<http://www.diva-portal.org>

Postprint

This is the accepted version of a paper published in *Empirical Software Engineering*. This paper has been peer-reviewed but does not include the final publisher proof-corrections or journal pagination.

Citation for the original published paper (version of record):

Munir, H., Wnuk, K., Runeson, P. (2015)

Open innovation in software engineering: a systematic mapping study.

Empirical Software Engineering, 21(2): 684-723

<https://doi.org/10.1007/s10664-015-9380-x>

Access to the published version may require subscription.

N.B. When citing this work, cite the original published paper.

Permanent link to this version:

<http://urn.kb.se/resolve?urn=urn:nbn:se:bth-13693>

Open Innovation in Software Engineering: A Systematic Mapping Study

Hussan Munir · Krzysztof Wnuk · Per Runeson

Received: date / Accepted: date

Abstract

Context: Open innovation (OI) means that innovation is fostered by using both external and internal influences in the innovation process. In software engineering (SE), OI has existed for decades, while we currently see a faster and broader move towards OI in SE. We therefore survey research on how OI takes place and contributes to innovation in SE.

Objective: This study aims to synthesize the research knowledge on OI in the SE domain.

Method: We launched a systematic mapping study and conducted a thematic analysis of the results. Moreover, we analyzed the strength of the evidence in the light of a rigor and relevance assessment of the research.

Results: We identified 33 publications, divided into 9 themes related to OI. 17/33 studies fall in the high-rigor/high-relevance category, suggesting the results are highly industry relevant. The research indicates that start-ups have higher tendency to opt for OI compared to incumbents. The evidence also suggests that firms assimilating knowledge into their internal R&D activities, have higher likelihood of gaining financial advantages.

Conclusion: We concluded that OI should be adopted as a complementary approach to facilitate internal innovation and not to substitute it. Further research is advised on situated OI strategies and the interplay between OI and agile practices.

H. Munir
Dept of Computer Science, Lund University, Sweden
E-mail: Hussan.Munir@cs.lth.se

K. Wnuk
Dept of Computer Science, Lund University, Sweden
E-mail: Krzysztof.Wnuk@cs.lth.se

P. Runeson
Dept of Computer Science, Lund University, Sweden
E-mail: Per.Runeson@cs.lth.se

Keywords Open innovation · collective innovation · collective invention · user innovation · literature review · research agenda · software engineering

1 Introduction

Open innovation (OI) and associated free exchange of information about new technologies are recognized as one of the main drivers for collective inventions in the 19th century by Allen [1]. Two decades after Allen’s paper from 1983, Chesbrough’s seminal book about OI [5] has initiated an unabated interest [21] among researchers in innovation management [29], economics, psychology, sociology, and also Software Engineering (SE) [57]. The work initiated by Chesbrough [5] forced both practitioners and scholars to rethink the design of innovation strategies in a networked environment [29]. The inherent flexibility of software, combined with increase of software cost and value for new products and services, puts SE into the hotspot of OI. Several trends, such as outsourcing, crowd-sourcing and funding, global software development, open source software, agility, and flexibility, rendered the *do it yourself* mentality outdated [19]. More courageous voices suggested even that closed innovation might have been the exception in the history, characterized mostly by open innovation practices [42].

OI is a relatively new field of research and a collective theoretical foundation is starting to emerge. Chesbrough [5] was the first to define OI as “*a paradigm that assumes that firms can and should use external ideas as well as internal ideas, and internal and external paths to market, as they look to advance their technology*”. OI encompasses various activities such as inbound, outbound and coupled activities [20], and each of these activities can be more or less open. Open Source Software (OSS) is the most straightforward application of OI to software development [29], although not the only one [66]. The tremendous success of OSS in the last twenty years have ignited and encouraged several new movements for collective innovation such as: outsourcing, global software development, crowd-sourcing and founding.

Despite the wide interest in several domains and the unquestionable potential that OI can bring to the software industry, OI remains greatly unexplored in the SE literature, while in the OI literature extensive interest is given to exploring OSS as one of the ways to incorporate external knowledge and innovation to internal product innovation activities [5]. Similarly in the early days of OSS, many interesting OI initiatives were performed, e.g. opening up software product organizations and utilizing open configurations [33]. However, there is a lack of systematic efforts that focus on summarizing the current state of the literature on the OI relation to SE. Previous reviews are either not systematic [29,66], partly systematic [61] or, for example, focus on the history or evolution of OSS or available innovation metrics [17]. Moreover, these reviews lack quality criteria to support the interpretation of the results in favor or against OI.

Therefore, we identified a need to systematically review OI research in SE with a specific focus on assessing the strength of the empirical evidence in the identified studies [31], highlighting the current themes and outlining implications for research and practice. For instance, a study might have high relevance (e.g. managerial implications for an industrial scale project), but at the same time have low rigor (e.g. having validity threats and lacking descriptions of the units of analysis). Consequently, these above mentioned needs lay the foundation for a systematic mapping study [45] to explore the concept of OI in the context of SE. Specifically, this mapping study makes the following contributions:

1. Identification of the existing themes and patterns in the literature for open innovation in software engineering.
2. Assessment of trustworthiness of the results with respect to rigor and relevance [31].
3. Based thereon, identification of knowledge that may inform industry practice on open innovation in software engineering
4. Identification of the research gaps for further exploration of open innovation in software engineering [36].

The remainder of the paper is structured as follows: Section 2 presents related work and Section 3 presents the research method (review protocol). Next, Section 4 highlights the results of the search and the analysis the synthesized research, followed by a discussion in Section 5 which results in a research agenda and advice for industry practice in Section 6. Section 7 concludes the paper.

2 Related work

We identified four secondary studies (literature reviews) [17, 29, 61, 66] on open innovation, relevant to this study, by applying forward snowball sampling, using the study by West and Bogers [61] as a baseline. The studies are summarized in Table 1.

Are the reviews systematic? Huizingh [29] and Wnuk and Runeson [66] conducted reviews on OI, however neither of them is systematic according to the guidelines stated by Kitchenham et al. [35]. The study conducted by West and Bogers [61] could be considered partly systematic, since the relevance can be seen in terms of data sources, inclusion/exclusion criteria and data extraction. On the other hand, the review conducted by Edison et al. [17] adheres to Kitchenham et al. [35] and Petersen et al. [45] guidelines. In this paper, we report a review conducted according to the guidelines by Kitchenham et al. [35].

What were the objectives behind conducting reviews? West and Bogers [61] conducted a review on OI with the main objective to define an agenda for OI research. They classified the studies into three main categories of OI, namely, inbound (outside in), outbound (inside out) and coupled, as suggested by Enkel

at al. [18]. Wnuk and Runeson [66] performed a study with the goal to propose a SE framework for OI.

Huizingh [29] also focused on exploring the notion of open innovation and on the degree of OI adoption by the firms. The study concluded that the knowledge about *how* to apply OI and *when* to do it is still incomplete. Edison et al. [17] centered their literature study around innovation measurement and innovation management aspects, e.g. definitions, frameworks and metrics. Our study limits its scope to SE and focuses on deriving existing OI themes and patterns using thematic analysis. Moreover, this study also focuses on exploring the strength of evidence under the light of rigor and relevance, and states the further course of action in terms of OI in SE.

What were the data sources used in the reviews? Were the used search terms appropriate? Huizingh [29] neither specified the database used, nor reported the used search terms. Likewise, West and Bogers [61] did not mention the search terms for their study, but provided the time scope of the survey (between 2003 and 2010) and the list of selected management journals, see Table 1. Conversely, the study conducted by Wnuk and Runeson [66] used Inspec and Compendex and used the following search terms “Open innovation, requirements engineering, testing, software and methodology”. However, the time span for the search is not reported. Edison et al. [17] used multiple data sources namely, Inspec and Compendex, Scopus, IEEE explore, ACM digital library, Science direct, Business Source Premier (BSP) and performed the search between 1949 and 2010, see Table 1. Their search terms aim at identifying innovation metrics, measurements, drivers and innovation attributes. Inspired by the previous reviews, we organized our search string into three main categories and employed the inclusion exclusion criteria after the search process, with keywords: i) related to OI, ii) on SE in order to restrict the results to the SE domain, and iii) pertaining to empirical evidence on OI (see Section 3.3). Furthermore, we complemented our search string with backward snowball sampling [32, 50] by scanning the reference list of all primary studies, see Section 3.2.

Did the reviews use any quality assessment criteria for primary studies before analyzing their results? Neither Wnuk and Runeson [66] nor Huizingh [29] used explicit criteria for quality assessment of the identified studies, which hinders interpretation. On the other hand, West and Bogers [61] included studies that focused on OI as per the definition by Chesbrough [5] and excluded book reviews, commentaries and editorial introductions. Edison et al. [17] used a set of questions for quality assessment and to evaluate if a study explains the aims, methodology and validity threats. We used a comprehensive set of guidelines that cover rigor and relevance of studies. We slightly tailored the criteria from Ivarsson et al. [31] to fit into the scope of this study, see Section 3.4.

How did the reviews extracted data from primary studies? Did they map data extraction with research questions? The data extraction strategy was not reported in three studies [29, 66, 61]. The information about the mapping between the data extraction properties and the research questions was also ab-

sent. However, Edison et al. [17] described the data extraction strategy which was piloted before the execution to ensure a common understanding among all involved researchers. We created a defined set of data extraction properties, and mapped them on research questions to avoid redundant information, outlined in Table 3.

How did the reviews synthesize the data from primary studies? Neither of the four studies followed an established procedure for the synthesis, such as thematic or cross-case analysis [9, 10]. Instead, West and Bogers [61] used a self-created four phase integrated model (i.e. obtaining, integrating, commercializing, interaction with communities) to guide the literature review and classified studies based on dimensions provided by Enkel et al. [18]. Similarly, Wnuk and Runeson [66] presented the synthesis in a table where studies are categorized in terms of research type (e.g. evaluation, proposal, opinion, solution, conceptual etc.) defined by Wieringa et al. [64]. Moreover, studies were also classified in terms of software techniques, process and methods, and presented a framework to foster OI with technical and methodological dimensions stated above.

Edison et al. [17] presented their synthesis in terms of different types of innovation definitions available in the literature, metrics used to measure innovation, and challenges related to existing innovation measurements. They developed a model to assist organizations to use the available measures to develop insights into their innovation program. Finally, Huizingh [29] wrote a literature review without synthesis.

In summary, this systematic study aims at exploring the OI in SE in a much more rigorous manner according to guidelines of Kitchenham et al. and Petersen et al. [35, 45] and focusing on systematic synthesis of the findings.

