LUND UNIVERSITY

How Firms Adapt and Interact in Open Source Ecosystems: Analyzing Stakeholder Influence and Collaboration Patterns

Linåker, Johan; Rempel, Patrick; Regnell, Björn; Mäder, Patrick

Published in:
Requirements Engineering: Foundation for Software Quality

DOI:
10.1007/978-3-319-30282-9_5

2016

Link to publication

Citation for published version (APA):
Linåker, J., Rempel, P., Regnell, B., & Mäder, P. (2016). How Firms Adapt and Interact in Open Source Ecosystems: Analyzing Stakeholder Influence and Collaboration Patterns. In *Requirements Engineering: Foundation for Software Quality: 22nd International Working Conference, REFSQ 2016, Gothenburg, Sweden, March 14-17, 2016, Proceedings* (Vol. 9619, pp. 63-81). (Lecture Notes in Computer Science; Vol. 9619). https://doi.org/10.1007/978-3-319-30282-9_5

Total number of authors:
4

General rights
Unless other specific re-use rights are stated the following general rights apply:
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.
• Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
• You may not further distribute the material or use it for any profit-making activity or commercial gain
• You may freely distribute the URL identifying the publication in the public portal

Read more about Creative commons licenses: https://creativecommons.org/licenses/

Take down policy
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

LUND UNIVERSITY

PO Box 117
221 00 Lund
+46 46-222 00 00

Download date: 01. Nov. 2026

How Firms Adapt and Interact in Open Source Ecosystems: Analyzing Stakeholder Influence and Collaboration Patterns

Johan Linåker[1], Patrick Rempel[2], Björn Regnell[1], and Patrick Mäder[2]

[1] Lund University, Sweden
{johan.linaker,bjorn.regnell}@cs.lth.se
[2] Technische Universität Ilmenau, Germany
{patrick.rempel,patrick.maeder}@tu-ilmenau.de

Abstract. [**Context and motivation**] Ecosystems developed as Open Source Software (OSS) are considered to be highly innovative and reactive to new market trends due to their openness and wide-ranging contributor base. Participation in OSS often implies opening up of the software development process and exposure towards new stakeholders. [**Question/Problem**] Firms considering to engage in such an environment should carefully consider potential opportunities and challenges upfront. The openness may lead to higher innovation potential but also to frictional losses for engaged firms. Further, as an ecosystem progresses, power structures and influence on feature selection may fluctuate accordingly. [**Principal ideas/results**] We analyze the Apache Hadoop ecosystem in a quantitative longitudinal case study to investigate changing stakeholder influence and collaboration patterns. Further, we investigate how its innovation and time-to-market evolve at the same time. [**Contribution**] Findings show collaborations between and influence shifting among rivaling and non-competing firms. Network analysis proves valuable on how an awareness of past, present and emerging stakeholders, in regards to power structure and collaborations may be created. Furthermore, the ecosystem's innovation and time-to-market show strong variations among the release history. Indications were also found that these characteristics are influenced by the way how stakeholders collaborate with each other.

Keywords: requirements engineering, stakeholder collaboration, stakeholder influence, open source, software ecosystem, inter-organizational collaboration, open innovation, co-opetition

1 Introduction

The paradigm of Open Innovation (OI) encourages firms to look outside for ideas and resources that may further advance their internal innovation capital [1]. Conversely, a firm may also find more profitable incentives to open up an intellectual property right (IPR) rather than keeping it closed. For software-intensive firms a common example of such a context is constituted by Open Source Software (OSS) ecosystems [2] [3].

The openness implied by OI and an OSS ecosystem makes a firm's formerly closed borders permeable for interaction and influence from new stakeholders, many of which may be unknown to a newly opened-up firm. Entering such an ecosystem affects the way how Requirements Engineering (RE) processes are structured [4]. Traditionally these are

centralized, and limited to a defined set of stakeholders. However, in this new open context, RE has moved to become more decentralized and collaborative with an evolving set of stakeholders. This may lead to an increased innovation potential for a firm's technology and product offerings, but also imply frictional losses [5]. Conflicting interests and strategies may arise, which may diminish a firms own impact in regards to feature selection and control of product planning [6]. Further, as an ecosystem evolves, power structures and influence among stakeholders may fluctuate accordingly. This creates a need for firms already engaged or thinking of entering an OSS ecosystem to have an awareness of past and present ecosystem governance constellation in order to be able to adapt their strategies and product planning to upcoming directions of the ecosystem [7].

Given this problematization, we were interested in studying how stakeholders' influence and collaboration fluctuate over time in OSS ecosystems. Researchers argue that collaboration is core to increase innovation and reduce time-to-market [8]. Hence, another goal was to study the evolution of OSS ecosystems' innovation and time-to-market over time. We hypothesize that this could be used as input to firms' planning of contribution and product strategies, which led us to formulate the following research questions:

RQ1 How are stakeholder influence and collaboration evolving over time?
RQ2 How are innovation and time-to-market evolving over the same time?

To address these questions, we launched an exploratory and quantitative longitudinal case study of the Apache Hadoop ecosystem, a widely adopted OSS framework for distribution and process parallelization of large data.

The rest of the paper is structured as follows: Section 2 presents related work. Section 3 describes the case study design and methodology used, limitations and threats to validity are also accounted for. Section 4 presents the analysis and results, which are further discussed in Section 5. Finally, Section 6 concludes the paper.

2 Related Work

Here we present related work to software ecosystems and how its actors (stakeholders) may be analyzed. Further, the fields of stakeholder identification and analysis in RE are presented from an ecosystem and social network perspective.

2.1 Software Ecosystems

Multiple definitions of a software ecosystem exists [9], while we refer to the one by Jansen et al. [3] - *"A software ecosystem is a set of actors functioning as a unit and interacting with a shared market for software and services, together with relationships among them. These relationships are frequently underpinned by a common technological platform or market and operates through the exchange of information, resources and artifacts."*. The definition may incorporate numerous types of ecosystems in regards to openness [10], ranging from proprietary to OSS ecosystems [9], which in turn contains multiple facets. In this study we will focus on the latter with the Apache Hadoop ecosystem as our case, where the Apache Hadoop project constitutes the technological platform underpinning the relationships between the actors of the Apache Hadoop ecosystem.

