Collaborative Requirements Elicitation in a European Research Project

Guglielmo De Angelis
CNR–IASI/ISTI, Italy
guglielmo.deangelis@iasi.cnr.it

Alessio Ferrari
CNR–ISTI, Italy
alessio.ferrari@isti.cnr.it

Stefania Gnesi
CNR–ISTI, Italy
stefania.gnesi@isti.cnr.it

Andrea Polini
University of Camerino, Italy
andrea.polini@unicam.it

ABSTRACT

A relevant part of the research activities performed in European computer science institutions is funded through European Union (EU) projects. Eliciting and defining requirements for a software system in a distributed environment with heterogeneous stakeholders is generally challenging. In EU projects partners can have different objectives and views, needs are not sharply defined and communication is hampered both by differences in native spoken languages and by the physical distance of the stakeholders.This paper presents the definition and the results of applying an innovative requirements elicitation and refinement approach in the context of an EU financed project (Learn PAd). The approach combines the KJ-method and collaborative wiki-based requirement sessions to come to a set of consolidated requirements. The paper includes the lessons learnt from this experience, also it discusses both the advantages, and the drawbacks of the instantiated approach.

Categories and Subject Descriptors

D.2.1 [**Software Engineering**]: Requirements/Specifications—*Methodologies*

Keywords

Elicitation, Experience Report, Wiki, KJ Method

1. INTRODUCTION

Since 1984 the European Commission (EC) finances research and innovation within the European Union (EU) through the "*Framework Programmes for Research and Technological Development*" (FP). In 2014 the 8th FP edition, named Horizon 2020, has been launched, and it will last until 2020.

From our experience, the most critical activity of a EU collaborative project actually relates to the *communication*, and to the elicitation of a clear and agreed set of requirements for the software to be developed and possibly prototyped. From this perspective, EU projects show issues that are common to global requirements engineering [7], such as communication difficulties, knowledge management, physical distance and cultural diversity, as illustrated in several research works (see, e.g., [29, 15, 10]). Among the factors that negatively influence the emergence of clear requirements, a key aspect of EU projects is the research and innovation context that asks for the definition and development of a software system never developed before. In such a context, when requirements and communication issues are not properly handled since the beginning, the result generally is a large project scattered in few small projects with subgroups of partners targeting different research and innovation objectives. Correspondingly complex integration issues, and a clear waste of resources, will emerge when the complete system will have to be delivered.

With some difference, all the authors have a quite long experience of work within several EU projects in the area of information and communication technologies (ICT), in different roles and with different responsibilities, and starting from Feb. 2014 they are collaborating within a EU financed project named Learn PAd (http://www.learnpad.eu). The paper describes the approach that has been defined and applied within the project, in order to reduce the risks related to communication issues during requirements elicitation. In particular, we describe how requirements workshops have been organized by adapting the KJ-method [16, 30], and how requirements have been refined and consolidated through the usage of a collaborative wiki infrastructure. The contribution of this work is threefold: (1) we highlight practical requirements elicitation issues in the context of a EU project, which can be considered a special case in the class of global software projects; (2) we address the demand for experience reports and lessons learnt in the usage of a wiki-based platform in global requirements engineering [29]; (3) we present a requirements process that combines *virtual* and *physical* communication among stakeholders in a global project.

The rest of the paper is structured as follows. Sect. 2 describes the complexity dimensions of EU projects. Then Sect. 3 outlines the applied process, while the following three sections describe the activities of planning and requirements elicitation (Sect. 4), collaborative refinement (Sect. 5), and requirements consolidation (Sect. 6). Sect 7 discusses the lessons learnt in the experience, and Sect. 8 reports relevant

[redacted]

related works. Finally Sect. 9 provides conclusions and final remarks.

2. COMPLEXITY DIMENSIONS

The overall goal of the Learn PAd project is to improve the quality of service of the Public Administration (PA) by providing a platform for process-driven, model-based learning (for details please visit: `http://www.learnpad.eu`). The very general idea of the project is to use the Business Process Model and Notation (BPMN) [14] to *teach* civil servants how the PA procedures shall be implemented in practice, and to complement the BPMN models with wiki documents that provide details about the procedures. Wiki documents can be accessed and modified by the civil servants, according to their daily experience. The envisioned platform includes also components to provide procedural learning through model simulation, components for learners' evaluation, and functionalities for *quality* evaluation of models and wiki documents.

