CSE - Annual Research Review

From Informal WinWin Agreements to Formalized Requirements

Hasan Kitapci

hkitapci@cse.usc.edu

March 15, 2005
Overview

- Introduction
- EasyWinWin Requirements Negotiation and Requirements Specification
- Research: Formalizing Informal EasyWinWin Agreements
- Preliminary Results
- Expected Results
- Future Work
Introduction

- Requirements related problems have a high impact on projects failure
  - Incomplete requirements
  - Lack of user involvement
  - Unrealistic expectations
  - Changing requirements and specifications

- Requirements related defects cost more than others when not removed as early as possible in the software development life-cycle

- Return on Investment for good requirements is higher than the other software artifacts
  - Value-prioritized, feasible, unambiguous
EasyWinWin Negotiation Process and Deliverables

Negotiation Inputs

High-level objectives of project

Negotiation Purpose

EasyWinWin Negotiation Process

1a. Welcome to EasyWinWin
1b. Identify Success-Critical Stakeholders
2. Explain the Mandate
3. Explain the Process
4. Review and Expand Negotiation Terms
5. Brainstorm Win Conditions
6. Evaluate Win Conditions
7. Reveal Constraints and Assumptions
8. Classify Win Conditions by Priority
9. Identify Issues
10. Propose Options
11. Negotiate Agreements
12. Check Completeness
Wrap-up: Report Results

Negotiation Results

Negotiation Topics

Negotiation Glossary

WinWinTree

Win condition

Involves

Issue

Addresses

Agreement

Covers

Option

Adopts

Development Project (Deliverables)

SRS Outline

Project Glossary

Project Plan

Requirements Specification

Project Contract
Current Requirements Specification Process

- **Human-initiative template with Inspection**
  - Member responsible for Requirements fills out the MBASE requirements template for agreed-upon win conditions
  - Team member(s), or Independent Validation and Verification person inspects the set of specifications to find misinterpretations and defects
  - Client representatives and outside experts review the reworked specifications in Architecture Review Boards in each anchor points and find fewer defects
Research Context Diagram

- **Requirements Negotiation**
  - Stakeholders
  - Transformation to WinWin Report

- **Requirements Documentation**
  - Output to SSRD

- **Natural Language Processing and Keyword Analysis**
  - Computer
  - Reviewers

- **Walkthrough**
  - Review negotiation data
  - Inspect WinWin report

- **Inspection**
  - Document level checking
  - Requirement level checking

- **Improvements to EasyWinWin Negotiation**
- **Automation of Requirements Specification Generation**

- **Template phase**
- **Language phase**
- **Inspection phase**

© 2005 USC Center for Software Engineering 6 3/15/2005
Activities for gap-bridging

- Generate report from the Negotiation tool
  - Use EasyWinWin output to fill requirements templates
    - Map WinWin Artifacts into Requirements template attributes
  - Final Output generated as a requirements specification document

- Use NLP and keyword analysis techniques to extract template-relevant meanings and support inspections
  - Extract verbs and nouns, priority, reference, etc.
  - Find ambiguous words and missing information
  - Feedbacks to stakeholders
Activities for gap-bridging (cont.)

- Inspect negotiation results and post-process them into precise specification
  - Use walkthroughs during the process (statement-level checking) and inspections on the outputs (document-level checking)
  - Find ambiguities, unstated assumptions, conflicting requirements, and other defects as early as possible
  - Avoid misinterpretations of WinWin artifacts

- Modify/Update the requirement templates
  - Use extracted data to fill attributes in MARS (Measurable, Achievable, Relevant and Specific) or Use Case forms
  - Use application and implementation domain knowledge
Defect Analysis of EasyWinWin

- **Searchable archive based on predefined criteria**
  - Unverifiable statement: indicator “predefined”
  - Found by speech tag: adjective

- **User authentication**
  - Unclear statement: indicator “not a sentence”
  - Found by speech tag: no subject and verb phrases

- **Search by specific fields e.g. title, author**
  - Missing information: indicator “no subject”
  - Found by speech tag: start with verb

- **More time is required to integrate with Z-bit or keep it as low priority**
  - Unclear term: indicator “Z-bit”
  - Found by speech tag: proper noun
EasWinWin vs. SSRD

Agreements

- Level of Service: 18%
- Interface: 14%
- Capability: 25%
- Evolution: 9%
- Project: 34%

Requirements

- Level of Service: 13%
- Interface: 8%
- Capability: 27%
- Evolution: 7%
- Project: 45%
**Ex: Capability Requirement**

<table>
<thead>
<tr>
<th>Requirement:</th>
<th>RG-2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title:</td>
<td>Users can upload/modify/download/delete authorized files</td>
</tr>
<tr>
<td>Priority:</td>
<td>Very High</td>
</tr>
<tr>
<td>Description:</td>
<td>File management, as a part of the system, will improve information sharing and accessing. This will help users to upload/modify/download/delete files needed in their project.</td>
</tr>
<tr>
<td>Input(s):</td>
<td>User's request (click the files button)</td>
</tr>
<tr>
<td>Source(s):</td>
<td>User</td>
</tr>
</tbody>
</table>
| Output(s):   | **Upload** – the new files  
**Modify** – the modified files  
**Download** – the receive files in the user’s own storage  
**Delete** – the file disappears in the project’s storage |
| Destination(s): | Given storage of project group and user's own storage. |
| Precondition(s): | User knows the files wanted to manipulate |
| Post condition(s): | User manipulates files |
| Proposed Activity: | **File sharing and management** as part of the collaboration services |
| Win Win Agreement(s): | A9, A10 |
| Mainstream Scenario: | 1. Upload files needed in their project;  
2. Modify files needed in their project;  
3. Download files needed in their project;  
4. Delete files needed in their project. |
| Exception Handling Scenario: | If current user has no right to modify the file he is requiring, the system will give an warning information to show that he does not have enough right to do this operation. |
| Reference:   | OCD 4.3 CAP-02 |

---

**Percentage of Automatically Template Filling => %33-42**
Expected Results

- An iterative and incremental process model with a quality assurance process
- All success-critical stakeholders participate in generating requirements specification
- Gather more information from stakeholders using the gap-bridging methods
- Remove ambiguities and misinterpretations from requirements
- Better traceability of where requirements come from
- Better change management for requirements
- Increase the usability of EasyWinWin process throughout the software life-cycle
Future Works

- Prototype the proposed techniques
  - Add improvements into the EasyWinWin process
  - Add activities to generate requirements specification
  - Add activities to inspect the artifacts
  - Integrate NLP tool with GroupSystems II using their APIs

- Generate user guidelines for the tool and help for the activities

- Experiment with the tool during Fall semester in CS577 Software Engineering course projects

- Analyze the results and improve the process
Questions and Comments

**Important Links** for further information

**MBASE**
- [http://sunset.usc.edu/cse/pub/research/mbase/](http://sunset.usc.edu/cse/pub/research/mbase/)

**EasyWinWin**
- [http://sunset.usc.edu/cse/pub/research/easy_win_win/](http://sunset.usc.edu/cse/pub/research/easy_win_win/)

**Formalizing Informal Stakeholder Inputs**
- [http://sunset.usc.edu/cse/pub/research/requirement.html](http://sunset.usc.edu/cse/pub/research/requirement.html)

**GroupSystems.com**