

Collaborative Requirements Negotiation with EasyWinWin

Paul Gruenbacher

Computer Science Department

University of Southern California, Los Angeles

941 W. 37th Place, Los Angeles, CA 90089-0781

Email: gruenbac@sunset.usc.edu

Abstract

The EasyWinWin methodology supports collaborative elaboration, prioritization, and negotiation of system requirements. The approach enhances the WinWin negotiation model by integrating group productivity techniques and collaborative tools. EasyWinWin has been used in real-client custom development projects and for real-world COTS product release planning. Our experience shows, that EasyWinWin helps to improve stakeholder involvement in the requirements engineering process and enhances the frequency, directness, and extent of stakeholder interactions. This helps to develop a deeper and broader set of deliverables in a shorter time and to identify and mitigate risks earlier in the life cycle.

1. Introduction

The involvement and interaction of success-critical stakeholders is crucial to any software project. The requirements engineering process in particular depends heavily on the collaboration of stakeholders with different backgrounds, objectives, and expectations. Stakeholder involvement, however, remains a challenging problem: On the one hand the used methodology and supporting tools have to be easy to understand by technically non-savvy participants, on the other hand the resulting deliverables have to meet the needs of software engineers. We have been addressing these challenges in the EasyWinWin project. EasyWinWin builds on lessons learned in previous WinWin groupware tools and in Group Support Systems research [5].

WinWin is a requirements negotiation approach that aims at making winners of all success-critical stakeholders to avoid win-lose and lose-lose situations in software projects [1, 2]. Stakeholders using the WinWin model express their needs and concerns using a set of artifacts with the overall goal of achieving mutually satisfactory agreements: *Win conditions* describe stakeholder objectives. *Issues* capture known constraints, problems, and conflicts detected among win conditions. *Options* are used to propose alternatives and solutions for identified

issues. *Agreements* describe achieved negotiation results. Furthermore, a glossary of *Terms* helps to capture and define the domain language. WinWin negotiation artifacts are organized and structured using a *Domain Taxonomy*. While the WinWin negotiation model is generally accepted it turned out that providing adequate tool support for heterogeneous collaborating stakeholders is a challenging task: USC's UNIX-based WinWin groupware system allows capturing and managing WinWin artifacts in a distributed, asynchronous work mode. The tool strictly enforces the negotiation model's artifacts, artifact states, and links. While this approach allows sophisticated traces of the negotiation process, it complicates the involvement of stakeholders that are not willing to pay the effort required to get familiar with the environment.

In the EasyWinWin project, we developed a 'light-weight' implementation of the WinWin approach to lower the entry barrier for all stakeholders [4]. We enhanced stakeholder guidance and interaction to improve the negotiation results. EasyWinWin is based on the off-the-shelf product GroupSystems by GROUPSYSTEMS.COM: This collaborative environment provides group productivity tools like e.g., electronic brainstorming, categorization of ideas, group outlining, and voting [9]. More specifically, we pursued the following objectives:

- *Eased involvement of stakeholders:* EasyWinWin provides comprehensible collaboration techniques and relaxes the strict enforcement of the negotiation model. We adopted and enhanced proven facilitation and collaboration methods. An example is the Free Brainstorming technique, which is used to gather stakeholder statements about their win conditions.
- *Enhanced stakeholder interaction:* Groupware properties like anonymity, rapid interaction, electronic transcripts, and the support for co-located as well as distributed interaction help to increase the directness and frequency of stakeholder communication.
- *Improved stakeholder guidance:* A process handbook provides detailed facilitation guidelines and helps a project manager to organize and moderate the EasyWinWin process [8].

- **Improved analysis and integration:** We have integrated the groupware environment with the Rational Rose CASE tool to improve the analysis of negotiation results and to enable traceability to other artifacts in the software life cycle (e.g., architectures, designs) [7].