Before presenting the approach for requirements elicitation that has been adopted in the context of the project, it is useful to list the peculiar characteristics of EU projects that we have taken into account while defining the approach. The following is probably a non exhaustive list of such characteristics:

number/distribution: EU projects involve a high number of partners physically distributed over Europe (at least). The number of partners could make a plenary discussion on the project needs rather ineffective. At the same time the distribution over many different countries limits the possibility of face-to-face meetings and introduces the need for managing remote requirements elicitation.

language/culture: in EU project partners come from different countries with different native speaking languages. As in other distributed projects [23, 10], this cultural difference might let emerge different attitudes and personalities, which can be interpreted differently by the partners. Clearly, this is a quite relevant problem since no mediator is foreseen within the consortium.

industrial vs academic mindsets: EU project consortia typically include partners with different working mindsets such as for instance industrial and academic. Industrial partners have daily experience with requirements elicitation for software development. Academic partners are normally more focused on theoretical aspects, and might be less rigorous and effective while performing practical requirements tasks. However, the combination of the different mindsets can enable the development of novel and interesting ideas [27].

background: in EU projects partners with different technical backgrounds have to strongly cooperate since the expected innovation or research results could be based on the intermix of their respective competences. This implies different backgrounds and vocabularies, and communication might be hampered by both the previous knowledge of the partners, and by the different terminology used.

objectives: partners can have different objectives and views on the project innovation and research directions to be taken. This is a real issue since requirements could be introduced in order to pursue partner specific interests. This can obviously negatively affect the innovation and research potential of the project and introduce complex effects for requirements management activities.

3. PROCESS OVERVIEW

The process for requirements elicitation applied in Learn PAd consists of three main phases aiming to mitigate the possible negative effects caused by the complexity dimensions discussed in the previous section. Fig. 1 details the different phases. Here, we will give a brief overview of each phase, while we will refer to Fig. 1 along the paper to describe to the specific steps in each phase.

Elicitation Workshops are the first collaborative phase of the process (see Sect. 4). This phase was performed according to a collaborative elicitation technique named KJ-method [16]. Such workshops have been planned to include both individual reflection and collaborative work activities. As suggested in [6], it is extremely important to have preliminary face-to-face meetings at the beginning of a globally distributed project, to mitigate undesirable partners heterogeneity issues and background effects.

Successively, the **Collaborative Refinement** phase (see Sect. 5) permitted to reflect and put in place a strategy to continue the work on requirements already started in the workshops, and to enable the cooperation among participants when they would have been back to their offices. In Learn PAd, a collaborative platform, based on a wiki infrastructure has been deployed and customized for the purpose. This peer-based collaborative refinement step was also oriented to encourage participation: ice-breaking is normally easier when a computer interface acts as a communication mediator [3].

The last phase of the process refers to **Requirements Consolidation** (see Sect. 6). Activities in this phase intended to homogenize collected requirements both with reference to the usage of words and syntactical structures. A restricted group of participants was then designated to form a core team to synchronously discuss the various requirements in a set of conference calls.

4. ELICITATION WORKSHOPS

In collaborative software projects, in which participants are distributed among different locations, there are not many opportunities to meet in face to face meetings. The first meeting for such a kind of projects is clearly a fundamental and critical step. It is important that, from the beginning of the project, members aim at deriving a common understanding on the software system that has to be developed, otherwise the project risks that contributors will work for a while targetting different objectives.

In Learn PAd, it has been decided to devote the first day meeting for project and partners presentations, and to permit to meeting participants to get in touch with each other. In the second day, instead, a couple of workshop sessions, lasting two hours each, were organized to let the partners define preliminary requirements.

4.1 Groups Definition

The 24 participants to the first Learn PAd meeting were organized in three different working groups each one focusing on a specific high-level view of the Learn PAd platform, namely "Modelling", "Quality" and "Learners Evaluation" (**Groups Definition** in Fig. 1). The assignment of the people to working groups was carefully planned and took into account expertises, and the possible involvement of the participant to the various WPs of the project. Clearly the objective is to define groups in which the discussion will be