An ecosystem may further be seen from three scope levels, as proposed by Jansen et al. [7]. Scope level 1 takes an upper perspective, on the relationships and interactions between ecosystems, for example between the Apache Hadoop and the Apache Spark ecosystems, where the latter's project may be built on top of the former. On scope level 2, one looks inside of the ecosystem, its actors and the relationships between them, which is the focus of this paper when analyzing the Apache Hadoop ecosystem. Lastly, scope level 3 takes the perspective from a single actor and its specific relationships.

Jansen et al. [7] further distinguished between three types of actors: dominators, keystone players, and niche players. Dominators expand and assimilate, often on the expense of other actors. Keystone players are well connected, often with a central role in hubs of actors. They create and contribute value, often beneficial to its surrounding actors. Platform suppliers are typically keystone players. Niche players thrive on the keystone players and strive to distinguish themselves from other niche players. Although other classifications exist [9] [10], we will stick to those defined above.

In the context of OSS ecosystems, a further type of distinction can be made in regards to the Onion model as proposed by Nakakoji et al. [11]. They distinguished between eight roles ranging the passive user in the outer layer, to the project leader located in the center of the model. For each layer towards the center, influence in the ecosystem increases. Advancement is correlated to increase of contributions and engagement of the user, relating to the concept of meritocracy.

2.2 Stakeholder Networks and Interaction in Requirements Engineering

To know the requirements and constraints of a software, one needs to know who the stakeholders are, hence highlighting the importance of stakeholder identification and analysis in RE [12]. Knowing which stakeholders are present is however not limited to purposes of requirements elicitation. For firms engaged in OSS ecosystems [3] [9], this is important input to their product planning and contribution strategies. Disclosure of differentiating features to competitors, un-synced release cycles, extra patch-work and missed out collaboration opportunities are some possible consequences if the identification and analysis of the ecosystem's stakeholders is not done properly [6] [2] [5]. Most identification methods however refer to the context of traditional software development and lack empirical validation in the context of OSS ecosystems [13].

In recent years, the research focus within the field has shifted more towards stakeholder characterization through the use of, e.g., Social Network Analysis (SNA) [13]. It has also become a popular tool in empirical studies of OSS ecosystems, hence highlighting potential application within stakeholder identification.

In regards to traditional software development, Damian et al. [14] used SNA to investigate collaboration patterns and the awareness between stakeholders of co-developed requirements in the context of global software development. Lim et al. [15] constructed a system based on referrals, where identified stakeholders may recommend others. Concerning RE processes within software ecosystems in general, research is rather limited [16] with some exceptions [17]. Fricker [16] proposed that stakeholder relations in software ecosystems may be modeled as requirement value chains *" ... where requirements emerge from and propagate with inter-stakeholder collaboration"*. Knauss et al. [17] investigated the IBM CLM ecosystem to find RE challenges and practices

used in open-commercial software ecosystems. Distinction is made between a strategic and an emergent requirements flow, where the former regard high level requirements, and how business goals affect the release planning. The latter considers requirements created on an operational level, in a Just-In-Time (JIT) fashion, commonly observed in OSS ecosystems [18].

In OSS ecosystems specifically, RE practices such as elicitation, prioritization, and selection are usually managed through open forums such as issue trackers or mailinglists. These are also referred to as informalisms as they are used to specify and manage the requirements in an informal manner [19], usually as a part of a conversation between stakeholders. These informalisms constitute an important source to identify relevant stakeholders. Earlier work includes Duc et al. [20] who applied SNA to map stakeholders in groups of reporters, assignees, and commentators to issues with the goal to investigate the impact of stakeholder collaboration on the resolution time of OSS issues. Crowsten et al. [21] performed SNA on 120 OSS projects to investigate communication patterns in regards to interactions in projects' issue trackers.

Many studies focused on a developer and user level, though some exceptions exist. For example, Martinez-Romeo et al. [22] investigated how a community and a firm collaborates through the development process. Orucevic-Alagic et al. [23] investigated the influence of stakeholders on each other in the Android project. Texiera et al. [24] explored collaboration between firms in the Openstack ecosystem from a co-opetition perspective showing how firms, despite being competitors, may still collaborate within an ecosystem.

This paper contributes to OSS RE literature by addressing the area of stakeholder identification and analysis in OSS ecosystems by investigating a case on a functional level [24]. Further it adds to the software ecosystem literature and its shallow research of RE [16] [17] and strategic perspectives[9] in general.

3 Research Design

We chose the Apache Hadoop project for an embedded case study [25] due to its systematically organized contribution process and its ecosystem composition. Most of the contributors have a corporate affiliation.

To create a longitudinal perspective, issues of the Apache Hadoop's issue tracking and project management tool were analyzed in sets reflecting the release cycles. The analysis was narrowed down to sub releases, spanning from 2.2.0 (released 15/Oct/13) to 2.7.1 (06/Jul/15), thus constituting the units of analysis through the study. Third level releases were aggregated into their parent upper level release.

Issues were furthermore chosen as the main data source as these can tie stakeholders' socio-technical interaction together [14] [20], as well as being connected to a specific release. To determine who collaborated with whom through an issue, patches submitted by each stakeholder were analyzed, a methodology similar to those used in previous studies [23] [22]. Users who contribute to an issue package their code into a patch and then attach it to the issue in question. After passing a two-step approval process comprising automated tests and manual code reviews, an authorized committer eventually

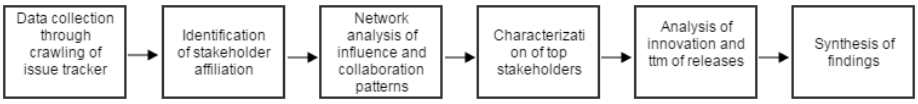

Fig. 1: Overview of the case study process

commits the patch to the project's source configuration management (SCM) system. The overall process of this case study is illustrated in Fig. 1 and further elaborated on below.

