

Friedrich-Alexander-Universität Erlangen-Nürnberg
Technische Fakultät, Department Informatik

ELCIN YENISEN YAVUZ

MASTER'S THESIS

Industry Success Factors for User Consortia

Submitted on 24.06.2019

Supervisor: Prof. Dr. Dirk Riehle, M.B.A.

Professur für Open-Source-Software

Department Informatik, Technische Fakultät

Friedrich-Alexander University Erlangen-Nürnberg

Versicherung

Ich versichere, dass ich die Arbeit ohne fremde Hilfe und ohne Benutzung anderer als der angegebenen Quellen angefertigt habe und dass die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen hat und von dieser als Teil einer Prüfungsleistung angenommen wurde. Alle Ausführungen, die wörtlich oder sinngemäß übernommen wurden, sind als solche gekennzeichnet.

ERLANGEN, 24.06.2019

License

This work is licensed under the Creative Commons Attribution 4.0 International license (CC BY 4.0), see <https://creativecommons.org/licenses/by/4.0/>

ERLANGEN, 24.06.2019

Abstract

The popularity of open source software development (OSS) strategies is increasing among organizations. Although it is not a new approach, the number of studies which concentrate on the open source user consortia strategy is limited. The goal of this thesis is to provide an overview of this phenomenon by investigating encountered problems in open source user consortia, and factors which influence the success of this strategy. Samples for the study are selected from higher education, automobile and information technologies industries. Sample projects are Sakai (higher education), openMDM (automobile), and GraphQL (information technologies). The multiple-case case study approach is performed in the study. The data sources are interviews, meeting notes, blog posts and conference videos of the key personnel in the project. The results show that the most important factors are related to governance policies in consortia.

Table List

Table 1: Related Literature about OSS Projects Success Factors.....	13
Table 2: Related Literature about Success Factors of Collaborations.....	17
Table 3: Case Selection Matrix.....	19
Table 4: Data Source Matrix.....	22
Table 5: Interview Matrix.....	22
Table 6: Explanation of the assigned codes to the data sources.....	24
Table 7: Summary of the Sample Backgrounds.....	28
Table 8: Problems in User Consortia.....	31
Table 9: Success Factors in User Consortia.....	32

1	INTRODUCTION	6
1.1	Thesis Goals	6
1.2	Change in the Thesis Goals	6
2	RESEARCH CHAPTER	7
2.1	Introduction	7
2.2	Related Work	8
2.2.1	Open Source User Consortium	8
2.2.2	Success Factors	11
2.3	Research Question	19
2.4	Research Approach	20
2.4.1	Research Method	20
2.4.2	Sampling	20
2.5	Used Data Sources	21
2.5.1	Data Collection	21
2.5.2	Data Analysis	23
2.5.3	Sample Characteristics	25
2.6	Research Results	29
2.6.1	Problems	29
2.6.2	Success Factors	37
2.7	Results Discussion	50
2.8	Conclusion	50
2.9	Acknowledgement	51
3	ELABORATION CHAPTER	52
3.1.1	User Consortia Structure and Characteristics	52
3.1.2	Success Measures	61
	Appendix A: Mapping Success Factors of Collaborations	62
	Appendix B: Data Sources	68
	Appendix C: More on Methodology	74
	References	75

1 INTRODUCTION

This section describes the thesis goals that were determined prior to the research and the decided changes during the research phase.

1.1 Thesis Goals

The goal of this thesis is to determine the factors which lead to success or cause problems in open source user consortia.

An open source software user consortium is a non-profit organization (foundation, consortium, working group) created for funding and managing the development of collaborative open source software for the use of consortium members and the general public. Its purpose is to establish a software ecosystem in which vendors and suppliers can provide products and services on an equal playing field to the software user companies. This approach to collaboration sometimes works well, and sometimes, it does not. This thesis makes the assumption about the effects of industry and other factors on the success potential of open source user consortia.

In order to provide an overview of the studies related to this topic, literature reviews about the open source software user consortia and success factors for collaboration will be conducted.

For the theory building, qualitative data analysis method will be followed for surveying known examples for the success factors. Expected results will be the theory of success factors.

1.2 Change in the Thesis Goals

During the literature review process, the methodology of the research has changed. From the qualitative data analysis method, a transmission to the multi-case study approach has decided. By using theoretical sampling, three cases are determined. Primary materials such as meeting notes, website contents, and published papers by the people who lead the sample projects have selected as data sources. Furthermore, interviews with the participants of the sample projects have been conducted.

2 RESEARCH CHAPTER

2.1 Introduction

Open Source Software (OSS) has a long history in computer science. Since the appearance of “open source” concept, the open source software development approach led innovations in both software development and business fields (Riehle, 2019). From the business point of view, open source software is considered to be as an example of open innovation due to its collaborative nature and openness (West and Gallagher, 2004).

The roots of the OSS are dated back to the Free Software Foundation (FSF). FSF was founded by Richard Stallman in 1985. Stallman is the creator of the “copyleft” (as opposition to copyright) and “General Public License” terms (von Hippel, 2003). In 1998, Bruce Perens, who was advocating the free software movement, created the “Open Source” definition by deriving the terms from the Debian Free Software Guidelines. At the same year, Bruce Perens and Eric Raymond founded the Open Source Initiative (Open Source Initiative, 2018; Wikipedia contributors, 2019). Open source initiative is “maintaining the Open Source Definition for the good of the community” (Open Source Initiative, n.d.).

Although in the beginning, the open source approach took attention of the software developers and engineers, with the change in time and conditions, not only individuals but also organizations started to involve OSS projects. Bailetti (2009) explains the relationship between OSS projects and organizations in a 5-level open source maturity curve. Companies start to interact with OSS by using and promoting it. In this phase, the company does not contribute to the code, only use and create awareness about it. The next step is contributing to the code, extending its features to reach a better quality. In the third stage, companies build new software and open source the developed code. Further levels are the co-creation and redefining the standards. The decisions about the first two stages are grouped as more engineering oriented. The next phases starting from the “build new” level are classified as more business-related decisions. Although most companies are at the first stage of this OS maturity curve, some companies are at the higher stages of this curve.

Some companies show a tendency to sponsor open source communities or open source their in-house developed codes because OSS development approach leads to a new form of value creation. By following this approach, organizations establish de facto standards, lead new improvements, create markets for complementary products and services, and generate goodwill among the audience (West and Mahony, 2005).

As a further form of benefiting the OSS development strategy is co-creation, which means collaborating for development. By this way, the involved parties can share resources and risk. First examples of this form of collaboration projects appeared in the higher education industry at the beginning of the 2000s (Wheeler, 2007). Since then, companies in different industries such as finance, automotive, energy are practicing collaboratively open source software development.

According to 2018 Open Source Program Management Survey (Hecht and Clark, 2018), 37% of the companies surveyed across all industries are using an open source program. Furthermore, 16% of the companies are planning to use an open source program in the next years. This usage ratio increases to 63% when the companies with more than 10,000 employees are considered. On the other hand, the companies which use the OSS are using it for non-commercial or internal reasons with a ratio of 80%. The ratio of companies which contribute the code upstream is 44%. The Results show that companies are interested in using open source software for non-commercial reasons and have the opportunity to contribute to the code.

Although various studies about the effects of OSS development approach in business are performed, scientific research about open source user consortia phenomenon is limited. This preliminary research aims to present the encountered problems, and factors which influence the success of the open source user consortia.

The structure of this thesis is as follows: library 2.2 presents the related literature about open source user consortia definition, examples, characteristics, success factors for open source software projects and success factors for collaborative projects. Sections 2.3, 2.4, 2.5 provide information about the research question, research approach, and data sources, respectively. The findings of the research are introduced in section 2.6. The research concludes with the discussion section and future research directions.

2.2 Related Work

This section presents an overview of two aspects of the research. The first part introduces the literature about the open source user consortia, and the second part provides literature about success factors for open source software and collaborations.

2.2.1 Open Source User Consortium

Related literature about the open source user consortium is grouped into definition, examples, and characteristics sub-sections.

