Lessons Learned from Integrating Specification Templates, Collaborative Workshops, and Peer Reviews

Marko Komssi[1,2], Marjo Kauppinen[1], Kimmo Toro[2], Raimo Soikkeli[3], and Eero Uusitalo[1]

[1] Software Business and Engineering Institute, Aalto University School of Science and Technology, P.O. Box 9210, 02015 TKK, Finland
[2] F-Secure Corporation, Finland
[3] Ilmarinen Mutual Pension Insurance Company, Finland
{Marko.Komssi,Marjo.Kauppinen,Eero.Uusitalo}@tkk.fi,
Kimmo.Toro@f-secure.com, Raimo.Soikkeli@ilmarinen.fi

Abstract. [Context & motivation] Specifying requirements and ensuring their quality are critical for the success of software development projects. A variety of practices have been suggested to manage these activities, including specification templates, collaborative workshops, and peer reviews, but few empirical studies exist on their inter-connection. [Question/problem] We studied the lessons learned from integrating these three practices: "what are the problems faced with the use of the three practices?" and "what kind of approach supports the integrated use of the three practices?" [Principal ideas/results] In the Finnish companies included in the study, the key problems with the use of the three practices seemed to be the following: 1) the use of the three practices was typically inadequately established to meet the needs of the particular projects and 2) the requirements were communicated to key participants late and insufficiently. To avoid these types of problems, it was found useful to use the practices in an integrated manner, so that the forms of collaboration between key participants were determined and the appropriate types of practices were selected and tailored in a project-specific way. [Contribution] The paper describes the success factors of the integration. The setup workshop is introduced to support the tailoring and integration of the practices.

Keywords: Requirements engineering, setup workshop, action research, industrial experience, integration, specification, quality control, best practices.

1 Introduction

Requirements engineering (RE) is a central part of software development, but despite this fact, actual knowledge of the RE process is lacking [1]. A large part of RE research concentrates on methods or techniques supporting a single activity instead of promoting an integrated view of RE and lacks reports on the connections between various good practices [1, 2].

Several best RE practices have been identified that contribute to software project success [1]. However, having the best RE practices in place may not be enough. A best RE practice may include various techniques. The proper selection and tailoring of

R. Wieringa and A. Persson (Eds.): REFSQ 2010, LNCS 6182, pp. 158–172, 2010.

a technique is typically needed in order to find an appropriate fit to a particular software project that has its own characteristics [3, 4]. Unfortunately, most RE literature does not sufficiently describe the suitable target areas or limitations of specific techniques [3]. Moreover, the lack of empirical research in method tailoring is surprising in the applied field of software engineering [4].

According to Katasonov and Sakkinen [2], requirements quality control is mostly a matter of communicating requirements. They emphasize that requirements are not discovered but constructed, and there is often some disagreement between stakeholders about goals. They argue that requirements quality control cannot be seen simply as a mechanical process of checking documents but should instead be studied as a coherent entity. Moreover, to overcome the deficiencies of requirements documents, communication links between stakeholders and requirement owners are needed [5].

This paper focuses on linking three practices of RE – specification templates, collaborative workshops, and peer reviews – to support a coherent view of specifying requirements with quality control. Our industrial research study was based on the following question: what lessons have been learned from integrating these three practices? In order to answer the question, an action research study was conducted in five companies that have applied the practices in their software projects.

Drawing on the lessons learned, this paper presents an approach to integrating the above-mentioned three practices. This approach allows the practices to be tailored to fit the operating environment of a software project and form a coherent whole that enables the practices to overcome each other's deficiencies. A key part of the approach is the setup workshop, where the application and timing of these practices is analyzed and decided upon collaboratively.

The paper is organized as follows. Section 2 presents the related work from the three practices and Section 3 explains the research design. Section 4 presents the problems faced with the use of the three practices. Section 5 presents the success factors of the integration ands proposes the setup workshop as a crucial component of the integration. Finally, Section 6 concludes the findings with suggestions for future research.

2 Related Work

It is important first to find out the strengths and weaknesses of individual practices in order to understand how these practices can work together. In this section, the previously reported characteristics of specification templates, collaborative workshops, and peer reviews are presented.

2.1 Specification Templates

The use of specification templates is a common practice that supports, in particular, the early phase of specification activity in software projects. Several specification templates, such as IEEE Recommended Practice for Software Requirements Specifications [6] and the Volere Requirements Specification Template [7], are available to provide a comprehensive structure for documentation. These templates contain predefined sections with instructions and examples as the basis for document writing.