Table 1 Summary of existing literature reviews

Facets	West and Bogers [61] (2013)	Edison et al. [17] (2013)	Wmuk et al. [66] (2013)	Huizingh [29] (2010)
Systematic	Partly	Yes	No	No
Data sources	1. Academy of Management Journal 2. Academy of Management Review 3. Administrative Science Quarterly 4. California Management Review 5. Harvard Business Review 6. IEEE Transactions on Engineering Management 7. Industrial and Corporate Change 8. International Journal of Technology Management 9. Journal of Product Innovation Management 10. Long Range Planning 11. Management Science 12. MIT Sloan Management Review 13. Organization Science 14. R&D Management 15. Research Policy 16. Research-Technology Management 17. Strategic Management Journal 18. Technological Forecasting and Social Change 19. Technovation	1. Inspece and Compendex 2. Scopus 3. IEEE Xplore 4. ACM Digital Library 5. ScienceDirect 6. Business Source Premier (BSP)	Inspece and Compendex	N/A
Repeatability	No	Yes	No	No
Quality Assessment	No	Yes	No	No
Inclusion/exclusion criteria	Partly	Yes	No	No
Data extraction properties	Partly	Yes	No	No
Validation of results	No	Yes (Piloted the criteria)	No	No
Data synthesis	Partly (without mentioning its type)	Yes (without mentioning its type)	Exploration instead of synthesis	No (Only conclusion)
Purpose	An attempt to define an agenda for open innovation research	This study explores various aspects relevant to innovation measurement ranging from definitions, measurement frameworks and metrics that have been proposed in literature and used in practice	This paper proposes a software engineering framework designed to foster open innovation by designing and tailoring appropriate software engineering methods and tools.	The study intends to explore the challenges that both practitioners and academics face in understanding the open innovation concept. The paper also focuses on less understood areas of open innovation that "require management attention and offer fruitful ideas for further academic research".
Outcome	The study defined the agenda for open innovation research and concludes with recommendations for future research that include examining the end-to-end innovation commercialization process, and studying the moderators and limits of leveraging external sources of innovation	A systematic literature review followed by an online questionnaire and interviews with practitioners and academics were employed to identify a comprehensive definition of innovation that can be used in software industry. The metrics for the evaluation of determinants, inputs, outputs and performance were also aggregated and categorized. Based on these findings, a conceptual model of the key measurable elements of innovation was constructed from the findings of the systematic review.	This study discusses the methodological and process dimensions and outlines challenge areas that should be reviewed when transitioning to software engineering driven open innovation.	The study shows that open innovation has been a valuable concept for so many firms and in so many contexts in innovation management. However, the knowledge about <i>how</i> to do it and <i>when</i> to do it remain dispersed and incomplete.

3 Research methodology

In this section, we present the literature review methodology, based on the guidelines provided by Kitchenham et al. [35] and Petersen et al. [45]. The study was conducted in six steps outlined in subsections below: I) identification of primary studies, II) search string development and database search, III) performing including and exclusion criteria, IV) data extraction, V) quality assessment through rigor and relevance, and VI) synthesis and reporting.

3.1 Research questions

The research questions for the mapping study are defined as:

RQ1: Which themes and patterns of OI in SE exist in the literature?

RQ2: How strong is the evidence in favor of or against OI in SE?

3.2 Identification of primary studies

In order to identify the primary studies, following steps were performed, see Figure 1.

1. Identification of 15 control papers [13] from forward snowball sampling [22, 32].
2. Extraction of studies from databases using a search string: 2805 papers were identified using a search string
3. Duplicate elimination at the database level: 305 studies were found to be duplicates, and hence removed.
4. Selection of studies based on abstract, titles and keywords: 2279 papers were not found relevant and excluded.
5. Filtering based on inclusion/exclusion criteria: 194 additional papers were excluded after applying the inclusion/exclusion criteria and 27 papers were found to be relevant and pertain to the scope of this study
6. Backward snowball sampling was applied to scan the reference list of 27 primary papers and enabled us to spot 6 more relevant papers.

As a result, we identified 33 studies that directly pertain to the scope of the study. The additional studies found by the snowball sampling confirms the usefulness of snowballing for identification of potential studies missed by database searches.

3.3 Search string strategy

In order to develop the search string, the keywords were aptly derived from 15 control papers, see Figure 1. The search terms are organized into three interventions: T1 includes terms related to open innovation, T2 related to outcomes, T3 related to the research methods.

Fig. 1 Identification of primary studies

1. **T1:** Open Innovation OR Open-Innovation OR OI OR innovation OR innovation management
2. **T2:** software OR software ecosystem OR product line OR requirement* engineer* OR requirement* management OR open source
3. **T3:** exploratory study OR lesson* learn* OR challenge* OR guideline* OR Empirical investigation OR case study OR survey OR literature study OR literature review OR interview* OR experiment* OR questionnaire OR observation* OR quantitative study OR factor*

The interventions are combined using boolean operators (**T1 AND T2 AND T3**) to achieve the desired outcome. We searched the following databases, using their command interfaces and utilizing expert or advanced search capabilities (the search strings used per database are reported in Appendix B):

1. ISI Web of Science
2. Inspec and Compendix (Engineering Village)
3. ACM Digital Library
4. IEEE Xplore
5. Science Direct (Elsevier)

The search string was refined, using the control papers as a benchmark, until the average acceptable level of precision and recall was achieved. A study conducted by Beyer and Wright [3] reported that the recall of the search strategies ranged from 0% to 87%, and precision from 0% to 14.3%. The final search string retrieved 13 out of 15 control papers which gives recall of 86.66%. The final search string achieved precision of 0.52% (13 out of 2500 papers, excluding duplicates). Both precision and recall scores are in range with the findings of Beyer and Wright [3]. The fact that two of the control papers were not captured by the final search string confirms the observations by Wohlin et al. [67] that using single search strategies leads to missing studies. Therefore, we combined database searches with snowball sampling.

3.4 Inclusion/exclusion criteria

The inclusion/exclusion criteria were derived and piloted and are outlined in Table 2. These criteria were applied simultaneously on studies to make sure we only include studies that pertain to SE domain and not, for example economics, management or psychology.

The selection of studies was accomplished independently by the two first authors, applying the inclusion/exclusion criteria. In case of uncertainty, the authors included the papers to next step in order to reduce the risk of excluding the relevant papers as suggested by Petersen and Bin Ali [44]. Kappa statistics [37] was calculated at multiple steps in order to check the agreement level between the authors. First, the Kappa coefficient was calculated on a 10% randomly selected sample of titles and abstracts and it was found to be 0.37. After discussing and resolving the disagreements, the Kappa value increased to 0.91 for the same sample. Second, Kappa was calculated on a sample of randomly

Table 2 Inclusion exclusion criteria

Inclusion Criteria	Exclusion Criteria
<ul style="list-style-type: none"> – Peer reviewed papers, and in case of duplicate publications, the priority follows the sequence: Journals, Conferences, Workshops – The study must be accessible in full text. – The study highlights the research-focused concept of OI in the context of software engineering. – The study that reports the benefits, disadvantages, limiting factors, and challenges of OI. – The studies pertaining to the scope of open source software used as OI examples – Factors limiting the adoption of open innovation in software engineering – Available tools used by the software community to support open innovation in software engineering – Studies that discusses the openness of software producing organization(SPO) – All studies from 1969 to 2013 	<ul style="list-style-type: none"> – All gray and white literature – Non-English articles – Studies about OI in the management and economics context – Intellectual property rights papers – Research on open innovation not related to software engineering – All papers that mentioned only the use of software to bring innovation in the fields other than software engineering. – All articles, which are not within the field of software engineering in terms of how to develop software – All duplicate studies

selected 50% of papers included into the full text reading phase while applying inclusion/exclusion criteria. Disagreements were identified as the Kappa value (0.48) was found to be below the *substantial agreement* range. Consequently, after discussing and resolving disagreements [44], the kappa value was calculated again and found to be 0.95. It is to be noted that the inclusion/exclusion criterion was applied simultaneously. However, for exclusion it is enough when one exclusion criterion holds.

3.5 Data extraction and synthesis strategy

The data extraction properties outlined in Table 3 were discussed and finalized before application. Moreover, a spreadsheet was created for the data extraction properties and also mapped with research questions, see Table 3. The first author performed the data extraction, supervised by the second and the third author.

The extracted data was synthesized by performed thematic analysis based on the guidelines by Cruzes et al.[10]. First, we identified patterns in the data and then grouped those patterns into distinct themes. Second, in order to check the trustworthiness of each paper, we used rigor and relevance criteria which helped us identifying whether or not results are generalizable to the software industry, see Section 4.3.

Table 3 The data extraction properties explained and mapped to the research question

Category	Properties	RQ Mapping
General information	Authors, Title, Year of Publication, Abstract	RQ1, RQ2
Study Type	Evaluation research, Solution research, Validation research, Proposal research	RQ1, RQ2
Research Methods	Case study, Tool proposal, Survey, Framework	RQ1, RQ2
Research Problem	Description of research questions	RQ1, RQ2
Outcomes	Benefits, limitation, strategies, patterns related to OI	RQ1
Context	Subjects Type (Students/ professionals/researchers/mixed), number of subjects, case description, validity threats to context.	RQ2

3.6 Quality assessment with respect to rigor and relevance

We used the assessment checklist for rigor and relevance of the research, proposed by Ivarsson et al. [31]. Two researchers reviewed the ratings and data extraction to ensure objectivity. Each paper was assigned a score using objective criteria tailored for this mapping study, see Appendix A.1 and A.2. The idea behind investigating rigor and relevance resembles the use of a rubric based evaluation in education [31]. Previous studies [34,41] have shown that rubrics increase the reliability of assessments in terms of inter-rater agreement between researchers.

Rigor can be defined as “the research methodology is carried out in accordance with corresponding best practices” [31]. Ivarsson et al. [31] state that rigor has two dimensions: following the complete reporting of the study, and best practices. Through aggregating of study presentation aspects from existing literature, they defined rigor as the degree to which study context (C), design (D), and validity threats (V) are described. All facets are rated on a scale, i.e. weak, medium, and strong description, see Appendix A.1.

Relevance deals with the impact of a study on industry [31]. It consists of manifold aspects, namely, relevance of the topic studied [52], ability to apply a solution in a real world industrial setting with degree of success [68], use of research methods that facilitate industrial realism [53], and provision of a realistic situation in terms of users, scale, and context [31]. We followed the suggestion of Ivarsson et al. [31] to decompose rigor into: users/subjects (U), scale (S), research methodology (RM), and context (C), see Appendix A.2.

3.7 Validity threats

This section highlights the validity threats associated with the systematic mapping and how they were addressed prior to the study in order to reduce their impact [48].

3.7.1 Internal validity

The key idea behind conducting the systematic mapping study was to capture available literature as much as possible without introducing any researcher bias thereby, internal validity seem to be a major challenge for the study. In order to address the internal validity concerns, a review protocol was created beforehand and evaluated by three researchers, which took on roles of quality assurance as well. The internal validity is enhanced by following the systematic mapping guidelines [45] and the guidelines for quality assessment criteria [31].

3.7.2 Construct Validity

Construct validity refers to the presence of potential confounding factors and whether or not a study was able to capture what was intended in terms of aims and objectives. One important concern for this study was the multiple definitions of OI. In order to minimize this threat and build on solid foundation, Chesbrough's concept of OI is adopted [5].

3.7.3 External validity

External validity refers to the ability to generalize the results to different settings, situation and groups. The majority of the studies fall into the category of case study methodology with high rigor and relevance, see Figure 6. Moreover, many studies were conducted in industrial contexts hence, the results are more generalized and industry relevant.

