How can Open Source Software Development help Requirements Management gain the potential of Open Innovation: An Exploratory Study

Krzysztof Wnuk, Dietmar Pfahl
Department of Computer Science
Lund University
Box 118, SE-221 00 Lund
Sweden
+46 46 222 45 17
Krzysztof.Wnuk@cs.lth.se,
Dietmar.Pfahl@cs.lth.se

David Callele
Department of Computer Science
University of Saskatchewan
111-116 Research Drive
+ 1 (306) 966-2073
callele@cs.usask.ca

Even-André Karlsson
Add a Lot, Sweden
even-andre.karlsson@addalot.se

ABSTRACT

A key component in successfully managing software products is to properly, and in a timely manner, identify and secure competitive advantage by innovation via feature differentiation. Although open source software (OSS) is not a new idea, several product development companies that operate in a market-driven context have started to use open source solutions as core software components in their products. Adopting open source core components implies a lower degree of control over software development and increased business risk associated with integrating differentiating contributions into the core release stream. Whether and how to adjust the current requirements management practices after the adoption of OSS components to fully benefit from the concept of open innovation has not yet been empirically explored. We outline experiences and challenges related to leveraging open innovation via engaging in OSS identified during 19 interviews with practitioners occupying different roles in the requirements management process at a large company followed by four validation interviews with other practitioners. We then propose a research agenda for requirements and decision management in the open innovation context and suggest which challenges in requirements engineering open innovation affects.

Categories and Subject Descriptors

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

General Terms

Documentation, Management

Keywords

Case study, open innovation, open source software, software requirements management,

1. INTRODUCTION

The principal goal of commercial software companies is to generate profit by actively exploiting viable business opportunities. Achieving and sustaining competitive advantage is becoming more and more challenging due to frequent (and uncontrollable) technology changes [1], shifting market needs and ubiquitous globalization of software production [2]. From the perspective of maximizing *return on investment* (ROI), software companies should focus on identifying and implementing the most profitable functionality, a goal that is strongly correlated with identifying requirements that fulfill customers' needs [1]. Unfortunately, this relationship appears to be underestimated: during the software engineering process, requirements are not necessarily ranked or prioritized from the perspective of value creation; rather, every requirement is often considered equally important [3].

Requirements engineering practice is predicated, in part, upon the belief that successful software product definition is inevitably related to accurate identification and implementation of customer needs, needs represented as various types of requirements [4]. Traditional requirements identification has focused on internal (from within the company) stakeholder interaction – analysis, research and development activities followed. This approach, called *closed innovation* by Chesbrough [5] is threatened by the need to adapt to a world of rapid change and fierce market competition. As noted by Chesbrough [5], companies need to "learn how to play poker as well as chess" with their innovation processes. Searching for, experimenting with, and ultimately using externally generated innovation can be a good short-term provider of growth opportunities [6], extending the lifecycle of existing products.

Companies can no longer rely only upon their internal resources to source innovation. Instead, they may be forced to look to the surrounding environment. Chesbrough [5] identifies the act of identifying and adopting externally generated innovations as *open innovation*, arguing that it may lead to new sources of technology and growth (open innovation is not open source). The open innovation approach supports both the adoption of externally acquired innovation and the active commercialization of internally generated innovations that are not aligned with the current business model (*e.g.* via licensing or sale). This approach attempts to address issues exemplified by the well-known "PARC Problem" experienced by the Xerox company [5]; the inability to

assess and capture value for (technology) innovations that were not directly related to Xerox products. A similar problem exists in some requirements engineering contexts (e.g,. market-driven requirements engineering) where not all requirements can be implemented [14], [15] in the products, potentially *wasting* those ideas that could not be commercialized due to, for example, a lack of alignment with the overall product portfolio of a company or limited resources.

The open innovation paradigm has been addressed by several researchers including explorations from the business management perspective [5], in terms of corporate venturing and valuation using a real options model [6], technology transactions [7] and finally to determine if open innovation is unique to large companies in so-called 'high-technology' industries [16]. However, to the best of our knowledge no study has attempted to investigate the role of open innovation in the requirements management process or to identify possible new challenges and process adjustments that the introduction of open innovation may enforce. As there appears to be a strong relationship between requirements engineering, the value creation process [1] and the innovation process, this study investigates open innovation in the requirements engineering context, simultaneously exploring "requirements management to sustain innovation", one of the future research topics identified by Kaupinnen [8].

The open innovation paradigm appears to be of interest to innovation management in large organizations [5], particularly in the context of maintaining a competitive position. In this paper, we investigate open innovation in a large organization that recently transitioned to an open innovation model by abandoning the development of a purely proprietary code base for their software product and making use of an *open source software* (OSS) project or code base (referred to here as a "platform") as a source of innovation (in both knowledge and technology). The code base is the main component of the company's products – embedded systems developed for the global market. The code base is also non-exclusive – it is also available to, and used by, competitors to the case company. In this scenario, each organization develops differentiating features based upon the common open source platform.