3.1 Data Collection

The Apache Hadoop project manages its issue data with the issue tracker JIRA. A crawler was implemented to automatically collect, parse, and index the data into a relational database.

To determine the issue contributors' organizational affiliation, the domain of their email addresses was analyzed. If the affiliation could not be determined directly (e.g., for @apache.org), secondary sources were used such as LinkedIn and Google. The issue contributors' full name functioned as keyword.

3.2 Analysis Approach and Metrics

Below we present the methodology and metrics used in the analysis of this paper. Further discussion of metrics in relation to threats to validity is available in section 3.3.

Network Analysis. Patches attached to issues were used as input to the SNA process. Stakeholders were paired if they submitted a patch to the same issue. Based on stakeholders' affiliation, pairings were aggregated to the organizational level. A directed network was constructed, representing the stakeholders at the organizational level as vertices. Stakeholder collaboration relationships were represented as edges. As suggested by Orucevic-Alagic et al. [23], edge weights were calculated to describe the strength of the relationships. Since stakeholders created patches of different size, the relative size of a stakeholder's patch was used for the weighting. We quantified this size as changed lines of code (LOC) per patch. A simplified example of calculating network weights without organizational aggregation is shown in Fig. 2. Each of the stakeholders A, B, and C created a patch that was attached to the same issue. A's patch contains 50 LOC. B's patch contains 100 LOC, while C's patch contains 150 LOC. In total, 300 LOC were contributed to the issue. Resulting in the following edge weights: A→B = 50/300, A→C = 50/300, B→C = 100/300, B→A = 100/300, C→B = 150/300, and C→A = 150/300.

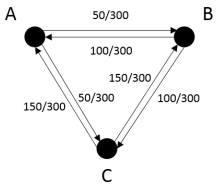

Fig. 2: Example of a weighted network with three stakeholders.

The following network metrics were used to measure the influence of stakeholders and the strength of the collaboration relationships among the stakeholders.

– *Out-degree Centrality* is the sum of a all outgoing edges' weights of a stakeholder vertex. Since it calculates the number of collaborations where the stakeholder has contributed, a

higher index indicates a higher influence of a stakeholder on its collaborators. It also quantifies the degree of contributions relative to the stakeholder's collaborators.
- *Betweeness Centrality* counts how often a stakeholder is on a stakeholder collaboration path. A higher index indicates that the stakeholder has a more central position compared to other stakeholders among these collaboration paths.
- *Closeness Centrality* measures the average relative distance to all other stakeholders in the network based on the shortest paths. A higher index indicates that a stakeholder is well connected and has better possibilities in spreading information in the network, hence a higher influence.
- *Average Clustering Coefficient* quantifies the degree to which stakeholders tend to form clusters (connected groups). A higher coefficient indicates a higher clustering, e.g., a more densely connected group of stakeholders with a higher degree of collaborations.
- *Graph Density* is the actual number of stakeholder relationships divided by the possible number of stakeholder relationships. A higher value indicates a better completeness of stakeholder relationships (collaborations) within the network, where 1 is complete and 0 means that no relationships exist.

Innovation and Time-To-Market Analysis. Innovation can be measured through input, output, or process measures [26]. In this study, input and output measures are used to quantify innovation per release. Time-to-market was measured through the release cycle time [27].

- *Issues* counts the total number of implemented JIRA tickets per release and comprises the JIRA issue types *feature*, *improvement*, and *bug*. It quantifies the innovation input to the development process.
- *Change size* counts the net value of changed lines of code. It quantifies the innovation output of the development process.
- *Release cycle time* is the amount of time between the start of a release and the end of a release. It indicates the length of a release cycle.

Stakeholder Characterization. To complement our quantitative analysis and add further context, we did an qualitative analysis of electronic data available to characterize identified corporate stakeholders. This analysis primarily included their respective websites, press releases, news articles, and blog posts.

3.3 Threats to Validity

Four aspects of validity in regards to a case study are *construct*, *internal* and *external validity*, and *reliability* [25].

In regards to *construct validity*, one concern may be definition and interpretation of network metrics. The use of weights to better represent a stakeholder's influence, as suggested by Orucevic-Alagic et al. [23] was used with the adoption to consider the net of added LOC to further consider the relative size of contributions. A higher number of LOC however does not have to imply increased complexity. We chose to see it as a simplified metric of investment with each LOC representing a cost from stakeholder. Other options could include consideration software metrics such as cyclomatic complexity. Further network metrics, e.g. the eigenvector centrality and the clustering coefficient could offer further facets but was excluded as a design choice.

Furthermore, we focused on input (number of issues) and output (implementation change size) related metrics [26] for operationalizing the innovation per release. Issues

is one of many concepts in how requirements may be framed and communicated in OSS RE, hence the term requirement is not always used explicitly [19]. Types of issues varies between OSS ecosystem and type of issue tracker (e.g., JIRA, BugZilla) [18]. In the Apache Hadoop ecosystem we have chosen the types feature, improvement and bug to represent the degree of innovation. We hypothesize that stakeholders engaged in bug fixing, are also involved in the innovation process, even if a new feature and an improvement probably includes a higher degree of novelty in the innovation. Even bugs may actually include requirements-related information not found elsewhere, and also relate to previously defined features with missing information. In future work, weights could be introduced to consider different degrees of innovation in the different issue types.

Release cycle times were used for quantifying the time-to-market as suggested by Griffin [27]. Since we solely analyzed releases from the time where the Apache Hadoop ecosystem was already well established, a drawback is that a long requirements analysis ramp up time may not be covered by this measure.

A threat to *internal validity* concerns the observed correlation of how the time-to-market and the innovativeness of a release is influenced by the way how stakeholders collaborate with each other. This needs further replication and validation in future work.

In regards to *external validity*, this is an exploratory single case study. Hence observations need validation and verification in upcoming studies in order for findings to be further generalized. Another limitation concerns that only patches of issues were analyzed, though it has been considered a valid approach in earlier studies [23] [22]. In future work, consideration should also be taken into account, for example, as this may also be an indicator of influence and collaboration. Further, number of releases in this study was limited due to a complicated release history in the Apache Hadoop project, but also a design choice to give a further qualitative view of each release in a relative fine-grained time-perspective. Future studies should strive to analyze longer periods of time.