2.2.1.1 Definition

In the literature, different terms are used to define user consortium model of open source development. Although different authors use various terms, the main characteristics of this type of consortium are explained as being driven by user organizations, not by volunteer individuals (Wheeler 2007; Liu, Wang, and Zhao, 2007) and developed to be used for the internal usage of the organizations instead of being part of the end-product (Riehle, 2018; Baldwin and von Hippel, 2011). The labor is mostly paid developers or developers from the partner institutions of the consortium (Fitzgerald, 2006).

Riehle (2018) defines open source user consortium as *“a consortium of companies who sponsor, steer, and possibly also develop open source software for their own use rather than as part of software products they sell.”*

In the higher education industry, this sort of inter-organizational collaborative open source development approach is named “community source.” Community source is a hybrid model which combines the open source development approach with in-house IT development. The staffs who work on the initial code development phase are coming from the partner institutions, and they funded by the shared pool resources. The voluntary contributions are accepted at the next steps of the project after the core code reached a maturity. The resulting software is open to use, distribute, change, or sell, and get the benefit for any organizations (Wheeler, 2007). Liu, Wang, and Zhao (2007) define community source as *“a unique type of open source that depends on significant financial and other support from a community of institutions in the development and deployment of the enterprise application.”*

Fitzgerald (2006) introduces “OSS 2.0” term as the metamorphosed version of open source software development strategy. OSS 2.0 is led by organizations to develop information system applications in the vertical domain, and the labors of the project are the paid developers who work in the partner institutions of the consortia (Fitzgerald, 2006).

“Federated Innovation” is another term for this kind of collaborations. Levy and Germonprez (2015) name define federated innovation as *“companies’ collaboration in a federated space with similar motive in an open innovation manner.”*

The focus of this thesis is user consortium as defined by Riehle (2018). However, I will sometimes refer to projects in the literature which meet the definition of user consortium.

2.2.1.2 Examples

The first examples of open source user consortia have appeared in higher education. uPortal, Sakai, Kuali, and Open Source Portfolio Initiative (OSPI) are the pioneers of this approach of software development. The common point of these projects is that they are initiated by research universities, their initial developers were university staff, and the projects were funded by the Andrew W. Mellon Foundation. Kuali and Sakai have both been described in detail by people intimately involved in the projects. (e.g., Wheeler, 2004; Wheeler, 2007; Severance, 2011). Mainly, the Kuali project has been a subject of many studies since 2007 (e.g., Liu et al. 2007; Liu et al. 2008; Liu et al. 2010; Liu et al. 2013; Liu et al. 2014). These papers generalize different characteristics of community source projects based on the findings from the Kuali project.

Besides higher education, community source projects took place in the library sector. Evergreen and Koha can be considered as examples from the library sector. These projects were created by the staff of the libraries. (Breeding, 2008).

In the past few years, interest in open source user consortia strategy has increased among commercial firms. Even competitive companies have started to collaborate for open source projects. In some cases, they create their own foundations to support their open source user consortium, and in other cases, they became members to the already established umbrella foundations such as Linux Foundation and Eclipse Foundation. The GENIVI Alliance is an open source user consortium as an example from the automotive industry (Winkelmann, 2015) and openKonsequenz from the energy and infrastructure industry. (Riehle, 2014; Schwab 2018). Both of these projects are incorporated their own foundation for governance. The openMAMA (finance sector) and SPDX projects were managed initially by their own foundations, and later they had joined the Linux Foundation (Levy and Germonprez, 2015). Polarsys Working Group is an open source user consortium as an example from the aerospace industry, which has hosted by Eclipse Foundation (Winkelmann, 2015).

2.2.1.3 Characteristics

Liu and colleagues published several research papers about the community source projects, especially around Kuali project (e.g., Liu et al. 2007; 2008; 2010; 2012; 2013; 2014) These findings provide information about some of the characteristics of open source user consortia.

The characteristics of community source projects show some differences from open source projects. These differences can be seen in partnership, control, and relationship aspects. Community source projects are virtual organizations with the individual participants who are working at the partner universities and collaboratively working on the same goal. In these kinds of projects, the relationship between the participating institutions and developers are more formal. The partner institutions have control over the development activities, and they compete to access the development skills. (Liu, Hansen, and Tu, 2014). Considering the differences, Liu et al. (2014) extend Raymond's Bazaar and Cathedral phenomena by categorizing community source approach under a new category, called "shopping mall."

Institutions' likelihood to join community source projects were examined in different studies: institution's size and internal development capability influence on this decision. Larger institutions have more resource than smaller ones, and they have the potential to influence the development process of the projects. These kinds of institutions tend to join community source projects as developer partners more likely than small institutions. Small institutions prefer to join projects which have lower investment costs such as for software, which shows low flexibility (Liu, Zeng, and Zhao, 2008). "Community Source Network Formation" model, which is created by Liu et al. (2013) presents the effects of individual and

institutional factors in decisions about joining the community source projects. Actors in the community source projects are partner institutions, managers (decision makers), and developers. Partner institutions are situated at the institutional level; managers and developers are at the individual level (Liu, Hull, and Hung, 2013).

Institutional factors are norms and monitoring the governance, which leads the social control, actor similarity, external funding, and external environment. Actor similarity between individuals eases communication and increases the trust factor in the community. External funding is a motivational factor. The external environment is related to the resource-dependence approach. This approach is about having power on the resources which is in need of by others and having a low dependency on the factors which other actors control. Based on this approach, community source projects increase the power for its members. Organizations tend to join community source projects if they need to decrease resource dependency on external actors (Liu, Hull, and Hung, 2013). The further incentives for being a development partner in a community source have the power to influence the application features, accelerate the development process, enhance the deployment of software, minimize the total cost of ownership and “reduce the variance of the system value” (Liu, Zeng, and Zhao, 2008).

On the other hand, motives, learning, and trust are the factors which are related to the individual level. Learning introduces the opportunity to developers to increase their collaboration experience, become dominant players in the community, learn more about the system, and become an expert in the developed system. Developers who have strong learning capabilities tend to join community source projects since they can benefit from others by networking. There are different views on the trust factor in community source projects. Although it is not easy to build trust between the members of virtual organizations, mostly these kinds of projects are built within a specific industry for a permanent time. This situation provides an opportunity for the players to meet potential contacts and partners for future projects (Liu, Hull, and Hung, 2013).

Although open source user consortia projects mostly start by depending on in-house development efforts, sharing financial resources and outsourcing the development is becoming a more common aspect of this kind of projects. When the projects have a higher number of partners and developers from different institutions such as in Kuali project, the management faces some problems. Coordination, unbalanced contributions, competence problems of developers, turnover of developers, the sustainability of the project are some of these problems. By outsourcing software development, the organization could concentrate on more community issues instead of technical issues. It is expected that outsourcing the development would be another era in the software development transformation (Liu, Wu, Zhao, and Zhu, 2010).

Partner organizations in open source user consortia projects have common goals, but they also have different needs and expectations from the software they develop. Flexible software architecture is a solution to fulfill the various needs of organizations (Liu, Wang, and Zhao, 2007). For example, in Sakai 3.0 Project (Aperio OAE) having flexibility provides customization, content sharing between organizations, supporting different scripting languages, attracting various web developers from different background, allowing the development teams to work in groups in parallel and practice agile method, and easing internationalization. These benefits help to ease managing the requirements of different member organizations (Liu, Wang, and Zhao, 2012).

2.2.2 Success Factors

2.2.2.1 Success Factors: Open Source Software

Factors which lead to success in OSS projects have been studied in different aspects. In this section, these aspects are grouped into software properties, development process, relations among the participants, and external factors.

License type, modularity, and complexity of code are the factors related to the software. Stewart et al. (2005) propose that having a non-restrictive license has a positive effect on OSS projects' popularity. Popularity is an indicator of OSS success among users (Stewart et al., 2005). From the maturity perspective, if the projects have highly restrictive license type such as GPL, the probability of reaching an advanced phase is lower for them. (Comino et al., 2007). On the other hand, further research suggests that non-restrictive license type has an only positive influence for the popularity of initial versions of the OSS projects. Developers prefer to contribute projects which have a restrictive license because it provides a safer environment by means of protecting their efforts (Midha and Palvia, 2012).