To investigate whether and how much the adoption of OSS components in the context of requirements management helps companies take advantage of the open innovation paradigm, we investigated the strengths and weaknesses of the requirements engineering practices in our case company. We formulated the following research question:

> *RQ1: Is the current requirements process in the case company designed to facilitate from open innovation?*

To tackle our research question, we conducted an exploratory interview study at a large-company that develops embedded systems for a global market using OSS components. In 19 interviews we explored challenges related to adjusting their current requirements management and decision-making processes to better benefit from the open innovation paradigm. The findings from 19 in-depth interviews were discussed, validated and complemented by four additional interviews at the case company. In particular, we focus in this study on requirements management and decision making processes [17]. The paper identifies research opportunities in creating effective contribution strategies as well as revisiting current prioritization and release planning methods to better benefit from open innovation. Furthermore, we analyzed the approach to innovation at the case company and identify which challenges in requirements engineering are impacted by open innovation and how the impact occurs.

In the remainder of this paper we present background and related work in Section 2; in Section 3 we outline the methodology and the case company context. Section 4 discusses the validity of the study. Section 5 presents the results of the study. The results are discussed in Section 6 and the paper is concluded in Section 7.

2. BACKGROUND AND RELATED WORK

Open innovation was defined by Chesbrough [5] as "the use of purposive inflows and outflows of knowledge to accelerate internal innovation, and expand the markets for external use of innovation, respectively." Lichtenthaler [7] proposed this refinement: "An open innovation approach refers to systematically relying on a firm's dynamic capabilities of internally and externally carrying out the major technology management tasks, *i.e.*, technology acquisition and technology exploitation, along the innovation process. Thus, open innovation processes involve a wide range of internal and external technology sources, and a wide range of internal and external technology commercialization channels." Both Chesbrough and Lichtenthaler set their work in the context of medium to large organizations, organizations that perceive innovation as one means (among many) to maintain or expand market share [16].

Lichtenthaler [7] surveyed the innovation environment within 154 European companies and plotted the results in the context of source of technology (internal *vs.* external) versus technology exploitation mechanism (products and services *vs.* licensing). Approximately 68% of the respondents were deemed *closed innovators* generating technology internally and exploiting that technology in products and services. A further 9% were *absorbing innovators*, acquiring technology externally and exploiting that technology in products and services. Approximately 6% were *desorbing innovators*, licensing internally developed technologies to third parties without developing related products or services and 8% were *open innovators*, acquiring technologies for use in products and services while also actively pursuing further technology commercialization such as licensing. Finally, 8% were *balanced innovators*, illustrating no significant bias in any direction.

To illustrate the range of perspectives on the issue, we note that Vanhaverbeke *et al.* [6] model innovation in terms of a *real option*, the right, but not the obligation, to take an action in the future. Instead of the commercialization strategies identified by Lichtenthaler, Vanhaverbeke *et al.* [[6], [7]] consider both closed and open innovation as considered risk management tactics that keep the organizations options open, postponing the need to make a decision until later in the development process. While sourcing innovation can be expensive, they consider this expense to be small compared to the overall costs of introducing the innovation to the organization or the marketplace.

Requirements engineering is a key activity of software engineering and management decision processes are cornerstones of business success [1], [4], [15]. After the requirements are captured, analyzed, specified, and sent to implementation, decision-making processes become the dominant activity [1], [14], [15]]. In the studied case company, the innovation is represented as identified needs (in the form of requirements) for technology acquisition followed by technology exploitation,

typically on a commercial basis. Closed technology exploitation occurs when the technology is held as a proprietary advantage and offered to third parties only in the form of a product or service. Open technology exploitation occurs when a technology is available to a third-party, via licensing or some other means. The company under study has recently transitioned from the closed innovation to open innovation exploitation.

