# Repeatable Quality Assurance Techniques for Requirements Negotiations

Paul Grünbacher Michael Halling
*Johannes Kepler University*
*Systems Engineering & Automation*
*4040 Linz, Austria*
*{pg | mh}@sea.uni-linz.ac.at*

Stefan Biffl
*Vienna University of Technology*
*Inst. for Software Technology*
*1040 Vienna, Austria*
*Stefan.Biffl@tuwien.ac.at*

Hasan Kitapci Barry W. Boehm
*University of Southern California*
*Center for Software Engineering*
*90089 Los Angeles, CA, USA*
*{hkitapci | boehm}@sunset.usc.edu*

## Abstract

*Many software projects fail because early life-cycle defects such as ill-defined requirements are not identified and removed. Therefore, quality assurance (QA) techniques for defect detection and prevention play an important role. The effectiveness and efficiency of QA approaches has been empirically evaluated. In this paper we discuss QA techniques optimized for requirements negotiations. In particular, we focus on negotiations using the EasyWinWin approach. We present (1) repeatable techniques for checking quality throughout negotiations as well as (2) role-oriented inspection techniques helping a project team to reduce unnecessary complexity and to mitigate risks stemming from defects in requirements negotiation results. We present the results of a thorough feasibility study we conducted to test our approach.*

**Keywords.** Pre-requirements, requirements negotiation, quality assurance techniques, GSS support, empirical evaluation.

## 1 Introduction

In any mission-critical task the quality of the resulting work products is of utmost importance. Adopting techniques emphasizing quality aspects and integrating these techniques into existing proven collaborative processes becomes very important.

Getting system requirements right for example, is crucial for project planning, architectural design, and ultimately to achieve customer satisfaction. Ensuring high-quality requirements should not be a single person effort but requires the participation and collaboration of numerous heterogeneous stakeholders with different roles. Designing repeatable techniques and work processes for quality assurance (QA) and providing technology support becomes critical in such a context: Deliverables emerging in requirements elicitation provide rationale and represent an important input to actual development [10][20]. The identification and elimination of defects in such 'pre-requirements' helps to mitigate risks and to avoid unnecessary complexity even before a more elaborated requirements specification is available.

Numerous techniques are available for capturing pre-requirements. Examples are the IBIS (Issue Based Information System) [17], DRL (Decision Representation Language) [21], Toulmin's model of argumentation [26], QOC (Question, Option, and Criteria) [22], the Potts & Burns model [23], or the WinWin negotiation model [3].

During the development of the EasyWinWin (EWW) requirements negotiation approach we have gained considerable experience in eliciting pre-requirements [4][6][11]. EWW is a collaborative methodology for requirements negotiation that adopts the WinWin negotiation model [3] and captures pre-requirements of success-critical stakeholders as win conditions, issues, options, and agreements. EWW uses a Group Support System and is based on stakeholder involvement and interaction to elicit goals and preferences. This is achieved by supporting the negotiation process with easy-to-use groupware tools and by using a human facilitator to move the process forward.

Although the outcome of a WinWin negotiation is typically not a complete, consistent, traceable, and testable requirements specification, it comprises jointly developed and agreed pre-requirements, potential risks and constraints, as well as important project-specific terms. Our lessons learned during real-world negotiations show that it is not effective to dampen the enthusiasm of stakeholders and the creative flow of ideas during a meeting with techniques that emphasize detailed wordsmithing, correctness, consistency, etc. [4].

During a negotiation rapid checks can be applied but detailed QA activities should be done after a session. The quality of requirements negotiation results is important e.g., for contract negotiation, initial project planning, and as input for requirements definition and management. The agreed results should therefore be as clear, correct, and

complete as possible with respect to a defined negotiation purpose.

This paper introduces techniques complementing the EWW requirements negotiation process and aiming at improving negotiation results. We discuss in-process as well as post-process QA techniques. Our approach is based on applying quality checks during the groupware-supported negotiation process and on an inspection-based technique for the negotiation results. The intended audiences are those stakeholders involved in requirements definition e.g., as project managers, engineers, executives, users, or customers.

The paper is organized as follows: Section 2 motivates the importance of QA techniques. Section 3 briefly discusses the EWW requirements negotiation approach and typical defects we saw in real-world negotiations. Section 4 presents rapid quality checks that can be applied during EWW sessions. Section 5 describes the inspection process and inspection techniques we have developed. In Section 6 we discusses the need for precise definition and automation of the QA techniques. Section 7 presents results from an initial feasibility study we carried out to empirically test our approach. Conclusions and an outlook on further work round out the paper.