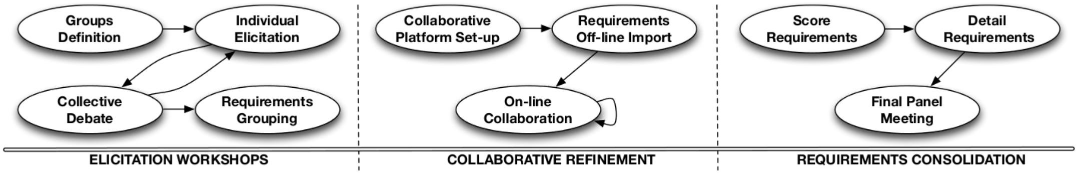

Figure 1: Process Outline

live and fruitful. Therefore it is useful to have in the same group people that could potentially have different, if not contrasting, opinions. More specifically, the partners were classified according to four categories: partners from academia; industrial partners practising open-source; closed-source industrial partners; and PA partners bringing their expertises on the demonstrators. Further considerations about the size, and the heterogeneity of the working groups led to the definition of the following criteria (even though it was not always possible to strictly abide by such rules): (a) each group should have counted around eight participants; (b) a project partner should have had representatives in at least two different groups.

Defined classifications and criteria permitted to have groups in which participants had good expertise on the topic assigned to the group. At the same time differences in their background and working contexts could let emerge different visions, and make discussions more fruitful.

4.2 Workshops – KJ Sessions

The activities of each working group proceeded according to the KJ method [16, 30]. This is an exploratory creativity technique [26], were requirements are first individually written in cards, and then collectively discussed and grouped. For each working group one moderator was appointed to drive the discussion and to involve all participants. All moderators participated to the planning of the workshops. In the first workshop session, according to the guidelines reported in [30], each moderator illustrated to the group members how the activity of the group had to proceed and the expected outcomes. Successively, each participant had the opportunity to write in post-it cards the requirements that he/she considered relevant for the subject of his/her group (one requirement for each post-it). This phase appears in Fig. 1 with the name **Individual Elicitation**. Collected requirements were discussed one by one by the group to understand their relevance and elicit novel ideas (**Collective Debate**).

In the second session, participants were asked to provide additional requirements in the cards, according to the discussion performed in the first workshop. Then, the moderator collected the post-its, briefly asked for further discussions on novel requirements, and put them on a blackboard. With the support of the participants, the requirements were grouped (**Requirements Grouping**) according to their *topic* (e.g., “Meta-modelling”, “Models Quality”, “Cooperation”, and other project-specific terms). The topics were decided along the discussion.

At the end of the second workshop session a wrap-up activity was included in the agenda. During the wrap-up session the vision of each group was presented to the other groups. The activities on requirements carried on during the first meeting permitted to identify 249 preliminary requirements (78 for the “Modelling” group, 81 for “Quality”, 90 for “Learners Evaluation”), which would have been the starting point for the successive collaborative activities. Notably each group defined a similar number of requirements confirming somehow the good splitting and relevance of the three aspects.

4.3 Observations

In general, all the sessions were successful and all group members were quite involved in the activity, permitting to suppose that the negative effects of some heterogeneity issue were rather limited. Besides the careful grouping of the participants, the success of the experience was mainly due to two factors: (1) the design of the workshops; (2) the moderators. Starting the workshops with an individual activity (i.e., card writing) and letting each participant present his/her point of view before the brainstorming was paramount in limiting age/role effects and mitigating objective discrepancies. Then, the partitioning in two workshop sessions let the participants have the time to re-think about their views. Indeed, in the second sessions, the participants experienced a sort of “second chance” to align the terminology with the others, and smooth out cultural and background differences. The smoothing of these aspects was also enabled by the choice of the moderators, and by their attitude. The moderators of the “Modelling” and “Learners Evaluation” groups were the scientific and technical leaders of the project, respectively, and they were familiar with almost all the participants. Therefore, they knew how to deal with them, and they could leverage their role to resolve conflicts. Conversely, the moderator of the “Quality” group, was unknown to most of the participants, and acted as an independent facilitator with the task of reducing relational problems thanks to his independent profile [25].

5. COLLABORATIVE REFINEMENT

At the end of the KJ-sessions, requirements were a set of post-its with associated topics. Our goal was to provide a refinement of these preliminary requirements, and continue the collaborative discussion and elicitation process in a virtual, distributed space, after that all partners went back to their offices. To this end, we have decided to set-up a collaborative wiki-based infrastructure for requirements management.