Successful teams have frequently transformed specification templates and examples from previous projects into rational RE activities [1]. In this study, the specification templates were identified as one of the best practices in RE. Specification templates have also been identified as one of the top ten RE guidelines and recommend their implementation in all organizations [8].

Although the use of specification templates is accepted as one of the best RE practices and most of the software companies apply specification templates in their software projects, the literature seems to provide surprisingly few empirical results from the use of specification templates in software projects. The benefits and strengths of specification templates are mainly introduced in RE books. The specification templates are intended to act as a guide to essential content and to help requirements analysts determine what belongs in the specification [7, 9]. The specification templates should contribute to documents that have a higher quality and lower cost [8]. Furthermore, the use of specification templates promotes consistent communication and helps software practitioners ask questions that they might otherwise ignore [10]. Peer reviewers can also capitalize on software templates that relate to reviewing work [8].

A few disadvantages and weaknesses of specification templates have been presented. One limitation is that they are not scalable for all types of projects. In other words, the templates often lack the flexibility that is required to respond to variability in the intended readership of a document [11]. Another drawback with specification templates is that they can lead to the production of specifications that are superficially attractive but limited in their content [12].

Some suggestions have been made about ways to improve the usefulness of specification templates. For example, it has been proposed that a suitable specification template with embedded guidance texts should be defined for each project type [10] or, similarly, that software practitioners should predefine a set of situation types and suggest an appropriate requirements document style for each type [13]. In addition, in order to develop a useful structure for a specification template, the existing documents of the organization should be investigated and ideas should be collected from document users [8].

2.2 Collaborative Workshops

The use of collaborative workshops seems to be a less common RE practice in software projects than the use of specification templates or peer reviews [1]. Collaborative workshops can vary from informal to formal and from short to long and can be used for several purposes to support different RE activities. In particular, a collaborative workshop can provide an efficient, controlled, and dynamic setting where the participants can quickly elicit, prioritize, and agree on a set of project requirements [14].

One formal approach to specifying requirements is the collaborative construction of the entire requirements document. For example, the RaPiD7 collaborative document authoring technique has been developed at Nokia [15]. In this technique, a document is created by a team in consecutive workshops, which reduces the risk of the document's content being based primarily on the judgment of the author. The RaPiD7 technique is supposed to enhance the communication and commitment of a project team and improve document quality. The reason for these advantages is the early involvement of the team in documentation work. RaPiD7 speeds up the document creation process in terms of calendar time.

RaPiD7 is similar to a more widespread method called Joint Application Development (JAD), which was developed at IBM in 1977 [16]. JAD enables technical and business specialists to learn about each other's domain knowledge, improves communication among interested parties, facilitates consensus management, and increases user acceptance of specifications [17].

One challenge these two techniques present is the common time required for workshop meetings. For example, a JAD procedure typically lasts for three to five days [18] and stakeholders have difficulty allocating common time. In fact, it was necessary to adapt the JAD technique in some organizations because the staff were sometimes unable or unwilling to commit to full-time participation in JAD workshops [19].

Cockburn [20] proposes a more informal process for a collaborative workshop, in which people work in a full group when there is a need to align or brainstorm and use the rest of their time in pairs or alone. Cockburn explains that a group is able to brainstorm and reach a consensus effectively, but when the group is split, more text is produced.

2.3 Peer Reviews

Peer reviews are a core practice of requirements quality control [2, 21]. A peer review consists of someone other than the author of a document examining it in order to discover defects and identify improvement opportunities [22]. Eventually, a software development organization may need to acquire deeper knowledge from the types, formalities, and feasibilities of peer reviews. In particular, mature software development organizations are advised to develop capability to determine what types of peer reviews are conducted and to tailor the peer reviews in the organizations' software projects [21]. Wiegers has defined several review techniques and types, both formal and informal [23], which are listed below using his definitions.

The most formal technique of a review, *inspection*, has several characteristics that distinguish it from other review techniques. For example, a trained moderator leads meetings and co-operates with a trained team. The moderator defines the goals, collects quality data, and distributes results using a reporting process.

A *team review* is slightly more informal and imprecise than an inspection. Team reviews concentrate more on detecting defects than preventing them. A team review may be chosen if no trained inspection leaders are available.