## 2 Early Life-Cycle Quality Assurance

Early life-cycle documents containing defects might ultimately cause even more defects in the final product. Although the impact of a defect in an early life-cycle document is typically rather minor and local initially, it can become more serious in later phases, since more rework is necessary and more people are involved to fix the defects. For example, an ill-defined capability defect that can be easily fixed during requirements elicitation and negotiation can become a major problem if it cannot be realized with the chosen system architecture. Consequently, before refining the negotiation results to other life-cycle artifacts like contracts, specification, project plans, or architectural models defects should be eliminated to reduce both the effort and probability of rework stemming from undetected defects.

Numerous QA techniques are available supporting the detection of defects in early life cycle documents such as requirement specifications. Examples include Fagan inspections [7], Gilb inspections [9], or reading techniques for inspection [1]. Empirical studies have demonstrated the effectiveness and efficiency of these techniques (see e.g., [18]). Inspection represents a QA technique that can be applied to any document produced in the software development process. Most organizations performing inspections follow a three-step procedure of preparation (individual defect detection), defect collection, and defect repair [7][9].

However, there has long been a focus on source code inspection complementing test efforts [7]. In the meantime researchers found that inspecting documents like requirements increases the benefits of inspection due to earlier defect removal and saved rework. Empirical studies in academic [2][13] and industrial [8][19] environments demonstrate the benefit of software inspections.

## 3 Eliciting Pre-Requirements with EWW

The EasyWinWin process helps success-critical stakeholders to jointly discover, elaborate and negotiate their requirements [4]. It builds on the WinWin negotiation model [3] and captures pre-requirements as *win conditions*, *issues*, *options*, *agreements*, and *terms*. Important negotiation results are negotiation topics organized in a taxonomy, the glossary of project-specific terms, and the WinWin tree, which organizes and represents win conditions, issues, options, and agreements (see shaded rectangles in Figure 1). These results are typically used in a project to develop crucial deliverables such as a project plan, requirements specifications, or contracts to name but a few. EWW uses a set of groupware tools supporting the negotiation process. These tools enable structured communication among the negotiation participants, help to elicit often tacit values and knowledge, and to create a common vision among the stakeholders [11].

An EWW negotiation is typically carried out in a series of co-located or dispersed sessions involving the identified success-critical stakeholders. A facilitator moderates the negotiation process following detailed guidelines [12]. The use of a technographer in charge of handling the groupware tools has proven to increase efficiency.

An EWW negotiation results in a significantly higher number of artifacts compared to traditional paper- or blackboard-based approaches: Our experience to date shows that typical negotiations about system requirements with 10+ stakeholders result in 300+ brainstorming ideas, 100+ win conditions, 50+ issues, 50+ options, and 100+ agreements. Due to this size and complexity systematic quality assurance techniques become even more important.

### 3.1 The Negotiation Process

The activities of the EWW process are summarized in Figure 1 (for a detailed description please refer to [4][6][11][12]). The results of each team activity in the process is a well-defined deliverable: (1) negotiation topics organized in a domain taxonomy, (2) definitions of key project terms, (3) agreements providing the foundation for further plans, (4) open issues addressing constraints, conflicts, and known problems, as well as (5) further decision rationale showing the negotiation history (comments, win conditions, issues, options, etc.).

Main results of the negotiation process are a list of agreements and a list of unresolved issues (e.g., caused by

stakeholder dissent), which have to be managed as potential projects risks. Agreements of success-critical stakeholders are input to the project contract and to refinement during requirements engineering activities. The WinWin negotiation history shows how agreements and open issues can be traced back to stakeholder win conditions.