We have applied full and tailored versions of EasyWinWin in different contexts of requirements negotiation: This includes the development of a shared product vision, the development of a requirements specification for custom development, COTS product release planning, as well as system transition planning. Recently, EasyWinWin has also been used for the collaborative definition of a business process.

2. EasyWinWin techniques

EasyWinWin defines a set of activities guiding stakeholders through a process of gathering, elaborating, prioritizing, and negotiating requirements. Figure 1 gives an overview about the EasyWinWin activities and deliverables. In this section, we will briefly describe these activities and demonstrate how collaborative tools and techniques support them. Screenshots are taken from the “USC bookstore site and ad management” requirements negotiation process (see Section 3).

Elaborate domain taxonomy: A domain taxonomy helps to structure and organize negotiation results and serves as a checklist during the process. Typical high-level taxonomy elements include *system capabilities* (features and services), *interfaces* (to the user interface and other systems), *properties* (‘non-functional’ requirements), *project & process* (cost, schedule, development tools, support), and *system evolution* [3].

In EasyWinWin, a domain taxonomy template is jointly refined and tailored to the specific needs of a project using the *GroupOutliner* tool. The could-be/should-be technique enables the collaborative development of the taxonomy helping the team both to understand and influence the agenda and the topics of the negotiation process. During the ‘could be’-step, stakeholders add comments about missing taxonomy elements and required changes to the initial outline. During the ‘should be’-step, the comments are used to refine and tailor the existing project-specific taxonomy.

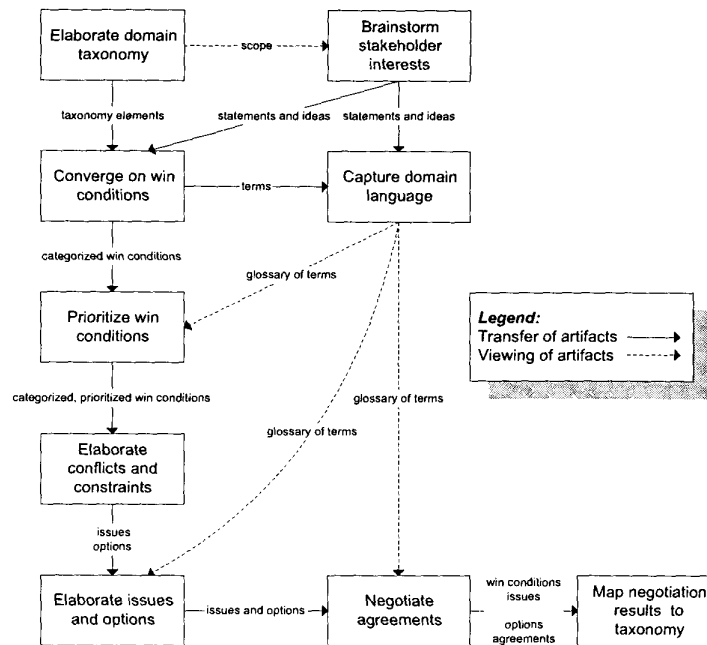


Figure 1: EasyWinWin activities

Brainstorm stakeholder interests: Stakeholders collect statements about their vested interests (win conditions) in a free brainstorming session. The *Electronic Brainstorming* tool (Figure 2) supports gathering ideas simultaneously and anonymously. This co-operative learning process allows participants to submit, view, and share their different opinions, backgrounds, and perspectives.

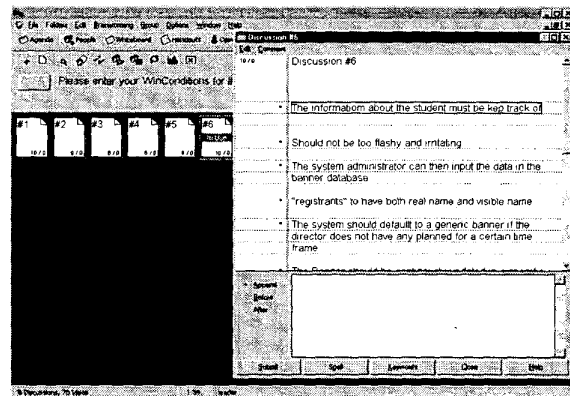


Figure 2: Free Brainstorming

In our first case studies, we asked participants to follow the brainstorming rules by Osborne, i.e., not to criticize, or comment ideas. However, we found that in an electronic brainstorming session, commenting on statements leads to improved interaction and to more meaningful negotiation results as stakeholders start to develop new

ideas based on other stakeholders' comments and concerns.