In a *walkthrough*, the author of the document explains it to colleagues and asks for their feedback. This review type is generally informal and does not involve data collection and reporting. However, the process steps and the role of each participant may be clearly defined.

In a *passaround*, the author of the document sends it to several colleagues and gathers their feedback. The passaround technique is useful, for instance, for obtaining ideas and corrections for a new project plan.

In a *peer deskcheck*, only one checker examines the document. While this review technique requires the smallest amount of resources, it is only appropriate for products that do not have very high quality expectations or are not to be reused.

In an *ad hoc review*, the author of the program presents a problematic part of the design to a fellow worker and asks for help. Although quite informal, this review type is useful for short and tricky cases.

A team can identify the strengths and weaknesses of the review types [23]. The purpose of this is to select the proper review type for each case with regard to the organizational culture, time constraints, and business objectives. In particular, the team is advised to select the least expensive review type that fulfills the objectives of the review [22].

3 Research Design

The goal of this study is to present lessons learned from the integration of specification templates, collaborative workshops, and peer reviews. The study was conducted using an action research approach in five Finnish companies. The data were collected from the case study companies during a period of ten years (1999 to 2009) and analyzed iteratively in three phases.

3.1 Research Approach

In order to gain a deep understanding of the three practices and their integration, we applied an action research approach. This research method was selected for two reasons: it has a unique ability to link research to practice, and as a qualitative method, it is also effective for explaining what is happening in a company [24]. The action research approach allows researchers to address complex real-life matters and study selected issues in detail [25]. Additionally, an "industry-as-laboratory" research approach, where researchers identify problems through close involvement with industrial projects and create and evaluate practices addressing the problem, is suggested in [26]. This lets researchers emphasize what people actually do or can do in practice, rather than what is possible in principle.

To access insider and historical data, as well as to engage practitioners in research, we also applied the insider action research approach [27]. In the insider action research approach, some of the researchers are internal members of practitioner organizations. As internal members of the organization, practitioner-researchers have the opportunity to collect data that are richer than what they would collect as external researchers. Gummesson [28] points out that a lot of information is stored in the minds of practitioners, who have often undergone central and dramatic changes. Therefore, Gummesson urges practitioners to act as researchers and reflect on what they had learned retrospectively.

3.2 Case Study Companies

Our research was conducted in five Finnish companies, which are introduced in Table 1. Three of the companies were of medium size, one was small, and one was large. Companies A, C, and E are internationally known and have a significant global market share in their fields. These three companies focus mainly on solutions developed for a large number of customers. Company B provides pension insurance

Table 1. Case study companies

Company	Number of employees	Application domain
A	700	Computer and information security solutions for companies and consumers
B	600	Earnings-related pension insurance services
C	500	Information management systems for buildings, public infrastructure, and energy distribution designers
D	50	Language technologies and services for companies and consumers
E	24000	Transportation systems and services for buildings

services in Finland, and Company D offers language technology solutions, mainly to large companies.

Table 2 summarizes the practices the case study companies have applied. Each of the companies defined at least one specification template for requirements. In order to shorten the requirements documents, some of the organizations opted for two templates – one for high-level requirements and one for more detailed requirements. We were able to gather data related to collaborative workshops from two companies, even though they were practiced in all of the companies to some extent. In particular, Company B had a long tradition of using workshops to define requirements collaboratively. All of the case study companies had applied some kind of peer reviews, with the most common being team reviews.

Table 2. Practices investigated in each company

	A	B	C	D	E
Specification templates	X	X	X	X	X
Collaborative Workshops	X	X			
Peer Reviews	X	X	X	X	X

3.3 Data Collection and Analysis

Our study was based on the following question: “what are the lessons learned from the integration of these three practices?“ Applying the “industry-as-laboratory” research approach [26], we divided the question into two more specific questions as follows: “what are the problems faced with the use of the three practices?” and “what kind of approach supports the integrated use of the three practices?” Figure 1 illustrates the three main phases of the study: 1) identification of problems faced in software projects and development of the integration approach, in Company A, 2) retrospective analysis of the problems faced in the five companies and refinement of the integration approach, and 3) validation of findings and refinement of the integration approach, in Company B.

