Open Tools for Software Engineering using the Theory of Openness: A Validation Study in the Automotive Industry

Hussan Munir, Per Runeson
Lund University, Department of Computer Science
P.O. Box 118, SE-221 00
Lund, Sweden
hussan.munir, per.runeson@cs.lth.se

Krzysztof Wnuk
Blekinge Institute of Technology, Software engineering research lab
Karlskrona, Sweden
krzysztof.wnuk@bth.se

ABSTRACT

Context: Open tools (e.g., Jenkins, Gerrit and Git) offer features or performance benefits that surpass their commercial counterparts. Many companies and developers from OSS communities create open tools in a collaborative effort in which software developers improve the code and share the changes within the community. We developed an empirically based theory for strategic choices on such tool development.

Aim: The aim of this study is to validate the theory of openness for tools in software engineering.

Method: We launched surveys in focus groups in two automotive industry companies and used the repertory grid technique to analyze the responses from participants in combination with qualitative data from discussions in the focus groups.

Results: We validated the theory in the *laggards* category (reactive, cost saving), as both companies belong to that category. Tools provided by suppliers are customized according to company needs to integrate into an already complex tool-chain and thereby incurs expensive licenses. The lack of central tool coordination leads to multiple variants of the same tools, causing additional costs to glue tools together. Further, the lack of legal frameworks to work with OSS tools communities hampers companies to engage developers in OSS tools.

Conclusion: Both companies need a centralized, proactive strategy to help software developers use open standardized tools to reduce integration issues. It may require companies to foster an internal champion, which serves as an interface between the legal department, software developers and top management, to drive the open tools strategy.

Categories and Subject Descriptors

D.2.8 [**Software Engineering**]: Metrics—*complexity measures, performance measures*

General Terms

Theory

Keywords

Open Innovation, theory of openness, open tools

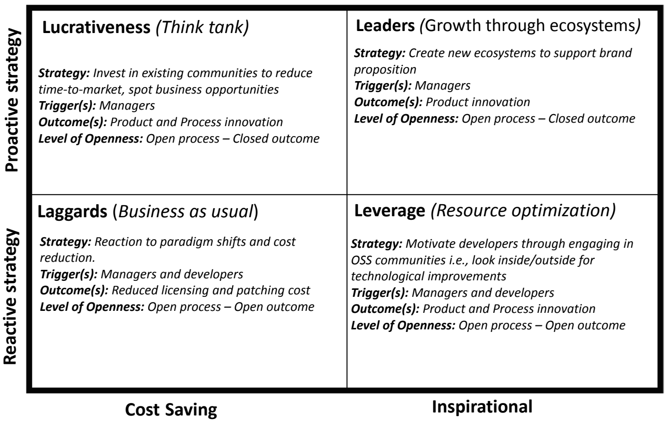
1. INTRODUCTION

Using and co-developing open tools with other organizations for proprietary product development is becoming an increasingly open process, thanks to the ever-growing size of Open Source Software (OSS). Software-intensive companies create open tools communities and develop these tools with other companies to share the development and maintenance cost. For example, the Gerrit code review tool, built on top of the git version control system, was developed by Google and made open source in relation to Android development. Jenkins is another OSS tool co-developed by many companies (e.g., Sony Mobile, Ericsson, Intel, SAP etc) to make their continuous integration process faster and more flexible. Open tools may help companies to reduce the licensing costs for proprietary tools, and save development costs by offering flexibility in the development environment, increased turnaround speed, faster upgrades/releases and sharing the maintenance cost [18]. However, open tools require investment in the communities if companies would like to be leaders and gain influence on the development direction of these OSS tools [24].

This study continues our previous research efforts where we: 1) conducted a mapping study to identify the existing evidence on the use of open innovation in software engineering [20], 2) performed an exploratory case study at Sony about the use of open tools (e.g., Gerrit and Jenkins) in proprietary product development [18] and 3) conducted a survey in OSS tools communities (e.g., Gerrit, Jenkins, Git) and combined with the existing evidence to develop a theory of openness for tools in software engineering [19]. Figure 1 shows that the theory presents four categories of openness in companies with their respective focus:

1. Laggards – Routine business
2. Leverage – Resource optimization
3. Lucrativeness – Acting as a think-tank
4. Leaders – Growth through ecosystems

Figure 1: Model of openness for tools



Lucrativeness *(Think tank)*

Strategy: *Invest in existing communities to reduce time-to-market, spot business opportunities*
Trigger(s): *Managers*
Outcome(s): *Product and Process innovation*
Level of Openness: *Open process – Closed outcome*

Leaders (Growth through ecosystems)

Strategy: *Create new ecosystems to support brand proposition*
Trigger(s): *Managers*
Outcome(s): *Product innovation*
Level of Openness: *Open process – Closed outcome*

Laggards (*Business as usual*)

Strategy: *Reaction to paradigm shifts and cost reduction.*
Trigger(s): *Managers and developers*
Outcome(s): *Reduced licensing and patching cost*
Level of Openness: *Open process – Open outcome*

Leverage *(Resource optimization)*

Strategy: *Motivate developers through engaging in OSS communities i.e., look inside/outside for technological improvements*
Trigger(s): *Managers and developers*
Outcome(s): *Product and Process innovation*
Level of Openness: *Open process – Open outcome*

Table 1: Propositions from the theory [19]

ID	Proposition
P1	Openness of tools provides opportunities to reduce development costs.
P2	Openness of tools provides opportunities to shorten the development time.
P3	Openness of tools complements internal processes and product innovation.
P4	The degree of investment in OSS communities has an affect on the outcome.
P5	Introducing a proactive strategy, in relation to openness of tools, requires conscious management involvement.