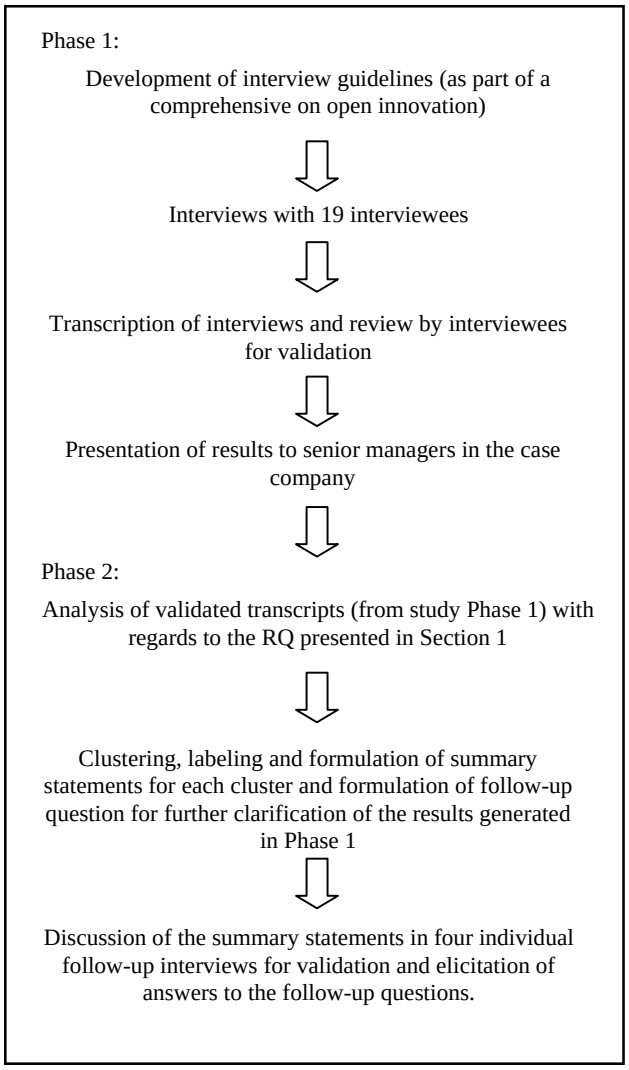

Figure 1. The phases of the study

An integral part of the *market-driven requirements engineering* (MDRE) context, in which our case company operates, is the constant flow of new requirements arriving from multiple sources [15]. MDRE also presents other challenges, e.g., balancing between market pull and technology push, cost-value estimation and release planning as well as overloaded requirements management [15]. Making decisions about which of the incoming requirements to implement is a vital part of developing software systems that meet stakeholders' needs and expectations [14], [15]. Given that only some requirements can be implemented due to limited resources, prioritization techniques need to be applied to find the most valuable requirements [14]. The task of prioritizing requirements; although considered generally challenging; could be further complicated in an open innovation context due to several reasons. Firstly, in a closed innovation model, the cost for implementation is estimated (e.g., by experts) and the market-value is estimated using market research or business intelligence activities. In an open innovation context (e.g., when a company uses an OSS solution as the basis for its software products), the cost could be low (only the integration cost must be incurred). However, the market value of the innovation could be limited by the fact that competitors may be using the same open source solution as the company.

The company must decide which version of an OSS project to use, how to make it compatible with the rest of the software and hardware elements of the product, and which new (differentiating) features to add in order to create or maintain competitive advantage (especially when competitors use the same open source code base). The decision is quite complex as there may be: (1) necessary features to continue to participate in a marketplace, (2) non-differentiating *vs.* differentiating features (differentiating features can confer a competitive advantage), (3) evolutionary *vs.* revolutionary features (*e.g.* abandon the current open source solution and move to another one).

In this case it appears to be critical to properly identify and prioritize the differentiating features that offer a competitive advantage. The lack of full control over the features and contents of releases of the OSS code base complicates software product management activities, especially scoping [11] and release planning [18] compared to the closed innovation context. Past research has not yet focused on these emerging issues [14], [11], [18], [17], [4]].

As the role of innovation is related to creating or eliciting requirements, several researchers have explored the role of innovation in requirements engineering. Kauppinen *et al.* observed six companies and discovered that creativity was not emphasized in their current RE processes [8]. According to Kaupinnen *et al.*, better support of innovation can be secured by ensuring that ideas about innovative features are not removed by the process constraints, as well as by discovering and developing hidden customer needs. Grube and Schmid [9] reviewed creativity techniques for the purpose of devising a systematic approach to those techniques in the requirements management process. Still, there is little empirical work on how requirements management processes can support innovation in general and open innovation in particular also in the case when open source software is used as a source of open innovation.

Open innovation based on OSS can be considered a tool for uncertainty reduction and risk management: if the requirement is already in the open source project, it can't be used as a differentiator by a competitor. However, OSS is also an unpredictable and only loosely controllable source of innovation. The innovation comes for free and is available to anyone using this solution. A company that uses open source can inject new requirements into the community to outsource the cost of prototype development. The company can 'contribute' by licensing outside of organization area of interest (as a *desorbing innovator* [6]) and can 'interfere' by deliberately placing an innovation in the freely available open context, thereby removing the contribution as a differentiator for any competitor. The company can simply 'extract' the already identified requirements and in this way reduce the risks related to their identification and implementation. Further, adopting an open source platform may not be a matter of choice: competitors may attempt to take competitive advantage via the platform, thereby forcing the

organization to follow rather than lead. While related papers described the requirements processes in open source software forums [19] or identified which industry sectors are significantly penetrated by open source software [20], little research has been conducted into identifying and understanding the new requirements engineering and decision making challenges posed by the open innovation paradigm.

3. CASE COMPANY DESCRIPTION AND RESEARCH METHODOLOGY

In this paper we investigate open innovation in the requirements management context at a case company using OSS and performing both requirements scoping and requirements management activities. Requirements scoping is defined as a process of deciding: (1) which version of the OSS product (*platform)* should be utilized, (2) what should be added to the platform in order to create product differentiation and competitive advantage, (3) what shall be contributed to the open source community (and when), and (4) how to influence the open source community and become the leading stakeholder to maximize ROI and reduce uncertainty. The associated requirements management aspects are related to understanding which customer requirements are going to be satisfied when a certain open source product is selected and how to ensure that previously satisfied customer requirements (and, therefore, no longer considered innovative) are implemented in the new version of the OSS platform.