Higher modularity and lower complexity of the code have positive relations with the OSS success in terms of attracting developers. Modularity allows developers to work in parallel in sub-parts of the software. In higher modularity structure, the probability that developers understand the problems easier and contribute more increases. Furthermore, developers show higher activity in the projects, which shows lower complexity (Midha and Palvia, 2012). The area of use for the software is a further factor related to its success. Applications for sophisticated users attract more developers since the developers can learn more from other participants and intellectually be satisfied. More sophisticated developers increase the possibility of driving the project to an advanced level and success (Comino et al., 2007).

The number of developers in a project is categorized both as a success measure and success factor. The size of the developer base only affects technical success in the initial versions of the developed software. This effect disappears when the community gets larger (Comino et al., 2007). A higher number of active developers means an increase in activity and creates a need for more effort for project management and coordination. This situation might lead to the goal conflict between the contributors and forking of the project (Midha and Palvia, 2012).

The network and coordination aspects of the open source projects are also essential to determine success factors. Relations among the project participants and external contacts influence project success. While higher internal cohesion, which means the knowledge sharing among the project participants, increases the success of a project, only moderate level of external cohesion (knowledge sharing among the external contacts) is beneficial for project success (Singh et al., 2011). Moreover, the small-world effect has a positive impact on project success (Singh, 2010).

Restricted access to the development team increases the coordination and safeguards exchanges in a project. Higher utilization of collective sanctions and the importance of reputation for the project members safeguard exchanges. The coordination of a project improves the performance of the software project and increases user and developer satisfaction. As a result, it has a positive effect on project success. On the other hand, safeguarding does not necessarily show a difference (Sagers, 2004).

Although the developers in the OSS projects work mostly voluntary and they are free to choose how to contribute, if the project administrators assign responsibilities to the correct participants with sufficient qualifications, the success possibility of projects increase (Midha and Palvia, 2012).

OSS project participants might join the project from different parts of the world with different backgrounds. Different diversity categories have different influences on project success. Heterogeneity in values, beliefs, and attitudes among the project participants, which is separation diversity creates barriers

to community engagement due to cultural differences and diversity in spoken language. However, separation diversity has a positive effect on the market success of the software. Variety diversity covers the differences in the functional background of the participants. These differences lead to greater information breath and higher creativity (Daniel et al., 2013). Having a moderate level of technological diversity among the participants has a positive effect on OSS success compared to a too low or too high level of diversity (Singh et al., 2011). A further diversity category is the disparity diversity, which is related to the power and resource differences between project members based on their contribution level. Both variety and disparity diversity have a positive influence on community engagement and market success (Daniel et al., 2013).

A high number of users and popularity have a positive influence on project success. Since the project popularity increases the likelihood of a software project to have support, proper documentation, and sustainable development, it increases the success possibility of the software (Midha and Palvia, 2012). Addition to support existence, effective bug-fixing performance has a positive relation with project success (Singh, 2010). Having language translations increases the project popularity and relatively, the success of the project (Midha and Palvia, 2012).

Hosted forge affects project characteristics and success. The study of Becher et al. (2008) shows that the same success factors lead to different characteristics and outcomes for the projects hosted on different forges, namely Debian and SourceForge.

OSS projects with organizational sponsors have more possibility to increase their popularity. The sponsor seems like a sign for technical support and sustainability of the software (Stewart et al., 2005).

Table 1 provides a summary about the reviewed literature in this section. Author names, year of these studies and the findings from these studies are presented in this table.

Table 1: Related Literature about OSS Projects Success Factors

Author	Year	Findings
Sagers G.	2004	<ul style="list-style-type: none">• Restricted access to the development team increases the coordination and safeguards exchanges in an OSS project.• Higher utilization of collective sanctions and the importance of reputation for the project members safeguard exchanges.• Coordination in an OSS project improves the performance, increases user and developer satisfaction, and has a positive effect on project success.• Safeguarding does not necessarily show a difference.
Stewart K. J., Ammeter A. P., Maruping L. M.	2005	<ul style="list-style-type: none">• Using a non-restrictive license has a positive effect on project popularity.• Having an organizational sponsor for the project has a positive effect on project popularity.• Project popularity has a positive influence on the OSS project vitality.
Comino S., Ma- nenti F. M., and Parisi M. L.	2007	<ul style="list-style-type: none">• Using restrictive license type lower the possibility to reach an advanced stage of development.• Applications for sophisticated users attract more developers who lead to the success of a project• Although a moderate number of developers increases the development progress; this effect disappears when the community gets larger.
Beecher, K., Boldyreff, C., Capiluppi, A., Rank, S.	2008	<ul style="list-style-type: none">• Same success factors may lead to different results in similar projects which are hosted on different forges (comparison between SourceForge and Debian)
Singh P.V.	2010	<ul style="list-style-type: none">• The small-world effect has a positive impact on project success.• Effective bug-fixing performance has a positive impact on project success.• Support existence has a positive impact on project success.
Singh P.V., Tan Y., Mookerjee V.	2011	<ul style="list-style-type: none">• Higher internal cohesion increases the success of a project.• A moderate level of external cohesion is beneficial for project success.• A moderate level of technology diversity among the project participants is better compared to a too high or too low level of diversity.

Author	Year	Findings
Midha, V., and Palvia, P.	2012	<ul style="list-style-type: none"> • Restrictive license type has a positive relationship with OSS success. • Size of the user base has a positive relationship with OSS success. • Language translation has a positive relationship with OSS success. • Responsibility assignment has a positive relationship with OSS success. • Lower complexity has a positive relationship with OSS success. • Higher modularity has a positive relationship with OSS success.
Daniel S., Agarwal R., Stewart K.J.	2013	<ul style="list-style-type: none"> • Disparity diversity has a positive impact on community engagement and market success. • Variety diversity has a positive impact on community engagement and market success. • Separation variety has a positive impact on market success. • Separation variety creates barriers to community engagement due to cultural differences and diversity in spoken language.

2.2.2.2 Success Factors: Collaborations

In the business literature, researchers concentrate on different types of collaborations and the factors which influence the success of these collaborations. Collaborations itself is an established field.

In order to provide a broad understanding about the success factors in collaborations, the studies which provide systematic literature review on this subject are selected. Each of these studies have a focus from different aspects. Topics examined here are about the success factors for collaborations and co-opetition in general, strategic partnerships, and collaborations in Information Technology. Table 2 presents the details about the reviewed literature in this section.

By comparing and mapping the findings of these six studies, 92 success factors are determined. Eighty-two of these factors are mentioned only in one or two studies, and ten of them in three or more studies. The mapping with whole success factors is presented in Appendix A. Following list represents the most mentioned factors in the considered literature:

- Clearly defined objectives agreed by all parties. (Mattessisch and Monsey, 1992; Bruce et al., 1995; Rai et al., 1996; Hoffmann and Schlosser, 2001; Rikkiev and Mäkinen, 2009)
- Mutual understanding and trust. (Mattessisch and Monsey, 1992; Bruce et al., 1995; Rai et al., 1996; Hoffmann and Schlosser, 2001; Rikkiev and Mäkinen, 2009)
- Having a unique purpose/ common goal / mutual benefit (Mattessisch and Monsey, 1992; Bruce et al., 1995; Rai et al., 1996; Chin et al., 2008)
- Having complementary expertise/strengths (Bruce et al., 1995; Hoffmann and Schlosser, 2001; Rikkiev and Mäkinen, 2009; Chin et al., 2008)
- Top management commitment/support (Bruce et al., 1995; Rai et al., 1996; Hoffmann and Schlosser, 2001; Rikkiev and Mäkinen, 2009)
- Equality in power (Bruce et al., 1995; Rai et al., 1996; Rikkiev and Mäkinen, 2009)
- Clearly defined responsibilities agreed by all parties / Agreement on operating procedures (Bruce et al., 1995; Rai et al., 1996; Hoffmann and Schlosser, 2001)
- Concrete, attainable goals/ aims/ objectives (Mattessisch and Monsey, 1992; Bruce et al., 1995; Hoffmann and Schlosser, 2001)
- Regular progress reviews (Bruce et al., 1995; Hoffmann and Schlosser, 2001; Chin et al., 2008)
- Having collaboration champions (Bruce et al., 1995; Rai et al., 1996; Rikkiev and Mäkinen, 2009)

These factors can be grouped into the Bruce et al. (1995) success categories which are partner selection (mutual understanding, having complementary expertise), establishing the ground rules (clearly defined objectives and responsibilities which are accepted by all parties, having concrete goals), ensuring equality (equality in power, having mutual benefits), people factors (top management commitment, having collaboration champions) and process factors (regular progress reviews).