Each category has the different levels of openness, based on their strategies (proactive or reactive) in relation to goals (cost saving or inspirational). Typically, *laggards* respond to paradigm shifts and all strategies are reactive, aiming to reduce the development cost (i.e. integration). As for the *leverage* category, organizations use the external sources of innovation by inspiring their internal developers to participate in various OSS tools communities, prior to internal R&D work. It not only adds to product and process innovation but also inspires developers to exchange ideas on discussion forums to develop competence. *Lucrativeness* refers to investing in existing OSS communities to be able to influence and steer these communities in the same direction as the organizational interests. The purpose is to support internal innovation and reduce costs by investing in OSS tools communities. *Leaders* are organizations that focus on creating new communities and ecosystems to strengthen their business model. As a part of defining the theory of openness for tools, we formulated five proposition shown in table 1. The main goal of this study is to validate the theory of openness for tools in software engineering by conducting workshops in two automotive companies. Propositions derived from the theory of openness are validated based on the data collected from the workshop. The first two authors were directly involved in conducting the focus groups while the other two authors reviewed the collected data.

In this paper, we introduce related work on open source tools, as well as the analysis methods used here, in Section 2. Section 3 introduces the case companies, data collection procedures, validity threats, and analysis method to validate the propositions. Section 4 presents the analysis and discussion based on the repertory grid technique, using focus groups in the case companies. Finally, section 5 wraps up the study with the conclusion.

2. RELATED WORK

Open Innovation (OI) builds on internal and external knowledge exchange that may or may not be associated with monetary transactions. OI can be realized as inside-out, outside- or coupled innovation processes [9]. OI has penetrated into several industries, as many companies discovered that their business may benefit by collaborating with OSS tools communities [6]. Companies apply OI in the area of OSS tools for company's internal product development. For example, our previous study [18] shows that Sony Mobile uses Jenkins and Gerrit ecosystems for knowledge flows between different companies' employees and Open Source Software (OSS) community developers, without monetary transactions. In contrast, companies like CloudBees monetize services related to the OSS tools. The stakeholders in the OSS tools communities not only share the innovation cost [5] and rewards, but also risks [18]. Companies may achieve that by revisiting their tool chains and look for open tools alternatives, as a natural and low-risk starting point in embracing openness.

The use of proprietary tools for software development leads to several challenges, e.g. delayed implementation of requirements, expensive licensing costs, inability of fixing things in-house, lack of customizability, and difficulty in finding solutions that meet current needs [18]. Companies use OSS tools communities to deal with several of these challenges [18]. However, in order to utilize open source tools communities for internal product development, companies need to invest their internal resources to use or contribute to OSS tools communites and have contribution strategies around it. The contribution strategies guides companies when to contribute and when to conceal in relation to their business model. The proposed theory of openness helps companies to choose the right level of openness while working with OSS tools communities.

Generating and acquiring data from software engineering activities is relatively easy. The challenge is to use that data in a meaningful way to present something that is true rather than spurious. Theories offer common conceptual frameworks to summarize, condense and accumulate knowledge. SE research community have raised concerns on the lack of theories in SE [2, 8, 11, 26, 25, 27, 4] and pointed on the limited nature of the work about SE theories. We used Sjøberg et al's method to design the theory of openness [26] and here we use the repertory grid technique to validate the theory. Repertory grid is a technique for identifying the ways that a person interprets or gives meaning to his or her experience, being used in software engineering already. Moynihan's exploratory study used the repertory grid technique to identify the prevalent risks factors in software development projects. [17, 16]. Lee and Truex [13] used repertory grid technique to explore whether or not training in information system development methods improves students' cognitive structures (e.g., focused, tight thinking). Baddoo's exploratory study [1] collected data from 13 collaborating software companies to investigate how groups of staff in three designated roles (i.e. developers, project managers and senior managers) saw themselves and the other roles as being involved in software process improvement. Ghazi et al. [10] proposed a decision support method, using the repertory grid technique, which aids practitioners to choose the right level of exploratory testing (i.e. freestyle testing, a high degree of exploration, medium degree of exploration, a low degree of exploration and scripted) in their context.

3. RESEARCH METHODOLOGY

Below, we describe the case companies, that data collection and analysis methods, and our discussion and actions taken to improve the validity of the study.

3.1 Case A

Company A is a major automotive industry manufacturer of heavy trucks and buses. It also manufactures diesel engines for heavy vehicles as well as marine and general industrial applications. The company employs approximately 42,100 people around the world. The company develops its software services for fleet management and is considering transport as a service perspective in their business model. Albeit their internally developed software is not monetized yet, the company has started seeing the software as an important strategic area for its core business. Therefore, the company has an abstract strategy of utilizing OSS tools to build competence in this area. The company uses Jenkins in the development of their fleet management system. The participants involved in the workshop were tools manager, product owners and technical teams leads and tools engineers.