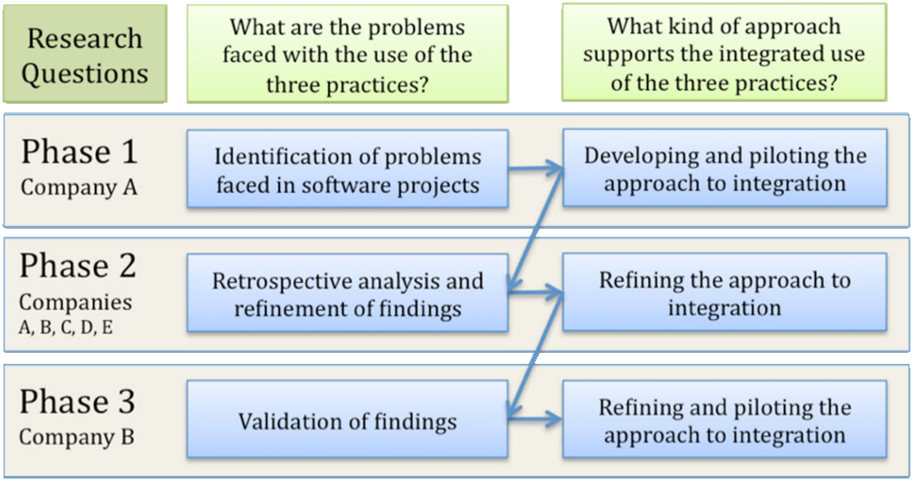
Research Questions
What are the problems faced with the use of the three practices?
What kind of approach supports the integrated use of the three practices?
Phase 1
Company A
Identification of problems faced in software projects
Developing and piloting the approach to integration
Phase 2
Companies A, B, C, D, E
Retrospective analysis and refinement of findings
Refining the approach to integration
Phase 3
Company B
Validation of findings
Refining and piloting the approach to integration

Fig. 1. Three research phases of the study

Phase 1 was performed in Company A between the years 2003 and 2006. Perceived problems of eleven software projects were first identified, and the related improvement ideas were collected and analyzed. Subsequently, a preliminary approach to the integration of the three RE practices was developed and piloted iteratively.

The goal of Phase 2 was to compare the preliminary results gained from Company A during Phase 1 with the experiences from the other four companies (B, C, D, and E). In this phase, retrospective analysis was used to examine previously collected data. The data had been collected in three ways. First, four of the authors had worked in one or two of the case study companies (A, B, D, and E), participating in software development projects and requirements process improvement work. Second, the authors conducted two research projects with the case study companies during 1999-2005. Within these research projects, data from Companies A, C, and E was gathered. Third, the authors interviewed a person who had been in charge of specification templates and peer reviews in Company A ten years ago. Based on the analysis, the authors refined the findings related to the problems faced with the use of the three practices and reflected on the approach to integration.

Phase 3 was conducted with Company B from 2006 to 2009. The company's goal was to develop a consistent yet tailorable set of RE practices that could be applied company-wide. This was accomplished by a series of 12 workshops and 19 meetings, where the findings of the previous phases were built on. The result of these activities was an approach that enables an organization to tailor and integrate specification templates, collaborative workshops, and peer reviews into a coherent entity. The approach was piloted during its development in 9 software projects. In addition, the company organized two training sessions, in which 28 project managers, requirements specialists, and group leaders participated. After the training, we asked for participants' comments on how suitable they perceive the approach as being for the types of projects they typically participate in. We used both a feedback form and group discussion to collect the data. In this phase, the collected data were analyzed and the findings were clustered into the previous findings iteratively. The findings were validated and new findings were merged with them. The final findings are described as the lessons learned. These lessons are described in the following sections.

Table 3 summarizes the data collection activities performed in the case study companies. The results of this study are based on the data collected through observations, formal semi-structured interviews, informal conversations, the analysis of requirements specification templates, the analysis of requirements documents, and questionnaires.

Table 3. Data collection activities of the study

	A	B	C	D	E
Observation	X	X	X	X	X
Interviews	X	X	X		X
Informal conversations	X	X	X	X	X
Analysis of specification templates	X	X	X	X	X
Analysis of requirements documents	X	X	X	X	X
Questionnaires	X		X		X

3.4 Threats to Validity

We apply the explanations of Yin [29] to construct and external validity. In our study, a threat to construct validity is the possibility that we were not able to correctly collect and evaluate the problems related to the use of the three practices and the benefits and success factors from applying the integration approach. As a result, our inferences concluded as lessons learned might not represent reality, in the companies. The threat to external validity is the possibility that lessons we have learned cannot be generalized to other software development organizations.