On the other hand, Bruce et al. (1995), Chin et al. (2008), and Rikkiev and Mäkinen (2009) introduce in their studies as the highest ranked factors based on their research results. The combined list contains the following factors:

- Management leadership (Chin et al., 2008)
- Top management support (Rikkiev and Mäkinen, 2009)
- Trust (Chin et al., 2008; Bruce et al., 1995; Rikkiev and Mäkinen, 2009)
- Clearly defined objectives agreed by all parties (Bruce et al., 1995; Rikkiev and Mäkinen, 2009)
- Commitment at all levels (Bruce et al., 1995; Rikkiev and Mäkinen, 2009)
- Frequent communication/ consultation/ information sharing between partners (Bruce et al., 1995; Rikkiev and Mäkinen, 2009)

- Clearly defined responsibilities (Bruce et al., 1995)
- Openness (Bruce et al., 1995; Rikkiev and Mäkinen, 2009)
- Mutual benefit (Bruce et al., 1995)
- Partner's complementary skills and resources (Rikkiev and Mäkinen, 2009)
- The product itself (Rikkiev and Mäkinen, 2009)
- Compatible strategy between the partners (Rikkiev and Mäkinen, 2009)
- Flexibility on changing pre-defined goals (Rikkiev and Mäkinen, 2009)
- Technology integration process (Rikkiev and Mäkinen, 2009)

Table 2: Related Literature about Success Factors of Collaborations

Author	Year	Study	Details
Mattessisch P.W., Monsey B.R.	1992	Collaboration: What Makes It Work	<ul style="list-style-type: none">• Success factors for collaborations in general• Grouped into six categories• Categories: Environment, membership, process/structure, communication, purpose, and resources
Bruce M., Leverick F., Littler D., and Wilson D.	1995	Success Factors for Collaborative Product Development: A Study of Supplier of Information and Communication Technology	<ul style="list-style-type: none">• Success factors for collaborative product development in ICT• Grouped into six categories• Categories: Choice of partners, establishing the ground rules, process factors, ensuring equality, people factors, and environmental factors• Most important factors: Clearly defined objectives agreed by all parties, mutual benefit, commitment at all levels, frequent communication/consultation, clearly defined responsibilities, mutual trust, and openness
Rai A., Borah S., and Ramaprasad A.	1996	Critical Success Factors for Strategic Alliances in the Information Technology Industry: An Empirical Study	<ul style="list-style-type: none">• Success factors for strategic alliances in Information Technology.• Factors: Partner congruity, partner evaluation, organizational advocacy, governmental policies, organizational issues, cultural concerns, human resource management practices, and partner dominance.
Hoffmann W.H., Schlosser R.	2001	Success Factors of Strategic Alliances in Small and Medium-sized Enterprises - An Empirical Survey	<ul style="list-style-type: none">• A framework about the critical success factors for Small Medium Enterprise collaborations.• This framework base on the transaction-cost-theory, the resource-based and knowledge-based strategic theory, and inter-organization theory

Author	Year	Study	Details
Chin K - S., Chan B. L., and Lam P - K.	2008	Identifying and Prioritizing Critical Success Factors for Coopetition Strategy	<ul style="list-style-type: none"> • Success factors for coopetition strategy • Grouped into three categories • Categories: Management commitment, relation development, and communication management. • Most important factors: Management leadership and the development of trust
Rikkiev A., Mäkinen S.	2009	Success Factors for Technology Integration Convergence Collaborations: Empirical Assessment.	<ul style="list-style-type: none"> • Success factors for collaborations in the Information and Telecommunication Technology • Grouped into five categories • Categories: Company strategy, management, process, people and offering in terms of products and services • Most important factors: Trust, partner's complementary skills and resources, the product itself, open communication, the commitment of the partners, information sharing between partners, compatible strategy between the partners, top management support, clear objectives of collaboration, flexibility on changing pre-defined goals and technology integration process.

2.3 Research Question

Although open source user consortia strategy is not a new approach, the research about this phenomenon is limited. The success factors for collaborations and open source software development are separately studied in the literature. However, the success factors for companies to collaboratively develop and use open source software is not approached effectively.

Since collaboratively open source software development strategy is gaining importance for businesses in order to use this software for their internal processes, it raises questions about how these collaborations can become successful. In order to provide an understanding of this phenomenon, the factors which lead to problems or success in open source user consortia are examined with the following research questions:

RQ 1: What kinds of problems occur in the open source user consortia?

RQ 2: What are the success factors for open source user consortia?

2.4 Research Approach

This section provides an overview of the followed research method and sampling strategy.

2.4.1 Research Method

The method of this study is a qualitative exploratory multiple-case study (Yin, 2013). The methodology of the research is adopted from the theory building framework which Eisenhardt (1989) suggests. Samples are selected theoretically. Since this is a master's thesis and the time allowed to study on this research is limited, the sample selection is limited to three. For data triangulation, multiple data sources are used. The data collection and analysis are performed in parallel and overlapped. The data analysis is conducted firstly within cases. After then, the cross-case pattern is searched by comparing similarities and differences in inter-case results. Constructs emerge during the data analysis from multiple sources and collected in tabulated form iteratively.

2.4.2 Sampling

In this research theoretical sampling method is performed. First, the user consortia projects which are hosted by the Apereo Foundation, the Eclipse Foundation, and the Linux Foundation are listed. Besides these umbrella organizations, a number of independent projects are examined, as well.

The considered dimensions for the sample selection were hosted foundation (in terms of different governance structure), industry, maturity level, and governance practices. The samples are selected from the following industries: Higher Education, Automotive, and Information Technology (IT). As a result, the Sakai Project, Eclipse openMDM Working Group, and GraphQL Linux Collaborative Project are chosen as sample cases. The properties of the cases are listed in the Case Selection Matrix.

Table 3: Case Selection Matrix

Sample	Umbrella Foundation	Driver	Maturity	Industry
Sakai	Apereo Foundation	User-led - User-developed	Mature	Higher Education
openMDM	Eclipse Foundation	User-led – service provider developed	Growing	Automotive
GraphQL	Linux Foundation	Vendor-led	Young	IT

2.5 Used Data Sources

In this section contains the details of data collection and analysis process, and background information about the samples used in this thesis.

2.5.1 Data Collection

The data collection process consists of collecting primary materials and conducting semi-structured interviews.

Data collection process started in December 2018 and lasted until June 2019. The searched data for the sample cases were meeting notes, websites, community wiki pages, code repositories. Furthermore, the insights of the people leading the sample user consortia are collected in the form of interviews, keynote speeches, blog posts, books, and published papers.

Data search is conducted online. Firstly, the websites of each sample cases are examined. The websites were useful for gaining general understanding about the cases. From the websites, community interaction channels are determined. Sample cases have different transparency strategies. openMDM EWG presents almost all the meeting notes (steering committee, architecture committee and quality committee) publicly in Eclipse Wikipage. The e-mail archives are also available. Some of the missing meeting notes and annual meeting presentations are collected received from the e-mails. For the Sakai case, online search lead to the book about Sakai (e.g. Severance, 2011) and blog of Severance. Published papers and presentations about the Sakai are also gathered by online search using search engines. GraphQL is not a new project but the corporation is very new. Since there are not many published information about the GraphQL Foundation, conference videos of the co-creators of the project are searched on Youtube and the newest videos are examined to find a connection with Foundation. On the other hand, GraphQL working group meeting notes are collected from GitHub repository of the project.

Furthermore, for data triangulation and gaining more understanding about each case, semi-structured interviews are conducted with project participants. In total, five semi-structured interviews were conducted in May 2019. One interview was performed in person; the other four interviews were conducted online. All interviews were performed in English. The focus of the interviews was the structure of the collaborations, problems confronted during the collaboration processes and the factors which influence the success of the collaborations. Moreover, considered success measures are questioned. Interview protocols were prepared before each of the interviews. Interviews were recorded with authorization, transcribed, and sent back for review.

The materials used as data sources for each project are presented in the Table 4, interview matrix is presented in Table 5.