3.7.4 Reliability

Reliability is concerned with to what extent the data and the analysis are dependent on a specific researcher. In order to enhance the reliability of this study, multiple strategies were taken into account. First, there is always a risk of missing out on primary studies with a single search string for all selected databases. Therefore, 15 control papers were identified through forward snowball sampling to verify the precision and recall of the search string. However, this only minimizes the selection bias that may impact further research steps. We believe that the potential effect of this bias have a lesser importance in mapping studies than in SLRs. To further substantiate the search process, backward snowball sampling was applied and resulted in additional studies pertaining to the context of OI in software engineering (see Figure 1).

Fig. 2 Distribution of studies over publication years

Second, quality assessment of the identified studies is sensitive on interpretation. Therefore, rigor and relevance criteria were applied to increase the objectivity of this step. The evaluation was performed by the first author and reviewed by the remaining authors. Moreover, we created a data spread sheet and mapped research questions with the data extraction properties in order to comply with the objectives of this study. Besides, all studies were rated according to the rigor and relevance criteria tailored from Ivarsson et al. [31] and data extraction properties from each paper were reviewed by two researchers in the study.

4 Results and analysis

In this section, the results of the mapping study analysis are reported. First, we give an overview of the time distribution, then categorize the studies based on research methodology used. An analysis of the themes studied is reported, followed by a detailed description of each theme.

4.1 Distribution of OI studies

33 primary studies about OI in software engineering were found, distributed by their publication year in Figure 2. The scholarly interest in OI seems to be growing at a steady pace since its introduction in 2003 with a maximum annual rate of 8 studies published in 2009. However, the trend declines after that, and it is hard to assess why, since the interest in OI seems to grow in general [29].

Fig. 3 Research methodology classification based on Runeson et al. and Wieringa et al. [49, 64]

4.2 Categorization based on research methodology

Primary studies found related to open innovation are categorized into the research methodology (i.e. case study, experiment, survey etc) and type of the study (i.e. evaluative, proposal, solution, opinion etc) dimensions. The horizontal axis in Figure 3 represents research methodologies defined by Runeson et al. [49] and vertical axis represents the classification of studies established by Wieringa et al. [64]. Evaluations, using case study research methodology dominate among the identified papers with 20 papers, among which two were interview studies that we consider qualitative case studies. Evaluations, using survey research methodology was found in 7 papers. We classified only 2 papers in each of the framework–proposal and case study–proposal categories. Finally, the categories case study–validation and tool proposal–solution received only 1 paper each and no papers were identified in the case study solution category.

4.3 Thematic analysis

The main objective behind conducting this analysis is to find the recurring themes in the identified primary studies. Based on the guidelines provided by Cruzes et al. [9,10], we performed the following analysis steps:

1. Extract data from the primary studies
2. Identify the interesting themes from the data
3. Group the themes into the distinct categories

Fig. 4 Identified Open Innovation themes in Software Engineering

4. Assess the trustworthiness of the identified themes using rigor and relevance criteria

The resulting 9 themes of OI in software engineering are depicted in Figure 4. Figure 5 provides a more detailed view on the identified themes using the mind map technique, where the 33 primary studies are referred to as **S_1** to **S_33**. In order to assess the trustworthiness of the identified themes, the rigor and relevance analysis is performed and its results are visualized in Figure 6. Details on the primary studies and the rigor and relevance scores are reported in the appendix, Table 7. The rigor and relevance scores are used to find the evidence in favor and against OI in SE (research question RQ2). The results from less relevant and less rigorous studies have weaker empirical support than those stemming from highly relevant and rigorously conducted primary studies. There can also be promising highly relevant studies that were conducted with low rigor.

Studies are organized into four quadrants (A, B, C and D) according to their rigor and relevance scores. The procedure for classification was as follows:

1. Studies with the score from (0–1.5) are considered as low rigor, while high rigor is defined for a score of 2 or above.
2. Studies with the score from (0–2) are considered as low relevance, while high relevance covers scores from 2.5 or above.

Fig. 5 Mind Map of OI in SE

We classified 17 studies as having the highest rigor and relevance, see area A in Figure 6, and these results are the most trustworthy. Moreover, we classified 12 studies into C category of studies with high relevance but low rigor. On the other hand, categories B and D contain two studies each and in for both categories the relevance scores were higher than the rigor scores, see Table 7. The identified themes are presented in the subsections below, sorted according to the number of categorized studies.

Fig. 6 Categorization of studies based on rigor and relevance

4.3.1 OI Strategies/Instruments

The software industry is characterized by frequent technological changes which force large incumbent firms to more rapidly innovate their strategies in the pursuit to maintain their current revenue levels. OI strategies focus on how innovation networks and strategies can be used to participate, orchestrate or govern this technologically unstable environment.

Research and development (R&D) collaboration strategies seem to help organizations to attract and establish communities and to stay competitive. This strategy is also visible among the firms that adopt OI to enhance their innovation process in nine primary studies (**S_3**, **S_5**, **S_6**, **S_13**, **S_15**, **S_19**, **S_25**, **S_26**, **S_29**). Six out of these studies (**S_3**, **S_6**, **S_13**, **S_15**, **S_25**, **S_29**) were conducted with high rigor and relevance, see category A in Figure 6, which increases their trustworthiness. On the other hand, the remaining three studies (**S_5**, **S_19**, **S_26**) were classified into category C which indicate that the studies have relative low rigor but still their results are highly relevant.

Looking at the primary studies with high rigor and relevance scores, the results of one study (**S_3**) indicated that firm's human capital affects the adoption of OI business strategy among the Finnish software companies. Consequently, the companies that have larger academically educated staff apply OI business strategies more often. Harison and Koski (**S_3**) stated the reason for that is the ties between the OSS communities and universities. Smaller companies (start-ups) tend to apply more open innovation strategies compared to large and older firms. This interpretation seems reasonable since smaller com-

panies often leverage OSS to acquire knowledge and substitute of a comparable depth as for the in-house R&D capabilities that they lack. Overall evidence suggests that a more positive attitude towards openness enables firms to better share in the benefits of open innovation processes (**S.6**).

In a study about implementing a private collective model at Nokia (**S.13**), a number of mitigation strategies were adopted. Nokia had the evidence of their competitors using their source code, therefore, they partially revealed their source code to retain control and information, and future plans leakage was protected through non-disclosure agreements. Moreover, the development control was compromised by involving communities, hiring key developers and upstream participation, which resulted in no single vendor being able to control the platform. Besides that, Nokia opened up and communicated the structure of its internal processes.

Dahlander and Magnusson (**S.15**) highlight that in order to address the emerging challenges of the public-private development model, such as attracting outsiders to work in their community, companies are releasing the code under open source licenses and in this way are establishing new communities or using existing communities. At the same time, companies often adopt licensing practices that clarify ownership, devoting resources to evaluate source code and give feedback on source code to communities.

One of the main conclusions of Grøtnes' study (**S.29**) is that the open innovation takes place in neutral arenas like standardization, and outside-in, inside-out and coupled processes are used to create new technological platforms. A more restricted membership gives a separate outside-in and inside-out process while open membership leads to a coupled process. A key difference can be explained by the example of Android that was available for invited firms only, while open membership is open for all. Open membership creates a modular innovation that embeds new radical innovations like mobile TV, while Android creates an architectural innovation with possibilities for further radical innovations.

Similarly, Deutsche Telekom (**S.25**) used Foresight workshops, executive forums, Customer integration, Endowed chairs (opening doors to academia world), Consortia projects (cost sharing of complex projects), Corporate Venture Capitalist (window to innovation in the start-up community and technology sourcing through co-investing), Internet platforms, Joined development, strategic alliances, spin-outs (external commercialization of internal R&D results in technologies, products or services) and test market (equipping a city with next generation infrastructure) to take advantage of open innovation, see Figure 5.

Looking at the studies performed with less rigor, West and Gallagher (**S.5**) argued that companies employing strategies such as pooled R&D/product development (firms sharing the R&D), spin-outs and selling complement and attracting donated complements, easier overcome the following challenges: 1) the generation and contribution of external knowledge (motivating), 2) incorporating the external innovation into firms resources and capabilities (incorpo-

rating), 3) diversifying the exploitation of intellectual property (IP) resources (maximizing).

The most noted example of pooled R&D is the Mozilla project, initiated by Netscape in response to the competitive pressure from Microsoft Internet Explorer (IE). Vendors such as IBM, HP and Sun needed a Unix based browser to increase sales of Internet connected workstations and therefore donated some of their IPs to the open source development lab (OSDL), while exploiting the common advantages of all the contributors to expedite the sale of related products. Similarly, spin-out (shared R&D between firms and a community) can also release the potential IP from the firm that is not creating the value anymore. Thereby, the firms transform internal development projects to externally visible open source projects.

Consequently, the donated IP generates demands for other products and services that (donor) firms continued to sell. Examples of spin-out is when IBM promotes the Java programming language, developed by Sun Microsystems, to compete with Microsoft. IBM was still able to generate revenue from sales of hardware and supporting services in the Java centric world. Selling complements is used by firms to build upon the already existing products and succeed through differentiation strategy and in contrast, donating complements are more feasible when selling to technically professional buyers, capable of making modification and improvements, such as hobbyist programmers or corporate engineers.

In addition, Dittrich and Duysters (**S_19**) also addressed the difference between exploration (seeking radical innovation) and exploitation (seeking incremental innovation) strategies adopted by firms to sustain their position in rapidly changing technological environments. Exploration networks make use of flexible legal organizational structures, whereas exploitation alliances are associated with legal structures that enable long-term collaboration. Nokia followed an exploitation (incremental innovation) strategy in the development of the first two generations of mobile telephony devices, and an exploration (radical) strategy in the development of technologies for the third generation. Such inter-firm networks seem to offer flexibility, speed, innovation, and the ability to adjust smoothly to changing market conditions and new strategic opportunities.

While studying the case of embedded Linux (**S_26**) Henkel found that hobbyists and developers in universities reveal nearly all of the code in contrast to companies. In particular, the more important it is to obtain external development support, the more code the respective firms reveal.

4.3.2 Challenges

This theme highlights challenges (**S_4**, **S_9**, **S_12**, **S_14**, **S_15**, **S_21**, **S_24**, **S_13**) when firms try to adopt open innovation, summarized in Table 4. The identified challenges are categorized into *business* and *process* related challenges. Business related challenges refer to business strategy (**S_9**, **S_13**,

S_14), entry barriers (**S_15**, **S_21**) and governance (**S_12**, **S_24**). Governance refers to establishing measurement and control mechanisms to enable project managers and software developers within the communities as well as others within a software development organization, to carry out their roles and responsibilities [6]. Process related challenges consider hinders in strategy realization.

As can be seen in Table 4, business and process level challenges are considered to be major hindering factors for the adoption of OI. Finding the right balance between contributing to community and reaping benefits at the same time is tough, and thus results in unclear business strategy (**S_14**). One of the biggest concerns is the difficulty in differentiation if a firm indulge itself in an OSS solution and guard its business secrets because its competitors have the same solution available for their products (**S_14**). Other challenges are: managing the conflicting needs (**S_9**) of all players involved in the process, aligning the firm's strategy with community (**S_15**) and achieving a common vision (**S_12**). Even if a firm has a clear business strategy where it manages to resolve the often conflicting stakeholders' needs, the challenge of community build up and survival remains (**S_21**). Therefore, firms and communities need to find the right balance of governance(**S_13**).