5.1 Collaborative Infrastructure

Wikis are a lightweight approach to produce documentation more powerful than plain office suites or collaborative tools, and easier to use and tailor than proprietary RE tools [11]. Moreover, wikis are regarded as promising tools for requirements elicitation/negotiation in distributed environments (see, e.g. [34, 29]). The adoption of a wiki in RE enables the various members of the project to contribute by adding, modifying, or deleting contents. In addition, a wiki platform natively supports the versioning of the handled documents. In this sense, contributors can always access to the history a requirement had, and they can trace its evolution. In our context, we have decided to use XWiki (http://www.xwiki.org/), an open-source and general-purpose collaborative platform (embedding a wiki). The platform, besides the collaborative editing capabilities typical of wikis, exposes a flexible data model (i.e., Classes, Properties, and Objects), which can be both queried and extended by the users. Indeed, though wikis facilitate collaborative work, customization is required to address RE-specific needs: a template requirements structure has to be defined, and requirements views have to be enabled to ease navigation. To this end, XWiki has been extended with a model of requirements that is based on the widely used VOLERE [33] template (**Collaborative Platform Set-up** in Fig. 1). In practice, each requirement was associated to a XWiki page with several VOLERE-like predefined fields such as type (e.g., Functional, Non-Functional), status (e.g. Proposed, Accepted, Discarded), description, justification, relations to other requirements (e.g. dependency, refinement and conflicts among requirements), semantic tags, *etc.*. Moreover, dynamic views for the requirements have been defined. Each view focuses on a specific aspect, for example: supporting the navigation of the requirements by some field (e.g. the initiator, the type, etc.); or showing the semantic dependencies among requirements tagged with the same meta-information. In addition, we provided views to query the data model and return analytics about the collaborative activities performed on each requirement.

5.2 Requirements Off-line Import

The results of the elicitation process pursued by means of the workshops have been imported into the collaborative infrastructure (**Requirements Off-line Import**). During this step, the moderators took care of uploading the preliminary requirements of the post-its into the platform, associating each requirement to the topic that was chosen during the KJ-sessions. The topic was stored as semantic tag in the VOLERE-like template. Moreover, to enable a uniform tagging, the moderators met in a set of sessions aiming to tag the requirements elicited by each thematic group with the categorizations proposed by the other groups. Requirements that could not be semantically categorised with a single tag were tagged with multiple tags of the set. This work permitted to all the other contributors more easily search and compare requirements on the base of semantic categorizations.

In addition, for each requirement, the moderators evaluated which work-packages of the project could be affected by that specific concern. This information was codified in the structured template in XWiki by adding the names of the WPs as items in the tags list field of the template. Interestingly we noted that the proportion among the number of requirements assigned to each WP was in line with the amount of work foreseen for each workpackge in terms of man months. In some sense this was considered a rough and preliminary assessment of a good distribution of effort among the WPs.

5.3 On-line Collaboration

Once the offline import activity was completed, we asked the whole consortium to access the XWiki platform in order to improve the specification of the requirements as originally proposed (**On-line Collaboration**). Indeed, the granularity of the information collected was minimal with respect to the VOLERE-like structure adopted for the requirements, and the sentences provided in the post-its were often unclear and with a free-form structure that made them hardly usable in an official requirements document. The moderators also provided a *glossary* to be followed by the contributors when modifying their requirements.

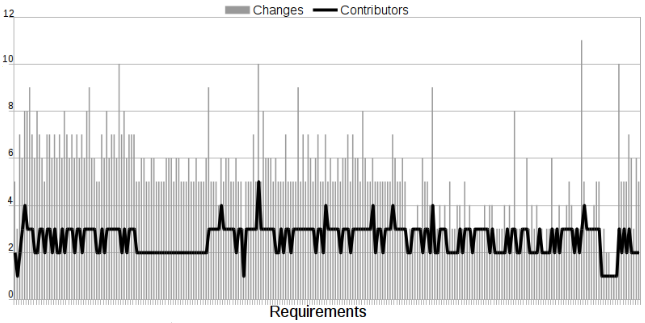

Figure 2: Analytics after collaborative refinement

In a first iteration, we asked the members of the consortium to improve and detail the information associated to each requirement. In particular, we asked to revise the text of the requirement according to a set of basic requirements editing guidelines, such as: use the verbs “shall” or “should” to indicate mandatory and optional requirements, respectively; use active sentences; the subject of the requirement should be the system, part of it, or an actor; *etc.*. Successively, the consortium collaboratively revised again the whole set of requirements. The goal of this second iteration was editing the “justification” field in order to provide detailed information about the rationale of each specific requirement. The final iteration foreseen aimed at identifying some existing relations among the requirements. Specifically, the members of the consortium were asked again to access the views given by the XWiki platform, and browse the requirements looking for dependencies, conflicts, or refinements.