To reduce the threat to construct validity of the study, we used of multiple sources of evidence and triangulation. We used a number of information sources and data collection techniques. We applied triangulation of data sources and data collection techniques by utilizing interviews, informal conversations, participant observation, and document analysis. In addition, the study covers a long period of time that improves the construct validity of our findings, as it was possible to analyze and validate the findings at different times. Finally, key informants from two companies reviewed our findings several times.

To reduce the threat to external validity of the research results, the study involved five separate case study organizations of different characteristics, such as size, solutions, and business environments. The integration approach was developed and piloted in two companies that have very different types of software development and business environments and solutions.

4 Problems Faced with the Use of the Three RE Practices

The use of specification templates fosters individual specification work. The application of specification templates was quite common in the companies. Software developers who used a specification template typically specified a document alone, which tended to lead to the writing of overly comprehensive specifications. Software developers often spent a lot of time on the writing phase when using a specification template. A significant problem was that they focused too much on writing requirements with high volatility. In addition, specification templates were occasionally found to cause superficially attractive specifications that satisfied guidelines but included irrelevant content. For example, interviews and informal conversations with practitioners in a case study company revealed to us that they, as document readers, had often met and disregarded long documents whose readership was unclear. The problem was identified as a result of an individual specifying work that was based on a specification template.

The timing of a peer review is typically too late. Several reviewers were dissatisfied with the fact that participating in a review of the requirements document was often their first involvement in the development of a software system or product. The peer review usually took place once the writer felt the document was, to all intents and purposes, finished. However, the late timing of reviews was often harmful for both the writer and the reviewers. Writers were often opposed to suggestions for major changes at that point and the reviewers' interest in suggesting changes for the document seemed to decrease as well. One company employee said, "Typically, the content of the document is so refined that I do not even dare to raise any issues in the review." In practice, documents that were too polished reduced reviewers' motivation to suggest changes, even if the reviewers had discovered several weaknesses in the documents' contents.

The difference between collaborative workshops and peer reviews is not clear in practice. Even if peer reviews were originally introduced to discover defects in the documents, the practitioners of three of the case study companies gradually began to use peer reviews more for sharing information and solving problems. The reasoning for this is as follows: For some key stakeholders, such as software testers and technical support personnel, a late peer review was often the first exposure to the project's requirements. Hence, the needs of these stakeholders were not aligned with the original intent of discovering defects, but rather gaining understanding or expressing their own views on the requirements. On the other hand, some collaborative workshops that the project teams held in the late phases of specifying requirements included aspects of peer reviews; in these projects, the teams skipped the actual peer reviews. Hence, the purpose of the two activities changed or became vague. The main difference seemed to be the timing. The term "collaborative workshop" was often used in the early phase and the term "peer review" in the late phase of specifying requirements.

The purpose and content of collaborative workshops and peer reviews are poorly communicated. Practitioners in one case study company were applying peer reviews for three different purposes, depending on the degree of completion of the documents. The purposes were to obtain comments from the domain experts, to share information with several stakeholders, and to achieve acceptance of the document. However, the fact that the practitioners referred to each of the three different types of sessions as 'reviews' caused confusion. In particular, the reviewers were irritated when the meeting was different than they had expected or wanted. In another company, reviews were part of the development process and the participants typically perceived the reviews as little more than a rubber stamp at the end of the software development procedure. On the other hand, collaborative workshops were not a defined practice in the case study companies. Undefined workshops were applied in one company and this was identified as a reason for the frustration of the participants. In particular, the goals of the workshops were not communicated and this meant that the participants had expectations of the course of action and outcomes of workshops that differed from those of the facilitators.

5 Integrating the Three RE Practices

In an attempt to mitigate the problems faced with the use of the three practices, we developed an approach for integrating these practices. In this section, we present the lessons we have learned from integrating the three practices. First, three success factors of the integration are presented. Second, we introduce the setup workshop as the means to perform the actual integration.

5.1 Success Factors of the Integration

Teamwork. We identified teamwork as the first success factor of the integration. By definition, teamwork produces synergy as people with different skills work together towards a common goal. Team members create shared meanings and arrange this knowledge into common frames. We found out that teamwork was essential for the identification of: 1) the project-specific needs of the requirements specification; 2) the project-specific needs for co-operation and communication; 3) the key information streams (informants and targets) in different stages of the project, and 4) the types of practices that are beneficial for the project. As a consequence of teamwork, the decisions were based on an extensive knowledge base and debate, they were far more willingly accepted by group members, and the reasons behind them were commonly understood.