On the other hand, process related challenges are also negatively impacting OI. For instance, Conboy and Morgan (**S_4**) suggest that agile and OI do not get along well, especially when dealing with the management of innovative requirements and release planning. Agile requirements backlogs do not have room for innovative requirements since short iterations, a feature backlog and stand up meetings make it extremely tough to play around or share ideas outside your team. The lack of control over release planning was also pointed out as a challenge in a study (**S_11**), for example, sometimes it is a better business decision to adopt the open source code, perform minimum changes, and sell it instead of spending time on developing differentiation features. This raises a question whether or not firms should have a separate requirements management process for innovative features (**S_11**), but nevertheless there is an inherent complexity in requirement management process while managing innovative features. Further process challenges include the lack of clear roadmaps for product highly dependent on OSS platforms and overcoming the "not invented here" problem.

The majority of the primary studies highlighting the challenges of open innovation lie in categories A (**S_15**, **S_24**, **S_13**, **S_14**) and C (**S_4**, **S_9**, **S_12**) suggesting that results are highly relevant to industry except of the one study (**S_21**) in category D.

4.3.3 Benefits

This category highlights the OI adoption benefits in terms of positive impacts associated with the inside-out, outside-in, coupled processes and the private collective model (**S_12**, **S_13**, **S_20**, **S_24**, **S_26**, **S_31**, **S_10**, **S_23**). The benefits are summarized in Table 5. As far as the strength of evidence is

Table 4 OI challenges categorized in business and process themes.

Facets	Challenges
Business (S_9, S_12, S_13, S_14, S_15, S_21, S_24)	Business strategy
	– Unclear content and contribution strategy (S_14)
	– Contribution time-line unclear (S_14)
	– Minimize modifications to the open source code (S_14)
	– Unclear relationship between the benefits from contributions in terms of strategy and business goals (S_14)
	– Be strategic when adopting innovative features (S_14)
	– Balancing the interests of those participants against those of the ecosystem leader (S_9)
	– Difficulty to differentiate (S_13)
	– Guarding business secrets (S_13)
	– Definition of core competencies (S_21)
	– Legal and property rights issues concerning the external knowledge (S_21)
Process (S_4, S_9, S_14, S_15, S_21, S_24)	Strategic OI entry barrier
	– Accessing communities to extend the resource (S_15)
	– Reducing community entry barriers base (S_13)
	– Aligning firm strategies with the community (S_15)
	– Community build-up and management (S_21)
	– Achieving a common vision (S_12)
	– Finding staff/ Competencies (S_24)
	– Lack of expertise (S_24)
	Governance
	– Expectation management of community (S_21)
Process (S_4, S_9, S_14, S_15, S_21, S_24)	– Increasing knowledge sharing and exchange (S_12)
	– Achieving a high level of commitment (S_12)
	– Giving up control (S_13)
	– Lack of support (S_24)
	– Lack of ownership (S_24)
	Agile processes
	– The new approach caused significant problems in terms of transferring the ideas outside the team (S_4)
	– Visibility as to what the new [agile] team were doing dropped quickly. The introduction of agile coincided with a rapid drop in the number of developers from that team attending the overall R&D meetings. (S_4)
	– The use of short iterations, a feature backlog and stand-up meetings reduced the amount of time you can spend playing around or sharing ideas outside your team (S_4)
	– Motivating the generation and contribution of external knowledge (Motivating) (S_4)
Process (S_4, S_9, S_14, S_15, S_21, S_24)	– Incorporating external innovation into firms resources and capabilities (Incorporating) (S_4)
	– Diversifying the exploitation of intellectual property (IP) resources (Maximizing)(S_4)
	Relation between process and innovation
	– Augmenting the requirements management process (S_14)
	– Manage innovative features in a separate process (S_14)
	– Top-down or bottom-up open innovation (S_14)
	Release planning and prioritization
	– Prioritization process needs modification (S_14)
	– Challenging acceptance criteria kills innovative features (S_14)
	– Need for special flow for innovative features to evolve to meet acceptance criteria (S_14)
Process (S_4, S_9, S_14, S_15, S_21, S_24)	– Release planning even more challenging (S_14)
	– Prioritizing the conflicting needs of heterogeneous ecosystem participants (S_9)
	– Assimilating communities in order to integrate and share results (S_15)
	– Efficient process management (S_21)
	– Lack of Road-maps with OSS Products (S_24)
	– Overcoming Not Invented Here (S_21)

concerned, five papers (S_13, S_24, S_31, S_10, S_23) lie in category A and two studies (S_12, S_26) fall into category C. The fact that only one study (S_20) has low rigor and relevance suggests that the identified OI adaption benefits are highly relevant for industry.

Table 5 OI Benefits categorized in business and process themes

Facets	Benefits
Process (S.12, S.13, S.20, S.24, S.31)	Knowledge building and exchange <ul style="list-style-type: none"> – Knowledge sharing and exchange (S.12) – Low knowledge protection costs (S.13) – Easy access to all information (S.12) – Increases organizational learning (S.12) – Improves collaboration with groups in Europe, USA, India (S.12) – Analyzing the firms revealing behavior, roughly one third of the firms (21) reveal no source code at all, another third (22) reveal an amount between 0% and 100%, and the rest (24) reveal all their drivers. Customer demand for source code has a significant (5%), positive effect on the decision to reveal at all (S.31)
	Platform and reuse <ul style="list-style-type: none"> – Improves platform use (S.12) – Promotes software reuse (S.12) – Increases trust in platform (S.12)
	Communication <ul style="list-style-type: none"> – Direct communications (S.12) – Supporting this process in an existing social network site lowers the hurdles for expressing and communicating ideas (S.20)
	Involvement and innovation support <ul style="list-style-type: none"> – Improves involvement of product teams (S.12) – Improves feedback by being open (S.12) – Avoidance of duplicate work (S.12) – Empowers developers and project leaders (S.12) – It brings together people with similar interests, but also introduces diverse people to each other, adding more heterogeneous viewpoints to ideas (S.20) – The process acts as a catalyst for ideas: while it does not help with the initial conception of an idea, it makes all following steps easier (S.20) – Executing the process successfully might result in the realization of ideas, providing more choices to consumers and possibly broadening the offerings of companies (S.20) – Developer/Tester Base (S.24) – Flexibility of use (S.24)
Business (S.10, S.12, S.13, S.23, S.24, S.26, S.31)	Time to market, cost, maintenance and efficiency <ul style="list-style-type: none"> – Reduces time to market (S.12) – Cost savings (S.12) – Increases efficiency in development (S.12) – Reduced maintenance effort (S.26) – Bug fixes by others (S.26) – Small firms reveal significantly more, likely because, due to resource scarcity, they expect to benefit more from external development support (S.26) – Further development by others (S.26)
	Innovation <ul style="list-style-type: none"> – Increases innovative capacity and speed (S.12) – Adoption of innovation (S.13) – Increased innovation at lower costs (S.13) – Encourages innovation (S.24) – The open innovation practice of technology scouting is positively associated to the SME's innovative performance (S.10) – Communities provide SME's with a munificent environment, rich of free-of-charge resources. Hence, communities offer an ideal context to investigate the mechanisms allowing SME's to access the resources of their open innovation partners and use them for diversification (S.23) – Increases collaboration (S.24)
	Improved competitiveness and other business gains <ul style="list-style-type: none"> – Extra business functionality (S.24) – Improves adoption rate of the platform (S.12) – Findings highlight a new competitive weapon for managers in companies that are not technology or market leaders. By being open, some of these companies may be able to significantly increase their competitiveness (S.31) – Reputation gain (S.13) – Revealing good code improves our company technical reputation (S.26) – Distribute ownership and control (S.12) – Learning effects (S.13) – De-facto standards (S.24)
	Culture change <ul style="list-style-type: none"> – Public success stories might create a culture of innovation spanning the whole social network site (S.20) – Experience with embedded Linux has a negative and revealing experience a positive effect (both at the 1% level in (2), and 10% and 5%, respectively. The likelihood that a firm reveals all of its drivers is positively related to the importance of technical benefits (1%) (S.31) – Findings suggest that external factors are less, and firm characteristics more important for this type of selective revealing. Bi-directional openness seems to require a much stronger cultural heritage or change than the simple disclosure of source code on the web (S.31)

Benefits are divided into the process and business related, see Table 5. OI allows firms to find a pool of skilled labor outside their boundaries without a significant cost. This external labor provides feedback and enables knowledge exchange between the community and the firms (S_12). Organizational learning is another important benefit, where OI often gathers diverse people with similar interests, adding more heterogeneous viewpoints to ideas (S_12). However, it is to be noticed that OI does not help with initial conception of an idea; rather it acts as a catalyst for ideas, and might also result in the idea realization. Consequently, OI provides opportunities to offer more choices to consumers and possibly broaden the firms' offerings. Furthermore, knowledge sharing and exchange lead to avoidance of duplicate work and encourages software reuse. Analyzing behavior of firms unveil that one third of the firms reveal no source code at all, and another one third of the firms reveal an amount between 0 to 100 %, while the remaining firms reveal all their source code. Customer demands are reported as the key factor that causes the firms to reveal the source code (S_31).

OI also brings business advantages, outlined in Table 5. OI involvement enables efficient development processes (S_12), reduces development cost (S_12), and increases innovation capacity (S_12). OI can also help to reduce time to market and can permit firms to build and maintain a good reputation from code revealing (S_13), public success stories and innovation culture (S_20). Finally, findings suggest that by being open, companies can significantly increase their competitive advantage and managers from the companies that are not market leaders may consider it as the competitive weapon against their competitors (S_12).

4.3.4 Enabling OI communities

This theme refers to communities as distributed groups of individuals, aiming at solving a general problem and/or developing a new solution supported by computer mediated communication. The solutions developed in the community can be used in conjunction with the firms' internal capability to develop competitive services and products. In particular, this theme uncovers strategies adopted by firms to use communities as complementary assets, positive impacts of the community on firms' innovation and challenges associated with it (S_1, S_6, S_8, S_21, S_33), see summary in Table 6.

As can be seen in Table 6, firms exploiting communities in their innovation process not only gain a good reputation but also influence the direction of development and legitimate the use of projects (S_1, S_6). Having an employee in the community seems to be the key to enabler of these advantages. Thus, companies use employee referrals or offer individuals monetary rewards to exploit communities. Besides, initiating OSS projects is an alternative way for attracting pragmatic OSS developers (S_6).