Finally, in regards to *reliability* one concern may be the identification of stakeholder affiliation. A contributor could have used the same e-mail but from different roles, e.g., as an individual or for the firm. Further, sources such as LinkedIn may be out of date.

4 Analysis

In this section, we present our results of the quantitative analysis of the Apache Hadoop ecosystem across the six releases R2.2-R2.7.

4.1 Stakeholders' Characteristics

Prior to quantitatively analyzing the stakeholder network, we qualitatively analyzed stakeholders' characteristics to gain a better understanding of our studied case. First, we analyzed how each stakeholder uses the Apache Hadoop platform to support its own business model. We identified the following five user categories:

- **Infrastructure provider**: sells infrastructure that is based on Apache Hadoop.
- **Platform user**: uses Apache Hadoop to store and process data.

- **Product provider**: sells packaged Apache Hadoop solutions.
- **Product supporter**: Provides Apache Hadoop support without being a product provider.
- **Service provider**: Sells Apache Hadoop related services.

Second, we analyzed stakeholders' firm history and strategic business goals to gain a better understanding of their motivation for engaging in the Hadoop ecosystem. We summarize the results of this analysis in the following list:

- **Wandisco** [Infrastructure provider] entered the Apache Hadoop ecosystem by acquiring AltoStar in 2012. It develops a platform to distribute data over multiple Apache Hadoop clusters.
- **Baidu** [Platform user] is a web service company and was founded in 2000. It uses Apache Hadoop for data storage and processing of data.
- **eBay** [Platform user] is an E-commerce firm and was founded in 1995. It uses Hadoop for data storage and processing of data.
- **Twitter** [Platform user] offers online social networking services and was founded in 2006. It uses Apache Hadoop for data storage and processing of data.
- **Xiaomi** [Platform user] is focused on smartphone development. It uses Apache Hadoop for data storage and processing of data.
- **Yahoo** [Platform user] is a search engine provider who initiated the Apache Hadoop project in 2005. It uses Apache Hadoop for data storage and processing of data. It spun off Hortonworks in 2011.
- **Cloudera** [Product provider] was founded in 2008. It develops its own Apache Hadoop based product *Cloudera Distribution Including Apache Hadoop* (CDH).
- **Hortonworks** [Product provider] was spun off by Yahoo in 2011. It develops its own Apache Hadoop based product *Hortonworks Data Platform* (HDP). It collaborates with Microsoft since 2011 to develop *HDP for Windows*. Other partnerships include Redhat, SAP, and Teradata.
- **Huawei** [Product provider] offers the Enterprise platform *FusionInsight* based on Apache Hadoop. FusionInsight was first released in 2013.
- **Intel** [Product supporter] maintained its own Apache Hadoop distribution that was optimized to their own hardware. It dropped the development in 2014 to support Cloudera by becoming its biggest shareholder and focusing on contributing its features to Cloudera's distribution.
- **Altiscale** [Service provider] was founded in 2012. It runs its own infrastructure and offers Apache Hadoop as-a-service via their product *Altiscale Data Cloud*.
- **Microsoft** [Service provider] offers Apache Hadoop as a cloud service labeled *HDInsight* through its cloud platform Azure. It maintains a partnership with Hortonworks who develops *HDP for Windows*.
- **NTT Data** [Service provider] is a partner with Cloudera and provides support and consulting services for their Apache Hadoop distribution.

Firms that belong to the same user category apply similar business models. Hence, we can identify competing firms based on their categorization.

4.2 Stakeholder Collaboration

Figure 3 shows all stakeholder networks that were generated for the releases R2.2 to R2.7. The size of a stakeholder vertex indicates its relative ranking in regards to the outdegree centrality. Table 1 summarizes the number of stakeholders and stakeholder relationships per release. It illustrates that the number of stakeholders and collaboration relationships varies over time. Except for the major increase from R2.2 to R2.3, the network maintains a relatively consistent size, though the number of collaborations are in the interval between 81 to 122 for R2.4 to R2.7.

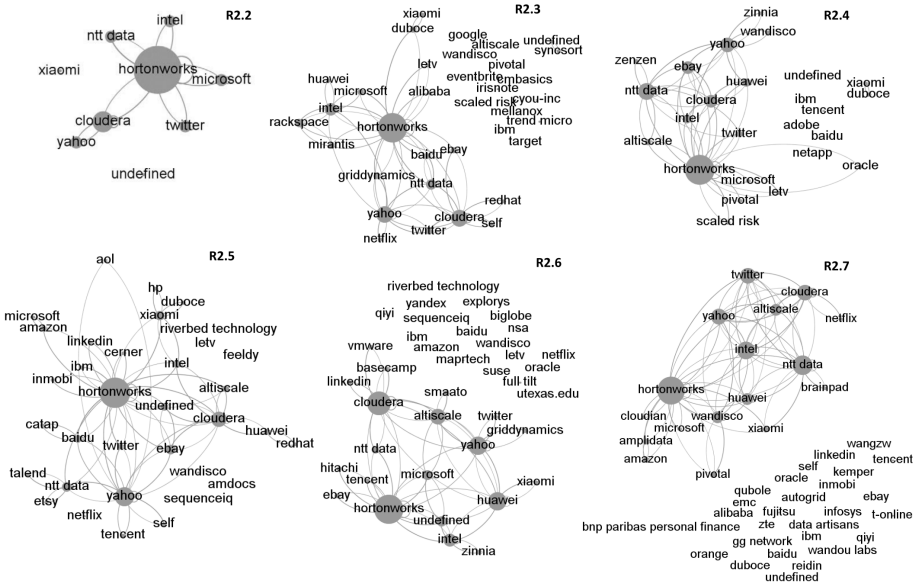

Fig. 3: Network distribution of releases R2.2-R2.7

Table 1: Number of stakeholder (vertices) and collaboration relationships (edges) per release

	R2.2	R2.3	R2.4	R2.5	R2.6	R2.7
Stakeholders	9	35	25	34	38	44
Collaboration relationships	21	97	81	108	96	122

A general observation among the different releases is the existence of one main cluster where a core of stakeholders is present, whilst the remaining stakeholders make temporary appearances. Many stakeholders are not part of these clusters implying that they do not collaborate with other stakeholders at all. The number of those stakeholders shows strong variation among the releases. This could imply that stakeholders implement their own issues, which is further supported by the fact that 65% of the patches are contributed by the issue reporters themselves.