3.2 Case B

Company B is one of the well-known global brands in automotive industry, heading towards an all-electric future. The company has 34,200 employees in five continents, and 2300 dealers globally. The increased use of software in the cars makes embedded software development a cutting-edge area for the company's business model. In order to facilitate the software development, the company buys tools to facilitate the software development from suppliers. However, it makes it difficult to buy a solution and fit it into the existing tool chain. Therefore, the company is shifting towards OSS tools to achieve standardization in all teams. Tools like Jenkins, Gerrit and Git are already utilized in the development process. The tool specially discussed in this study is an internally developed tool named *Awesome framework* by the company. It is an automated testing framework which enables test automation for hardware-in-the-loop, software-in-the-loop, model-in-the-loop test environments to be able to fit these into a continuous integration chain. It provides an integrated development environment for developing such tests at component and system level. The participants of the workshop in the case company include tools manager, product owners, business analysts, technical teams leads and tools engineers.

3.3 Data collection and analysis methods

Focus groups [23] were conducted at the two companies involved in this study. There were 12 and 10 participants in the focus groups conducted at the companies A and B, respectively. Furthermore, we used a repertory grid technique [12] to analyze and validate the theory of openness. Kelly proposed the personal construct theory (PCT) and the associated repertory grid technique in the 1950's to elicit and analyze personal constructs [12]. The idea behind the theory is that participants have their own view of the world based on their observation of surroundings. Therefore, each participant builds his own conceptual framework which leads to different opinions about the same problem. Participants constantly observe and react to their understanding of the surroundings. Consequently, participants reform their personal theories and assumptions [7].

Kelly's repertory grid technique is used to elicit, evaluate and analyze the constructs [12]. The grid is comprised of following three basic concepts: 1) Elements elicitation, 2) Constructs elicitation, 3) Ratings. *Elements* refer to individual aspects or objects of a topic, which participants try to understand. A *construct* is made of two contrasting concepts that are equally weighted on a bipolar scale. There are two essential ways to select grid elements: a) elicit elements from participants, b) provide participants with elements.

We used repertory grid to validate the theory of openness by providing the elements and constructs derived from the theory of openness [19]. Each elements was rated against each construct in the focus groups. To further support our choice, a number of studies have taken this approach [22, 28, 21, 13] by providing elements and constructs to participants for the ratings. The description of elements, constructs and their ratings are as follow:

Elements derived from theory of openness: This study uses four elements from the theory of openness mentioned below:

1. Inspiration
2. Reactive
3. Cost saving
4. Proactive

The elements can be seen in figure 2 where we used the same elements for companies A and B.

Constructs derived from the theory of openness: We designed constructs from the theory of openness by creating the contrasting poles of each construct, see A1, A2 ... in Table 2. These constructs are derived from the existing literature in the theory of openness [19]. The constructs asses company's perspective on creating new communities to facilitate internal product development by reducing the development, development cost, new creation of roles, approval from top management for OSS tools adoption and access to skilled workforce etc.

Ratings: In this step, elements are then rated against each construct in the focus groups at companies A and B. We used a three-point Likert scale to rate each element against each construct, see Table 3.

First, the focus group participants in both companies were given an introduction of constructs and elements to develop a common understanding in the whole group. Second, participants from company A picked *Jenkins* and participants from company B selected an internal tool called *Awesome framework* for the focus groups. Third, a survey link was distributed among all the participants to rate each element against the constructs, based on the selected tools from their internal development environment. Fourth, each element was rated against each construct using the scale in table 3.

Table 2: Constructs derived from theory of openness with their contrasting poles

ID	Similarity Pole (+)	Contrast Pole (-)
A1	Creating new open source communities to facilitate internal product development	Creating new open source communities not required to facilitate internal product development
A2	Creating new open source communities to explore emerging technologies	Creating new open source communities not required to explore emerging technologies
A3	Leveraging OSS to increase market share	Leveraging OSS to increase market share not required
A4	Using existing open source communities to identify the emerging technologies	Using existing open source communities to identify the emerging technologies not required
A5	Using existing open source communities for technological improvements	Using existing open source communities for technological improvements not required
A6	Investing in the open source communities to steer the community's development towards organizational benefits	Investing in the open source communities not required to steer the community's development towards organizational benefits
A7	Defining new work roles and teams to work with open source communities	Defining new work roles and teams not required to work with open source communities
A8	Adopting OSS tools need approval from top management	Adopting OSS tools does not need approval from top management
A9	Contributing to open source communities need approval from top management	Contributing to open source communities does not need approval from top management
A10	Educating the organization about OSS culture and ways of working	Educating the organization about OSS culture and ways of working not required
A11	Using OSS tools helps to reduce development time	Using OSS tools helps to reduce development time
A12	Working with open source communities gives access to free labor	Working with open source communities gives access to free labor
A13	Using OSS tools keeps developers updated with best tools practices	Using OSS tools keeps developers updated with best tools practices
A14	OSS tools reduces licensing cost in relation to proprietary tools	OSS tools reduces licensing cost in relation to proprietary tools
A15	The benefits of engaging in open source communities outweighs the risk of losing Intellectual property rights	The benefits of engaging in open source communities does not outweighs the risk of losing Intellectual property rights
A16	Engaging in open source communities provides fun ways of working for developers	Engaging in open source communities provides boring ways of working for developers
A17	Working with open source communities gives access to external knowledge from communities	Working with open source communities does not gives access to external knowledge from communities

Table 3: Scale used for the ratings

	Scale		
Elements	**1**	**2**	**3**
Inspirational	Inspirational	Somewhat	No Inspirational
Reactive	Reactive	Somewhat	Not Reactive
Cost saving	Cost saving	Somewhat	Not Cost saving
Proactive	Proactive	Somewhat	Not Proactive