Using the wisdom of crowds, and direct articulation of user wishes (S_8, S_21) help organizations receive new features from communities instead of extensions of already existing features. Albeit OI can bring benefits to both

Table 6 Studies in the OI community theme

Ref	Positive Impacts	Negative findings	Strategies
S.1	<ul style="list-style-type: none"> Creates good reputation Legitimizes the use of the project Companies can influence the development direction for these communities 	<ul style="list-style-type: none"> No clear evidence that firm sponsored individuals are able to orchestrate or stimulate debate within these communities Individuals with affiliations with large incumbents in the software industry have no significant effect in the community Other software companies may not devote their best employees to working in the community or may only passively screen developments 	<ul style="list-style-type: none"> A man on the inside to be able to gain access to communities
S.6	<ul style="list-style-type: none"> A more positive attitude towards revealing will enable firms to better share in the benefits of open innovation processes 	<ul style="list-style-type: none"> Too open behavior by firms programmers would be commercially harmful Management is not always informed about this sharing and has broad, but nonetheless limited means of monitoring it Management may overestimate the risk of critical code leaking out 	<ul style="list-style-type: none"> Sponsor provides monetary rewards to contributors Employee referrals to attract contributors The focal firm might consider launching its own public OSS project in order to attract pragmatic OSS developers
S.8	<ul style="list-style-type: none"> Feature gifts (new features instead of extension of existing features) 	<ul style="list-style-type: none"> N/A 	<ul style="list-style-type: none"> Participants having an activity (i.e. report bugs, offer bug fix etc.) are more likely to be granted access to the developer community
S.21	<ul style="list-style-type: none"> Increase of ideators and therefore ideas Strong customer orientation since users can articulate wishes directly Possibility to use wisdom of the crowds to handle high number of ideas New forms of evaluation with better results More and better ideas, concepts and products Increase in efficiency and effectivity 	<ul style="list-style-type: none"> Community build-up and management Overcoming Not Invented Here Technical realization of external interfaces Legal and property rights issues concerning the external knowledge Expectation management of community Definition of core competencies Efficient process management 	<ul style="list-style-type: none"> N/A
S.33	<ul style="list-style-type: none"> OI communities produce complementary assets that are of significant value to firms Provide firms with the tacit knowledge to address some of the tensions that arise in firm-community interaction Limited resource base of small firms exerts a <i>ceiling-effect</i> on the optimal level of community involvement Above-average levels of technical community participation limit the financial performance of small OSS firms for small firms, initial increases in involvement in the communities has a positive impact on their financial performance 	<ul style="list-style-type: none"> Contributing entails significant costs in terms of resource Investments and loss of strategic assets that may result in decreasing returns 	<ul style="list-style-type: none"> Internal and external sources of innovation are complements rather than substitutes

small and large companies, small firms with limited resources exert a ceiling effect on community involvement. OI should, in those cases, be used as a complementary asset to accelerate internal innovation and R&D processes of the organization (**S_33**). On the other hand, OI does have its cost when companies might procure the outcome of the community participation, but at the same time not be willing to devote their best resources to work in the communities (**S_1**, **S_6**). In addition, it remains unclear how to orchestrate or stimulate debates within communities, thereby making it hard for firms to achieve their goals. Consequently, too open behavior might be potentially harmful and contributions without selective revealing strategy could entail significant cost in terms of programming resources and loss of strategic assets that may result in decreasing returns (**S_21**, **S_33**). As far as the trustworthiness of results of these studies is concerned based on rigor and relevance (see Figure 6), 5 studies (**S_1**, **S_6**, **S_8**, **S_33**) lie in category A except for one study (**S_21**) in category D.

4.3.5 Managerial implications

This category includes six studies that focus on the recommendations for managers how and when to indulge in open innovation, in order to increase firms innovative performance (**S_7**, **S_15**, **S_17**, **S_22**, **S_32**, **S_33**), see Figure 4. The primary study (**S_17**) suggests that firms working in open source settings can pursue differentiated strategies to achieve openness, without really distancing their developers from the communities. The openness is most realized at the component level and differs significantly between software and hardware components. Openness of software seems to be more important to the community than openness of hardware. Thus, companies may get involved in open source software initiatives and secure their competitive position by capturing more value or differentiation in hardware. Managers could enhance the degree of innovation and performance of their firms in a number of ways. Among them, firms should consider getting access to skilled resources, and learning by encouraging their employees to participate in the communities, instead of free riding (**S_33**).

Moreover, firms operating in hostile environments, motivate managers to draw knowledge out of end users and communities (**S_32**). However, most often it is not a straightforward decision for managers to participate in communities or draw knowledge from end users. A survey conducted in Dutch software industry revealed that managers are confronted with too little available time, resources, lack of commitment, and often the wrong strategy to indulge themselves in the communities (**S_22**). Furthermore, firms need to develop sufficient absorptive capacity to benefit from external knowledge and to find interesting tasks for community participants to keep them motivated (**S_15**). To underline the strengths of the evidence, Figure 6 depicts four studies (**S_17**, **S_33**, **S_32**, **S_15**) classified in category A with high rigor and relevance, while two studies (**S_7**, **S_22**) fall in category C that have high industry relevance but relatively low rigor.

4.3.6 OI Models/Framework

This theme includes the models or frameworks (**S_7**, **S_11**, **S_13**, **S_20**, **S_27**). As far as the mapping of these studies with rigor and relevance is concerned, two studies (**S_27**, **S_13**) lie in category A and three studies (**S_7**, **S_11**, **S_20**) fall into category C, B and D respectively, see Figure 6. Jansen et al. presents an open software enterprise model (OSE) for determining the openness of a software producing organization (**S_27**). An organization can choose to be open on both supply and demand sides of the supply chain. This happens typically by opening up development on the side of software developers and contributors, or opening up service delivery on the side of service partners who deploy, configure, and service the software platform produced by the organization. However, the paper lacks clear guidelines how to execute these activities using software engineering techniques or processes.

Stuermer et al. (**S_13**) focuses on the private-collective innovation model which proposes incentives for individuals and firms to privately invest resources to create public goods innovations. Such innovations are characterized by non-exclusivity and non-rivalry in consumption. Stuermer et al. examined Nokia's Internet Tablet development and identified five hidden costs for the company involved: difficulty to differentiate, guarding business secrets, reducing community entry barriers, giving up control, and organizational inertia.

Ebner et al. (**S_7**) highlight the idea of competition as a method to nurture a virtual community for innovations. Similarly, Wnuk et al. (**S_11**) proposed a software engineering framework, designed to foster open innovation by designing and tailoring appropriate software engineering methods and tools. The framework is divided into the technical (e.g. requirements engineering, software design, development and testing techniques etc.) and methodological dimensions. Singer et al. (**S_20**) envisions a 7 step innovation process as a conceptual solution. The process covers an idea life-cycle from its creation to its realization and is exemplified on an IT-related example.

4.3.7 Degree of openness

Openness of software producing organization in studies is explicated by revealing of the proprietary information. Existing and potential intellectual property rights are voluntarily given up to interested parties in order to make them accessible. This theme comprises three studies that not only contain different forms of open strategies (**S_16**, **S_17**), but also presents an open enterprise software (OSE) model developed in order to assess the openness of organizations (**S_27**). As far as strategies for openness are concerned, West (**S_16**) claims that proprietary platforms are more suitable for market leaders and open standards are more feasible when propriety strategies fail. Besides, differentiation can be achieved through opening some parts, by disclosing technology under such conditions that it will only provide value to customers, without really giving away the advantage to competitors. Open source pro-

vides direct benefits to many users who lack the requisite technical skills to do their own development.

Balka et al. (S_17) state that transparency, accessibility and replicability are important to open design communities. They present an open software enterprise model that suggests that openness can quickly create critical mass of developers or partners around the software product if the surrounding partners are prepared to enter in the ecosystem in any of roles, such as developers, values added resellers, service partners or customers. However, Balka et al. also suggest that openness is not always beneficial to the organization and mention the role of partnerships in software producing organizations as a form of openness (S_17). It is also noticed that openness often leads to creation of new business models. All above mentioned results carry more industry relevance, since two studies (S_17, S_27) lie in category A, and one study (S_16) in category C with high relevance and low rigor.

4.3.8 Intellectual property (IP) strategies

This theme refers to strategies used by firms to share IP among stakeholders in the OI context. Rayna and Striukova (S_18) investigated open source vs. patent pools as innovation structures, however the study has low relevance according to Figure 6. Patent pools are comprised of multi-party ownership and include not only current patents, but may also include future changes to these patents. Typically, all patents in a patent pool are available to each member of the pool. In contrast, the open source structure is based on the copy-left paradigm instead of intellectual monopoly rent paradigm, where the source code as well as any subsequent modifications and improvements are released, not only to the members of the project, but to the whole community. This study (S_18) compares two OI structures in terms of risks, cooperation, financial/non-financial benefits, standards and their feasibility.

Rayna and Striukova (S_18) argue that patent pools and open source have common risks and benefits. For instance, the key risks are associated with intellectual property right (IPR) infringement, bad publicity and discouraged further investment. On the other hand, benefits can be reaped in terms of decreased R&D expenditure and transaction cost, access to skilled resources and increased future business opportunities and reputation. Besides that, open source is exclusive in application but universal in access while patent pools are universal in application but exclusive in access. Therefore, it is more suitable for large companies to initiate or adopt patent pools compared to small companies or start-ups. Small companies may find additional benefits in terms of having a chance to set a standard in open source and give them access to highly skilled work force, and thereby reduce the development cost. Finally, patent pools are often formed based on prior knowledge unlike open source that generates new knowledge based on skills and competences.

4.3.9 OI toolkits

This theme includes the toolkits developed in order to involve end users into firms' internal innovation process. Given that international firms often operate in hostile environments, limited evidence (**S_2**, **S_28**, **S_32**) was found related to the use of user innovation toolkits and its impacts on firms innovative performance. As far as the strength of evidence is concerned, one study (**S_32**) was found in category A and remaining two studies (**S_2**, **S_28**) fall into category C, see Figure 6. Wang et al. concluded that innovation toolkits improve the innovation outcome and productivity for users with knowledge and experience. We identified only one toolkit, namely *INOVEX* (**S_28**), that is used by software producing organizations to extract knowledge from end users. When it comes to utilizing the end user knowledge, evidence suggests that larger firms seem to exploit end users online less than the smaller firms (**S_32**). This could be due to the fact that small firms are having more open search strategies caused by a lack of skilled resources and the need to reduce the development cost.

5 Discussion

The synthesized evidence in this study suggests that smaller companies (start-ups) have higher tendency to adopt OI compared to incumbents. This trend makes sense when we consider start-ups engaging themselves in OSS solution in order to quickly acquire knowledge and R&D capabilities within limited resources. OI provides initial financial gains for small companies, but limits their financial performance when the level of participation increases above the average. Thus, in order to reap the financial benefits, it seems important to have high absorptive capacity to properly catch the technical know-how from the available knowledge. Large companies should encourage their developers to participate in communities for improved knowledge sharing and for obtaining heterogeneous viewpoints on ideas.

The primary studies classified in the category OI strategies (Section 4.3.1) indicate that both small and large companies explore the potential in OI, but in different ways. For smaller companies, adapting OSS solutions seems to provide the most benefits, while larger companies also benefit from adapting their code ownership strategies and internally adapt OSS practices via so called inner-sourcing [28]. Therefore, it is possible to hypothesize that larger companies should dedicate more effort into the analysis of possible OI strategies and options. Moreover, companies that own implemented assets have more possibilities to capitalize their innovative potential via OI strategies, than companies that have no implemented assets. Still, for companies owning only intangible innovations, there exist strategies to share these assets via, for example, pooled IPR forums.