The visual observation from the networks being weakly connected in general is supported by the Graph Density (GD) as its values are relatively low among all releases (see Table 2). The values describe that stakeholders had a low number of collaborations in relation to the possible number of collaborations. The Average Clustering Coefficient (ACC) values among all releases (see Table 2) further indicate that the stakeholders are weakly connected to their direct neighbors in the releases R2.2 - R2.6. This correlates

Table 2: Average Clustering Coefficient (ACC) and Graph Density (GD) per release.

	R2.2	R2.3	R2.4	R2.5	R2.6	R2.7
ACC	0	0.207	0.303	0.198	0.237	0.552
GD	0.292	0.082	0.135	0.096	0.068	0.064

Table 3: Stakeholder collaborations among the different user categories.

	Infrastructure provider	Platform user	Product provider	Product supporter	Service provider
Infrastructure provider	0	2	4	1	0
Platform user	2	24	73	6	14
Product provider	4	73	124	23	50
Product supporter	1	6	23	0	3
Service provider	0	14	50	3	10

with the observation that there are many unconnected stakeholders and only a few core stakeholders collaborating with each other. The ACC value however indicates a significantly higher number of collaborations for release R2.7.

Table 3 summarizes stakeholder collaborations among the different user categories. It shows that collaborations took place among all user categories, except between infrastructure providers and service providers. The product providers were the most active and had the highest number of collaborations with other product providers. They also have the highest amount of collaborations with other user categories. These results show that stakeholders with competing (same user category) and non-competing (different user category) business models collaborate within the Apache Hadoop ecosystem.

4.3 Stakeholder Influence

To analyze the evolving stakeholder influence over time, we leveraged the three network centrality metrics: outdegree centrality, betweeness centrality, and closeness centrality.

The left graph in Fig. 4 shows the outdegree centrality evolution for the ten stakeholders with the highest outdegree centrality values. These stakeholders are most influential among all Apache Hadoop stakeholders in regards to weighted issue contributions. The graph also shows that the relative outdegree centrality varies over time. To further investigate this evolution, we created a stakeholder ranking per release using the relative outdegree centrality as ranking criteria. This analysis revealed that Hortonworks was

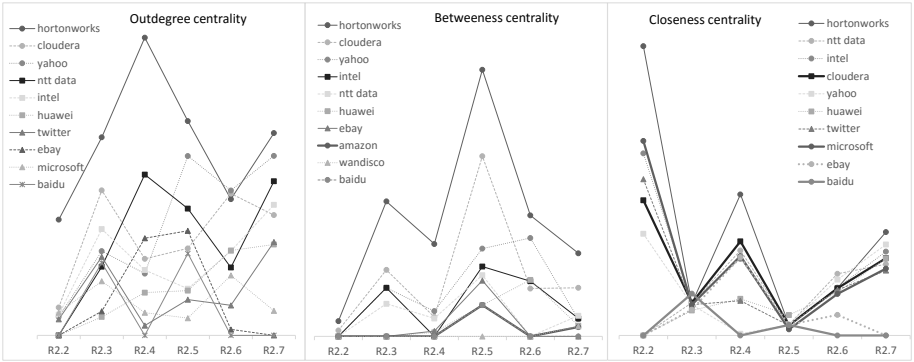

Fig. 4: Evolution of stakeholders' outdegree, betweeness, and closeness centrality across the releases R2.2-R2.7

most influential in terms of issue contributions. It was five times ranked first and once ranked third (average ranking: 1.3). The other top ranked stakeholders were Cloudera (average ranking: 3.3) and Yahoo (average ranking: 3.3). The stakeholders NTT Data (avg ranking = 4.7) and Intel (average ranking: 4.8) can be considered as intermediate influencing among the top ten outdegree centrality stakeholders. The stakeholders Huawei (average ranking: 8.2), Twitter (average ranking: 8.5), eBay (average ranking: 9.0), Microsoft (average ranking: 9.5), and Baidu (average ranking: 10.2) had the least relative outdegree centrality among the ten stakeholders.

The center graph in Fig. 4 shows the betweeness centrality evolution of the ten stakeholders with the highest accumulated values. As the metric is based on the number of shortest paths passing through a stakeholder vertex, it indicates a stakeholder's centrality with regards to the possible number of collaborations. The resulting top ten stakeholder list is very similar to the list of stakeholders with the highest outdegree centrality. The top stakeholders are Hortonworks (average ranking: 1), Cloudera (average ranking: 2.7), and Yahoo (average ranking: 3.0). Intel (average ranking: 4.2), NTT Data (average ranking: 4.7), and Huawei (average ranking: 5.3) are influencing among the top ten beweeness centrality stakeholders. eBay (average ranking: 6.7), Amazon (average ranking: 6.7), WANdisco (average ranking: 7.0), and Baidu (average ranking: 7.2), the group of stakeholders with the least betweeness centrality among the top ten stakeholders differs compared to the group of stakeholders with the least outdegree centrality. The stakeholders Twitter and Microsoft were replaced by Amazon and WANdisco.

The right graph in Fig. 4 shows closeness centrality evolution of the ten stakeholders with the highest accumulated values. A higher degree of closeness centrality indicates higher influence, because of closer collaboration relationships to other stakeholders. The resulting top ten closeness centrality stakeholder list differs compared to the outdegree and betweeness centrality list. Our analysis results do not show a single top stakeholder with the highest closeness centrality. The stakeholders Hortonworks (avgerage ranking: 3.2), NTT Data (average ranking: 4.0), Intel (average ranking: 4.3), Cloudera (average ranking: 4.8), and Yahoo (average ranking: 5.5) had relatively similar closeness rankings among the releases. This is also reflected in Fig. 4 by very similar curve shapes among the stakeholders. Also the remaining stakeholders with lower closeness centrality values had very similar average rankings: Huawei (average ranking: 7.7), Twitter (average ranking: 8.0), Microsoft (average ranking: 8.3), eBay (average ranking: 9.2), Baidu (average ranking: 9.3).