Table 4: Data Source Matrix

Data Sources	openMDM	Sakai	GraphQL
Interview	X	X	X
Meeting notes	X		X
Book		X	
Published papers		X	
Keynote speeches			X
Blog posts (blogs of people from the community)		X	X
Community wiki pages and Github repositories	X	X	X
Project website	X	X	X
Other websites & blog posts		X	X

Table 5: Interview Matrix

Interview Partner	Organization	Consortium	Date
Ralph Müller	Eclipse Foundation	openMDM	02.05.2019
Angelika Wittek	Eclipse Foundation	openMDM	24.05.2019
Wilma Hodges	Apereo Foundation	Sakai	17.05.2019
Joshua Wilson	Longsight	Sakai	20.05.2019
David Rudin	Joint Development Foundation	GraphQL	13.05.2019

2.5.2 Data Analysis

Qualitative data analysis on collected online data and interview transcripts were performed by using MaxQDA data analysis tool. In the coding phase, open, axial, and selective method was followed, as Straus and Corbin (1990) suggested. For each of the cases codes are grouped into user consortia characteristics, problems, success factors and success measures.

In the result section, the references of data sources are present in assigned codes. Each of the code is assigned by the sample category, data source category, and a number. For example, the assigned code for the “Sakai: Building an Open Source Community” book is SB1 (Sakai – Book – 1). The list of assigned codes and data sources are presented in the Appendix section in detail. A summary of the codes to provide an understand about the research results are presented in Table 6.

Table 6: Explanation of the assigned codes to the data sources

Code	Explanation	Code	Explanation	Code	Explanation
S	Sakai	M	openMDM	G	GraphQL
SB	Sakai Book	MA	Annual meeting presentation	GC	Charter
SI	Sakai Interview	MC	Charter	GP	Podcast
SP	Sakai Presentation	ME	Email	GV	Keynote & Interview video
SBP	Sakai Blog Post	MI	openMDM Interview	GBP	Blog Post
SPP	Published Paper	MW	Webpage	GWi	Wikipedia
SWi	Wikipedia	MWi	Wikipedia	GWG	Working Group Meeting Notes
		MAN	Annual meeting notes		
		MSC	Steering Committee Meeting Notes		

2.5.3 Sample Characteristics

Sakai, openMDM, and GraphQL communities have different characteristics and structures. This section aims to provide background information for each of these projects.

2.5.3.1 Sakai Brief History

Sakai is a Learning Management System (LMS) which is used in a number of universities in different countries.

Sakai project was informally initiated by the members of four universities (The University of Michigan, Indiana University, Massachusetts Institute of Technology, Stanford University) and two research projects (uPortal and Open Knowledge Initiative (OKI)) in the USA in June 2003. At that point in time, these universities were working on development of their own in-house LMSs separately. They decided to join their resources and develop a system collaboratively which would meet their needs [SB1, SPP1].

The formal starting date of the project was January 2004. From January 2004 to December 2015, the collaborative effort funded by the Mellon Foundation. Furthermore, the project received funding from the Hewlett Foundation. In February 2004, the project partners founded the Sakai Educational Partners Program (SEPP) with the aim of ensuring the sustainability of the project both in financial and functional aspect [SB1, SPP1].

In the first 2 years, Sakai community released first two versions of the LMS. At the end of 2005, the funding process of the Sakai project came to an end. After that period, Sakai Foundation is incorporated. The goal of the Foundation was handling the organizational complications related to the universities governance policies. In 2012, Sakai Foundation merged with Jasig Foundation and incorporated the Apereo Foundation. Since this merger, Sakai Community has been working under the umbrella of Apereo Foundation [SB1].

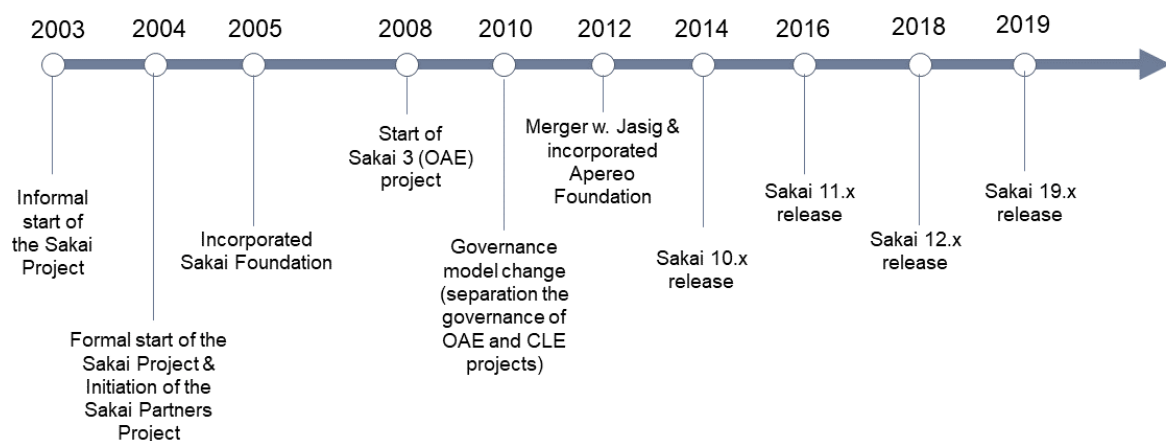


Figure 1: Milestones of the Sakai project (Source: Personal Collection based on Qualitative Data Analysis [SB1, SWi1])

2.5.3.2 OpenMDM EWG Brief History

The MDM project was initiated by Audi AG in 1999. The Technical Development Department of the Audi AG started the MDM project for measured data management systems. In 2008, Audi open-sourced the software to the use of other vehicle manufacturers and suppliers such as BMW, Daimler, Bosch, DAF, and FEV [MW1]. Throughout the time, demands of the users increased, and a need for an “Equal Partnership” model arose. This model indicated equality both in decision making and funding. In 2012, Audi AG and Eclipse Foundation started work on structuring the openMDM project. In 2014, the openMDM community officially joined to Eclipse Foundation and became an Eclipse Working Group [MI1].

The announced goal of the community is “*promoting the development and distribution of open source tools for measurement data management based on the ASAM ODS¹ standards*” [MW3].

The founding members were Audi, BMW, Daimler, HighQSoft, Gigatronik, Science+Computing, Canoo Engineering, Peak Solutions [MW3].

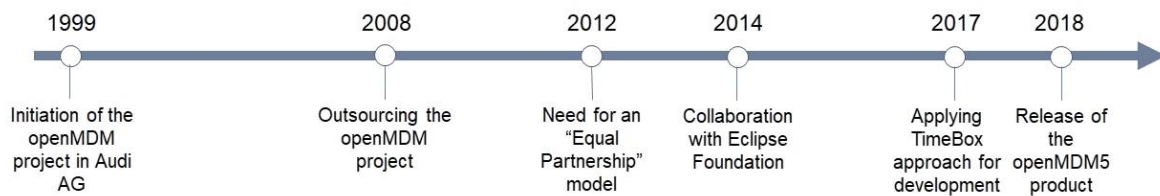


Figure 2: Milestones of the openMDM project (Source: Personal Collection based on Qualitative Data Analysis [MI1])

¹ ASAM (Association for Standardization of Automation and Measuring Systems) provides standards for data exchanges between research, development, and validation of vehicles and their components. ODS (Open Data Services) is an ASAM standard used in the automotive industry which allows creating application-specific data models (ASAM, 2019).

2.5.3.3 GraphQL Brief History

GraphQL is a data query language which was an internal Facebook project from 2012 to 2015. In 2015 the project specifications (spec) and a reference implementation of the project in JavaScript were open-sourced. Since 2015, besides Facebook, other organizations such as Github, Pinterest, Intuit, Coursera, Shopify are using GraphQL [GW1]. In September 2017, GraphQL spec was released under Open Web Foundation Agreement [GB1, GB2, GN1, GN2].