The primary studies summarized in the enabling communities for open innovation category (Section 4.3.4) lead to an interpretation that communi-

ties offer significant benefits that companies should exploit. In particular, it seems that initiating OSS projects is equally important from the OI perspective as joining or governing an OSS project or the entire ecosystem. It remains an important aspect to further explore what strategy is optimal, given the company's size, domain and product characteristics. Furthermore, our results suggest that firms are able to influence the direction of development (governance) in communities to some extent, with one exception. Companies that sponsor individuals in their involvement in OSS projects were not able to effectively stimulate or orchestrate debates in these projects. One important reason could be that communities believe that companies have their vested interests in participation. This could explain the difference between OSS and OI well, where in OI organizations decide to open up when they see a potential benefit in opening up, while in OSS the community contributes with the mind set of *free software ideology* without expecting any benefits in return.

The results regarding the interplay between OI and agile methods provide interesting interpretations. It seems that openness is often compromised due to lack of transparency between competitors, and even business units within an organization. Combining agile and OI seems to create barriers in transferring the ideas outside the team's boundaries, primarily due to the use of short iterations, minimum documentation, stand-up meeting, and a feature backlog that reduces the amount of time *you can spend trying new things or sharing ideas outside your team*. The resulting lack of overall R&D group overview disables the innovation opportunities when using agile practices (S_4). Further, the introduction of agile with OI caused a rapid decline in teams attending R&D meetings, due to the lack of tolerance for prolonged meetings as stated by senior anonymous developers, *using the old plan-driven approach we would have been going to meeting after meeting, but since going to agile, every minute you spend in one of these meetings you just think about all of the work not being done* (S_4).

To further demonstrate the challenges of OI in the agile context, developers quoted that *on-site customer practice seem to be the most telling barrier since you feel accountable to person there at all time and its harder to justify taking a half day out to sit with folks in other projects for benefits of other customers*. At the same time, managers experienced lower quality of ideas due to focus on daily work.

Managerial implications is another aspect worth discussing (Section 4.3.5). Managers can enhance the financial situation and innovativeness of their firms, by encouraging their employees to participate in communities. To gain further advantage, managers can consider the learning and resource advantages attached to community participation, instead of just free riding. The identified evidence suggests that participation is more strongly related to the performance of those firms that exhibit high level of social participation. However, the literature also underlines the inherent complexity for organization to initiate, build and nurture an external community as a complementary asset to their internal R&D process. To be more specific, managers have too few re-

sources available in order to indulge them in communities. This may lead to too little time and commitment from resources to contribute in the communities.

Business strategies also play an important role in embracing open innovation, thus companies can pursue differentiation strategies with the controlled degree of openness towards communities. Nonetheless, transparency and accessibility are important factors when talking about openness of firms. Consequently, from the firms' point of view, OI does not substitute the already existing R&D process, but it complements the existing internal innovation processes.

Regarding OI models or frameworks, fostering competing ideas seems to be promising. At the same time, companies may use social networks to lower the hurdle of sharing ideas, but since the primary study (S_20) presents preliminary work and therefore lacks rigor and relevance, more empirical research is needed to ensure that. Similarly, the framework presented by Wnuk and Runeson (S_11) is preliminary and lacks specific guidelines about which SE techniques are applicable for which contexts.

When it comes to the benefits and challenges of applying a collective innovation model (S_20) there is a need for further studies that directly connect benefits and challenges with SE techniques, as the current evidence is incomplete and largely anecdotal. Similarly, Jansen et al. (S_27) describe in great detail *what* to do rather than *how* to do it, especially on the operational level, where appropriate SE techniques can provide great support. To summarize, there seems to be a lot of interesting techniques or processes that foster OI, but the ways how to operationalize them remain unspecified and requires further research effort.

When looking at the results in the IP strategies theme, it appears that patent pools is an alternative solution for the companies that may not necessary have innovation implemented in software (S_18). Both patent pools, and OSS share many benefits and challenges, but differ in that OSS provide universal access but is exclusive in application, while patent pools restrict the access but enable application. Thus, large companies should use their IPR capital for enabling OI via patent pools.

The results indicate little research focus on the OI toolkits since only one toolkit was found among the primary studies. Moreover, primary studies suggest that extensive experience is required to unlock the full potential of these toolkits. Therefore, it remains to be explored how to enable less experienced practitioners to be more innovative and in this way to leverage their innovative potential. We believe that enabling newcomers is important to fully benefit from OI, since many OI contexts are characterized with high turnover for contributors that often contribute once in a project.

6 Implications for Research and Practice

6.1 Research Agenda for Open Innovation in Software Engineering

In line with the advice by Kitchenham et al. [36], we use the systematic mapping study to derive an agenda for further research. We interpret the increased scholarly interest in OI since the launch of Chesbrough's book in 2003 as a sign of increased importance of OI. Still the number of publications that focus on OI in SE remain small and therefore we believe that focusing on OI in SE should be highlighted on the research agenda in SE. In particular, based on the results our interpretation, the following areas should be put on the research agenda:

- Further exploring suitable software development methodologies that foster OI. The results outlined in Section 4.3.2 suggest that combining agile and OI provides additional challenges that may have a ceiling effect on the potential benefits from OI. Thus, it is important to direct research efforts into better understanding of which development methods or processes best suits OI and what changes need to be implemented to unlock OI's full potential.
- Providing clear managerial guidelines on how to adapt OI depending on the context factors, with a special focus on which SE techniques, processes and methods can be applied depending on the selected managerial strategy. In this way the findings reported in Sections 4.3.1 and 4.3.5 will be complemented by guidelines on the operational level to form more complete solutions for adapting to OI.
- Exploring the balance between community involvement and in-house SE activities. This study identified several benefits from OI and OSS community involvement, see Section 4.3.3. However, the process-related benefits should be further explored, with a focus on uncovering where involvement brings most benefits. In particular, the role of OI involvement in improved testing remains unexplored, where we believe that OI provides not only significant reduction of the test effort but also can be a source of innovation. We base this assumption on a premise that testing uncovers unexpected behavior of software, which could be inspirational in the innovation process.
- Focusing on the role of requirements engineering in OI both during and beyond innovation discovery. OI offers access to a wide and heterogeneous communities of potential stakeholders which puts pressure on the current techniques for key stakeholder identification and domain understanding. Advances in current techniques are required for supporting the identification of commodity and competitive advantage requirements sources. Beyond innovation discovery, there is a need for a decision making support that can combine both strategic and operational levels and provide run-time requirements triage support for capturing and incorporating OI potential into product planning and requirements decision making. Despite that, researching if unimplemented requirements that represent valuable

IPRs, can be shared with others in a similar way as for example patents, and what benefits this approach brings is important.

- We encourage researchers to develop and publish more solution and validation research in OI as these remain underrepresented, see Figure 3. The large number of evaluation research is definitely positive but, at the same time, highlights the immaturity of the OI in SE research area. Thus, more solutions in terms of tool proposals or frameworks and their validations are needed to advance to the next maturity level.

6.2 Implications for industry practice

Although we summarize the empirical evidence in the field of OI in SE being scarce, there is some evidence that may be used to guide software companies in their innovation strategies:

- The identified conflict between agile and OI principles should be given special attention. Agile principles focus developers attention and communication in order to meet specific project goals. However, the innovation process benefits from the noise of leaks in the information flow from multiple sources, internal as well as external. Companies should make sure that this information flow is regained, using other practices.
- Open innovation strategies seem to be more beneficial for smaller and newer actors in a market. They may apply OI and thus can gain significant competitive advantage against competitors by more quickly absorbing potential innovation. However, there are also examples of major corporations that manage a software ecosystem, based on open or semi-open innovation. The take-away for companies is that they need to define and monitor their OI strategy to make sure their actions are relevant, given their current and future expected market position.
- An implication for industry, based on the literature findings, is that OSS and OI is *not* for free. In order to gain the full and long term benefits from OI, companies must invest in the open communities, and since these are complex networks with a multitude of actors, these companies must have a clear resources investment plan, just as they need for closed innovations.
- IPR management is different for OI. The studied research recommend large companies using patent pools to manage their IPR capital in relation to the open innovation community.

7 Conclusions

Open innovation (OI) becomes significantly important for companies developing software-intensive products and services. It provides several benefits that force these companies to re-think and often significantly change their current innovation strategies. The external availability of innovations combined with the flexibility of their realization generate potential for providing value to the

customers. OI pushes software industry into a new ground where well-known and checked software development and management strategies need to be revisited. At the same time, OI remains greatly unexplored in the SE literature, focusing greatly on exploring OSS, resulting in a lack of systematic efforts to summarize OI literature in relation to software engineering.

We conducted a systematic mapping study on open innovation in software engineering with the aim to identify the existing themes in the literature and evaluate them based on the rigor and relevance analysis.

Answering research question *RQ1* we identified nine themes in the surveyed literature. The dominant themes are related to OI strategies, OI challenges and benefits, enabling OI communities, managerial implications of adaption OI and OI models or frameworks. The degree of openness, OI toolkits and IP strategies are less frequently represented in the surveyed papers.

Our findings for *RQ2* suggest that a majority of the studies are conducted with high rigor and high relevance (17/33) and as many as 29/33 were considered industry relevant. This strongly indicates that OI in SE is industry practice oriented. Further, 27/33 studies are of evaluation type, which is unusually high for a mapping topic. Therefore, we encourage more solution and validation research, see Table 7. The high rigor and relevance also implies generalizability of the results derived from thematic analysis in answer to *RQ1*.

This mapping study leads to a proposal to further explore the field of SE to foster OI, for example, development methodologies that interplay with OI, situated managerial guidelines for OI adaptation, as well as exploring the balance between open community involvement and in-house development. Specifically, the roles of testing and requirements engineering in OI remain unexplored.

Acknowledgement

This work is funded by the Swedish National Science Foundation Framework Grant for Strategic Research in Information and Communication Technology, project Synergies (Synthesis of a Software Engineering Framework for Open Innovation through Empirical Research), grant 621-2012-5354.

References

1. Robert C. Allen. Collective invention. *Journal of Economic Behaviour and Organization*, 4(1):1 – 24, 1983.
2. Kerstin Balka, Christina Raasch, and Cornelius Herstatt. How open is open source? - software and beyond. *Creativity and Innovation Management*, 19(3):248–56, 2010*.
3. Fiona Beyer and Kath Wright. Comprehensive searching for systematic reviews: a comparison of database performance. Technical report, Centre for Reviews and Dissemination, The University of York, 2011.
4. Stelian Brad, Mircea Fulea, Bogdan Mocan, Adina Duca, and Emilia Brad. Software platform for supporting open innovation. In *2008 IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR 2008)*, volume vol.3, pages 224 – 9, Piscataway, NJ, USA, 2008*.