It is important to remark that, at any iteration, any member of the consortium had the possibility of both accessing, and modifying *any* of the collected requirements. During each iteration, members were also encouraged to add requirements. At the end of the collaborative refinement phase, the set of requirements increased from 249 to 337.

5.4 Observations

A set of specific views has been developed in order to get analytic data about the behaviour of the different editors during the collaborative refinement process. Among the others, a view provided the total number of modifications each requirement experienced. The result of this indicator showed that the average number of changes on a given requirement is 5.14 (variance 3.01). Another view aimed at counting the number of different contributors who edited each requirement. In this case, we have an average of 2.58 editors per requirement (variance 0.39). Fig. 2 plots the number of versions, and the number of different contributors for each requirement.

From these data emerges that in average a requirement has been modified more than the number of iterations planned by the process. The participants to the RE process actually perceived some need in order to progressively evolve the set of requirements, for example by aligning the ideas expressed in a requirement according to the current status of the whole set. This observation becomes even more evident if we analyze only those requirements having a number of modifications greater than the number of iterations we requested. In this case, the average of changes applied on a requirement rises to more than 6. It is worth noticing that this sub-set counts more than the 70% of the collected requirements. Thus, the collaborative refinement approach can be considered successful at least in stimulating *further reasoning* on the requirements. The remaining 30% includes both requirements that have been modified only according to the instructions given (i.e., almost the 7.5%), and requirements resulted with less modifications than the number of planned iterations (22.5%).

A further comment about the data from Fig. 2 concerns the number of contributors per requirement (dark line in the plot). In this case, although having 2.58 editors per requirement in average might show a rather good degree of collaboration, from a more fine-grained analysis we saw that only four out of 25 participants edited the majority of the requirements. It is worth noting that three of these participants are the moderators of the KJ sessions. If we exclude these four major collaborators – and the requirements touched solely by them, namely the 39, 3% over the whole requirement set – the average number of editors per requirement decreases to 1.04. This result suggests that the largest majority of the editors tended not to modify the requirements initially promoted by other participants.

Our interpretation of this result comes from the fact that research projects are mainly organized as peer-based projects with a clear separation of competences. Thus, while the initial promoter of a requirement is somehow mostly available in order to reconsider and to let evolve his/her conception about some aspects of the project (i.e. expressed by a requirement), at the same time he/she may be fill *uncomfortable* in a direct modification of the contribution of some other promoter.

6. REQUIREMENTS CONSOLIDATION

After the refinement task, a core team of key partners led by the coordinator of the requirements activities (referred in the following as "RE coordinator") performed a manual analysis of the requirements in XWiki. The goal of this task was to come to a final set of consolidated requirements. Indeed, though the collaborative refinement experience was rather lively, many requirements still needed both syntactic adjustments – to adhere to the guidelines provided – and semantic refinements – to clarify their content. The analysis started with a requirements scoring activity, in which each requirement was associated to a score by the RE coordinator (**Score Requirements** in Fig. 1): (a) ACCEPT, in case the requirement is clear and the RE coordinator evaluates the implementation effort as acceptable; (b) PROVISIONALLY ACCEPT, in case the RE coordinator cannot evaluate the implementation effort; (c) DETAILS REQUIRED, in case the requirement is unclear; (d) REJECT in case the implementation effort is not acceptable or the requirement is equivalent to others. Then, each requirement that was scored with (c) was sent back to the stakeholder who originally produced the requirement to provide further details (**Detail Requirements**). In a **Final Panel Meeting**, the core team discussed each requirement and marked it as (a) or (d). The set of consolidated and accepted requirements consists of 191 unique elements.

ACCEPT	PROV. ACCEPT	DETAILS REQ.	REJECT	Total	**Total (ACCEPT)**
110	144	55	28	337	**191**

Table 1: Requirements per score given by the RE manager.

6.1 Observations

Table 1 shows the number of requirements associated to each score, and the final number of accepted requirements. We see that, among the requirements marked with (c) and (b), only 81 (i.e., 41%) have been finally accepted. This implies that, for many requirements, the implementation effort was considered not acceptable. This result is however tolerable. Indeed, KJ sessions are a creative space that can naturally lead to a large number of ideas that are hardly applicable. However, we conjecture that limiting such ideas would hamper also the emerging of those solutions that, in the end, result to be more interesting.