Correct timing of the RE practices. The second success factor of the integration is the correct timing of the RE practices, so that each practice is used at the right phase of the project – maybe a seemingly trivial task, but not so. As an example, one specialist commented that the utilization of the specification template too early restricts the creative work and collaboration too much. The selection of the proper sequence of the practices also involves planning when and how the different stakeholders are to be engaged in specifying requirements. Without this kind of planning, the first involvement of some key stakeholders was often in peer reviews, late in the phase of specifying requirements, causing frustration on the part of the stakeholders and extra development costs.

Tailoring the RE practices. Tailoring the selected practices was crucial, as none of them was found to be optimal as such. Here, tailoring means a purposeful activity in which a practice or practices are adapted to fit a particular purpose. We found out that two types of tailoring were used. First, the case study organizations performed company-level tailoring of the practices. The most typical type of company-level tailoring was to define the structures and scrutinize examples of different types of specification templates by adapting a standard version like the IEEE Recommended Practice for Software Requirements Specifications [6]. Second, project-specific tailoring was needed.

Purposeful, project-specific tailoring seemed to require at least one skilled person to take ownership of the tailoring. In particular, tailoring the three practices as integrated required an understanding of their overlapping characteristics. Increased knowledge about the variety of practices and their strengths and weaknesses seemed to help the team to choose the appropriate types of practices and adapt them to their context. When software project teams participated in collaborative workshops to a greater degree, specification templates were less necessary for guiding the writing work and peer reviews for information sharing.

As a result of the tailoring, the variability of the collaborative workshops increased. The goals and required participants of the collaborative workshops varied. Hence, it was considered crucial to choose and communicate the type, procedure, and goal of each collaborative workshop in order to avoid unnecessary participants and false expectations.

5.2 Setup Workshop as a Crucial Component of the Integration

A setup workshop is a collaborative and facilitated workshop used for planning and communicating how to utilize the three practices in the creation of requirements specification in software projects. The setup workshop was originally developed and piloted in Company A, and later refined and piloted in Company B. The setup workshop seems to be a key component for integrating the three practices. The current version of the setup workshop is presented in the following and illustrated in Figure 2.

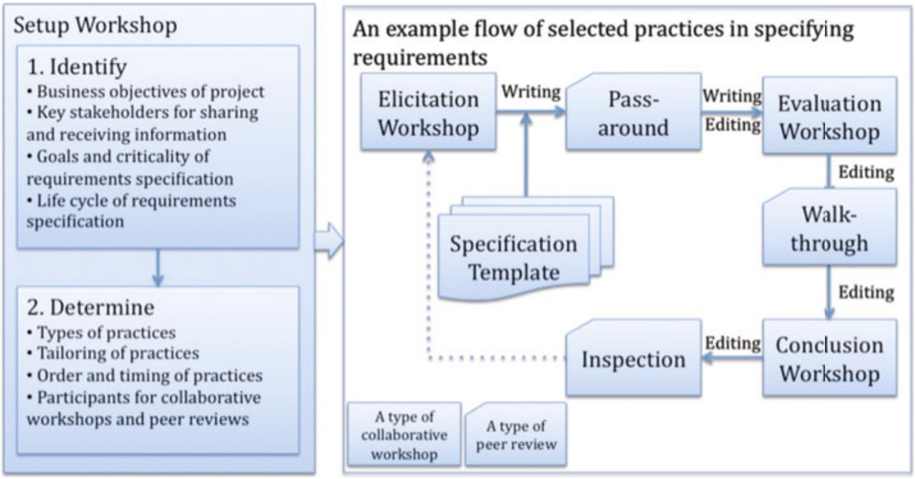
Setup Workshop
1. Identify
• Business objectives of project
• Key stakeholders for sharing and receiving information
• Goals and criticality of requirements specification
• Life cycle of requirements specification
2. Determine
• Types of practices
• Tailoring of practices
• Order and timing of practices
• Participants for collaborative workshops and peer reviews
An example flow of selected practices in specifying requirements
Elicitation Workshop
Writing
Pass-around
Writing
Editing
Evaluation Workshop
Editing
Walk-through
Editing
Conclusion Workshop
Editing
Inspection
Specification Template
A type of collaborative workshop
A type of peer review

Fig. 2. Using a setup workshop for planning the integrated deployment of the three RE practices

The setup workshop is performed in the early stages of specifying requirements. In the setup workshop, the facilitator enables the participants to identify or clarify the following:

- the business objectives of software project,
- the key and minor informants of requirements in different stages of the project,
- the project-specific goals of the requirements specification and the needs of its readers,
- the criticality of the requirements specification and the role of oral communication, and
- the life-cycle of the requirements specification and the estimated volatility of requirements.