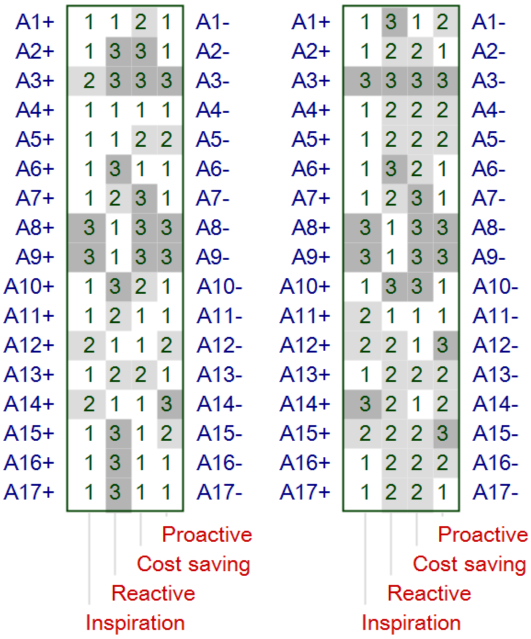
For example, Figure 2 shows that adopting OSS tools need approval from top management **(A8)** is rated as not inspirational (3), reactive (1), not cost saving (3) and not proactive (3) by the participants in both companies. Similarly, the use of OSS tools helps to reduce the development time **(A11)** is rated as a cost saving (1) and proactive (1). Third, we held a discussion among participants based on the ratings to further explore their ratings. The discussion part was recorded and transcribed to further explore the rationales for the participant's ratings.

Outcome: The outcome of this step (ratings) is Figure 2, which shows ratings of focus groups from companies A and B.

3.4 Validity threats

We discuss validity threats with respect to internal validity, external validity, construct validity, and conclusion validity [23].

Internal validity: Internal validity threat may relate to confounding factors that may have influenced the outcome without researcher's knowledge. One possible threat is that the participants in companies A and B may have misunderstood the constructs and elements. Therefore, an introduction presentation was given to participants before rating each element against constructs. Constructs and elements were exemplified during the introduction presentation session. Furthermore, we also had a follow-up discussion based on the ratings from the participants to clarify misunderstandings. Another threat to internal validity was that the participants might had a difficulty in interpreting the scale for the ratings. To minimize the risk of scale's misunderstanding, the examples were given in the introduction prior



	Inspiration	Reactive	Cost saving	Proactive	
A1+	1	1	2	1	A1-
A2+	1	3	3	1	A2-
A3+	2	3	3	3	A3-
A4+	1	1	1	1	A4-
A5+	1	1	2	2	A5-
A6+	1	3	1	1	A6-
A7+	1	2	3	1	A7-
A8+	3	1	3	3	A8-
A9+	3	1	3	3	A9-
A10+	1	3	2	1	A10-
A11+	1	2	1	1	A11-
A12+	2	1	1	2	A12-
A13+	1	2	2	1	A13-
A14+	2	1	1	3	A14-
A15+	1	3	1	2	A15-
A16+	1	3	1	1	A16-
A17+	1	3	1	1	A17-

(a) Company A

	Inspiration	Reactive	Cost saving	Proactive	
A1+	1	3	1	2	A1-
A2+	1	2	2	1	A2-
A3+	3	3	3	3	A3-
A4+	1	2	2	2	A4-
A5+	1	2	2	2	A5-
A6+	1	3	2	1	A6-
A7+	1	2	3	1	A7-
A8+	3	1	3	3	A8-
A9+	3	1	3	3	A9-
A10+	1	3	3	1	A10-
A11+	2	1	1	1	A11-
A12+	2	2	1	3	A12-
A13+	1	2	2	2	A13-
A14+	3	2	1	2	A14-
A15+	2	2	2	3	A15-
A16+	1	2	2	2	A16-
A17+	1	2	2	1	A17-

(b) Company B

Figure 2: Ratings of elements in relation to constructs from focus group participants, using the three point Likert scale presented in table 3.

to ratings of elements and constructs.

External validity: Both companies are experiencing a paradigm shift with ever increasing adoption of software in cars and trucks to stay on the competitive edge. Validating the theory in an automotive industry gives more weight to the external validity of the theory. However, automotive industry does not represent the whole software industry and therefore, the scope of the findings are limited. More studies are intended to perform in the software industry to improve the external validity of the theory.

Construct validity: Both constructs and elements in the theory are relevant to discuss in relation to construct validity. Neither of the case companies comes from an OSS development and community participation background and therefore, do not have established procedures to work with open tools communities. As a consequence, neither company has a well-defined procedure to map all the constructs of the theory. This threat was partially met by keeping the discussion on a higher level to the company's specific context.

Conclusion validity: First, participants in the focus groups may have tried guessing the propositions during the introduction presentation, which introduces a threat to the conclusion validity. This threat was mitigated by keeping elements and constructs as discrete as possible. Second, a researcher's bias while interpreting and recording focus groups mat threaten the conclusions. To reduce this threat, multiple researchers were involved in the focus groups (observer triangulation). Furthermore, focus groups were recorded, transcribed, and the results were validated by multiple researchers.