Due to the exploratory nature of this study, we decided to use a combination of maximum variance and convenience sampling [12] to gather the data. A case company sample population of different roles that are related to the requirements management process was created. The 19 study participants who came from requirements engineering, product planning, high-level management, and portfolio management. The requirements-related experience of interviewees varied between 6 months and 10 years. Given the limited related work and the exploratory nature of this study, a qualitative interview study was used to understand the issues raised by the change in the experimental context [10], 12].

The interviewees are employees of a global company with approximately 5,000 employees currently undergoing a transition from a waterfall-based methodology to an agile methodology [13]. As a part of the transformation to the agile methodology the company introduced the following innovations: continuous release planning flow, cross-functional development teams, iterative detailing of requirements and integrated requirements engineering. The company uses *software product line* (SPL) management [11] in the embedded systems domain and there are more than 20,000 feature and system requirements defined across all the product lines. New projects on the product line typically add 60 to 80 new features with an average of 12 new system requirements per feature. Approximately 20 to 25 different development teams (with 40 to 80 developers per team) work on implementing these features. At the same time as undergoing this methodology transformation, the company started using an open source solution as a base for software products. The open source solution, referred to here as the *platform*, is the base for the software product line projects and derived products.

The research process is outlined in Figure 1. In the first part of the study (Phase 1), each of the 19 participants was interviewed. The first author of this paper interviewed each participant individually. Each interview was recorded and transcribed. The transcripts were returned to the participants for validation before coding and grouping the transcribed responses. The transcripts were later analyzed and categorized by the first authors based on his subjective judgment. Next, the categories were grouped into 12 clusters, labeled, and assigned summary statements S1 to S12 as described in Section 5 below. These statements serve as the basis for developing a better understanding of the interplay between requirements management, engaging in OSS development, and facilitating open innovation. The completeness and consistency of the clusters and their summary statements were discussed between three authors of this paper. One concern was, for example, that statements S7 and S11 as well as statements S3 and S12 had seemed to have some overlap. The argument to keep them as separated categories, nevertheless, was that they approach the issues raised by our research question from different angles, either from the process or contribution angle or from the release planning and prioritization angle.

At the end of Phase 1, the study results were summarized in a presentation and presented to high-level management at the case company and their feedback was collected.

Table 1. Interview questions used in study Phase 1, grouped into topics (with topics related to the study reported in this paper highlighted in gray)

1. Background
1.1 What is your role?
1.2 How long have you been working with this role?
1.3 How much experience do you have working with requirements?
1.4 How much experience do you have working with scoping and scoping processes?
2. The business goal of the current requirements management process
2.1 What is according to you the goal of the business goal of the current requirements management process?
2.2 How the part of the process you are mostly involved in can contributes to this overall business goal?
2.3 How do we know that the overall business goal of the requirements management process is achieved?
3. Current metrics of the requirements management process
3.1 Do you have any metrics to measure the performance of the process?
3.2 Can the exiting metrics be traced to the business goal of the process?
3.3 How much time do you spend on the metrics right now?
3.4 How do you collect the metrics? Manually or automatically?
3.5 How do you analyze and interpret the metrics? Do you have any issues with analysis and interpretation?
4. Desired metrics of the requirements management process
4.1 What metrics would you like to have to better measure the process?
4.2 How much time would you spend on collecting and analyzing these metrics?
5. Visualization of the requirements management process
5.1 Do you see the need to visualize the requirements management and especially the scoping process?

6. Requirements management and open innovation
6.1 Do you think that the current requirements management process is designed to facilitate open innovation?
6.2 What actions you suggest to improve the current requirements management process to better benefit from open innovation?

The interview questions used in Phase 1 of the study are listed in Table 1. The questions were sent to each of the participants in advance to help them prepare for the interview. The results reported in this paper are exclusively related to the questions under topics 1 and 6. The remaining topics discussed during the interviews will be reported in a separate publication. The questions directly concerned with open innovation are questions 6.1 and 6.2. The open nature of these questions allowed interviewees to express their opinions about the readiness of the current process to support open innovation as well as to provide their views of the process improvements that are required to fully benefit from open innovation.

In the second part of the study (Phase 2), interviews were conducted with one high-level business development manager, two requirements engineers performing feature scoping, and one process manager. The 4 respondents were presented with the list of 12 statements S1 to S12 derived from the 19 interviews and were asked to agree or disagree with the statements and to provide comments if they were so inclined. The meaning of each statement was described in detail to the interviewees. In addition, the interviewees were also asked to classify their company using the classification scheme reported in [6] – both as it currently operates and how the respondent would like their company to operate. Finally, the respondents were asked to assess whether open innovation impacts requirements engineering challenges and, if so, in what way.