Immediately after identifying the above factors, the facilitator supports the participants in determining the types of practices, the ways they are tailored, and their sequence. At the same time, they determine the participants for the planned collaborative workshops and peer reviews.

The benefits of this approach were obvious. First, the readers of the requirements specification considered that this approach improved the relevance of specifications. Second, from setup workshop participants' point of view, proper planning and communication of the goals of each collaborative workshop and peer review was considered to reduce false expectations and frustration on the part of participants and to promote early information sharing between stakeholders. Overall, 20 out of 28 practitioners who participate in specifying requirements for information systems in Company B perceived the idea of the setup workshop as very useful for their work. The remaining eight practitioners perceived it as useful. Remarkably, all three team leaders and seven out of eight project managers found the setup workshop very useful.

A setup workshop was found useful, at least, in software projects where a requirements specification had an important role. An informant commented that the use of the three practices in an intertwined way was also found beneficial in certain software projects in Company B, even without applying the setup workshop as such. This implies that the integration of the practices can be successfully performed in several ways and the setup workshop is just one approach to perform it.

During our study, Company A was transformed from a traditional software organization to an agile one, which changed the company's software development culture. Moreover, the software developers increasingly opposed documentation and peer reviews. Consequently, the use of requirements specification was reduced and peer reviews nearly discontinued. As a result of these changes, we were not able to fully apply setup workshops as initially piloted; oral communication took the key role as the main information channel. Interestingly, software developers appeared to apply collaborative workshops informally. Even though the applicability of setup workshops as such can be limited in agile software organizations, we, in fact, suggest their use as a tool to help agile software development teams towards efficient information sharing between cross-functional teams.

6 Discussion and Conclusions

Although specification templates, collaborative workshops, and peer reviews have all been recommended, they are typically treated as independent RE practices in the literature. Our findings indicate that the independence of the practices leads to several problems in practice. The use of specification templates often leads to individual specifying work, resulting in relatively long documents. Peer reviews are typically performed too late and reviewers are not motivated to contribute.

The integration of the three practices was identified as a rational way to reduce such problems. As each of the three practices has strengths and weaknesses that partly overlap, using the practices in a tailored and intertwined way helps the team to reduce the negative influences of their weaknesses. As a means to perform integration, our findings suggest using the setup workshop for planning the workflow of specifying requirements, identifying project-specific ways to collaborate, and selecting and tailoring the appropriate types of practices. The use of a setup workshop can improve the applicability of specification templates and promote early information sharing between stakeholders. Proper planning and communication of the goals of each collaborative workshop and peer review should reduce false expectations and frustration on the part of participants.

The proposed integration approach includes limitations and challenges when adopting it in a software organization. Software developers may not easily adopt the proposed integration approach, if they already oppose meetings, documentation, and peer reviews. In addition, the tailoring of the RE practices, as a key element of the integration, requires more RE skills than the use of standardized practices. A software organization needs to consider whether they have the necessary skills or willingness to acquire them. Furthermore, the integration approach will increase the variability of the used RE practices in the software organization. The work practices and requirement specifications in different software projects will become less comparable. Consequently, a process owner and management may find it more difficult to observe the progress and quality of software projects.

The significance of our findings is to be confirmed in future studies. The role of certain authors as active participants in the companies may have affected the construct validity of the results. It should be noted that the development of the integration approach partly occurred as everyday work in two companies and was not solely organized to support the research purpose. Furthermore, we were able to apply the integration approach only in two out of the five companies. This weakens the external validity of the findings. New studies in several organizations are needed to evaluate whether the integration approach really addresses to the problems faced with use of the three RE practices.

While this paper presented the three RE practices as integrated, in future, it would also be worth studying how to integrate larger sets of RE practices that are adopted for use in software companies. Indeed, evaluating the usefulness of the setup workshop for planning and tailoring the entire RE process of a software project appears to be a promising idea.