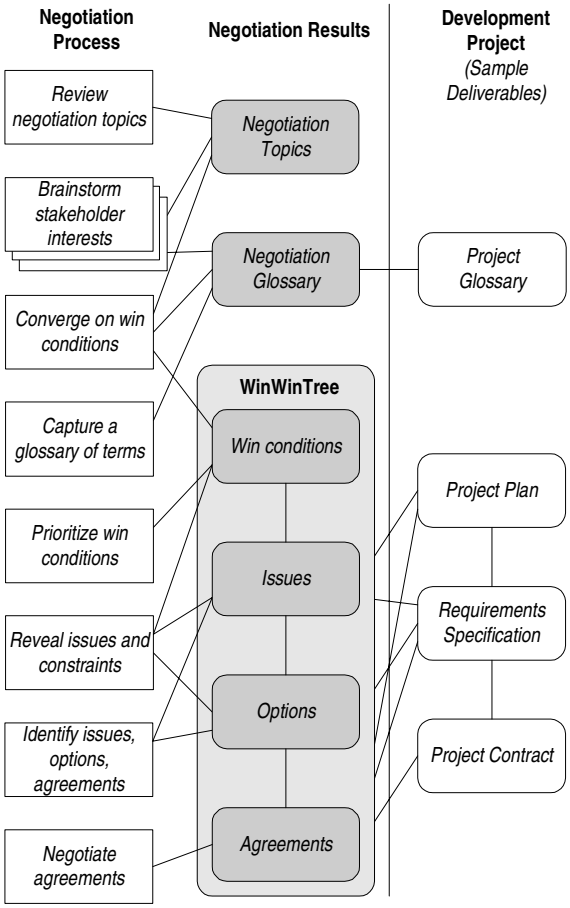


**Figure 1: EasyWinWin activities and products with relationships to important life-cycle deliverables.**

## 3.2 Typical Defects

Defects in a requirements negotiation can appear on the statement level or on the negotiation level. Typical faults we experienced are vague or ambiguous statements, missing information, wrong level of detail, or inconsistencies. All defects have to be interpreted with respect to the defined negotiation purpose, which states the context and the desired level of detail.

***Statement-Level Defects.*** Negotiation artifacts are statements written in natural language and therefore error-prone. Table 1 shows some examples of defects that can we identified in individual statements.

**Table 1: Statement-Level Defects (Examples).**

| Statement | Defect Type | Defect Description |
|---|---|---|
| *Win condition:* List search results in alphabetical order | Unclear terms / statement or missing information | Unclear by which fields search results should be sorted |
| *Glossary definition:* ISBN is the abbreviation of International Standard Book Number. Every journal has a unique ISBN | Incorrect statement | Journals have an ISSN (unlike books) |
| *Win condition:* User interface should be simple and straightforward | Unverifiable statement | This statement is not measurable and testable |
| *Win Condition:* The administrator is responsible for changing data formats | Ambiguous term | The term administrator is ambiguous, as different types of administrators exist in this context (e.g., sys admin, db admin) |

***Negotiation-Level Defects.*** A challenge of the EWW process lies in understanding a large number of interrelated artifacts and their relation to the overall negotiation purpose. The quality criteria *completeness* and *consistency* relate the negotiation results to a set of expectations (what should be in the negotiation results) and span multiple statements. A typical severe defect is the incomplete coverage of the negotiation purpose statement or the agreed negotiation topics (see Table 2).

**Table 2: Negotiation-Level Defects (Examples).**

| Statement | Defect Type | Defect Description |
|---|---|---|
| *Win condition:* Non-fulltext journals' information will not be in the journal title searchable database *(categorized as a system capability)*. | Incorrect relationship of statement to topic | This is not a system capability but a goal describing the intended use of the system. |
| *Win condition:* Use tool X to build the […] database. | Inconsistency | Conflict with overall project constraint that tool Y should be used. |

# 4 Quality Assurance During Negotiations

One inherent trade-off in a real-world negotiation situation is precision for speed: a major challenge for the facilitator is to watch the trade-off between generating many ideas and delivering consistent high-quality negotiation result. It is also important to keep the participants focused on the right level of detail according to the negotiation purpose in order to elicit as complete and consis-

tent information as possible in a given negotiation situation.

This section discusses QA techniques that are applied before negotiation (Section 4.1) and in the course of a negotiation (Section 4.2).

## 4.1 Negotiation Preparation

The facilitator has the responsibility to ensure that the preconditions of a negotiation are satisfied. It is crucial to develop a statement summarizing the major purpose of the negotiation, context information, and major objectives of the system to be developed.

The second central issue is to get all success-critical stakeholders to attend and contribute. A success-critical stakeholder is any individual whose interests must be accommodated in order for the project to succeed [25]. Success-critical stakeholders are people who can make agreements about the requirements, and make those agreements stick. Typical categories of stakeholders that should be considered success-critical are people designing and developing a system, people interested in system use (e.g., end-users or customers), people having a financial interest, or people responsible for system introduction and maintenance.

There are some additional success factors for selecting stakeholders for a negotiation. They should be empowered and have the official authority or legal power to negotiate agreements. They should be committed to the decisions that are jointly developed. Stakeholders should be representative when serving as a delegate or agent for a team or organization. They should be collaborative and have the willingness and perceptiveness required for developing mutually satisfactory solutions in a team process. Stakeholders should also be knowledgeable and well-informed about the negotiation domain.

## 4.2 Joint Rapid Review

The EWW process basically contains two types of collaborative activities: In some activities stakeholders focus on generating ideas and statements ("diverge"), in other activities stakeholders consolidate the created results ("converge"). During "diverging" activities the primary goal of the process is to produce new content. During "converging" activities the results are consolidated and also checked for defects [5]. This separation avoids confusion from switching the context between production and consolidation/analysis processes.