5. Henry William Chesbrough. *Open innovation: the new imperative for creating and profiting from technology*. Harvard Business School Press, Boston, Mass., 2003.
6. Sunita Chulani, Clay Williams, and Avi Yaeli. Software development governance and its concerns. In *Proceedings of the 1st International Workshop on Software Development Governance*, SDG 08, pages 3–6, New York, NY, USA, 2008. ACM.
7. Massimo G. Colombo, Evila Piva, and Cristina Rossi-Lamastra. Open innovation and within-industry diversification in small and medium enterprises: The case of open source software firms. *Research Policy*, 2013*.
8. Kieran Conboy and Lorraine Morgan. Beyond the customer: Opening the agile systems development process. *Information and Software Technology*, 53(5):535 – 542, 2011*.
9. Daniela S. Cruzes and Tore Dybå. Research synthesis in software engineering: A tertiary study. *Information and Software Technology*, 53(5):440 – 455, 2011.
10. Daniela S. Cruzes, Tore Dybå, Per Runeson, and Martin Höst. Case studies synthesis: A thematic, cross-case, and narrative synthesis worked example a thematic, cross-case, and narrative synthesis worked example. *Empirical Software Engineering*, Accepted for publication, 2014.
11. Linus Dahlander and Mats Magnusson. How do firms make use of open source communities? *Long Range Planning*, 41(6):629 – 649, 2008*.
12. Linus Dahlander and Martin W. Wallin. A man on the inside: Unlocking communities as complementary assets. *Research Policy*, 35(8):1243 – 1259, 2006*.
13. Jorge Calmon de Almeida Biolchini, Paula Gomes Mian, Ana Candida Cruz Natali, Tayana Uchôa Conte, and Guilherme Horta Travassos. Scientific research ontology to support systematic review in software engineering. *Advanced Engineering Informatics*, 21(2):133 – 151, 2007. Ontology of Systems and Software Engineering; Techniques to Support Collaborative Engineering Environments.
14. Paul M. Di Gangi and Molly Wasko. Steal my idea organizational adoption of user innovations from a user innovation community: A case study of dell ideastorm. *DECISION SUPPORT SYSTEMS*, 48:303–312, 2009*.
15. Koen Dittich and Geert Duysters. Networking as a means to strategy change: The case of open innovation in mobile telephony. *Journal of Product Innovation Management*, 24(6):510 – 521, 2007*.
16. Winfried Ebner, Jan Marco Leimeister, and Helmut Krcmar. Community engineering for innovations: the ideas competition as a method to nurture a virtual community for innovations. *R&D Management*, 39(4):342 – 56, Sept. 2009*.
17. Henry Edison, Nauman Bin Ali, and Richard Torkar. Towards innovation measurement in the software industry. *Journal of Systems and Software*, 86(5):1390 – 1407, 2013.
18. Ellen Enkel, Oliver Gassmann, and Henry Chesbrough. Open r&d and open innovation: exploring the phenomenon. *R&D Management*, 39(4):311–316, 2009.
19. Oliver Gassmann. Opening up the innovation process: towards an agenda. *R&D Management*, 36(3):223–228, 2006.
20. Oliver Gassmann and Ellen Enkel. Towards a theory of open innovation: three core process archetypes. In *R&D Management Conference (RADMA)*, pages 1–18, Lisbon, Portugal, 2004.
21. Oliver Gassmann, Ellen Enkel, and Henry Chesbrough. The future of open innovation. *R&D Management*, 40(3):213–221, 2010.
22. Trisha Greenhalgh and Richard Peacock. Effectiveness and efficiency of search methods in systematic reviews of complex evidence: audit of primary sources. *BMJ*, 331(7524):1064–1065, 2005.
23. Endre Grøtnes. Standardization as open innovation: two cases from the mobile industry. *Information Technology & People*, 22(4):367–381, 2009*.
24. Elad Harison and Heli Koski. Applying open innovation in business strategies: Evidence from finnish software firms. *Research Policy*, 39(3):351 – 359, 2010*.
25. Joachim Henkel. Selective revealing in open innovation processes: The case of embedded linux. *Research Policy*, 35(7):953–969, 2006*.
26. Joachim Henkel. Champions of revealing—the role of open source developers in commercial firms. *Industrial and Corporate Change*, 18(3):435–471, December 2008*.
27. Joachim Henkel, Simone Schöberl, and Oliver Alexy. The emergence of openness: How and why firms adopt selective revealing in open innovation. *Research Policy*, September 2013*.

28. Martin Host, Klaas-Jan Stol, and Alma Orlucevic-Alagic. Inner Source Project Management. In G. Ruhe and C. Wohlin, editors, *Software Project Management in a Changing World*. Springer, 2014.
29. Eelko K.R.E. Huizingh. Open innovation: State of the art and future perspectives. *Technovation*, 31(1):2 – 9, 2011.
30. Stefan Husig and Stefan Kohn. Open cai 2.0 - computer aided innovation in the era of open innovation and web 2.0. *Computers in Industry*, 62(4):407 – 13, 2011*.
31. Martin Ivarsson and Tony Gorschek. A method for evaluating rigor and industrial relevance of technology evaluations. *Empirical Software Engineering*, 16(3):365–95, 2011.
32. Samireh Jalali and Claes Wohlin. Systematic literature studies: database searches vs. backward snowballing. In *2012 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement, (ESEM 2012)*, pages 29–38, 2012.
33. Slinger Jansen, Sjaak Brinkkemper, Jurriaan Souer, and Lutzen Luinenburg. Shades of gray: Opening up a software producing organization with the open software enterprise model. *Journal of Systems and Software*, 85(7):1495 – 1510, 2012*.
34. Anders Jonsson and Gunilla Svingby. The use of scoring rubrics: Reliability, validity and educational consequences. *Educational Research Review*, 2(2):130–144, 2007.
35. Barbara Kitchenham, Pearl Brereton, and David Budgen. Mapping study completeness and reliability - a case study. In *Proceedings of the 16th International Conference on Evaluation & Assessment in Software Engineering (EASE 2012)*, pages 126–135, 2012.
36. Barbara A. Kitchenham, David Budgen, and O. Pearl Brereton. Using mapping studies as the basis for further research - a participant-observer case study. *Inf. Softw. Technol.*, 53(6):638–651, June 2011.
37. Harold L. Kundel and Marcia Polansky. Measurement of observer agreement. *Radiology*, 228(2):303, 2003.
38. Mikko O.J. Laine. Using knowledge from end-users online for innovations: Effects of software firm types. In *Lecture Notes in Business Information Processing*, volume 114 LNBP, pages 70 – 78, Cambridge, MA, United states, 2012*.
39. Lorraine Morgan, Joseph Feller, and Patrick Finnegan. Exploring inner source as a form of intra-organisational open innovation. In *19th European Conference on Information Systems*, Helsinki, Finland, 2011*.
40. Lorraine Morgan and Patrick Finnegan. Open innovation in secondary software firms: An exploration of managers perceptions of open source software. *DATA BASE FOR ADVANCES IN INFORMATION SYSTEMS*, 41(1):76–95, 2010.
41. Barbara Moskal, Keith Miller, and LA King. Grading essays in computer ethics: rubrics considered helpful. *ACM SIGCSE Bulletin*, 34(1):101–105, 2002.
42. David C. Mowery. Plus ça change: Industrial R&D in the third industrial revolution. *Industrial and Corporate Change*, 18(1):1–50, February 2009.
43. Vinit Parida, Mats Westerberg, and Johan Frishammar. Effect of open innovation practices on smes innovative performance: An empirical study. In *Annual ICSB World Conference*, Stockholm, Sweden, 2011*.
44. Kai Petersen and Nauman Bin Ali. Identifying strategies for study selection in systematic reviews and maps. In *Proceedings of the 5th International Symposium on Empirical Software Engineering and Measurement, ESEM 2011, Banff, AB, Canada, September 22-23, 2011*, pages 351–354, 2011.
45. Kai Petersen, Robert Feldt, Shahid Mujtaba, and Michael Mattsson. Systematic mapping studies in software engineering. In *12th International Conference on Evaluation and Assessment in Software Engineering*, volume 17, page 1, 2008.
46. Thierry Rayna and Ludmilla Striukova. Large-scale open innovation: open source vs. patent pools. *International Journal of Technology Management*, 52(3-4):477 – 96, 2010*.
47. René Rohrbeck, Katharina Hölzle, and Hans Georg Gemünden. Opening up for competitive advantage - how deutsche telekom creates an open innovation ecosystem. *R & D Management*, 39(4):420 – 30, 2009*.
48. Per Runeson, Martin Höst, Austen Rainer, and Björn Regnell. *Case Study Research in Software Engineering - Guidelines and Examples*. Wiley, 2012.
49. Per Runeson, Martin Host, Austen Rainer, and Bjorn Regnell. *Case Study Research in Software Engineering: Guidelines and Examples*. John Wiley & Sons, March 2012.

50. Per Runeson and Mats Skoglund. Reference-based search strategies in systematic reviews. In *Proceedings 13th International Conference on Empirical Assessment & Evaluation in Software Engineering (EASE)*, Durham University, UK, 2009. British Computer Society.
51. Leif Singer, Norbert Seyff, and Samuel A. Fricker. Online social networks as a catalyst for software and IT innovation. In *4th International Workshop on Social Software Engineering, SSE'11, September 5, 2011 - September 5, 2011*, SSE'11 - Proceedings of the 4th International Workshop on Social Software Engineering, pages 1–5, 2011*.
52. Dag I. K. Sjøberg, Tore Dybå, and Magne Jørgensen. The future of empirical methods in software engineering research. In *Workshop on the Future of Software Engineering (FOSE 2007)*, pages 358–378, 2007.
53. Dag I. K. Sjøberg, Jo Erskine Hannay, Ove Hansen, Vigdis By Kampenes, Amela Karahasanovic, Nils-Kristian Liborg, and Anette C. Rekdal. A survey of controlled experiments in software engineering. *IEEE Trans. Software Eng.*, 31(9):733–753, 2005.
54. Wouter Stam. When does community participation enhance the performance of open source software companies? *Research Policy*, 38(8):1288 – 1299, 2009*.
55. Matthias Stuermer, Sebastian Spaeth, and Georg von Krogh. Extending private-collective innovation: a case study. *R&D Management*, 39(2):170–191, 2009*.
56. Han van der Meer. Open innovation ? the dutch treat: Challenges in thinking in business models. *Creativity and Innovation Management*, 16(2):192–202, 2007*.
57. Georg von Krogh and Sebastian Spaeth. The open source software phenomenon: characteristics that promote research. *Journal of Strategic Information Systems*, 16(3):236 – 53, 2007.
58. Georg von Krogh, Sebastian Spaeth, and Karim R. Lakhani. Community, joining, and specialization in open source software innovation: A case study. *Research Policy*, 32(7):1217 – 1241, 2003*.
59. Yonggui Wang, Ruqiong Tong, and Shanji Yao. An empirical study on external influencing factors of user innovation performance. In *Management and Service Science, 2009. MASS '09. International Conference on*, pages 1–4, 2009*.
60. Joel West. How open is open enough?: Melding proprietary and open source platform strategies. *Research Policy*, 32(7):1259 – 1285, 2003*.
61. Joel West and Marcel Bogers. Leveraging external sources of innovation: A review of research on open innovation. *Journal of Product Innovation Management*, 2013.
62. Joel West and Scott Gallagher. Challenges of open innovation: the paradox of firm investment in open-source software. *R & D Management*, 36(3):319 – 31, 2006*.
63. Joel West and David Wood. Creating and evolving an open innovation ecosystem: Lessons from symbian ltd. *Available at SSRN 1532926*, 2008*.
64. Roel Wieringa, Neil Maiden, Nancy Mead, and Colette Rolland. Requirements engineering paper classification and evaluation criteria: a proposal and a discussion. *Requirements Engineering*, 11(1):102–107, 2006.
65. Krzysztof Wnuk, Dietmar Pfahl, Davide Callele, and Even André Karlsson. How can open source software development help requirements management gain the potential of open innovation: an exploratory study. In *Proceedings of the ACM-IEEE international symposium on Empirical software engineering and measurement*, pages 271–279, Piscataway, NJ, USA, 2012*.
66. Krzysztof Wnuk and Per Runeson. Engineering open innovation—towards a framework for fostering open innovation. In Georg Herzwurm and Tiziana Margaria, editors, *Software Business. From Physical Products to Software Services and Solutions*, volume 150 of *Lecture Notes in Business Information Processing*, pages 48–59. Springer, 2013*.
67. Claes Wohlin and Rafael Prikladnicki. Systematic literature reviews in software engineering. *Information & Software Technology*, 55(6):919–920, 2013.
68. Marvin V Zelkowitz, Dolores R Wallace, and D Binkley. Culture conflicts in software engineering technology transfer. In *NASA Goddard Software Engineering Workshop*, page 52, 1998.