7. LESSONS LEARNT

Overall the requirement related activities were judged quite successful by the consortium partners. In the following we report our perception on what really worked or did not work in the applied approach to global requirements engineering, trying to identify the rationale behind the observed results, and providing suggestions for possible improvements.

Shared understanding: the content of the wiki is accessible to all the participants. This enables a better understanding of the needs of all the stakeholders involved in requirements refinement. Indeed, as observed in Sect. 5.4, refinements were often provided to align the initial ideas to the status of the whole project, taking into account the requirements of other stakeholders.

Motivation: the lively atmosphere of the small elicitation groups facilitates motivation. In the refinement phase, hiding behind the peer-based wiki motivates everyone to contribute and provide novel ideas. Indeed, as shown in Sect. 5.3, during on-line collaboration, the number of requirements increased consistently (about 35% additional requirements).

Collaboration: collaboration was effective during the face-to-face meetings, where people were called to discuss the requirements of the others, but collaboration was less effective during refinement. Moderators had a key role in the face-to-face meetings. Therefore, we argue that the role of the moderators as *facilitators* [8] have to be enforced also during the refinement activities, to foster collaboration and also to improve the quality of the requirements.

Uniformity: the possibility of defining requirements templates (VOLERE templates in our project) allows structuring the requirements in a uniform manner. On the other hand, we have seen that, to have a uniform structure at the level of the natural language requirements, different tools are needed. In particular, defining a glossary already at requirements elicitation stage can help having fixed reference concepts and a shared vocabulary.

Control: the possibility to perform tagging and data analytics through the wiki enables greater control over the set of requirements, and over the refinement process itself. In particular, the possibility to monitor single contributions helps in adjusting the process of requirements refinement during its execution. The following observations highlight the peculiar aspects of our wiki infrastructure, which enable greater control on the requirements process.

Semantics Links: the navigation across the requirements is enabled by a collection of tags implemented using the data model of the XWiki platform. Each tag is intended to classify a collection of similar requirements. A set of predefined tags was given, but users were able to formulate new ones. The flexibility of the XWiki platform supports the definition of scripts that can dynamically query the pool of requirements in order to classify/reclassify the information. For example, in such a way contributors do not have to remember to explicitly cross-link the pages were requirements are displayed. It is worth to note that requirements relationships such as dependency, refinement and conflicts were elements of the VOLERE template, and could be navigated like the other semantic links.

Traceability: given the flexibility of the XWiki platform, different views can be implemented as queries over the requirements pool. For example, we inspected the requirements by type, tags, contributors, modifications, as well as we elaborated statistics on them. Also the data model of XWiki supports the dynamic modification on the requirements templates. In this sense new fields can be added, for example in order to express emerging links with other specifications. In this case, views can implement traces of the requirements on such specifications.

History: wikis support pages versioning. In our experience each requirement is attached to a page for its display. Each modification to a requirement results as a modification on a page that the wiki will store as a passed revision. In this sense, the wiki automatically supports to trace the evolution of a requirement.

Navigation: wikis tend to grow in an unstructured manner, and tend to be chaotic and hard to navigate. If one wishes to have an abstract view of the content, this is hard to achieve in current wikis. This issue has been addressed by providing views in XWiki, which have helped both the requirements contributors and the requirements managers to have a clear understanding.

8. RELATED WORKS

The literature counts several papers describing challenges [10, 6, 5] experiences [8, 9, 2], and tools [4, 22, 29, 11] in the field of global requirements engineering. In particular, some practice-based works (see, e.g., [10, 8, 6]) stress the importance of communication difficulties and knowledge management in distributed teams, which can lead to poor shared understanding of the problem domain, and to obstacles in decision making. The challenges outlined in these works brought to the definition and application of techniques for collaborative requirements elicitation (e.g., [12, 19]) and distributed requirements engineering (e.g., [6, 23, 3, 21]). Tools have been also developed to support requirements elicitation in global teams, both based on social network-based systems [4, 22], online forums [19, 21], and wikis [29, 17, 11, 32, 34, 1, 31].