Converge on win conditions: The objectives of this step are twofold:

(1) Elaborating a non-redundant list of unambiguous, succinct descriptions of win conditions based on the brainstorming results. (2) Organizing the win conditions into high-level taxonomy elements.

In most cases, several brainstorming statements are merged into a new win condition. If a statement covers several aspects (e.g., it describes a system capability and a system property), the statement gives rise to several win conditions. The brainstorming statements are attached to the resulting win conditions to preserve the brainstorming rationale (see Figure 3).

We have been using two different techniques so far using the Categorizer tool:

a) *Offline:* In this case the project manager carries out the categorization together with the requirements engineer.

b) *Fast Focus:* This technique is used in a meeting by all stakeholders for converging on a list of win conditions. Team members looking at brainstorming ideas are asked to summarize and select the most important ideas, which are then collected in a public list of win conditions by the moderator.

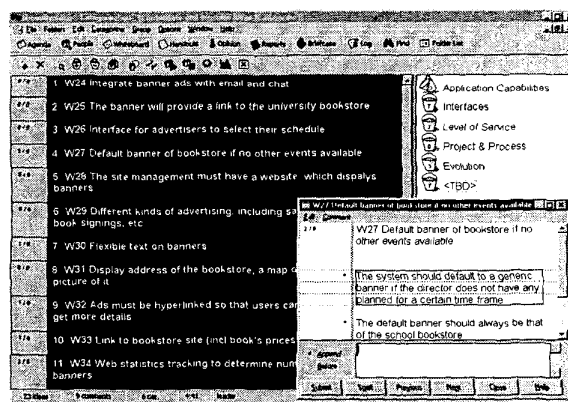


Figure 3: Converge on win conditions

Capture domain language: EasyWinWin allows defining and sharing important terms of the domain language. During the categorization and consolidation of win conditions, the facilitator collects a list of important terms appearing in the ideas. The electronic brainstorming tool automatically groups ideas based on this list of keywords (and defined synonyms). Stakeholders then develop,

jointly review, and comment the definitions for the different terms.

Prioritize win conditions: In this task, we classify the collected win conditions from two perspectives: The *business importance* expresses the relevance of a win condition to clients and users. The *ease of realization* indicates the perceived and expected difficulties of implementing a win condition (e.g., technical or financial constraints). Stakeholders rate win conditions using these two criteria on a scale from 1 (=unimportant, difficult) to 10 (=important, easy). In the voting process, developers focus on technical issues while clients and users rate the business relevance. The Alternative Analysis tool supports voting on a set of choices, using a customizable list of criteria and voting methods. It groups the win conditions into four categories:

Low hanging fruits: Win conditions that are important and easy to accomplish.

Important with hurdles: High-priority win conditions, which are difficult to implement.

Maybe later: Low-priority win conditions that may be considered later because of the low difficulty of realization.

Forget them: Unimportant win conditions that are difficult to achieve.

Elaborate conflicts and constraints: By analyzing the prioritization poll, the team is revealing reasons for discrepancy in the stakeholder perceptions. In the "crowbar" technique, the vote spread of elements showing discrepancy is presented to the team with the instruction to find reasons for the different opinions. In the following negotiation process, which is recorded by a scribe, stakeholders identify puzzling win conditions, surface issues, and reveal constraints.