For local defects on the statement level (see Section 3.2) a simple form of alleviating the effects of imperfect products is to check each individual statement. Beyond such simple checks we have developed joint rapid checking activities to be performed by a team to spot and resolve defects during a negotiation. At certain points in the process all participants step back from the negotiation and check the quality of the products developed so far to eliminate local defects that have been identified. Fixing defects in the process is typically straightforward as it is possible to clarify issues with the author. The process is not an inspection as the participants (and authors) themselves review the products. The list of checking activities is in Table 3 and explained in detail below.

**Table 3: Checking activities in EasyWinWin requirements negotiation.**

| Preceding Step | Checking Focus |
|---|---|
| Review negotiation topics | Context and negotiation topics |
| Capture a glossary of terms | Initial set of win conditions and glossary terms |
| Identify issues and options | WinWinTree |
| Negotiate Agreements | Completeness of negotiation results |
| Negotiate Agreements | Completeness of documentation |

**Check the context and negotiation topics.** The moderator and the stakeholders check the *list of negotiation topics* with respect to the *mission statement* and *statement of negotiation purpose*. The negotiation topics shall be consistent with the scope and boundaries of the system as defined in the mission statement. In addition, they shall sufficiently address the system environment, interfaces; functions, nominal and off-nominal cases; risks, technical and organizational constraints and conditions; system quality issues, level of service, performance; as well as project issues such as budget, time, resources, economic constraints, and conditions.

**Check the initial set of win conditions and glossary terms.** The next check in the process is after converging on win conditions and defining glossary terms. Important conditions to be checked are:

- All *win conditions* express a stakeholder goal and are related to a specific negotiation topic. Further they must be within the scope of the mission statement. If a win condition covers more than one topic, it should be split up.
- Important and possibly *unclear terms* in the win conditions are defined in the glossary.

**Check the WinWinTree.** After the identification of issues, options, and agreements the checks from the preceding stages can be repeated. Rules for describing agreements, risks, and glossary terms include:

- All *win conditions* and *options* are defined as complete, active, and positive sentences.
- An *issue* states a risk, constraint or objection to a win condition, and not a solution or alternative.
- An *option* states a solution or an alternative to overcome an issue.
- An *agreement* is either derived from a *win condition*

that did not raise an issue or from an *option*.

- Important and possibly unclear terms in the issues, options or agreements are defined in the glossary.
- Agreements and potential risks can be traced back to the initial win conditions in the WinWin tree.

**Check the completeness of negotiation results.** At the end of the negotiation agreements and potential risks are organized by negotiation topics. Sorted and reduced results are checked for completeness and comprehensibility.

- *Agreements* express goals that can be more general than requirements.
- *Potential risks* are *open issues* that could not be resolved during the negotiation.
- The *glossary* defines all potentially unclear terms in an unambiguous way (e.g., does not contain undefined terms).

**Check the completeness of documentation.** The completeness of the documentation can be checked for project management and requirements engineering purposes at the end of a negotiation:

- For each negotiation topic there is at least one win condition. Otherwise the list of negotiation topics should be revised or missing win conditions should be identified.
- The set of agreements should provide a useful basis for developing plan, contract, requirements document, etc.

## 5 Inspection Process

In addition to the joint and rapid checks performed by the stakeholders described in the preceding section, this section presents an inspection process following the standard process [9] consisting of the following stages:

1. *Inspection preparation* to check the entry criteria of completeness and sufficient quality for understanding the inspection context.
2. *Individual Reading* supported with reading techniques optimized for negotiation results.
3. *Meeting of inspectors* or some other form of defect collection.
4. *Report and rework* to clarify the issues raised with the author if possible, or to document the problem and the resolution in a traceable way. Possible rework strategies depend on defect severity.

In this section we describe steps 1 and 2 of the process for defect detection, i.e., preparation (Section 5.1) and individual reading (Section 5.2). The main differences to the checks described in Section 4 are that (a) only consolidated results and no temporary work products get checked and (b) the inspectors might have no knowledge of the negotiation process, so traceability in the results and clear definition of important terms are of specific importance. The inspectors are not necessarily system stakeholders, but can be independent expert reviewers whose purpose is to expedite the transition from quickly-determined top-level stakeholder agreements to a more clear, correct, complete, and consistent requirements description with respect to the defined negotiation purpose. Thus the inspection of pre-requirements can also support stakeholders who did not participate in the negotiation and who need to understand the negotiation results.