Table 2. The interview instrument used in study Phase 2

1. Please agree or disagree with the following statements related to open innovation and requirements management and to provide comments if they were so inclined
Contributions to the OSS community
(S1) Unclear content and contribution strategy
(S2) Contribution timeline unclear
(S3) Minimize modifications to the open source code
(S4) Unclear relationship between the benefits from contributions in terms of strategy and business goals
(S5) Be strategic when adopting innovative features
Relation between process and innovation
(S6) Augmenting the requirements management process
(S7) Manage innovative features in a separate process
(S8) Top-down or bottom-up open innovation
Release planning and prioritization
(S9) Prioritization process needs modifications
(S10) Challenging acceptance criteria kills innovative features
(S11) Need for special flow for innovative features to evolve to meet acceptance criteria
(S12) Release planning even more challenging
2. Please assess whether open innovation impacts the following requirements engineering challenges and, if so, in what way
2.1 Identify stakeholders needs
2.2 Requirements traceability
2.3 Changing requirements
2.4 Quality requirements
2.5 Communication
2.6 Release planning
2.7 Requirements prioritization
2.8 Requirements overload
3. Please classify your company using the following classification scheme (based on [6])
Closed innovator – generating technology internally and exploiting that technology in products and services
Absorbing innovator – acquiring technology externally and exploiting that technology in products and services
Open innovator – acquiring technologies for use in products and services while also actively pursuing further technology commercialization such as licensing
Balanced innovator – illustrating no significant bias in any direction

4. VALIDITY ISSUES

Following the typology suggested by Maxwell for qualitative studies [21] we briefly discuss the five categories of validity relevant for qualitative studies: description validity, interpretive validity, theoretical validity, generalizability and evaluative validity.

Description validity is concerned with the factual accuracy of our account of what the interviewees actually said. In order to mitigate threats to description validity, we recorded the interview sessions, transcribed them, where necessary removed repetitions or obviously irrelevant statements, and then asked our interviewees to check the accuracy of the cleaned transcription.

Asking interviewees to check the cleaned transcriptions was a measure to mitigate threats to *interpretive validity*.

Theoretical validity is concerned with the relationship between the researchers' observations during the study and their attempts to relate their observations to existing theories, e.g., for the purpose of confirmation, or for the purpose of modifying existing and generating new theory. In other words, theoretical validity has to do with reasoning about the descriptions and interpretations. It has two aspects, firstly, the application of theoretical constructs to the descriptive and interpretive understanding of what has been observed, and secondly, the creation of semantic relationships, narrative structures, and causal relationships that help explain what has been observed. The first aspect of theoretical validity is often called *construct validity*; the second aspect is often called *internal* or *causal* validity. We tried to mitigate threats to theoretical validity. For example, to avoid

bias due to unclear questions, we had our interview guideline and the list of statements we presented to the interviewees reviewed by four experienced researchers and practitioners familiar with both open innovation and requirements management research and practice. This contributed also to minimizing the risk of response bias, i.e., the possibility that questions or our list of statements are formulated in a way that they impose a particular answer. We also were aware of the risk of reflexivity during the interview session, i.e., the possibility that an interviewee responds according to a perception of what the interviewer wants to hear. We took precautions that the interviewer expressed neutrality when, for example, talking about the concepts mentioned in our list of summary statements S1 to S12 during the second interview round (Phase 2). Since the study was purely exploratory, we did not have specific hypotheses, theories, or conceptual frameworks in mind to which we tried to connect the responses given by the interviewees. However, we tried to develop an understanding of how the adoption of OSS in the context of requirements management facilitates open innovation. To what extent our conceptualizations and conclusions derived from the interviews are correct (e.g., the generation and clustering of statements S1 to S12) remains to a certain degree unclear and calls for inspection by other researchers in the field as well as follow-up studies conducted to corroborate or disprove our findings. The least we can claim, however, is that the investigated problem is authentic as it originates directly from the company under study.

Generalizability has two sides, internal generalizability, concerned with generalizing within the case company studied to persons that were not directly observed or interviewed, and external generalizability, concerned with generalizing to other companies. These two sides of generalizability correspond to what is often called statistical conclusion validity and external validity in quasi-experimental research. Both aspects of generalizability are – as it is typical for qualitative studies - strongly limited, since our study was conducted in one single company, involving a relatively small number of interviewees. Nevertheless, we can say that we took care through our contacts within the case company to sample representative individuals as interviewees. Furthermore, our case company is comparable to their direct competitors and thus could be considered typical case for that group of companies.

Finally, we can say that evaluative validity didn't play a role in the context of our study, since our study was purely exploratory, and we didn't refer to any kind of evaluative framework when investigating our research question.

5. RESULTS

After we had transcribed and interpreted responses from the first round of interviews, we extracted key statements and grouped them into 12 clusters. For each cluster, a summary statement was synthesized for use in the second round of interviews. We present the summary statements and our observations below and briefly sketch the opinions shared by the interviewees participating in the validation sessions (second interview round, Phase 2 of the study).