A Rigor and Relevance Criteria

A.1 Rigor

Context (C)

1. **Strong description:** The context is described to the extent where it becomes comparable to other settings [31]. In particular, we emphasized subject type (graduate, undergraduate, professionals, researcher), development experience, development methodology, duration of the observation. If all these aforementioned factors are highlighted, then C is evaluated to 1.
2. **Medium description:** If any of the above mentioned factors is missing in the study, then C is evaluated to 0.5.
3. **Weak description:** If no description of context is provided in the study, then C is evaluated to 0.

Design (D)

1. **Strong description:** The research design is described to the extent where it becomes transparent and detailed enough for the reader to understand the design [31]. To be specific, if the study underlined the outcome variables, measurement criteria, treatments, number of subjects, and sampling, then D is evaluated to 1.
2. **Medium description:** If a study is missing out on any of the factors related to design and data collection is missing (see above), then D evaluates to 0.5.
3. **Weak description:** If no design description is provided at all then, D is evaluated to 0.

Validity threats (V)

1. **Strong description:** If different types of validity (i.e. internal, external, conclusion and construct validity) are evaluated and reflected upon then, V is evaluated to 1.
2. **Medium description:** If a study only highlights the subset of the relevant threat categories then, V is evaluated to 0.5
3. **Weak description:** If a study is missing out on validity discussion completely, then V is evaluated to 0.

A.2 Relevance

Users/Subjects (U)

1. **Contribute to relevance:** If the subjects used in the study are from industry (professionals) then, U is evaluated to 1 for industry.
2. **Partially contribute to relevance:** The subjects are partially representative, i.e. they are master(Msc.) or graduated students then, U is evaluated to 0.5
3. **Does not contribute to relevance:** If the subjects are bachelor/undergrad students or the information is missing then, U is evaluated to 0

Scale (S)

1. **Contribute to relevance:** If an industrial size application is used in the study then, S is evaluated to 1.
2. **Does not contribute to relevance:** The application is down-scaled or a toy example hence, S is evaluated to 0.

Research Methodology (RM)

1. **Contribute to relevance:** The chosen research methodology is suitable to scrutinize real world contexts and situations with relevance for practitioners (action research, case study, industry interviews, experiment investigating a real situation, and surveys/interviews). If study belongs to any of the aforementioned research methodologies then, RM is evaluated to 1

2. **Does not contribute to relevance:** If a Study is using Lab experiment (human subjects/software) or missing information then, RM is evaluated to 0.

Context (C)

1. **Contribute to relevance:** If a study is executed in a setting that matches real industrial usage (industrial setting) then, C is evaluated to 1.
2. **Does not contribute to relevance:** If a study is investigated under under artificial setting (e.g. lab) or others that do not represent a context matching real world situations, or not reported then, C is evaluated to 0.

Table 7 Rigor and relevance scores with category

Study_ID	Ref.	C	D	V	Rig. Sum	U	S	RM	C	Rel. SUM	Category
S_1	[12]	1	1	0.5	2.5	1	1	1	1	4	A
S_2	[59]	0.5	0.5	0.5	1.5	1	1	1	1	4	C
S_3	[24]	1	1	0	2	1	1	1	1	4	A
S_4	[8]	1	0.5	0	1.5	1	1	1	1	4	C
S_5	[62]	1	0.5	0	1.5	1	1	1	1	4	C
S_6	[26]	1	0.5	0.5	2	1	1	1	1	4	A
S_7	[16]	0.5	0.5	0.5	1.5	0.5	1	1	1	3.5	C
S_8	[58]	1	1	0.5	2.5	1	1	1	1	4	A
S_9	[63]	0.5	0	0	0.5	1	1	1	1	4	C
S_10	[43]	1	1	0.5	2.5	1	1	1	1	4	A
S_11	[66]	1	0.5	0.5	2	0	0	1	1	2	B
S_12	[39]	0.5	0.5	0.5	1.5	1	1	1	1	4	C
S_13	[55]	1	1	0	2	1	1	1	1	4	A
S_14	[65]	0.5	0.5	1	2	1	1	1	1	4	A
S_15	[11]	1	1	0	2	1	1	1	1	4	A
S_16	[60]	0.5	0.5	0	1	1	1	1	1	4	C
S_17	[2]	1	1	0.5	2.5	1	1	1	1	4	A
S_18	[46]	0.5	1	0.5	2	0	1	1	0	1	B
S_19	[15]	0.5	0.5	0.5	1.5	1	1	1	1	4	C
S_20	[51]	0	0.5	0	0.5	0	0	1	0	1	D
S_21	[30]	0	0.5	0	0.5	0	0	0	1	1	D
S_22	[56]	0.5	0.5	0	1	1	1	1	1	4	C
S_23	[7]	1	1	0.5	2.5	1	1	1	1	4	A
S_24	[40]	1	0.5	0.5	2	1	1	1	1	4	A
S_25	[47]	0.5	0	0	0.5	1	1	1	1	4	C
S_26	[25]	1	0.5	0	1.5	1	1	1	1	4	C
S_27	[33]	1	0.5	0.5	2	1	1	1	1	4	A
S_28	[4]	0	0.5	0	0.5	1	1	1	0	3	C
S_29	[23]	1	0.5	1	2.5	1	1	1	1	4	A
S_30	[14]	1	1	0.5	2.5	1	1	1	1	4	A
S_31	[27]	1	0.5	0.5	2	1	1	1	1	4	A
S_32	[38]	1	1	0	2	1	1	1	1	4	A
S_33	[54]	1	1	1	3	1	1	1	1	4	A

B Database search strings

Search string used for the Compendex and Inspect database (years 1969 to 2013):

(((((Open Innovation WN KY OR Open-Innovation OR OI WN KY OR innovation WN KY OR innovation management WN KY) AND (software WN KY OR software ecosystem WN KY OR product line WN KY OR requirement* engineer* WN KY OR requirement* management WN KY OR open source WN KY) AND (exploratory study WN KY OR lesson* learn* WN KY OR challenge* WN KY OR guideline* WN KY OR Empirical investigation WN KY OR case study WN KY OR survey WN KY OR literature study WN KY OR literature review WN KY OR interview* WN KY OR experiment* WN KY OR questionnaire WN KY OR observation* WN KY OR quantitative study WN KY OR factor* WN KY) AND (ENGLISH) WN LA))))

Search string used for the ACM Digital Library database (years 1969 to 2013):

((((((((((((((("Title": "Open Innovation" OR "Title": "Open-Innovation" OR "Title": OI OR "Title": innovation OR "Title": innovation management) AND ("Abstract": software OR "Abstract": software ecosystem OR "Abstract": requirement* engineer* OR "Abstract": open source OR "Abstract": product line) AND ("Abstract": exploratory study OR "Abstract": challenge* OR "Abstract": guideline* OR "Abstract": Empirical investigation OR "Abstract": case study OR "Abstract": survey OR "Abstract": literature study OR "Abstract": literature review OR "Abstract": interview* OR "Abstract": experiment* OR "Abstract": questionnaire OR "Abstract": observation* OR "Abstract": quantitative study OR "Abstract": factor*))))) and (FtFlag:yes)) and (FtFlag:yes)) and (PublishedAs:journal OR PublishedAs:proceeding OR PublishedAs:transaction) and (FtFlag:yes)))) and (PublishedAs:journal OR PublishedAs:proceeding OR PublishedAs:transaction) and (FtFlag:yes))))))

Search string used for the IEEE Explore database (years 1969 to 2013):

((("Index Terms": "Open Innovation" OR "Index Terms": "Open-Innovation" OR "Index Terms": OI OR "Index Terms": innovation management OR "Index Terms": innovation) AND (Search_Index.Terms: software OR "Index Terms": ecosystem OR "Index Terms": product line OR "Index Terms": requirement* engineer* OR "Index Terms": requirement* management* OR "Index Terms": open source) AND (p_Abstract:case study OR "Abstract": exploratory study OR "Abstract": lessons learn* OR "Abstract": survey OR "Abstract": Empirical investigation OR "Abstract": guidelines "Abstract": literature study OR "Abstract": interview OR "Abstract": experiment OR "Abstract": factors OR "Abstract": questionnaire)))

Search string used for the ISI Web of Science database (years 1969 to 2013):

((((TI=("Open Innovation" OR "Open-Innovation" OR OI OR innovation OR innovation management) AND TS=(software OR software ecosystem OR product line OR requirement* engineer* OR requirement* management OR open source) AND TS=(exploratory study OR lesson* learn* OR challenge* OR guideline* OR Empirical investigation OR case study OR survey OR literature study OR literature review OR interview* OR experiment* OR questionnaire OR observation* OR quantitative study OR factor*)))) AND Language=(English) Refined by: Web of Science Categories=(COMPUTER SCIENCE INFORMATION SYSTEMS) Timespan=1969-2013. Databases=SCI-EXPANDED, SSCI, A&HCI, CPCI-S, CPCI-SSH

Search string used for the Science Direct database (years 1969 to 2013):

(open innovation OR open-innovation OR OI OR innovation OR innovation management) AND (software OR software ecosystem OR product line OR requirement* engineer* OR requirement* management OR open source) AND (exploratory study OR lesson* learn* OR challenge* OR guideline* OR Empirical investigation OR case study OR literature study OR literature review OR interview* OR experiment* OR case study OR questionnaire OR observation* OR quantitative study OR factor*)[All Sources(Computer Science)]