The inspection approach we have developed adopts a reading technique. Reading is a key activity in defect detection to understand a given software artifact and compare it to a set of expectations regarding structure, content, and desired qualities. *Ad hoc* inspections depend largely on inspector experience and do not facilitate a repeatable process. Defect detection should thus be supported with repeatable reading techniques [1] guiding the inspectors through the inspected document and instruct them what target defect classes to uncover and how to perform quality checking. An example of a general reading technique is a checklist showing all defect types or symptoms to look for in a particular document type, usually independent from a specific notation.

Perspective-based reading [1] exploits different viewpoints like, e.g., users, designers or testers of an inspection object. For each perspective there can be a scenario, which prescribes procedures (e.g., to produce a model like a user manual for the user's view, high-level design sequence diagrams for the designer's view, or test cases for the tester's view). These models can be analyzed to answer questions based on the particular perspective's qualities. The underlying assumption is that the union of perspectives provides extensive coverage of the range of defects present, while each reader is responsible for a narrowly focused view of the document. This is supposed to lead to a more in-depth analysis of potential defects in the document.

### 5.1 Planning and Preparation for Inspection

The inspection leader has to make sure that the material is complete and has sufficient quality on the superficial level to start the inspection: Negotiation results should state agreements with clear responsibilities as well as open issues and how they are going to be resolved. Also the importance and feasibility of the agreements should be available. The negotiation results are then organized and the inspection object with the following outline is prepared:

- *Context and overview*: Mission statement for the negotiation (to assess level of details of negotiation contributions), statement of negotiation purpose, reference material, list of identified stakeholders, their roles, and acronyms;
- A list of *agreements* together with the thread of win conditions, issues, and options leading to each agree-

ment organized by negotiation topics;
- A list of *potential risks*, i.e., unresolved issues sorted by negotiation topics;
- A *glossary* of terms, i.e., definitions for the language used in the project/domain.

Further the inspection manager has to check, whether all success-critical stakeholders participated in the negotiation to ensure commitment to agreements and potential risks. If a stakeholder could not participate, it is crucial to inform him/her about the negotiation and solicit information, which might be missing in the pre-requirements. Candidates for inspectors are either expert reviewers or success-critical stakeholders who were not present at the workshop. Performing an inspection of the pre-requirements by the stakeholder is an excellent preparation for follow-up activities and an opportunity to strengthen stakeholder involvement.

### 5.2 Reading Technique for Management and Technical Perspectives

For individual reading we suggest to use several perspectives to minimize the overlap between the inspectors and to exploit different stakeholder viewpoints [1]. Therefore this should include persons covering the management perspective (e.g., a project manager) as well as the technical focus (e.g., a requirements engineer). The perspectives give general guidelines (relevant for all readers) to check for clarity, and specific content focus to (a) the project manager on project and process management parts and (b) requirements developers on technical content and external understandability.

Each inspector is asked to perform the following defect detection procedure individually and independently. It aims at preparing the inspector with an overview and then continuing with detailed checks for clarity, correctness, completeness, and consistency in consecutive stages. The strategy is to support the inspector with easily digestible steps to avoid cognitive overload.

In the following we present a condensed description of the steps from the operational version of the reading technique:

**Get an overview.** Inspectors are asked to get an overview on the material handed over to them and check it for completeness, i.e., availability of a mission statement, negotiation topics, a list of agreements, a list of potential risks (open issues), and a glossary describing all terms, which may be unclear or have a special meaning in the project (other than usual in general, in the profession, or in the organization).

**Understand the introduction.** In this task inspectors are asked to (a) read the mission statement to understand the main project and negotiation goals and (b) to list important information they would expect to find in the negotiation results. This list of expected information is later compared with actually available information. Furthermore they quickly scan through the lists of agreements and potential risks, and mark the portions of text, which are of particular interest to their point of view (project manager vs. requirements engineer) with respect to the project goals. Inspectors also check the list of negotiation topics for clarity and completeness by comparing it to a standard requirements engineering reference.

**Check for clarity.** Inspectors check each statement and mark potentially unclear/ambiguous terms and statements. They look up unclear terms in the glossary, and report missing definitions. Inspectors further report unclear statements and mark win conditions, belonging to several negotiation topics or listed under the wrong negotiation topic.

**Check for correctness and completeness.** Inspectors check each statement relevant for their perspective and mark incorrect statements. They also check whether each win condition really presents a goal, each issue really is a constraint or objection, and each option really suggests a solution. Furthermore they mark statements, which do not describe a goal (e.g., a design solution) and list missing information by comparing the actual negotiation to the things they expected to find (see above).