5.1 Contributions to the OSS community

(S1) Unclear content and contribution strategy. In our study context, contribution strategy is defined as a management strategy defining when and what to contribute back to the OSS community. The contribution unit is in this case features. Several interviewees mentioned issues and challenges in relation to the contribution process and contribution strategy; four respondents stressed the lack of a clear contribution strategy. One respondent stated that there exists a contribution process description at the case company, a statement supported by three out of four respondents in the validation phase. For one respondent (the business development manager) the contribution strategy was clearly identified. This suggests that the contribution strategy may be present at the case company but not clearly communicated to the operational levels of the company.

(S2) Contribution timeline unclear. One interviewee expressed uncertainty as to when the developed innovative features should be contributed back to the open source community and all four validation respondents agreed. Given that guidelines do not exist, we take this as evidence that deciding when to contribute is challenging.

(S3) Minimize modifications to the open source code. Two interviewees pointed out that adopting the open source code, making changes, and not contributing these changes back to the community creates a risk of creating additional effort to perform maintenance and gap analysis. One interviewee stressed that, regardless of the risk of losing a competitive advantage, the company should contribute as much as possible to the open source community in order to minimize the maintenance effort. Two of the validation respondents agreed, one expressed a neutral opinion and one suggested that this is *not* an issue since the current requirements management process prevents too much code differentiation during the architecture analysis phase.

(S4) Unclear relationship between the benefits from contributions in terms of strategy and business goals. One interviewee expressed the concern that some competitors are more successful in the market even though the company is contributing more than they are contributing. All validation interviewees agreed, explaining that the company actually contributes many low level features that may not be directly recognized as short-term business value. Further, contribution to the community is a long-term investment.

(S5) Be strategic when adopting innovative features. Another interviewee stated that sometimes the company shouldn't be the first one to release certain features based upon open source code but should instead wait for the open source release to provide the functionality and be the second company on the market. The participants of the validation interviews partially agreed or disagreed with this statement – explaining that it depends on whether the features are a part of differentiating technological advance. In this case, the company should be the first one to release new technology or functionality to the market. In areas not considered of *core technological advantage*, being second in the market would probably not harm profitability. In other words, early commitment to customer requirements seems to be not beneficial for non-core technological features as OSS may provide these features in the next release minimizing the effort required to implement them.

5.2 Relation between process and innovation

(S6) Augmenting the requirements management process. Most interviewees suggested that the process needs to be upgraded to fully benefit from open innovation, but some interviewees disagreed. For example, commitments to customers' requirements early in the development process and delivery time agreements were felt to hinder open innovation: the current process is tailored

for well-defined features for a specific product definition and "you get locked in, with the agreed functionality, sometime ahead". However, the validation interviewees disagreed with this statement explaining that the current requirements management process is *independent* or *agnostic* to the innovative features as long as they are technically feasible and could prove to have the market potential required to by the demanding prioritization process.

(S7) Manage innovative features in a separate process. One interviewee stated that their process is designed for handling mature feature concepts that can be designed and implemented by developers in a straightforward manner. This interviewee proposes a separate process for maturing innovations so that they can be implemented when ready. All four participants of the validation interviews agreed with this statement, we reckon the interviewer to their response to S6 as the reason (see above).

(S8) Top-down or bottom-up open innovation. Two interviewees stressed that, to enable more contributions, open innovation should be handled at the developer's level; the company should be much more technology-driven, asking the developers "what can be done next with the same code base". The respondents of the validation interviews disagreed with this statement explaining that the developer may have difficulties understanding the overall business and product portfolio strategy. Therefore, open innovation should, according to them, be handled both by developers and managers. These results complement the viewpoint of Chesbrough who focused on introducing open innovation in a top-down fashion [5].

5.3 Release planning and prioritization

(S9) Prioritization process needs modification. One interviewee stressed that the business value prioritization currently used by the case company, designed for handling mature concepts within a defined product, hinders the acceptance of innovative features. Although the problem of applying wrong prioritization criteria doesn't seem to be specific only for OSS and open innovation contexts, in the studied context the challenges seems to be related to the fact that the values for the selected criteria may change depending on what the next release of the OSS provides and the contribution strategy.

(S10) Challenging acceptance criteria kills innovative features. According to another interviewee, the company has tried to include the innovative features in the normal feature decision making and release planning flow but this resulted in an inability to meet the tight deadlines dictated release planning as well as by the market and competitors. Therefore, only mature features are currently considered.

(S11) Need for special flow for innovative features to evolve to meet acceptance criteria. Two interviewees suggested extending the current process to have a special process for handling these features, perhaps by giving them more time to mature (see Section 4.2). All four participants of the validation interviews agreed with statements S9, S10 and S11 – pointing out that the current feature management process is a "controlled factory" that facilitates the development of the features with greatest potential value, a defined customer, and reliable estimates of system impact and development effort. All respondents supported a separate process for introducing and maturing innovative features to prepare them for joining the tight release deadlines of the "controlled factory" process.