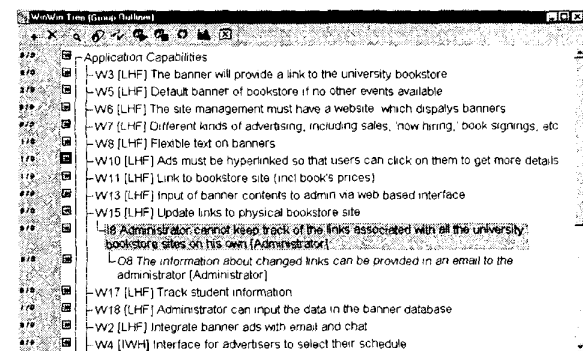


Figure 4: WinWin negotiation tree

Elaborate Issues and Options: Conflict identification and resolution in the WinWin negotiation model is based on Issues and Options. Participants populate the tree in several iterations of identifying and reviewing issues and options.

The WinWin negotiation tree see (Figure 4) is used to organize all win conditions, issues, options, and agreements. In this shared outline the level in the hierarchy together with a tag determines the type of an artifact. Links between artifacts are established by creating sub-items.

Negotiate Agreements: Once the stakeholders have finished elaborating possible options for each issue, they negotiate agreement for every win condition and every open issue. However, the discussion can be focused on the available possibilities, which are already explicitly captured in the win-win tree. The group continues to work until the WinWin equilibrium is achieved and all issues and win conditions are covered by agreements.

Map artifacts to taxonomy: In this activity stakeholders associate all WinWin artifacts to taxonomy elements in the Categorizer tool. In the first step, the team collaboratively sorts artifacts into buckets (=taxonomy elements). In the second step the contents of the buckets is reviewed to address one to many relationships between artifacts and taxonomy elements as well as incorrect associations.

3. Case Studies

We validated the approach in several real-client projects at USC. During the Fall 99 software engineering class, EasyWinWin was used in requirements negotiations for three web-based information systems and three collaboration systems. Clients, users, and developers participated in the requirements negotiation process. The starting point for each team was a short project description developed by the clients, which outlined the project motivation, background, and major objectives. A facilitator moderated several sessions for users, clients, and developers of each project team. During the sessions, the facilitator introduced the different activities of the EasyWinWin methodology and explained the use of the technology.

The focus of these projects has been as follows:

- **Instruction and research support system:** A system managing links to multimedia objects stored in USC's 'Center for Scholarly Technology' databases that allows to create and maintain databases for instruction and research.

- **Business/reference Q&A's:** A web-based system using question-matching-answer strategies to provide customer support for clients of the USC Library.
- **Bookstores virtual calendar:** A web-based system for a consortium of college bookstores supporting private student calendars that are populated with academic and school-specific events.
- **Bookstores Email & student-student communications:** A web-based system hosted by the USC bookstore providing e-mail and student-to-student chat capabilities over the WWW.
- **Bookstores site and advertisement management:** A backend system allowing USC Bookstore administrators to customize the placement of banner ads.
- **EasyWinWin web viewing and commenting:** A web-based system allowing to baseline, view and comment WinWin negotiation results and to create customizable views and reports.

Subsequently, EasyWinWin has been used in several real-world COTS product requirements negotiations: release planning for an existing product as well as strategic product definition for a new product. These negotiations involved sales people, executives, and developers and were organized as multi-day workshops. Remote stakeholders contributed via web-clients and audio links.

Collaboration Scenarios: We have developed recommendations for using EasyWinWin in various collaboration scenarios (see Table 1):

Table 1: Collaboration scenarios

<i>Scenario</i>	<i>Activities</i>
Same time/ same place or Same time/ different place with audio link	- Elaborate taxonomy ("should be") - Brainstorm stakeholder interests - Converge on win conditions - Elaborate conflicts and constraints - Review issues and options - Negotiate agreements
Different time/ different place	- Elaborate taxonomy ("Could be") - Prioritize win conditions - Capture domain language - Identify issues and options
Offline by requirements engineer	- Elaborate taxonomy ("Could be") - Converge on win conditions - Capture domain language

Observations: Compared to results from requirements negotiations done with previous WinWin systems [6], the teams using EasyWinWin were able to develop a broader and deeper set of results in a shorter time:

- **Reduced cycle time:** Parallel contributions and rapid interaction result in a fewer stakeholder hours. For ex-

ample, the brainstorming sessions lasted between 20 and 75 minutes, depending on the number of participants, the complexity of the problem, and the familiarity of the people with the domain.