One of the main contributions of this paper resides in the usage of a collaborative/wiki-based tool to support RE activities in a global requirements enginering context. Similarly to other technologies (e.g. SOP-Wiki [31, 11], WikiReq [1], and Softwiki [32]), our wiki-based approach relies on semantic links among requirements. Moreover, our approach structures the requirements according to an object-oriented data model which does not reflect the page structure of the wiki (i.e., the data presentation layer). The consequence is a more flexible approach in elaborating groups and views over the collected requirements. This feature contributes to the final goal of achieving a shared understanding and improve structured participation, besides easing data analysis and control.

With respect to the previous papers concerning wiki-based approaches for RE, the main contribution of the current work resides in the *evidence* provided regarding the usage of a wiki in requirements elicitation and refinement. Indeed, as highlighted in [29], very little contribution exists in the literature concerning practical experiences associated to the usage of such wikis.

Another central contribution of this paper is the description of a practical experience in employing a workshop technique [13] for collaborative requirements elicitation [24]. Our approach provides only slight adaptation of the canonical KJ-method [16], and it would be beyond our scope to compare such method with other approaches presented in the literature (e.g. [20][28]). However, it is worth to highlight that, with respect to other elicitation techniques, the main advantage of the KJ method resides in its *simplicity*, which drove our choice of applying it for collaborative requirements elicitation.

9. CONCLUSIONS AND FUTURE WORK

In this paper, we have presented the combination of a face-to-face requirements elicitation technique (i.e., the KJ method) with a wiki-based approach for distributed requirements elicitation/refinement in the context of a EU research project. We have seen that these projects show issues that are common to global requirements engineering (e.g., communication difficulties, knowledge management, see [10, 6]) and that can be addressed with tools that are suggested by previous experiences (i.e., initial face-to-face meetings [6] combined with asynchronous wiki interaction [11]). However, we have also seen peculiarities that need specific solutions: among them, the need to enforce the role of moderators also during on-line activities, to foster collaboration, and the need to define a preliminary glossary already during the face-to-face meetings, to improve the uniformity of the future requirements. Advanced techniques to distributed requirements clarification (e.g., [18]) are also foreseen to increase the quality of the elicited requirements.

Part of our observations stem from quantitative analyses performed through XWiki. However, our lesson learnt are mainly qualitative, and based on our perception of the strength and weaknesses of the approach, compared to our experiences in other EU projects, where requirements were defined by a selected group of participants. In depth quantitative analysis and comparison with these previous projects are foreseen as future research.

Acknowledgements

This paper describes work undertaken in the context of the European Project FP7 STREP 619583: Learn PAd.

10. REFERENCES

[1] L. Abeti, P. Ciancarini, and R. Moretti. Wiki-based requirements management for business process reengineering. In *WIKIS4SE'09*, pages 14–24. IEEE, 2009.

[2] C. Bird, N. Nagappan, P. Devanbu, H. Gall, and B. Murphy. Does distributed development affect software quality?: an empirical case study of windows vista. *Communications of the ACM*, 52(8):85–93, 2009.

[3] F. Calefato, D. Damian, and F. Lanubile. Computer-mediated communication to support distributed requirements elicitations and negotiations tasks. *Empir. Soft. Eng.*, 17(6):640–674, 2012.

[4] C. Castro-Herrera, J. Cleland-Huang, and B. Mobasher. Enhancing stakeholder profiles to improve recommendations in online requirements elicitation. In *RE'09*, pages 37–46. IEEE, 2009.

[5] B. H. Cheng and J. M. Atlee. Research directions in requirements engineering. In *FSE'07*, pages 285–303. IEEE CS, 2007.

[6] D. Damian. Stakeholders in global requirements engineering: Lessons learned from practice. *Software, IEEE*, 24(2):21–27, 2007.

[7] D. Damian and D. Moitra. Guest editors' introduction: Global software development: How far have we come? *Software, IEEE*, 23(5):17–19, 2006.

[8] D. E. Damian, A. Eberlein, M. L. Shaw, and B. R. Gaines. An exploratory study of facilitation in distributed requirements engineering. *Requir. Eng.*, 8(1):23–41, 2003.

[9] D. E. Damian and D. Zowghi. The impact of stakeholders' geographical distribution on managing requirements in a multi-site organization. In *RE'02*, pages 319–328. IEEE, 2002.

[10] D. E. Damian and D. Zowghi. RE challenges in multi-site software development organisations. *Requir. Eng.*, 8(3):149–160, 2003.