(S12) Release planning even more challenging. One interviewee considered release planning more challenging in the open innovation context, stressed the inherent difficulties. She provided an example where it turned out to be a better business decision to just adopt the open source code, perform minimal adaptation, and sell it rather than spending time and effort on creating differentiating features. The lack of control over release planning was cited as a complicating factor. Two validation respondents agreed with this statement but the other two pointed out that release planning is not more difficult if the innovative features are releasable (possible to implement within budget and hold promising business potential). To summarize, the respondents felt that release planning could be more challenging in the open innovation context.

5.4 Challenges in requirements engineering

The four interviewees that participated in the validation interviews were asked to suggest which requirements engineering challenges [8] are addressed by, impacted by, or made more challenging by the open innovation context. Three out of four interviewees indicated that open innovation makes the challenge of identifying stakeholders' needs more manageable and one indicated that this challenge is not related to open innovation. Regarding the challenges of requirements traceability and release planning, two interviewees thought that it is more challenging while the other two felt that this challenge is not related to open innovation. Further, all four respondents confirmed that the challenges of changing requirements and requirements overload are more manageable in the open innovation context. Regarding the challenge of managing quality requirements, two interviewees indicated that this challenge would become more manageable in the open innovation context (since the company will use open source code with potentially fewer bugs) while the other two respondents suggested that this challenge is not related to the open innovation context. Further, three out of four interviewees indicated that the challenge of communication in requirements engineering is unrelated to open innovation and all four interviewees suggested that requirements prioritization is more challenging in the open innovation context.

The validation interviewees were asked to identify which type of innovative company the case company is and which type of innovative company it should be. All four interviewees suggested that the company is currently mostly an *absorbing innovator* [[6]] and one interviewee also suggested that the company is behaving a bit like a *desorbing innovator* when releasing the implemented features back to the open source community. All interviewees pointed out that the ultimate goal for the company is to become an *open innovator.*

6. DISCUSSION

Answering our research question RQ1 (see Section 1), based on the results presented in the previous section, we found that the current requirements management process isn't designed to fully benefit from open innovation context. In the following, we summarize and discuss the suggestion for adaptations of the current requirements management process in the case company.

The overall results from both sets of interviews suggest the need for creating a contribution strategy that clearly identifies what should be contributed, when, as a part of the requirements management process. Further we identified a need for finding the right balance for contribution that secures successful product differentiation. Moreover, our respondents suggested that *open*

innovation should be introduced in both a top-down and bottom-up fashion.

When it comes to the requirements management process, our results highlight the need for introducing a separate requirements process for handling (often immature) innovative feature concepts. Further, our interviewees expressed the need for creating a method for prioritizing requirements (or features) that is more suitable for the open innovation context. Our interviewees suggested that release planning and prioritization methods should be revisited and optimized for the open innovation context as planning releases is definitely more challenging in this context. Finally, our results suggest that challenges of identifying stakeholders' needs, changing requirements and requirements overload are more manageable in an open innovation context while release planning and requirements prioritization are more challenging.

Based on the results from the two rounds of interviews we highlight two areas where we believe further research should be focused on:

- *Requirements management processes for open innovation (this area emerged based on summary statements S1, S2, S3, S6 and S7).* Our respondents clearly state the need for improved requirements management that supports planning and execution of feature contributions that will be sent back to the open source community. Early or frequent contributions create a risk of losing competitive advantage while late or rare contributions can greatly increase the maintenance cost. Thus, further research is needed to better understand the balance between limited and generous contribution strategies. Moreover the right level of adjustments and differentiations has to be preserved in order to minimize the maintenance costs. Additionally, there seems to be a need for augmenting the requirements management process by a separated flow for introducing innovative features where they can mature and become integrated into the main process pipeline.
- *Revisit release planning and prioritization models (this area emerged based on summary statements S3, S9 and S11).* There is a need for revisiting current release planning and prioritization techniques with the goal of understanding how those two tasks can be performed in the open innovation context. For example, the market value and the implementation cost commonly used as criteria in the Analytic Hierarchy Process (AHP) method [[14]] should be considered in the context of both the company and the open source community. The cost of functionality developed individually by a company and later contributed back to the open source community will be decreased by sharing the ongoing maintenance effort (the dominant cost factor in the case company). Moreover, the release planning methods should be reviewed and possibly augmented by the necessary risk and dependency analysis when features provided by open source products (with release times that can't be controlled) are key components of a company's software products.

7. CONCLUSIONS

We have presented initial results from a study that explores challenges regarding requirements scoping and requirements management in the open innovation context. The results obtained through a first set of 19 interviews followed by a second set of 4 interviews, highlight that managing requirements in an open innovation context is challenging as requirements are freely available for several potentially contributing companies and, even more interestingly, their implementation is freely available.

Future work includes further empirical studies in understanding the impact of open innovation on requirements engineering processes, tools and techniques. Further, we plan to focus on exploring new ways of prioritizing requirements that could potentially be more suitable for the open innovation context.