**Check for consistency.** In this task inspectors mark statements that contradict each other and any other forms of inconsistency (e.g., win condition vs. glossary definition; errors in the structure of a WinWin tree).

**Check the glossary.** Each entry in the glossary is checked for clarity and correctness of the definition, especially whether unclear terms in the definition are explained appropriately in the glossary.

## 6 Automating the QA Techniques

We have been defining and automating both in-process QA techniques (Section 4) and post-negotiation inspections (Section 5) [5]. The quality assurance techniques discussed in Section 4 are described in the EWW process guide [12], which explains in detail how to prepare a negotiation and facilitate such quality assurance activities in a meeting. The guidebook also shows how groupware tools can be used to support the checking activities.

In order to provide appropriate tool support for the inspection technique we customized the GroupSystems.com GSS suite for defect collection and inspection support. A detailed description is available in [14][15]. The major benefits of using a GSS for inspections are as follows:
- Its flexibility allows customizing it to different inspection process designs (e.g., different reading techniques, perspectives, etc.).
- The inspection process can be continuously monitored, which allows the inspection manager to continuously optimize the process [16].
- It supports both individual defect detection tasks as well as inspection meetings.

- It can be nicely integrated with the EWW methodology.
- It supports different inspector roles, i.e. inspectors performing different defect detection tasks.

## 7 Feasibility Study

This section discusses quantitative results and qualitative experiences we gained in a study carried out to test the usefulness of the reading techniques.

### 7.1 Study Process

We used negotiation results from projects carried out in Fall 2000 at USC and asked ten persons to inspect them. The two selected inspection objects dealt with web-based library systems so that inspector familiarity and domain knowledge could be assumed. Inspection object 1 contained 41 agreements and 27 glossary entries. Inspection object 2 contained 39 agreements and 15 glossary entries. Both EWW negotiation documents followed the structure described in Section 5.1. One additional smaller negotiation document was used during a tutorial explaining the structure of negotiation results and examples of possible defects.

Neither the designers of the inspection technique nor the inspectors were familiar with the selected negotiations before the actual inspection. The inspectors included experienced project managers, researchers, professional developers, and students. All inspectors received a short tutorial on the EWW negotiation process, the overall inspection process, and the reading techniques. During the tutorial the inspectors were asked to practice reading techniques for both stakeholder views. After this dry run we clarified arising questions, assigned a stakeholder perspective to each inspector, and had them proceed with the real inspection. Each inspector inspected two EWW negotiations and completed a feedback questionnaire.

### 7.2 Study Results

Table 4 provides the average number of defects found in the inspection objects. Although the selected documents are relatively small, inspectors identified a considerable number of defects. Because of the small sample size we do not provide sophisticated statistical measures but focus on qualitative results instead.

*Inspector experience.* A qualitative result we expected is that experience with a certain reading perspective has a strong impact on inspection effectiveness. It turns out that detecting defects is more difficult in requirements negotiation results than in requirements specifications or designs where defect types can be defined more precisely. Our results show that experienced inspectors found more true defects and at the same time also needed more time for inspection. Inexperienced inspectors reported difficulties to clearly identify defects and to assess defects severity.

*Defect types.* The most frequent defect type detected by experienced inspectors was missing or unclear information. Experienced inspectors started the inspection with higher expectations (i.e., they were checking if certain important topics were addressed adequately) compared to inexperienced inspectors. An interesting result is however that inexperienced inspectors did a better job in finding inconsistencies.

**Table 4: Number of inspectors (I) per perspective and mean number of normal/major/critical defects found (N, M, C) for each inspection object.**

| *Perspective* | *Inspection Object 1* | | | | *Inspection Object 2* | | | |
|---|---|---|---|---|---|---|---|---|
| | *I* | *N* | *M* | *C* | *I* | *N* | *M* | *C* |
| *Project Manager* | *3* | 6.7 | 2.7 | 3.3 | *3* | 5.3 | 2 | 3.3 |
| *Requirements Engineer* | *7* | 6.6 | 1.6 | 1 | *7* | 6.7 | 0.9 | 0.6 |

*Effort.* The effort required to perform the inspections was relatively low. The mean effort for project managers/requirements engineers for inspection object 1 was 43/76 minutes. The mean effort for project managers / requirements engineers for inspection object 2 was 48/66 minutes. This also confirms our expectations that requirements engineers who had to take a more thorough look at the entire document (as defined in the reading technique) needed considerably more time than project managers.

As the focus of this feasibility study was put upon qualitative feedback we used a questionnaire and interviews to collect feedback data. Main qualitative results regarding the EWW *negotiation results* are:

*Acceptance of inspection objects.* The acceptance of the negotiation document was very high. All inspectors agreed on the usefulness of EWW for negotiating and documenting requirements.