4. RESULTS AND DISCUSSION

The outcomes from the repertory grid technique are the grid and a PCA analysis. Figure 2 shows the formation of the repertory grid for companies A and B, based on the ratings from the participants in the focus group meetings. The vertical axis shows the contrasting poles of *constructs* derived from the theory of openness, mentioned in Table 2. The horizontal axis shows the *elements* derived from the theory of openness. Each element was rated by the participants in the workshop in relation to constructs using the scale presented in Table 3. Figure 3 shows the statistical repertory analysis based on the principal components analysis (PCA) [3]. PCA is a data reduction technique, used to find the dimensions of maximum variability in data. Figure 3 represents the spatial distances between and among the elements and constructs, and suggests how they might be related to each other.

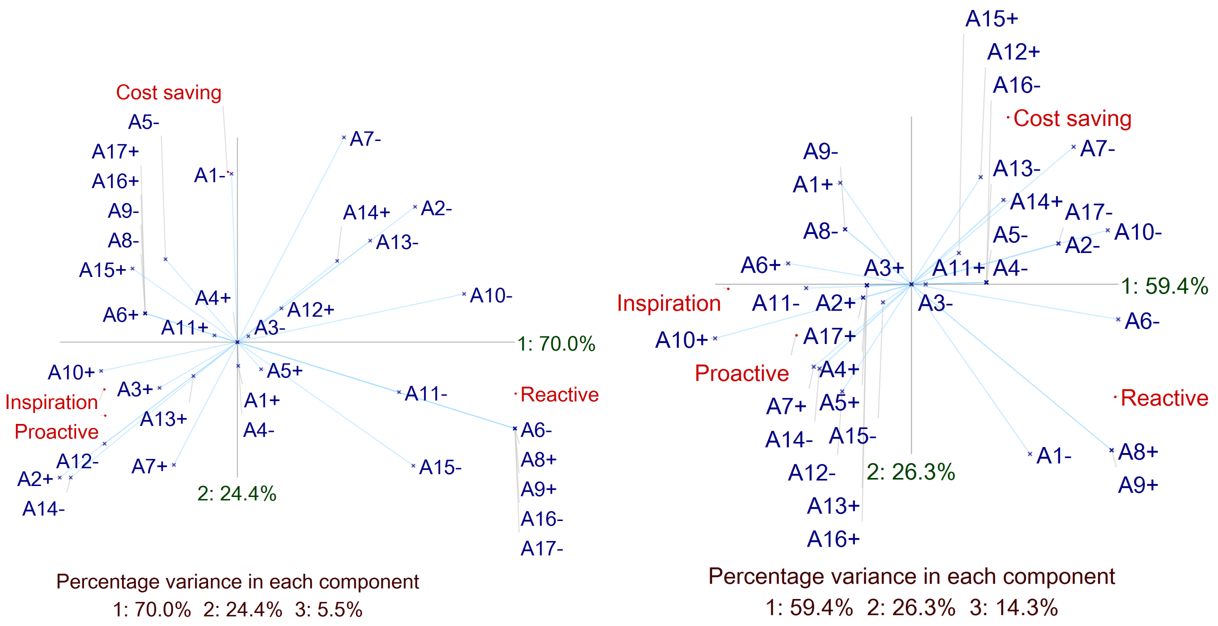
4.1 Relation between strategies

The PCA shows that *proactive* and *inspiration* in companies A and B are spatially closer to each other. However, the *cost saving* and *reactive* strategies in company A seem to have a higher variance in contrast to company B. It can be explained by both companies' current strategy of buying OSS enterprise solutions instead of actively using or contributing to OSS tools communities, because they believe it gives them more control over the contract as the companies think in the old way of subcontracting. The ratings in company A were not able to distinguish between OSS tools, and tools provided by suppliers based on OSS. The reason for this may be the lack of OSS understanding and competence at company A, since the company does not come from an OSS development background. One of the participants stated, *"one of the major reasons for not using OSS tools is the lack of competence and understanding. We do not know how to protect our intellectual property rights. We tend to go for buying solutions when there are intellectual property rights involved"*.

4.2 Validation of propositions

Further, we validate the five propositions (see table 1) from the theory of openness [19] in the light of the PCA and discussions in the focus groups.

For proposition **(P1)** about the reduction in development cost, the PCA shows that access to free labor **(A12+)** and reduced licensing cost **(A14+)** is closer to cost saving in case of company B as opposed to company A (see fig 3a and 3b). One possible explanation is that Company A does not see faster automated build servers as a competitive edge for its internal developed software. Furthermore, none of its software is provided as a service to the customers yet. However, they started seeing value in services and connections, where licensing costs might come into play. One of the participants stated that *"the percentage of money spent on licenses is actually not that high at the moment. However, if we start producing hundreds of micro-services with each of them need their own database, buying Oracle licenses might become really expensive"*. On the other hand, it is not uncommon for company B to buy tools and these tools can be expensive. One of the participants in company B stated that *"customized tools provided by big vendors like MathWorks for the company are really expensive. Therefore, we have a window of opportunity to update our environment with open tools."*. In conclusion, openness of tools has the



(a) Company A

(b) Company B

Figure 3: Principal components analysis (PCA) of ratings from focus groups.

potential to reduce the development cost, which validates our proposition **P1**.