- *More deliverables:* The approach allows participation of more people and elicits more from everybody involved. We were able to increase the number of artifacts the teams created during the process. For example, in the USC projects, the stakeholders collected an average of 100 statements during the 20-30 minute brainstorming sessions, which resulted in 40-50 win conditions. In the COTS project examples, participants collected about 300 statements resulting in approximately 100 win conditions.

4. Conclusions

EasyWinWin has been successfully applied to real-client custom development projects and large-scale COTS product definition efforts. The methodology has helped to achieve win-win solutions and to build trust among stakeholders.

EasyWinWin enhances the WinWin negotiation approach in several dimensions:

- *Improved involvement and interaction:* The approach relaxes modeling constraints and builds on groupware supported creativity and facilitation techniques (e.g., brainstorming, crowbar, fast focus, etc.), which significantly eases the involvement, participation, and interaction of stakeholders.
- *Defined process:* EasyWinWin provides a repeatable, documented process [8] guiding stakeholders through a negotiation and thus improves the practicability of the approach.
- *Support for different collaboration scenarios:* EasyWinWin fosters the variety and frequency of stakeholder interaction, both computer-based and face-to-face, by combining different modes of collaboration (same time/different time, same place/different place, offline/facilitated group).
- *Improved prioritization and elaboration of issues:* EasyWinWin improves the prioritization of win conditions by providing a two-dimensional voting method (business importance, ease of realization). By analyzing the prioritization poll, teams are able to identify more issues, conflicts, and constraints than in previous generations of WinWin systems.

5. Acknowledgements

This work is supported by the Austrian Science Fund (Erwin Schrödinger Grant 1999/J 1764 "Collaborative Requirements Negotiation Aids"). I would like to thank Barry Boehm and Bob Briggs for many valuable discussions and their support of the EasyWinWin project.

6. References

- [1] Boehm B., In H., "Identifying Quality Requirement Conflicts", IEEE-Software, 13(2):25ff, March 1996.
- [2] Boehm B., Egyed A., Kwan J., Port D., Shah A., Madachy R., "Using the WinWin Spiral Model: A Case Study", IEEE Computer, 7:33-44, 1998.
- [3] Boehm B., Port D., "Escaping the Software Tar Pit: Model Clashes and How to Avoid Them", Software Engineering Notes, Association for Computing Machinery, pp. 36-48, January 1999.
- [4] Boehm B., Gruenbacher P., "Supporting Collaborative Requirements Negotiation: The EasyWinWin Approach", International Conference on Virtual Worlds and Simulation VWSIM 2000, San Diego, to appear: SCS Conference Proceedings.
- [5] Boehm B., Gruenbacher P., Briggs B., "EasyWinWin: Lessons Learned from Four Generations of Groupware for Requirements Negotiation", Technical Report, USC-CSE, 2000.
- [6] Egyed A.F., Boehm B., Comparing Software System Requirements negotiation patterns, Journal for Systems Engineering, John Wiley & Sons, 1999.
- [7] Gruenbacher P., Egyed A., Medvidovic N., Dimensions of Concerns in Requirements Negotiation and Architecture Modeling, Workshop on Multi-Dimensional Separation of Concerns in Software Engineering (ICSE 2000).
- [8] Gruenbacher P., Rai S., "The EasyWinWin Process Guide", to appear: USC-CSE and GROUPSYSTEMS.COM.
- [9] GroupSystems.com, 1430 E. Fort Lowell Rd., Suite 301 Tucson, AZ, USA 85719, <http://www.groupsystems.com>