*Varying level of detail.* The main problem reported when inspecting a results document was the varying level of detail in the investigated artifacts. Some aspects are discussed and presented in great detail while other aspects of the project are not touched at all or only on a superficial level. We have meanwhile updated to the EWW facilitation guidelines [12] to address this problem.

*Interdependencies.* Inspectors also reported that they had difficulties in dealing with similar or overlapping agreements, which negatively impacted the readability of the document. We have updated the EWW process guide to better address interdependencies among agreements and other artifacts.

Main experiences regarding the *inspection process* include the following:

*Acceptance of Inspection Technique Effectiveness.* There was a strong consensus among inspectors that the inspection process and the reading technique are suitable for the inspection objects and that they support the detection of defects. Inspectors also agreed that the EWW results were considerably improved through fixing the detected defects.

*Assessment of Defect Severity.* The assessment of defect severity turned out to be difficult due to the broad and large variety of topics addressed. Often fairly general information is included and details are left for further analysis steps, which make it often hard to estimate defect severity.

*Precision.* The defect detection technique to quickly mark all parts of a negotiation that are relevant for a specific stakeholder perspective seems to favor overlooking errors. Participants reported that important information might easily be overlooked.

*Project Size.* Some inspectors argued that the reading technique might be inefficient for small and simple inspection objects (such as the tutorial document) as the separation of different defect types via reading techniques might lead to redundant defect detection tasks thus creating overhead. However, the EasyWinWin results we typically find in practice are quite complex, justifying a well-structured, systematic defect detection approach.

## Conclusion and Further Work

This paper describes collaborative and repeatable quality assurance techniques complementing the EasyWinWin requirements negotiation approach. As requirements negotiation results represent valuable information for all success-critical stakeholders, it is important to ensure that they are of high quality to expedite the development of other crucial life-cycle deliverables like specifications, plans, etc. It is essential to remove defects from requirements negotiation results to increase the chance to finish the project successfully on time and within budget. Therefore we provide a set of structured and well-defined techniques aiming at defect reduction.

In a feasibility study we show that the inspection of requirements negotiation results with reading techniques is feasible and effective, as the inspection reveals numerous defects. Compared to the inspection of other documents [2] the efficiency of pre-requirements inspection is high. We believe that our approach helps to improve communication, trust, and shared vision, where detailed issues and conflicts may of course occur, but in a well-set framework, rather than in an unclear context.

We believe that, although the process is currently optimized for EasyWinWin results, it can also be tailored with little effort to other approaches for capturing pre-requirements.

Further work will focus on using and evaluating the described techniques in industry projects. We also identified potential improvements for the EasyWinWin tools (e.g., improved formatting support). Such relatively simple support would allow a moderator to focus even more on quality assurance activities and would also prevent errors.

Beyond that we will concentrate on extracting valuable information on the importance of system features for project and quality planning from high-quality negotiation results. Project managers could for example select those working products (e.g., specific sections of requirements or design documents) for inspection, which represent major sources of risk in a project. It is also important to apply inspection techniques in order to ensure that working products created during development comply with agreements made earlier during requirements negotiation.

## Acknowledgements

The authors would like to thank all inspectors who participated in the feasibility study especially Andreas Bubla, Patrick Braunsberger, Klaus Brunner, Lorenz Froihofer, Gerhard Chroust, Christoph Hoyer, Franz Luef, Hans Prochaska, Gerwin Szabo, and Stephan Taborsky. Furthermore we would like to thank all stakeholders who participated in the EasyWinWin negotiations selected for the study, i.e., the students and customers of the software engineering class at University of Southern California's Center for Software Engineering. Michael Halling has been supported by the Austrian Science Fund, Grant P-14128-COSIMIS. Stefan Biffl has been funded in part under the Austrian Science Fund grant J-1948-INF.

## 8 References

[1] Basili, V., Green, S., Laitenberger, O., Lanubile, F., Shull, F., Soerumgaard, S., and Zelkowitz, M. The Empirical Investigation of Perspective-Based Reading, *Empirical Software Engineering: An International J.*, vol. 1, no. 2, 1996, pp. 133-164.

[2] Biffl, St., Halling, M. Software Product Improvement with Inspection, *Proc. Euromicro 2000 Conference Software Product and Process Improvement Track, Maastricht,* IEEE Comp. Soc. Press, Sept. 2000.

[3] Boehm, B., Bose, P., Horowitz, E., Lee, M.J. Software Requirements Negotiation and Renegotiation Aids: A Theory-W Based Spiral Approach, *Proc. ICSE'95*, IEEE CS Press, Los Alamitos, Calif., 1995.

