section 4.1. The following table illustrates the results they find out.

TABLE IV. Evaluation of WikiWinWin [1]

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 [3]		***	**	***	*
Multiple webpage (re)classification [3]		---	**	***	*

The comparison is executed according to the number of stars listed in table 4.

Both EasyWinWin and WikiWinWin closely follow Win Win collaboration negotiation approach, by which they provide a series formats of activities for stakeholders to take up in order to get requirements satisfied by everyone. SOP-wiki and regular wiki such as TWiki, obviously, does not support Win Win methods.

Wiki structure boasts its being easy to exchange ideas and share knowledge. Furthermore, wiki structure is extremely advantageous for stakeholders to update contents from time to time and keep trace of vision. So WikiWinWin, SOP-wiki and regular wikis gain advantage on these sides.

V. CONCLUSION AND FUTURE WORK

Wikis techniques are becoming increasingly useful nowadays for stakeholders interactivities on the purpose of requirements negotiation. From the experiment and the comparison in the section of Evaluation, we can see clearly that wiki is an effective platform which facilitates stakeholder's collaborative negotiation on requirements.

We are glad to recognize the positive points WikiWinWin created on requirement engineering. However, this system also has some limitations or

shortcomings which need the future efforts. Its working mode for synchronous collaboration is not perfect. The level of automation is not high enough as it still requires a great number of facilitation efforts made concurrently by Shaper. Besides, we can hardly tell whether WikiWinWin system will also have so positive a result on large or medium scale project as the available experiment executed only on a small-scale project in which only 6 stakeholders were included.

Last but not the least, WikiWinWin, for the next step, might add new features to help the requirement engineering become more efficiently. For example, how to export domain models automatically according to the agreements and requirements worked out on WikiWinWin.

ACKNOWLEDGMENT

This work is supported by Wenzhou Science Technology Bureau with Plan Project No.H20100037 and Wenzhou Vocational & Technical College with Project No.WZY2010035.

REFERENCES

- [1] Da Yang, Di Wu, Koolmanojwong, S. Winsor Brown, A. Boehm, B.W., Chinese Academy of Science, "WikiWinWin: A Wiki Based System for Collaborative Requirements Negotiation"
- [2] Vibha Sinha, Bikram Sengupta, Satish Chandra, "Enabling Collaboration in Distributed Requirements Management," IEEE Software, vol. 23, no. 5, pp. 52-61, Sep/Oct, 2006
- [3] 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
- [4] M.J.Lee, "Foundations of the WinWin Requirements Negotiation System," Ph.D.theis, Computer Science Department, University of Southern California, Los Angeles, CA 90089, August 1996
- [5] P.Lourideas, "Using Wikis in Software Development", IEEE Software, vol.23, no.2, 2006, pp.88-91
- [6] A.Majchrzak, C.Wagner, and D.Yates, "The role of shapers in virtual firm-based practice networks," under review at Management Science.
- [7] Early Medieval East Asian Tombs Project, <http://greenbay.usc.edu/csci577/spring2007/projects/team14>