Regarding whether openness shortens the development time **(P2)**, PCA for both companies in Figures 3a and 3b shows that the benefits of engaging in OSS tools communities outweigh the risk of losing IPRs **(A15+)**, and reduced development time **(A11+)** is spatially closer to cost saving. This indicates that participants think that engaging in open tool communities outweighs the risk of losing intellectual property rights, although the lack of maintenance support in the company is a point of concern for developers. Software developers in company B think that using open tools speed up their development in relation to seeking permission to buy a tool. One participant stated that *"regarding speed as a developer, it takes a few minutes to download an open framework rather than seeking an approval from the manager to buy a tool, which takes forever"*. However, both companies are careful in selecting and integrating these open tools to make sure that these tools won't die over time. There is a risk that software developers might not get the required funding and enough management backing to maintain these open tools internally if the community dies. A participant stated that *"we are dependent on some really large python projects that we know won't die anytime soon."*. Therefore, the proposition **P2** about open tools may reduce the development is confirmed by the data.

In relation to process and product innovation **(P3)**, it should be mentioned we have earlier observed that process innovation leads to product innovation [14]. However, neither case company considers tools front leading innovation in the development of their core products. In particular, the tools development team of company B does not get enough funding and management backing, although other departments in the company need tool support to facilitate product innovation in the core products. This is a general problem with the project focused companies, where the budget is allocated for the given project only, with limited consideration for long-term investment. The tools department does not have an allocated budget for the development of tools and the maintenance of these to achieve standardization. One of the participants in company B stated *"when we buy tools, it is part of the R&D budget and there is no designated budget for maintenance. For example, if we get the funding for developing an internal tool like Awesome framework, there is no allocation of budget for the maintenance of that tool"*. On the other hand, both companies identify the need for process innovation by creating new roles [15] to deal with the challenges of open tools platform. Consequently, there have been initiatives in both companies to create new open innovation roles. For example, a participant of company A mentioned that *"the company has actually employed a new role of open source cloud manager to create procedures and guidelines regarding how to be active in an open space"*. Furthermore, a participant in company B stated that *"there is a new initiative called Nordic foundation partly driven by Ericsson to spread open source usage and with the already formed legal framework among Nordic companies. It could be a fast track for us but it needs to be championed by R&D in order to reap benefits"*. Therefore, the proposition **P3** related to the openness of tools complements internal process and product innovation is confirmed by the data.

In both companies, PCA shows that adopting **(A8+)** and contributing to OSS tools **(A9+)** are spatially close to the *reactive* strategy, meaning that participants consider taking approval from top management in either adopting or contributing to open tools communities as a reactive strategy. One possible explanation for seeking approval is that neither company has a legal framework to work with open tools communities. One of the participants in company A stated that *"we have a top-level statement saying* Open Source First *but* *what does that exactly mean? It does not state exactly what, why and how should we go open and how these decisions are supported".* Similarly, a participant in company B also stated that *"we do not have any internal procedure or legal framework to create new or work with the existing open tools communities".* Therefore, **P4** about the relation between investment in OSS tools communities and its impact, is neither confirmed nor rejected by the data, since both case companies lack internal procedures to invest their internal resources in open tools communities.

Finally, proposition **P5**, about the role of management in Table 1, needs to be extended since not only the proactive strategy but also the reactive strategy, requires management involvement, as software developers need a legal framework to work with open tools.

4.3 Cross analysis with Sony Mobile

After the focus group meeting, we conducted a cross case analysis with our previous Sony Mobile case [18]. That study explored Sony Mobile's transition from closed tools to open tools platform [18], which constitutes one of the cases for an empirical grounding of the theory of openness for tools. Therefore, in this cross case analysis, we have found three similar patterns in companies A and B in relation to Sony Mobile.

First, there was a paradigm shift when Sony Mobile moved from Windows to Linux, which in turn required Sony to move from proprietary tools (e.g, ElectricCommander) to open tools platforms (e.g., Gerrit, Jenkins, Git). The complex integration tool chain triggered the move towards open tools platforms and it resulted in increased development speed and reduced time to market. Both companies A and B are experiencing a paradigm shift in the automotive industry, where software is becoming the key area for the development of vehicles such an autonomous driving features and providing transportation as a service. Both companies are very reactive in their approach when opting for open tools. A participant in the company said, *"the change generally depends on how painful the situation is. A few years ago, we had 10 different teams working with an internal platform using different tool chains and had a great difficulty in glowing the tool chain together. We hired an external company that glued our tools chain together with massive scripts using an open tool solution".* Therefore, the open tools chosen in a proactive way of standardizing the tool chain in all teams, may play an important role in the development of software-intensive vehicles.

Second, when Sony Mobile realized that they lacked legal procedures to work with open tools communities, they created an internal open tools department consists of legal experts, developers and managers to create a well-established procedure for using or contributing to open tools platforms. Companies A and B may follow the same steps by creating an internal process to educate the developers and develop competence to work with open tools communities. As one of the participants in company A said, *"we lack competence and streamlined decisions on how to work with open tools communities".* Another participant in company B said that *"we need improvements in the areas of governance, training and policies."*

Third, Sony Mobile had an internal champion to derive the whole open tool-chain movement with the support of software developers and legal team. This played an important role in convincing the top management to support and fund open tools platforms, however, it took three years for Sony Mobile to make it. The current case companies may need a similar, internal champion to act as an interface between the legal team, software developers, and the top management.

Key takeaway: Companies A and B may choose a centralized proactive strategy similar to Sony Mobile (reactive in that case) to make internal legal policies and education for employees to meet the ever-growing size of software in automobiles.