8. REFERENCES

[1] Aurum, A., Wohlin C. 2007. A Value-Based Approach in Requirements Engineering: Explaining Some of the Fundamental Concepts. In *Proceedings of the REFSQ 2007 Conference* (Montpellier, France, June 20-23, 2007). 109-115. Springer, Heidelberg (2007) 109-115. DOI = 10.1007/978-3-540-73031-6_8

[2] Herbsleb, J. D. 2007. Global Software Engineering: The Future of Socio-technical Coordination. In: Proccedings of the *FOSE'07 Future of Software Engineering* (Minneapolis, MN, United states, May 23, 2007 - May 25, 2007). IEEE Computer Society, Washington, DC, USA, 188-198. DOI = 10.1109/FOSE.2007.11.

[3] Biffl, S., Aurum, A., Boehm, B., Erdogmus, H., Grunbacher, P. (eds.). 2005. *Value-Based Software Engineering*. Springer, Heidelberg, ISBN 3-540-25993-7.

[4] Sommerville, I., Sawyer P. 1997. *Requirements Engineering: A Good Practice Guide.* John Wiley & Sons Ltd., Chichester.

[5] Chesbrough, H. 2003. *Open Innovation The new Imperative for Creating and Profiting from Technology*, Harvard Business School Press, USA.

[6] Vanhaverbeke, W., Van de Vrande, V., Chesbrough, H. 2008. Understanding the Advantages of Open Innovation Practices in Corporate Venturing in Terms of Real Options. *Creativity and Innovation Management*. 251 (Mar. 2008), 251-258. DOI: 10.1109/IEEM.2008.4738011.

[7] Lichtenthaler, U. 2008. Open Innovation in Practice: An Anlaysis of Strategic Approaches to Technology Transactions. *Trans on Eng Management* 55 (May 2008), 148-156. DOI: 10.1109/TEM.2007.912932.

[8] Kauppinen, M., Savolainen, J., Mannisto, T. 2007. Requirements Engineering as a Driver for Innovations. In *Proceedings of the 15th IEEE International Requirements Engineering Conference* (New Delhi, India, 15--19 October 2007). IEEE Press, New York, 15-20. DOI: doi: 10.1109/RE.2007.47

[9] Grube, P., Schmid, K. 2008. Selecting Creativity Techniques for Innovative Requirements Engineering. In *Proceedings of the Third International Workshop on Multimedia and Enjoyable Requirements Engineering - Beyond Mere Descriptions and with More Fun and Games* MERE '08 (Barcelona, Spain 3 September), IEEE Press New York, 32-36. DOI: 10.1109/MERE.2008.6.

[10] Myers, M. D., Avison D. 2002. *Qualitative Research in Information Systems*. Sage Publications, USA.

[11] Pohl, C., Böckle, G., Linden, F. J. van der. 2005. *Software Product Line Engineering: Foundations, Principles and Techniques.* Springer-Verlag, New York USA.

[12] M.Q. Patton. 2002. *Qualitative Research and Evaluation Methods.* Sage Publications, USA.

[13] Agile manifesto `http://agilemanifesto.org/`, accessed January 2012

[14] Karlsson, J., Ryan, K. 1997. A cost-value approach for prioritizing requirements. *IEEE Softw*. 14 (Sept. 1997), 67-74. DOI: 10.1109/52.605933.

[15] Regnell, B., Brinkkemper, S. 2005. Market–Driven Requirements Engineering for Software Products. In: *Aurum, A., Wohlin, C. (Eds.), Managing and Engineering Software Requirements*, 287-308. Springer- Verlag, Heidelberg.

[16] Chesbrough, H., Adrienne Kardon C. 2006. Beyond high tech: early adopters of open innovation in other industries. *R&D Management*. 36 (June 2006), 229-236. DOI: 10.1111/j.1467-9310.2006.00428.x.

[17] Hood, C., Wiedemann., S., Fichtinger, S., Pautz, U. 2008. *Requirements Management: The Interface Between Requirements Development and All Other Systems Engineering Processes*. Springer, Berlin.

[18] Svahnberg, M., Gorschek, T., Feldt, R., Torkar, R., Bin Saleem, S., Usman M. 2010. A systematic review on strategic release planning models. *Inf. Softw. Technol.* 52 (Mar 2010), 237-248. DOI: 10.1016/j.infsof.2009.11.006.

[19] Laurent, P., Cleland-Huang, J. 2009. Lessons learned from open source projects for facilitating online requirements processes. In *Proceedings of the 15th International Working Conference Requirements Engineering: Foundation for Software Quality* (Essen, Germany, 25 – 28 march 2009). Springer Verlag, Berling, 240-55. DOI: 10.1007/978-3-642-02050-6_21.

[20] Glynn, E., Fitzgerald, B., Exton C. 2005. Commercial adoption of open source software: An empirical study. In *Proceedings of the International Symposium on Empirical Software Engineering*, (17-18 November, Noosa Heads, Australia , 2005), IEEE Press, New York , 225-234. DOI: 10.1109/ISESE.2005.1541831.

[21] Maxwell, J. A. 1992. Understanding and validity in qualitative research. *Harvard Educational Review, 62*(Fall 1992), 279-300.