The results of our analysis also show that the stakeholders with the highest outdegree centrality, betweeness centrality, and closeness centrality were distributed among different stakeholder user categories: 4 platform user, 3 product provider, 2 service provider, and 1 product supporter. However, it is notable that the average ranking differs among these user categories. Product providers had the highest average influence ranking. Platform users and service providers had lower influence ranking. This implies that product providers are the most driving forces of the Apache Hadoop ecosystem.

4.4 Innovation and Time-To-Market over Time

The evolution of the degree of innovation and time-to-market from release R2.2 to R2.7 is summarized in Figure 5 by three consecutive graphs. The first graph in Fig. 5 shows

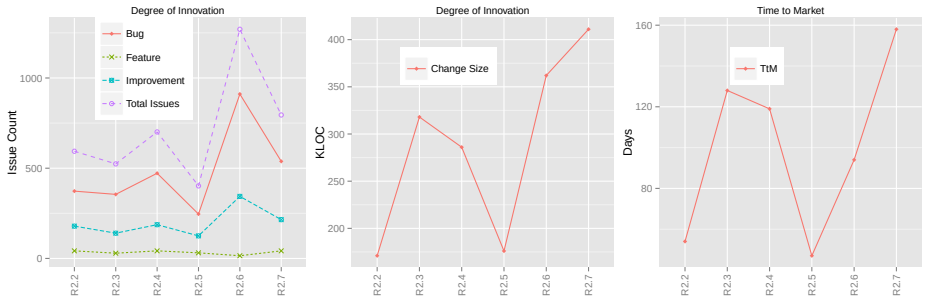

Fig. 5: Evolution of the degree of innovation over time with respect to implemented JIRA issues and changed lines of code and time to market.

the number of issues that were implemented per release. The illustrated number of issues is broken down into the issue types: bug, improvement, and feature. The number of implemented features (avg: 33.5, med: 37, std: 9.88) remains steady across all analyzed releases. This is reflected by a relatively low standard deviation. Similarly, the number of implemented improvements (avg: 198.3; med: 183; std: 71.62) remains relatively steady across the releases with one exception. In release R2.6, the double amount of improvement issues was implemented compared to the average of the remaining releases. The number of implemented bugs (avg: 482.5; med: 423; std: 212.52) features stronger variation among the releases.

The second graph in Fig. 5 shows the number of changed lines of code per release. The total number of changed lines of code per release (avg: 287,883.33; med: 302,257; std: 89,334.57) strongly varies across the analyzed releases. Each of the analyzed releases comprises code changes of significant complexity. Even the two releases R2.2 and R2.5, with the lowest change complexity (R2.2: 171 KLOC; R2.5: 176 KLOC), comprised more than 170 KLOC. The remaining releases comprised change complexities of more than 250 KLOC. Further, the graph indicates that the change complexity scatters randomly among the studied releases. A steady trend cannot be determined.

The third graph in Fig. 5 depicts the time between the start and the end (time-to-market) of each analyzed release. Analogous to the evolution of the changed lines of code, the time-to-market scatters randomly among the analyzed releases.

5 Discussion

Stakeholder Collaborations (RQ-1). The number of collaborating stakeholders remains on a relatively stable level. However, as indicated by the GD and ACC, the networks are weakly connected in regards to the possible number of collaborations. Only a core set of stakeholders is engaged in most of the collaborations. This may indicate that they have a higher stake in the ecosystem with regards to their product offering and business model, and in turn a keystone behaviour [7]. From a requirements value chain perspective, collaborations translate into partnerships and relationships. This may prove valuable in negotiations about requirements prioritization and how these should be treated when planning releases and road maps [16]. The results also show that many

stakeholders do not collaborate at all. This is supported by the fact that 65% of the reported issues are implemented by reporters themselves without any collaboration. This indicates that a lot of independent work was performed in the ecosystem. Reasons for this could be that issues are only of interest for the reporter. It also indicates that the ecosystem is relatively open [10] in the sense that it is easy for stakeholders to get their own elicited requirements implemented and prioritized, but with the cost of own development efforts.

Another aspect of the collaborations can be inferred from the different user categories. Firms with competing business models collaborate as openly as non-rivaling firms do, as presented in Table 3 and reported in earlier studies [24]. Some of the collaborations may be characterized through the partnerships established between the different stakeholders, as presented in our qualitative analysis of stakeholder characteristics. One of Hortonworks many partnerships include that with Microsoft through the development of their Windows-friendly Apache Hadoop distribution. Cloudera's partnerships include both Intel and NTT Data. None of these partnerships, or among the others identified in this study, occurs within the same user category. Yet still, a substantial part of the ecosystem collaboration occurs outside these special business relationships.

Independent of business model, all firms work together towards the common goal of advancing the shared platform, much resembling an external joint R&D pool [2]. As defined through the concept of co-opetition, one motivation could be a joint effort to increase the market share by helping out to create value, and then later diverge and capture value when differentiating in the competition about the customers [28]. Collaboration could further be limited to commodity parts whereas differentiating parts are kept internal, e.g. leveraged through selective revealing [29].

Stakeholder Influence (RQ-1). Although the distribution of stakeholders' influence fluctuated among the releases, we identified that the group of most influential stakeholders remained very stable. Even the influence ranking within this group did not show high variations. It can be concluded that the development is mainly driven by the stakeholders Hortonworks, Cloudera, NTT Data, Yahoo, and Intel, which may also be referred to as keystone players, and in some cases also niche players relative to each other [7]. Due to this stable evolution, it can be expected that these stakeholders will also be very influential firms in the future. The stakeholder distribution represents multiple user categories, although the product providers Hortonworks and Cloudera tend to be in the top. This may relate to their products being tightly knit with the Apache Hadoop project. In turn, service-providers may use the product-providers' distributions as a basis for their offerings.