4.4 Business implications

Automotive industry sees software services as a new emerging area in terms of transportation as a service that will be software-enabled. This pattern can be related to Sony and Android case in which natural drive for saving money and reducing ownership costs was to create industry wide standard platforms and Google capitalized on the first mover advantage by offering Android for free and it got widely adopted by vendors. Initially, Sony underestimated the value in services and apps So everyone let Google develop Android and occupy services such as Gmail, drive, gaming, camera etc. Now Android is provided as an empty shell and vendors just provide devices to use services for money. Similar risk may apply to automotive industry, if the companies do not own their code base by establishing OSS communities and ecosystems. Companies may provide vehicles to run the services and make money on the services instead of selling the vehicles.

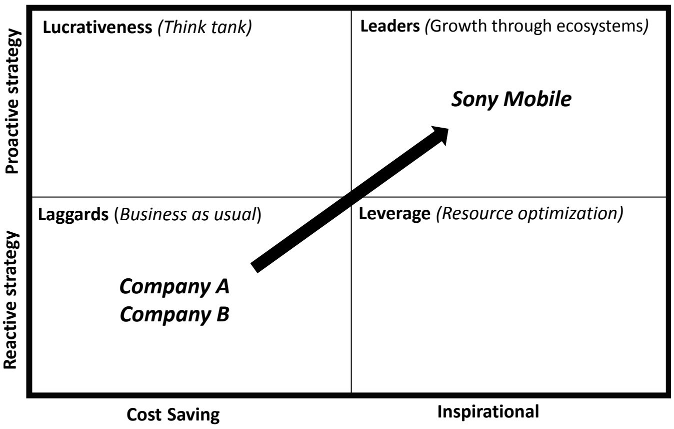
4.5 Implications for theory of openness

Figure 4 shows the mapping of companies A and B on the model of openness for tools [19]. Both companies qualify as **laggards** since they have no internal procedures to facilitate developers to contribute to OSS tools communities. Both companies react when the integration of tools becomes painful for developers, rather than standardize the ways of working in the companies. Also for laggards, we noticed that it is important to have the management support and approval. In case company B, the internally developed tool named *Awesome framework* is meant to be shared among different partners of the company and it is used in core product development teams. However, the development does not have enough resources to support core product development teams who have requirements and need support. This suggests the importance of the development of the tools but lack of funding and resources hampers the work of core product development teams who are dependent on the tool.

The arrow in figure 4 shows that Sony Mobile went through the same transition as companies A and B are going through. Sony Mobile started as a **laggard** with its complex continuous integration tools chain, before actively using and contributing to OSS tools communities like Jenkins and Gerrit. However, this transition required Sony Mobile to create a tools department with the legal framework to utilize the OSS tools communities.

In the case of two studied companies, company B is found to be more in need of this transition due to their complex integration tool chain compared to company A. Therefore, both companies may have processes and legal frameworks for developers, to encourage them to take more initiative towards open tools in order to standardize the ways of working. Currently, in the company B, all teams have their in-

Figure 4: Mapping of companies on the model of openness for tools.


Lucrativeness (Think tank)
Leaders (Growth through ecosystems)
Sony Mobile
Laggards (Business as usual)
Leverage (Resource optimization)
Company A
Company B
Proactive strategy
Reactive strategy
Cost Saving
Inspirational

ternally developed tools with all possible technologies (e.g., C#, Python, Java etc.) which leads to loss of resources on writing more glue code to integrate the tool chain. A participant from company B stated, *"I think we have to change our way of thinking and we never share anything because there is always some kind of minor internal change that drives our testing and we do not want to share that. However, 80% of the rest is the same that can be shared. At the moment, it is all or nothing and we need to calm down a little bit and a share the common parts atleast"*.

Once the processes for contribution and using OSS tools communities are in place, both companies may consider becoming new open tools ecosystems **(leaders)** by making the code open, to attract other manufactures in the industry with the same needs. One of the participants stated, *"I think it is a question of survival in the near future and we do not have time to deal with the increase in the number of variants. It will be nice to replace the closed source tools with the open ones so that we can actually fix tools without glue scripts"*. It is to be noted that the data collected from the case companies does not address or validate the leverage and lucrativeness category in Fig 4. The studied companies only validates the laggards category, thus the theory required more validation studies to validate the lucrativeness and leverage category as well.

Key takeaway: Both companies may learn from Sony Mobile's transition from **laggards** to **leaders** by innovating their process in terms of creating a legal framework. The framework will help companies to engage their developers in OSS tools communities together with the legal team to eventually be able to build new communities (**leaders**) to facilitate their core product development. However, **laggards** also require management to look at tools as a long term standardization of processes perspective rather than a project based approach.

5. CONCLUSIONS

This study aims to validate our theory of openness for software engineering tools. We use the repertory grid technique to discuss five propositions derived from the theory of openness and presents a cross analysis with Sony Mobile. We were able to validate the theory for the laggards category, with two companies from the automotive domain.

The findings suggest that both case companies lack internal procedures to work with the open tools communities. This leads to frustration among employees for not knowing why and how to work with open tools communities. It might be because both companies come from a closed background of manufacturing cars and trucks. However, both companies are currently experiencing a paradigm shift in the automotive industry with more software introduced in the vehicles. This may require the management to rethink and revise their strategy to extract the external knowledge by using open tools communities.

The strategy should address creating a new role which works as an interface between legal department, software developers and management like Sony Mobile. Furthermore, it may entail creating processes regarding how and when to use, contribute or create new open tools platforms for company's benefits. As for future work, more validation studies need to be conducted to validate the propositions derived from the theory of openness.