[4] Boehm, B., Grünbacher, P., Briggs, B. Developing Groupware for Requirements Negotiation: Lessons Learned, *IEEE Software*, pp. 46-55, May/June 2001.

[5] Briggs, R.O., de Vreede, G.-J., and Nunamaker, J.F. Thinklets: Achieving Predictable Repeatable Patterns of Group Interaction with Group Support Systems (GSS), *Proc. HICSS 2001 (Hawaii Int'l Conf. System Sciences)*, IEEE CS Press, Los Alamitos, Calif., 2001.

[6] Briggs, R.O., Grünbacher, P., EasyWinWin: Managing Complexity in Requirements Negotiation with GSS,

*Proceedings Hawaii International Conference on System Sciences*, IEEE Computer Society, 2002.

[7] Fagan, M. Design and Code Inspections To Reduce Errors In Program Development, *IBM Systems J.*, vol. 15, no. 3, 1976, pp. 182-211.

[8] Genuchten, M., Dijk, C., Scholten, H., Vogel D., "Industrial Experience in Using Group Support Systems for Software Inspections," *IEEE Software*, vol. 18, no. 3, May/June 2001, pp. 60–65.

[9] Gilb, T., Graham, D., *Software Inspection*, Addison-Wesley, 1993.

[10] Gotel, O.C.Z., Finkelstein, A.C.W., An Analysis of the Requirements Traceability Problem. *Proc. 1st Int. Conference on Requirements Eng.*, pp. 94-101, 1994.

[11] Grünbacher, P., Briggs, B. Surfacing Tacit Knowledge in Requirements Negotiation: Experiences using EasyWinWin, *Proc. Hawaii International Conference on System Sciences*, IEEE Computer Society, 2001.

[12] Grünbacher, P., *EasyWinWin OnLine: Moderator's Guidebook, A Methodology for Negotiating Software Requirements*. JKU Linz, USC-CSE 2000, GroupSystems.com.

[13] Halling, M., Biffl, St. Using Reading Techniques to Focus Inspection Performance, *Proc. Euromicro 2001 Conference - Software Product and Process Improvement Track, Warsaw,* IEEE Comp. Soc. Press, Sept. 2001.

[14] Halling, M., Grünbacher P., Biffl St., Tailoring a COTS Group Support System for Software Requirements Inspection, *Proc. 16th IEEE International Conference on Automated Software Engineering*, San Diego 2001.

[15] Halling M., Grünbacher P., Biffl St., Groupware Support for Software Requirements Inspection, *Workshop on Inspection in Software Engineering* at the CAV'01, 2001.

[16] Halling M., Biffl St., Grünbacher P., A Groupware-Supported Inspection Process for Active Inspection Management, *Proc. Euromicro'02 Conference*, Dortmund Germany, IEEE CS 2002.

[17] Kunz, W., and Rittel, H., *Issues as elements of information systems*, Working Report 31, Center for Planning and Development Research, Univ. of California Berkeley, 1970.

[18] Laitenberger, O., *Cost-effective Detection of Software Defects through Perspective-based Inspections*. PhD thesis, University of Kaiserslautern, Germany, May 2000.

[19] Laitenberger, O., and DeBaud, J.-M., An encompassing life cycle centric survey of software inspection, *Journal of Systems and Software*, vol. 50, no. 1, 2000, pp. 5-31.

[20] Lamsweerde, A. van, Goal-Oriented Requirements Engineering: A Guided Tour, *Int. Conf. on Requirements Engineering'01 Tutorial Notes*, 2001.

[21] Lee, J., and Lai, K. Y., *A comparative analysis of design rationale representations*, Technical report, MIT, Center for Coordination Science, CCS 121, 1991.

[22] MacLean, A., Young, R. M., and Moran, T. P., "Design rationale: The argument behind the artifact," *Proc. CHI '1989* Austin, TX.

[23] Potts, C., and Burns, G., "Recording the reasons for design decisions, *Proc. .10th International Conference on Software Engineering*, pp. 418-427, IEEE Comp. Soc. Press, Sept. 1988.

[24] Robinson, W.N., Volkov, V. Supporting the Negotiation Life Cycle. 95-102, *Communications of the ACM*, Vol. 41, 1998.

[25] Sharp, H., Finkelstein, A., Galal, G., Stakeholder Identification in the Requirements Engineering Process, *Proc. 10th Int. Workshop on Database & Expert Systems Applications*, 1998.

[26] Toulmin, S., 1958, *The Uses of Argument*, Cambridge Univ. Press, England.