References

1. Hofmann, H.F., Lehner, F.: Requirements Engineering as a Success Factor in Software Projects. IEEE Software 18(4), 58–66 (2001)
2. Katasonov, A., Sakkinen, M.: Requirements Quality Control: a Unifying Framework. Requirements Engineering 11(1), 42–57 (2006)
3. Tsumaki, T., Tamai, T.: Framework for Matching Requirements Elicitation Techniques to Project Characteristics. Software Process Improvement and Practice 11(5), 505–519 (2006)
4. Fitzgerald, B., Russo, N.L., O'Kane, T.: Software Development Method Tailoring at Motorola. Communications of the ACM 46(4), 64–70 (2003)
5. Uusitalo, E.J., Komssi, M., Kauppinen, M., Davis, A.M.: Linking Requirements and Testing in Practice. In: Proceedings of the 16th IEEE International Requirements Engineering Conference, pp. 265–270. IEEE CS Press, Barcelona (2008)
6. IEEE Recommended Practice for Software Requirements Specifications (IEEE Std-830), pp. 207–244 (1998)
7. Robertson, S., Robertson, J.: Mastering the Requirements Process, 2nd edn. Addison-Wesley, Boston (2006)
8. Sommerville, I., Sawyer, P.: Requirements Engineering: A Good Practice Guide. Wiley, Chichester (1997)
9. Davis, A.M.: Just Enough Requirements Management: Where Software Development Meets Marketing. Dorset House Publishing, New York (2005)
10. Wiegers, K.E.: Software Requirements, 2nd edn. Microsoft Press, Redmond (2003)
11. Brockmann, R.J.: Where Has the Template Tradition in Computer Documentation Led Us? In: Proceedings of the 2nd Annual International Conference on Systems Documentation, pp. 16–18. ACM, Seattle (1983)
12. Kaner, C., Bach, J., Pettichord, B.: Lessons Learned from Software Testing–A Context-driven Approach. Wiley, New York (2002)
13. Power, N., Moynihan, T.: A Theory of Requirements Documentation Situated in Practice? In: Proceedings of the 21st Annual International Conference on Documentation, pp. 86–92. ACM, San Francisco (2003)
14. Gottesdiener, E.: Requirements by Collaboration: Getting It Right the First Time. IEEE Software 20(2), 52–55 (2003)
15. Kylmäkoski, R.: Efficient Authoring of Software Documentation Using RaPiD7. In: Proceedings of the 25th International Conference on Software Engineering, pp. 255–261. IEEE Computer Society Press, Portland (2003)
16. Carmel, E., Whitaker, R.D., George, J.F.: PD and Joint Application Design: a Transatlantic Comparison. Communications of the ACM 36(6), 40–48 (1993)
17. Purvis, R., Sambamurthy, V.: An Examination of Designer and User Perceptions of JAD and the Traditional IS Design Methodology. Information & Management 32(3), 123–135 (1997)
18. Wood, J., Silver, D.: Joint Application Development, 2nd edn. Wiley, New York (1995)
19. Davidson, E.J.: Joint Application Design (JAD) in Practice: The Journal of Systems and Software 45(3), 215–223 (1999)
20. Cockburn, A.: Writing Effective Use Cases. Addison-Wesley, Upper Saddle River (2001)
21. Chrissis, M.B., Konrad, M., Shrum, S.: CMMI: Guidelines for Process Integration and Product Improvement. Addison-Wesley, Boston (2003)
22. Wiegers, K.E.: Peer Reviews in Software: A Practical Guide. Addison-Wesley, Massachusetts (2001)

23. Wiegers, K.: When Two Eyes Aren't Enough. Software Development 9(10), 58–61 (2001)
24. Avison, D., Lau, F., Myers, M.D., Nielsen, P.A.: Action Research. Communications of the ACM 42(1), 94–97 (1999)
25. Avison, D.E., Baskerville, R., Myers, M.D.: Controlling Action Research Projects. Information Technology & People 14(1), 28–45 (2001)
26. Potts, C.: Software-Engineering Research Revisited. IEEE Software 10(5), 19–28 (1993)
27. Coghlan, D.: Insider Action Research Projects: Implications for Practising Managers. Management Learning 32(49), 49–60 (2001)
28. Gummesson, E.: Qualitative Methods in Management Research, 2nd edn. Sage Publications Inc., Thousand Oaks (2000)
29. Yin, R.K.: Case Study Research – Design and Methods, 3rd edn. Sage Publications Inc., Thousand Oaks (2003)