In November 2018 GraphQL Foundation is established. The Foundation is hosted by the Linux Foundation. The goal of the foundation is combining the resources to provide a vendor-neutral home to the community, increasing the involvement of developers and attracting more organizations [GW1]. Founding members of the GraphQL foundation are Facebook, AWS, IBM, Intuit, Neo4j, Salsify, Apollo, and Hasura [GY1]. The primary responsibility of the foundation is delegating the resources for the support of GraphQL community and GraphQL OSS projects in terms of supplying budget for infrastructure and service costs, events and training [GW1, GC1, GN3]. Since March 2019, The GraphQL Foundation has been collaborating with Joint Development Foundation [GW2].

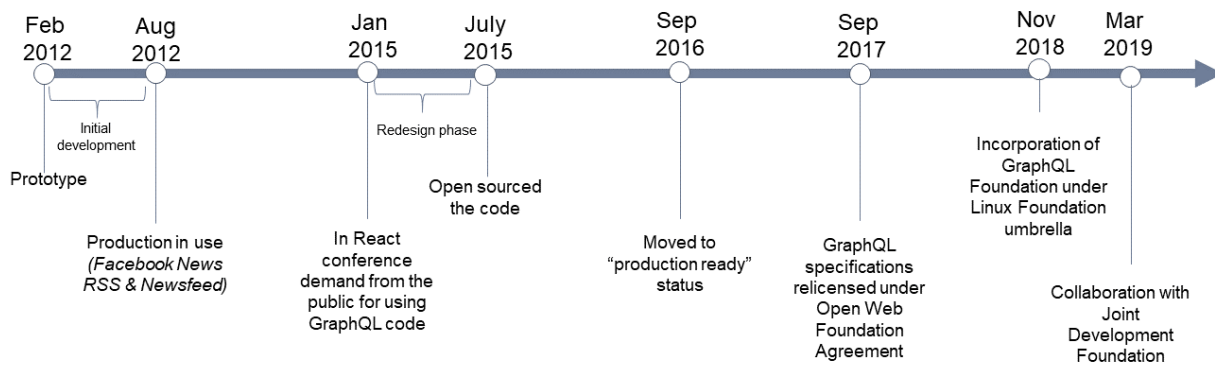


Figure 3: Milestones of the GraphQL project (Source: Personal Collection based on Qualitative Data Analysis [GV2, GV3])

Table 7: Summary of the Sample Backgrounds

	Sakai	openMDM	GraphQL
Umbrella Foundation	Aperio Foundation	Eclipse Foundation	Linux Foundation
Founding Members	The University of Michigan, Indiana University, Massachusetts Institute of Technology, Stanford University	Audi, BMW, Daimler, HighQSoft, Gigatronik, Science+Computing, Canoo Engineering, Peak Solutions	Facebook, AWS, IBM, Intuit, Neo4j, Salsify, Apollo and Hasura
Formal Incorporated Year	2004 (SEPP) 2005 (Sakai Foundation) 2012 (Aperio Foundation)	2014	2018
Creator of the seed code	Michigan University (CHEF project)	Audi	Facebook
Initial Sponsor of the project	Mellon Foundation, Hewlett Foundation, 4 Founder Universities	Audi	Facebook
Governance Structure	Do-ocracy	Hierarchical	Hierarchical
Initial Developers	Developers from the founder universities	Developers from the service provider members	Developers from Facebook
Current Developers	Commercial affiliates, member organizations, volunteer contributors, students	Service provider members, developers from driver member's third parties	Volunteer contributors, developers from Facebook, commercial affiliates
License	Educational Community License (ECL)	Eclipse Public License 2.0 (EPL 2-0)	MIT License (Publisher Massachusetts Institute of Technology)

2.6 Research Results

In section 2.6.1, results of RQ1 are presented. Section 2.6.2 contains RQ2 results. Research questions are as follow:

RQ1: What kinds of problems occur in the open source user consortia?

RQ2: What are the success factors for open source user consortia?

For the RQ1, results are grouped into four categories: Governance, Process Management, Product, and Environment. On the other hand, for the RQ2 results are grouped into five categories: Governance, Process Management, Product, Environment and People. Each of these categories are presented in details.

2.6.1 Problems

In the section 2.6.1.1 open source user consortia problems are presented as the results of cross-case analysis. In addition, the case results are presented separately in sections 2.6.1.2, 2.6.1.3, and 2.6.1.4. The section 2.6.1.2 shows the problems encountered in Sakai Project. 2.6.1.3 provides an overview about the problems of openMDM EWG. In the last sub-section, problems about 2.6.1.4, GraphQL project are presented.

2.6.1.1 Open Source User Consortia Problems

Sakai, openMDM EWG, and GraphQL are at different times of their lifecycles. Sakai is an active project since 2003. openMDM is an open source user consortium history since 2012. On the other hand, GraphQL is a newly incorporated project.

In different times, these projects faced various problems. Some problems led to further problems, and some others led to changes in the consortium management and structure.

This section presents the results of the cross-case analysis. Table 7 shows the analysis results with related data source categories.

2.6.1.1.1 Governance Problems

One of the main responsibilities of the governance team in the user consortia is planning the strategies. Creating and following wrong governance strategies cause problems in the health of the community. When a user consortium creates an ecosystem around itself, it is important to find a balance the needs of the community and member organizations. Otherwise, this situation leads problems.

Finance Management is a further responsibility of the government bodies of the collaborations. Open source user consortia are working under non-profit foundations. These collaborations depend on membership fees and conference incomes. In some cases, these incomes are not enough for development efforts. In other cases, although the financial resources seem to be enough at the beginning, not monitoring the financial situation and spending more than income lead to problems financial problems.

Losing members means losing both financial resources and developer contributions. In addition, it has a negative effect on the public image of the projects.

IP problems cause uncertainty among the developers and users of the projects. The wrong choice of license cause doubts in the community and leads loss of contributors. These situations lead to a decrease in the pace of development. In order to avoid these problems, it is better for user consortia to follow established IP management strategies and commonly accepted licenses instead of creating their own.

Although the main goal of the open source user consortia strategy to develop a system for their own use, increasing the members in the foundation is important. The members help the sustainability of the projects by providing financial resources or developers. When user consortia do not explain and promote their projects, they hardly gain new members. Attending the conferences, having a user-friendly web site, and sharing use cases are some of the ways to overcome this problem.

2.6.1.1.2 Process Management

Splitting the core-code development process into different parties which are not coordinated with each other makes the process management difficult. Furthermore, the lack of a project manager on top who would monitor the whole process cause inefficiency in the development process.

Turnover in the developers cause know-how and time lost in the development process. Depending on volunteer contributors in the core-code might affect the timeline of the development.

2.6.1.1.3 Industry

The barriers in the automobile industry lead to problems about promoting the projects. People in this industry do not prefer to talk publicly about the projects they involved. Not promoting the projects cause not increase in the member numbers

2.6.1.1.4 Product

Not having an English interface of the software is a barrier for increasing user and member numbers. Not having a working version of the product, after working for a long time in the project demotivates the project participants and cause loss of members.

Table 8: Problems in User Consortia

Category	Sub-Category	Problem	Data Source ²
Governance	Strategy	Driving the product roadmap, which is coming from the foundation, not from the community	SB-SBP
Governance	Strategy	Having short-term focuses	SB-SBP
Governance	Strategy	Having one organization in the center	GI1, GV, GBP
Governance	Finance Management	Lack of financial resources	MI1, MM, SI1, SI2
Governance	Finance Management	Not monitoring the financial situation	SB-SBP
Governance	Finance Management	Spending more than income	SB-SBP
Governance	Member Management	Losing members	MI1, MM, SI2, SB-SBP
Governance	Legal issues	IP problems	SB-SBP
Governance	Legal issues	License problems	GBP
Governance	Marketing	Not promoting / explaining the project	MI1, SI1, SI2
Process Management	Coordination	Splitting the development job	MI1
Process Management	Coordination	Working with different 3rd parties without coordination	MI1
Process Management	Coordination	Not having a central control mechanism / not monitoring the whole process	MI1
Process Management	Labor	Turn-over in the service provider personal	MM
Process Management	Labor	Relying on volunteer contributions for core developments	SI1
Process Management	Development	Integration problems	MI1, MI2, MM
Process Management	Development	Not having more bug-reports	MI1

² Explanation of Data Source Codes: MI: openMDM Interview; SI: Sakai Interview; GI: GraphQL Interview; MM: openMDM Meeting Minutes; SB – SBP: Sakai Book and BlogPosts by the same author; GV: GraphQL Keynote videos

Category	Sub-Category	Problem	Data Source³
Industry		Not talking about the project in public due to the industry dynamics	MI1
Product	Usability	Not having an English GUI*	MI1, MM
Product	Function	Not having a working version of the software	MI1

³ *Explanation of Data Source Codes:* MI: openMDM Interview; SI: Sakai Interview; GI: GraphQL Interview; MM: openMDM Meeting Minutes; SB – SBP: Sakai Book and BlogPosts by the same author; GV: GraphQL Keynote videos

2.6.1.2 Sakai Project Problems

Sakai Project has almost a sixteen-year history. Although the Sakai Community faced many problems in its lifecycle, the community has managed to survive since 2003. This section provides an overview of the problems in Sakai Project in different periods.