Tracking that influence may be useful to identify groups and peers with key positions in order to create traction on certain focus areas for the road map, or to prioritize certain requirements for implementation and release planning [16]. Further, it may help to identify emerging stakeholders increasing their contributions and level of engagement [11], which may also be reflected in the commercial market. Huawei's increase in outdegree centrality, for example, correlates with the release of their product FusionInsight, which was launched in the beginning of 2013.

The fact that the network metrics used revealed different top stakeholders, indicates the need of multiple views when analysing the influence. For example, the betweeness

centrality Xiaomi, Baidu, and Microsoft in the top compared to the outdegree centrality. This observation indicates that they were involved in more collaboration but produced lower weighted (LOC) contributions relative to their collaborators.

Evolution of Ecosystem in Regards to Innovation and Time-To-Market (RQ-2). The analysis results indicate that the number of implemented features does not vary among the analyzed releases. A possible reason for this could be the ecosystem's history. From release R2.2 to R2.5, the project was dominated by one central stakeholder (Hortonworks). Although, additional stakeholders with more influence emerged in release R2.6 and R2.7, Hortonworks remained the dominating contributor, who presumable continued definition and implementation of feature issues. Another potential reason for the lack of variance among features could be the fact that our analysis aggregated all data of third level minor releases to the upper second level releases.

However, our results indicate that the number of implemented improvements show variations among the releases. From release R2.2 to R2.5, the number of implemented improvements per release remained at a steady level. For release R2.6 and R2.7, the number of implemented improvements increased (double the amount). A possible reason for the observed effect could be the fact that other stakeholders with business models get involved in the project to improve the existing ecosystem with respect to their own strategic goals that helps to optimally exploit for their own purpose. The number of implemented bugs varies among all analyzed releases. The high variance of the number of defects could be a side effect of the increased number of improvement issues that potentially imply increase in overall complexity within the ecosystem. Further, the more stakeholders get actively involved in the project to optimize their own business model the more often the ecosystem is potentially used, which may increase the probability to reveal previously undetected defects.

The analysis results with respect to the evolution of the change size indicate a strong variance among all analyzed releases. Similarly to the change size, the time-to-market measure showed great variance among the analyzed releases. Co-variances of stakeholder collaboration, degree of innovation, and time-to-market measure among the analyzed releases may indicate relationship between these variables. However, to draw this conclusion a detailed regression analysis of multiple ecosystems is required.

Implications for Practitioners. Even though an ecosystem may have a high population, its governance and project management may still be centered around a small group of stakeholders [11], which may further be classified as keystone and in some cases, niche players. Understanding their evolving composition and the influence of these stakeholders may indicate current and possible future directions of the ecosystem [7]. Corporate stakeholders could use this information to better align their open source engagement strategies to their own business goals [24]. It could further provide insights for firms, to what stakeholders' strategic partnerships should be established to improve their strategic influence on the ecosystem regarding, e.g., requirement elicitation, prioritization and release planning [16]. Here it is of importance to know how the requirements are communicated throughout the ecosystem, both on a strategic and operational level for a stakeholder to be able to perform the RE processes along with maximized use of its

influence [17]. Potential collaborators may, for example, be characterized with regards to their commitment, area of interest, resource investment and impact [30].

The same reasoning also applies for analysis of competitors. Due to the increased openness and decreased distance to competitors implied by joining an ecosystem [7], it becomes more important and interesting to track what the competitors do [5]. Knowing about their existing collaborations, contributions, and interests in specific features offer valuable information about the competitors' strategies and tactics [24]. The methodology used in this study offers an option to such an analysis but needs further research.

Knowledge about stakeholder influence and collaboration patterns may provide important input to stakeholders' strategies. For example, stakeholders may develop strategies on if or when to join an OSS ecosystem, if and how they should adapt their RE processes internally, and how to act together with other stakeholders in an ecosystem using existing practices in OSS RE (e.g., [19] [18]). This regards both on the strategic and operational level, as requirements may be communicated differently depending on abstraction level, e.g., a focus area for a road map or a feature implementation for an upcoming release [17]. However, for the operational context in regards to how and when to contribute, further types of performance indicators may be needed. Understanding release cycles and included issues may give an indication of how time-to-market correlates to the complexity and innovativeness of a release. This in turn may help to synchronize a firm's release planning with the ecosystem's, minimizing extra patchwork and missed feature introductions [6]. Furthermore, it may help a firm planning their own ecosystem contributions and maximize chances for inclusion. In our analysis, we found indications that the time-to-market and the innovativeness of a release is influenced by the way how stakeholders collaborate with each other. Hence, the results could potentially be used as time-to-market and innovativeness predictors for future releases. This however also needs further attention and replication in future research.

6 Conclusions

The Apache Hadoop ecosystem is generally weakly connected in regards to collaborations. The network of stakeholders per release consists of a core that is continuously present. A large but fluctuating number of stakeholders work independently. This is emphasized by the fact that a majority of the issues are implemented by the issue reporters themselves. The analysis further shows that the network maintains an even size. One can see that the stakeholders' influence as well as collaborations fluctuate between and among the stakeholders, both competing and non-rivaling. This creates further input and questions to how direct and indirect competitors reason and practically work together, and what strategies are used when sharing knowledge and functionality with each other and the ecosystem.

In the analysis of stakeholders' influence, a previously proposed methodology was used and advanced to also consider relative size of contributions, and also interactions on an issue level. Further, the methodology demonstrates how an awareness of past, present and emerging stakeholders, in regards to power structure and collaborations may be created. Such an awareness may offer a valuable input to a firm's stakeholder management, and help them to adapt and maintain a sustainable position in an open source ecosystem's governance. Consequently, it may be seen as a pivotal part and enabler for a

firm's software development and requirements engineering process, especially considering elicitation, prioritization and release planning for example.

Lastly, we found that innovation and time-to-market of the Apache Hadoop ecosystem strongly varies among the different releases. Indications were also found that these factors are influenced by the way how stakeholders collaborate with each other.

Future research will focus on what implications stakeholders' influence and collaboration patterns have in an ecosystem. How does it affect time-to-market and innovativeness of a release? How does it affect a stakeholder's impact on feature-selection? How should a firm engaged in an ecosystem adapt and interact in order to maximize its internal innovation process and technology advancement?