[11] B. Decker, E. Ras, J. Rech, P. Jaubert, and M. Rieth. Wiki-based stakeholder participation in requirements engineering. *IEEE Software*, 24(2):28–35, 2007.

[12] M. Geisser and T. Hildenbrand. A method for collaborative requirements elicitation and decision-supported requirements analysis. In *Advanced software engineering*, pages 108–122. Springer, 2006.

[13] E. Gottesdiener. *Requirements by collaboration: workshops for defining needs.* Addison-Wesley Professional, 2002.

[14] O. M. Group. Business Process Modeling Notation (BPMN) version 2.0., 2011.

[15] J. D. Herbsleb. Global software engineering: The future of socio-technical coordination. In *FSE'07*, pages 188–198. IEEE Computer Society, 2007.

[16] J. Kawakita. The KJ method: a scientific approach to problem solving. Technical report, Kawakita research institute, Tokyo, 1975.

[17] E. Knauss, O. Brill, I. Kitzmann, and T. Flohr. Smartwiki: Support for high-quality requirements engineering in a collaborative setting. In *WIKIS4SE'09*, pages 25–35. IEEE, 2009.

[18] E. Knauss, D. Damian, J. Cleland-Huang, and R. Helms. Patterns of continuous requirements clarification. *Requir. Eng.*, pages 1–21, 2014.

[19] H. Lai and Y. Ni. Evaluating a lightweight forum-based tool: Empirical studies on requirements elicitation process. *Journal of Software*, 9(12):2989–2997, 2014.

[20] V. Laporti, M. R. Borges, and V. Braganholo. Athena: A collaborative approach to requirements elicitation. *Computers in Industry*, 60(6):367–380, 2009.

[21] P. Laurent and J. Cleland-Huang. Lessons learned from open source projects for facilitating online requirements processes. In M. Glinz and P. Heymans, editors, *REFSQ*, volume 5512 of *LNCS*, pages 240–255. Springer, 2009.

[22] S. L. Lim, D. Damian, and A. Finkelstein. Stakesource2. 0: using social networks of stakeholders to identify and prioritise requirements. In *ICSE'11*, pages 1022–1024. ACM, 2011.

[23] W. J. Lloyd, M. B. Rosson, and J. D. Arthur. Effectiveness of elicitation techniques in distributed requirements engineering. In *RE'02*, pages 311–318. IEEE, 2002.

[24] L. Macaulay. Requirements capture as a cooperative activity. In *RE'93*, pages 174–181. IEEE, 1993.

[25] L. A. Macaulay. Seven-layer model of the role of the facilitator in requirements engineering. *Requir. Eng.*, 4(1):38–59, 1999.

[26] N. Maiden, S. Jones, K. Karlsen, R. Neill, K. Zachos, and A. Milne. Requirements engineering as creative problem solving: A research agenda for idea finding. In *RE'10*, pages 57–66. IEEE, 2010.

[27] N. Maiden, J. Lockerbie, K. Zachos, A. Bertolino, G. De Angelis, and F. Lonetti. A requirements-led approach for specifying qos-aware service choreographies: An experience report. In *Proc. of REFSQ*, volume 8396 of *LNCS*, pages 239–253, Essen, Germany, Apr. 2014. Springer.

[28] L. Mich, C. Anesi, and D. M. Berry. Applying a pragmatics-based creativity-fostering technique to requirements elicitation. *Requir. Eng.*, 10(4):262–275, 2005.

[29] R. Peng and H. Lai. DRE-specific Wikis for Distributed Requirements Engineering: A Review. In *APSEC*, pages 116–126, 2012.

[30] K. Pohl. *Requirements engineering: fundamentals, principles, and techniques.* Springer Publishing Company, Incorporated, 2010.

[31] E. Ras. Investigating wikis for software engineering-results of two case studies. In *WIKIS4SE'09*, pages 47–55. IEEE, 2009.

[32] T. Riechert and T. Berger. Leveraging semantic data wikis for distributed requirements elicitation. In *WIKIS4SE'09*, pages 7–13. IEEE, 2009.

[33] S. Robertson and J. Robertson. *Mastering the Requirements Process (2Nd Edition).* Addison-Wesley Professional, 2006.

[34] D. Yang, D. Wu, S. Koolmanojwong, A. W. Brown, and B. W. Boehm. WikiWinWin: A wiki based system for collaborative requirements negotiation. *HICSS'08*, 0:24, 2008.