2.6.1.2.1 Governance

Wrong Governance Strategy: Following the grant-funded phase, the Sakai Foundation was incorporated. Governance authority of the Foundation was the Board of Directors. Until the second half of 2010, the governance team followed a top-down management approach and acted as if they were directing a commercial LMS product. They created a roadmap without considering the expectations of the community and tried to manage the voluntary developers as if they were paid staff of the project. In 2008, the Sakai Foundation started to work on a new project, Sakai 3.x⁴. Although the Sakai 3.x and Sakai 2.x were different efforts, the governance structure, the Product Council, was the same for both of them. The management decided to transfer the financial resources and community source to the Sakai 3.x project instead of continuing the development work on Sakai 2.x. This approach caused problems because voluntary contributors and partner institutions were used to working based on their priorities, not on the Board's demands [SB1, SBP19, SBP26, SBP21].

Later in 2010, a further council for Sakai 2.x project was established. This council was Technical Community Council, and the members were from the contributors of the community. This separate governance approach instead of a single governance structure ensured both projects to survive [SBP19, SBP11, SBP21].

Having Short-Term Focuses: The priorities of the founder organizations and Mellon Foundation, which funded the project, were different. The Mellon Foundation requirements had to be completed until the end of 2005, and the requirements of the founder organizations were expanding. This situation caused a time pressure for the project development team. In the beginning, software development was done for “short-term survival.” On the other hand, the marketing activities of the project was showing results even before a reliable product was formed [SB1].

While the number of member organizations was growing, the expectations of the software were increasing. On the other hand, there was not enough developer resource to meet the demands. The development prioritizations were decided based on the organizations which contribute to the project with their developers. These priorities had mostly “short-term” focus. The lack of resources and funding made it less possible to work for long-term benefits [SB1].

Not Monitoring the Financial Situation: After the grant-funded phase had been over, the Sakai Community started to face financial problems. Between the years 2008 and 2010, the Sakai Foundation had negative cash flow. The reason for that situation was not monitoring the financial resources and expenditures of the Foundation. Conference costs, labor costs, maintenance, and improvement costs of the software were the main budget items. Although the member numbers and income were decreasing, the Sakai Foundation expanded the spending on hiring new staff such as a product manager, marketing person, and user experience person. In order to handle the negative cash flow situation, the Foundation started to bill the membership fees up to three years, decrease yearly conference numbers from two to one, and increase the attendee fees for the conference. In the second half of 2010, executive director of the Foundation had changed and followed a new finance and government policy, which helped to overcome the financial crisis of the Sakai [SB1, SBP16, SBP19, SBP4, SBP25, SBP11].

Lack of Financial Resources: Sakai shows innovative features and evolves continuously considering

⁴ Name of Sakai 3.0 has been changed to Apereo Open Academic Environment (OAE) project.

the end-user requirements. However, the number of member institutions are not increasing. The reason for that seems to be not having enough marketing budget. Marketing and sales activities are not the priority of the community.

On the other hand, competitive products such as Canvas are showing aggressive sales activities and gaining new members [SI1, SI2].

Charles Severance explains this situation with the following words: *“Our impact on overall market innovation is *extremely high* through Sakai-led innovations like LTI and Common Cartridge. Our contribution and impact and end-user satisfaction unfortunately does not correlate to rapidly growing market share because after we meet end-user needs year after year with a best-of-breed 100% open source product, we don’t have any money to hire sales people to visit every university on the planet and buy free lunches for the IT staff”* [SBP1].

Losing Members: When the Sakai Educational Partners Program started in 2004, universities and institutions became members for three years. At the end of these three years, the Sakai Community started losing members. The reasons for that problem were various. One of the reasons for this situation was that some of the partner organizations had never adopted the software, although they joined the program. A further reason was the financial crisis and budget cuts in the years 2008 and 2009 [SB1].

Starting from 2012, the Founder Universities of Sakai began to replace Sakai Software with other Learning Management Systems from the market or their own in-house developed LMS. Although they were not doing the core system development for Sakai anymore, losing these members created bad publicity for the Sakai [SB1, SI2, SBP6, SBP8].

2.6.1.2.2 Process management

Relying on Volunteer Contribution: In the Sakai community, commercial affiliates and universities are working together for development. For some commercial affiliates, Sakai code development is in priority, such as Longsight. On the other hand, universities have other priorities. Although some developers from the universities are very active in the Sakai community, if some priorities occur at their university job, their local prioritize come earlier than Sakai. This situation leads to unpredictable timeline problems [SI1].

2.6.1.3 openMDM EWG Problems

From its initial phase (2014) to 2019 Eclipse openMDM Working Group has faced a number of problems and overcome most of them. The solutions to these problems increased the likelihood of project’s success. In this section, these problems and in most cases their solutions are presented.

2.6.1.3.1 Governance

Lack of Financial Resources: Since 2016, the consortium has had five driver members and one user member. The collaboration relies on the membership fees for sustainable development. In 2017 and 2018, there were not any new driver member who joined the collaboration. This means the income has not been increased [MA2, MA3].

On the other hand, at the Annual Meeting of 2018, MBBM and Siemens shared a projection about the required amount for a minimal working model as 500k. Considering the yearly income to be around 206k, projection showed that income is not enough for the expenses [MAN4]. The problems with financial resources slow down the development process.

Losing Members: Between 2015 and 2018, the working group experienced turn over in the service provider member. The missing resources are replaced with the new member organizations. However, it

led to some problems. These problems are both from a development perspective and a marketing perspective. Losing active members weaken the collaboration status from the sight of publicity.

Not Promoting the Project: openMDM collaboration does not explain itself to the public enough. The website of the working group is not user-friendly and informative to make people want to learn more about collaboration goals and dynamics [MI1]. There are not user stories which can attract organizations with similar needs.

2.6.1.3.2 Process Management

Splitting the Code Development Responsibility: In 2015, at the initial phases of the project, the software development responsibility was allocated to the driver members. These were Audi, BMW, and Daimler at that time [MA2]. The job was split, and each of the driver members coordinated their part of the development with different service provider members. Companies were paying the developers separately, and each of the driver members was monitoring only its part, not the whole process. This approach led to problems in terms of not having a complete product at the end of the defined process [MI1].

Not Having Collaboration Between the Third Parties: Although there was a dependency between the jobs of service providers, the service providers were not working in a collaborative approach. As a result, the failure of one of the service providers had an effect on the jobs of other providers [MI1].

Not Monitoring the Whole Process: At the beginning of the collaboration, no one was responsible for the coordination and monitoring of the whole process. Due to this approach, it was not possible to realize the problems earlier [MI1].

Personal Turnover in the Service Provider Members: The base code of the openMDM project is developed by the service provider members. The turnover in one of the service provider personal cause problems at the pace of the development process, which began in July 2017 and continued until August 2018. Due to the loose of experienced staff on the project, a loss of know-how and time occurred during the code integration process. [MSC1, MAC1].

Integration and Know-how Transfer: One of the problems faced during the development process was integrating the codes to the Eclipse environment, which were developed in other repositories. Using different frameworks in the externally developed code was a further challenge with similar outcomes. Integrating these codes took more time and effort of the development team than expected [MAC2, MAC3, MAC4].

2.6.1.3.3 Product

Changing the License: In 2018 the license model of the component was changed from EPL1 to EPL2 which created a requirement to check and change all of the documents and libraries for IP. This process increased the workload of the Toolkit Management Team [MAC5].