6. ACKNOWLEDGMENTS

We thank the company participants for sharing their insights in the focus groups. This work was partly funded by the EASE industrial excellence center.

7. REFERENCES

[1] N. Baddoo and T. Hall. Practitioner roles in software process improvement: an analysis using grid technique. *Software Process: Improvement and Practice*, 7(1):17–31, 2002.

[2] V. R. Basili, F. Shull, and F. Lanubile. Building knowledge through families of experiments. *IEEE Transactions on Software Engineering*, 25(4):456–473, 1999.

[3] R. C. Bell. Analytic issues in the use of repertory grid technique. *Advances in personal construct psychology*, 1:25–48, 1990.

[4] E. Bjarnason, K. Smolander, E. Engström, and P. Runeson. A theory of distances in software engineering. *Information and Software Technology*, 70:204–219, 2016.

[5] H. Chesbrough. Why companies should have open business models. *MIT Sloan management review*, 48(2), 2012.

[6] H. Chesbrough, W. Vanhaverbeke, and J. West, editors. *New Frontiers in Open Innovation*. Oxford University Press, Nov. 2014.

[7] H. M. Edwards, S. McDonald, and S. M. Young. The repertory grid technique: Its place in empirical software engineering research. *Information and Software Technology*, 51(4):785–798, 2009.

[8] A. Endres and H. D. Rombach. *A handbook of software and systems engineering: Empirical observations, laws, and theories*. Pearson Education, 2003.

[9] E. Enkel, O. Gassmann, and H. Chesbrough. Open R&D and open innovation: exploring the phenomenon. *R&D Management*, 39(4):311–316, 2009.

[10] A. N. Ghazi, K. Petersen, C. Wohlin, and E. Bjarnason. A decision support method for recommending degrees of exploration in exploratory testing. *arXiv preprint arXiv:1704.00994*, 2017.

[11] J. D. Herbsleb and A. Mockus. Formulation and preliminary test of an empirical theory of coordination in software engineering. In *ACM SIGSOFT Software Engineering Notes*, volume 28, pages 138–137. ACM, 2003.

[12] G. A. Kelly. *The psychology of personal constructs. Vol. 1. A theory of personality. Vol. 2. Clinical diagnosis and psychotherapy.* Norton & Co., 1955.

[13] J. Lee and D. P. Truex. Exploring the impact of formal training in isd methods on the cognitive structure of novice information systems developers. *Information Systems Journal*, 10(4):347–367, 2000.

[14] J. Linåker, H. Munir, P. Runeson, B. Regnell, and C. Schrewelius. A Survey on the Perception of Innovation in a Large Product-focused Software Organization. *6th International Conference on Software Business - ICSOB*, 2015.

[15] C.-E. Mols, K. Wnuk, and J. Linåker. The open source officer role – experiences. In F. Balaguer, R. Di Cosmo, A. Garrido, F. Kon, G. Robles, and S. Zacchiroli, editors, *Open Source Systems: Towards Robust Practices*, pages 55–59, Cham, 2017. Springer International Publishing.

[16] T. Moynihan. An inventory of personal constructs for information systems project risk researchers. *Journal of information technology*, 11(4):359–371, 1996.

[17] T. Moynihan. How experienced project managers assess risk. *IEEE software*, 14(3):35–41, 1997.

[18] H. Munir, J. Linåker, K. Wnuk, P. Runeson, and B. Regnell. Open innovation using open source tools: a case study at Sony Mobile. *Empirical Software Engineering*, pages 1–38, 2017.

[19] H. Munir, P. Runeson, and K. Wnuk. A theory of openness for software engineering tools in software organizations. *Information and Software Technology*, 97:26 – 45, 2018.

[20] H. Munir, K. Wnuk, and P. Runeson. Open innovation in software engineering: a systematic mapping study. *Empirical Software Engineering*, 21(2):684–723, Apr 2016.

[21] N. Niu and S. Easterbrook. Discovering aspects in requirements with repertory grid. In *Proceedings of the 2006 international workshop on Early aspects at ICSE*, pages 35–42. ACM, 2006.

[22] G. J. Phythian and M. King. Developing an expert support system for tender enquiry evaluation: A case study. *European Journal of Operational Research*, 56(1):15–29, 1992.

[23] P. Runeson, M. Höst, A. Rainer, and B. Regnell. *Case Study Research in Software Engineering: Guidelines and Examples.* John Wiley & Sons, Mar. 2012.

[24] P. Runeson, H. Munir, and K. Wnuk. It is more blessed to give than to receive–open software tools enable open innovation. *Tiny Transactions on Computer Science*, 4, 2015.

[25] C. Sauer, D. R. Jeffery, L. Land, and P. Yetton. The effectiveness of software development technical reviews: A behaviorally motivated program of research. *IEEE Transactions on Software Engineering*, 26(1):1–14, 2000.

[26] D. I. Sjøberg, T. Dybå, B. C. Anda, and J. E. Hannay. Building theories in software engineering. In *Guide to advanced empirical software engineering*, pages 312–336. Springer, 2008.

[27] W. F. Tichy. Should computer scientists experiment more? *Computer*, 31(5):32–40, 1998.

[28] S. M. Young, H. M. Edwards, S. McDonald, and J. B. Thompson. Personality characteristics in an xp team: a repertory grid study. In *ACM SIGSOFT Software Engineering Notes*, volume 30, pages 1–7. ACM, 2005.