Acknowledgments. This work was partly funded by the SRC in the SYNERGIES project, Dnr 621-2012-5354, and BMBF grant 01IS14026B.

References

1. Henry William Chesbrough. *Open innovation: The new imperative for creating and profiting from technology*. Harvard Business Press, 2006.
2. Joel West and Scott Gallagher. Challenges of open innovation: the paradox of firm investment in open-source software. *R&d Management*, 36(3):319–331, 2006.
3. Slinger Jansen, Anthony Finkelstein, and Sjaak Brinkkemper. A sense of community: A research agenda for software ecosystems. In *31st International Conference on Software Engineering*, pages 187–190. IEEE, 2009.
4. Johan Linåker, Björn Regnell, and Hussan Munir. Requirements engineering in open innovation: a research agenda. In *Proceedings of the 2015 International Conference on Software and System Process*, pages 208–212. ACM, 2015.
5. Linus Dahlander and Mats Magnusson. How do firms make use of open source communities? *Long Range Planning*, 41(6):629–649, 2008.
6. Krzysztof Wnuk, Dietmar Pfahl, David 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 271–280. ACM, 2012.
7. Slinger Jansen, Sjaak Brinkkemper, and Anthony Finkelstein. Business network management as a survival strategy: A tale of two software ecosystems. *Proccedings of the 1st International Workshop on Software Ecosystems*, pages 34–48, 2009.
8. Ellen Enkel, Oliver Gassmann, and Henry Chesbrough. Open r&d and open innovation: exploring the phenomenon. *R&d Management*, 39(4):311–316, 2009.
9. Konstantinos Manikas and Klaus Marius Hansen. Software ecosystems–a systematic literature review. *Journal of Systems and Software*, 86(5):1294–1306, 2013.
10. 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.
11. Kumiyo Nakakoji, Yasuhiro Yamamoto, Yoshiyuki Nishinaka, Kouichi Kishida, and Yunwen Ye. Evolution patterns of open-source software systems and communities. In *Proceedings of the international workshop on Principles of software evolution*, pages 76–85. ACM, 2002.
12. Martin Glinz and Roel J Wieringa. Guest editors' introduction: Stakeholders in requirements engineering. *IEEE Software*, 24(2):18–20, 2007.
13. Carla Pacheco and Ivan Garcia. A systematic literature review of stakeholder identification methods in requirements elicitation. *Journal of Systems and Software*, 85(9):2171–2181, 2012.

14. Daniela Damian, Sabrina Marczak, and Irwin Kwan. Collaboration patterns and the impact of distance on awareness in requirements-centred social networks. In *15th IEEE International Requirements Engineering Conference*, pages 59–68. IEEE, 2007.
15. Soo Ling Lim, Daniele Quercia, and Anthony Finkelstein. Stakenet: using social networks to analyse the stakeholders of large-scale software projects. In *Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering*, pages 295–304. ACM, 2010.
16. Samuel Fricker. Requirements value chains: Stakeholder management and requirements engineering in software ecosystems. In *Requirements Engineering: Foundation for Software Quality*, pages 60–66. Springer, 2010.
17. Eric Knauss, Daniela Damian, Alessia Knauss, and Arber Borici. Openness and requirements: Opportunities and tradeoffs in software ecosystems. In *IEEE 22nd International Requirements Engineering Conference (RE)*, pages 213–222. IEEE, 2014.
18. Neil Ernst and Gail C Murphy. Case studies in just-in-time requirements analysis. In *IEEE Second International Workshop on Empirical Requirements Engineering*, pages 25–32. IEEE, 2012.
19. Walt Scacchi. Understanding the requirements for developing open source software systems. In *Software, IEE Proceedings-*, volume 149, pages 24–39. IET, 2002.
20. Anh Nguyen Duc, Daniela S Cruzes, Claudia Ayala, and Reidar Conradi. Impact of stakeholder type and collaboration on issue resolution time in oss projects. In *Open Source Systems: Grounding Research*, pages 1–16. Springer, 2011.
21. Kevin Crowston and James Howison. The social structure of free and open source software development. *First Monday*, 10(2), 2005.
22. Juan Martinez-Romo, Gregorio Robles, Jesus M Gonzalez-Barahona, and Miguel Ortuño-Perez. Using social network analysis techniques to study collaboration between a floss community and a company. In *Open Source Development, Communities and Quality*, pages 171–186. Springer, 2008.
23. Alma Orucevic-Alagic and Martin Höst. Network analysis of a large scale open source project. In *40th EUROMICRO Conference on Software Engineering and Advanced Applications*, pages 25–29, Verona, Italy, 2014. IEEE.
24. Jose Teixeira, Gregorio Robles, and Jesús M González-Barahona. Lessons learned from applying social network analysis on an industrial free/libre/open source software ecosystem. *Journal of Internet Services and Applications*, 6(1):1–27, 2015.
25. Per Runeson and Martin Höst. Guidelines for conducting and reporting case study research in software engineering. *Empirical software engineering*, 14(2):131–164, 2009.
26. Dan Knight, Robert M Randall, Amy Muller, Liisa Välikangas, and Paul Merlyn. Metrics for innovation: guidelines for developing a customized suite of innovation metrics. *Strategy & Leadership*, 33(1):37–45, 2005.
27. Abbie Griffin. Metrics for measuring product development cycle time. *Journal of product innovation management*, 10(2):112–125, 1993.
28. Barry J Nalebuff and Adam M Brandenburger. Co-opetition: Competitive and cooperative business strategies for the digital economy. *Strategy & leadership*, 25(6):28–33, 1997.
29. Joachim Henkel, Simone Schöberl, and Oliver Alexy. The emergence of openness: How and why firms adopt selective revealing in open innovation. *Research Policy*, 43(5):879–890, 2014.
30. Jesus M Gonzalez-Barahona, Daniel Izquierdo-Cortazar, Stefano Maffulli, and Gregorio Robles. Understanding how companies interact with free software communities. *IEEE software*, (5):38–45, 2013.