The license change might lead to more problems if there was not exist a contributor agreement among the contributors and consortium.

2.6.1.3.4 Industry

Not promoting the project: Industry dynamics has negative influence on the promoting the project and gaining new members. Since it is the automotive industry, it is not easy for the members to talk in public about the projects they involved [MI1].

2.6.1.4 GraphQL Problems

2.6.1.4.1 Governance

Having One Organization in the Center: GraphQL is a Facebook initiated project. Although it is an open source project and has a community around it, Facebook wanted to involve heavily [GP1, GWG6].

On the other hand, GraphQL is adopted by various organizations such as Pinterest, Github, and Shopify. The user companies have concerns that by a change in the GraphQL, they would be affected [GBP3, GI1, GP1]. The rights of the other users should be concerned. Incorporating a vendor-neutral foundation is expected to be a solution to this situation [GBI, GP1].

License Problems: In June 2015, Facebook open sourced the GraphQL code, and developers started to use the code [GBP3]. GraphQL had been licensed under BSD+Patents [GBP2, GBP3]. In June 2017, Apache Software Foundation announced that software with Facebook BSD+Patents license type would be banned from Apache projects because it was incompatible with the Apache License [GBP4]. This situation caused discussions in the community, and users reacted. In September 2017, GitLab announced that they were freezing using GraphQL code due to the license concerns. [GBP3].

In September 2017, GraphQL adopted the Open Web Foundation Agreement and changed GraphQL specification's license type to MIT license. However, not all of the concerns are dismissed. The patent grants related to MIT license were a question for the community.

2.6.2 Success Factors

Factors which influence the success of user consortia are various. In this research, these factors are grouped into five categories: Governance, Process Management, Product, Industry, and People. In the section 2.6.2.1 the results of cross-case analysis are presented. The sub-sections 2.6.2.2, 2.6.2.3, and 2.6.2.4 show results of the Sakai, openMDM EWG and GraphQL cases, respectively.

2.6.2.1 Open Source User Consortia Success Factors

This section presents the results of the cross-case analysis. The analysis results are listed in the Table 8 with related data sources.

2.6.2.1.1 Governance

Strategies followed in an open source user consortium is significantly related to success. Having a common goal is the basis of almost all collaborations.

Setting clearly defined rules and boundaries at the beginning of the collaboration helps to avoid conflicts in further phases. Making collective choice of arrangements and having collective responsibility are the further factors affects positively the health of the collaboration.

Equality, in terms of resource sharing and decision making in the community, strengthens the commitment to the collaboration. Having a sustainable product is important both for the members of the user consortia and potential users. In some cases, besides using the core system, members build in-house systems based on the core code of the consortium's software. The sustainability gains more importance in these cases. In order to provide the sustainability of the project, having a vendor-neutral environment and not having the dominance of one or two organizations is important.

Transparency, openness and trust are musts of an open source software project. For the open source user consortia, the situation is similar. Transparency is important to provide information flow between community members. Openness increases the interaction in the community, helps to build trust and paces the innovation.

Learning from the previous experiences of other projects and applying the best practices in every stages of development are further success factors for open source user consortia. Adopting established governance and legal infrastructure shortens the incorporation process and prevents potential problems which may occur after the foundation phase.

Building an ecosystem is important for the survival of open source projects. For the open source user consortia, it is also important to have commercial affiliates, volunteers, and other members besides the founder partners for the sustainability of the project.

From the marketing perspective, promoting the project has positive outcomes in terms of increasing adopters, members, and developers of the project. Building the community around the project has positive influences both on financial and human resources aspects.

2.6.2.1.2 Process Management

It is important to monitor the whole process of the development process. Splitting the development responsibility into different parties and not having coordination between these parties lead to problems. On the contrary, having a project manager on top who coordinates the different parties increase the likelihood of success. Having a continuous team of developers is a further success factor.

Following a roadmap and setting milestones have positive effects on successful product development. Having sanction mechanisms or motivating contributors with kind gestures are used ways to ensure quality of the code development.

Periodic communication and using linked tracking tools have a positive influence on collaborative software development.

Reviewing the code, using one repository and doing the changes on that repository, conducting pilot tests, having a quality assurance process are significant factors for developing successful and qualitative product.

2.6.2.1.3 Product

The usability of end-product is important to reach more users. Having user interfaces in different languages effects the adoption of the product in international context. A further factor is, ability of the product code to be used in different programming language. For example, publishing the specs of the software project with a reference implementation, increase its adoption among different developers and companies.

Offering opportunity to the community to customize the software based on their preferences affects positively the integration of software.

2.6.2.1.4 Industry

Industry dynamics influence the success possibility of the open source user consortia in different ways.

One factor is the prior collaboration experiences. When the project participants know each other from previous collaborations, building social relations and trust become easier. Collaboration opportunities change in different sectors.

The power of the member organizations inside their industry has positive influence on the project success.

The scope of the project in the industry has influence on the project success. For example, the Sakai LMS is adopted from different universities in the world. It reaches both students and developers in the universities, which is a huge community.

2.6.2.1.5 People

Working with motivated, passionate, and knowledgeable people increase the likelihood of the success of the collaboration.

Table 9: Success Factors in Open Source User Consortia

Category	Sub-category	Success Factor	Data Source⁵
Governance	Strategy	Clearly defined boundaries	MI1
Governance	Strategy	Clearly defined rules	MI1
Governance	Strategy	Making collective choice of arrangements / Collaborative Decisions and planning	MI1, SI2, SB-SBP, SPP
Governance	Strategy	Collective responsibility / Distributed responsibility	SB-SBP, GV
Governance	Strategy	Transparency	MI1, MM
Governance	Strategy	Openness	MI1, SI1, SB-SBP, GV
Governance	Strategy	Equality	MI1, MI2, SB-SBP, SPP
Governance	Strategy	Commitment of the members to the project	MI1, MM
Governance	Strategy	Building Ecosystem	MI2, SI1, SB-SBP, SPP, GV
Governance	Strategy	Adopting previously established governance infrastructure	MI2, SB-SBP, GV
Governance	Strategy	Adopting previously established legal infrastructure	MI2, SB-SBP, GI1
Governance	Strategy	Being innovative	SI2, SB-SBP
Governance	Strategy	Being responsive to the users' needs	SI2, SB-SBP, MI2
Governance	Strategy	Building Trust	MI2, MM, SI1
Governance	Strategy	Motivating the contributors	SB-SBP
Governance	Marketing	Promoting the project	MI1, SI1, SI2, SB-SBP, SPP
Governance	Marketing	Having a self-explaining, user friendly web site	MI1, SI2

⁵ *Explanation of Data Source Codes:* MI: openMDM Interview; SI: Sakai Interview; GI: GraphQL Interview; MM: openMDM Meeting Minutes; SB – SBP: Sakai Book and BlogPosts by the same author; GV: GraphQL Keynote videos

Category	Sub-category	Success Factor	Data Source⁶
Governance	Legal issues	Having GPL- friendly license	SI1, SB-SBP
Governance	Strategy	Learning from other projects / Benchmarking	SB-SBP, GV
Governance	Strategy	Having a sustainable product	SI2, SB-SBP
Process Management	Coordination	Monitoring the whole process	MI1, MI2, SI2
Process Management	Coordination	Having sanction mechanisms	MI1
Process Management	Coordination	Following timebox development approach	MI1, MI2
Process Management	Coordination	Having milestone releases	MI1, MI2, MM
Process Management	Coordination	Following product line approach	MI1
Process Management	Coordination	Proper documentation	MI2, MM, GBP
Process Management	Coordination	Periodic communication	MI2, MM, SI1, SB-SBP, GV
Process Management	Coordination	Quick response to the bug reports	MI2
Process Management	Coordination	Coordination by linked tracking tools	MI2, SI2
Process Management	Labor	Having a continues team of developers	MI1, MI2, MM
Process Management	Labor	Having a dedicated project manager	MI1, MM
Process Management	Development	Code Review	MI2
Process Management	Development	Using one repository making changes on it	MI1
Process Management	Development	Delivering testable and remarkable releases	MI1, MI2

⁶ *Explanation of Data Source Codes:* MI: openMDM Interview; SI: Sakai Interview; GI: GraphQL Interview; MM: openMDM Meeting Minutes; SB – SBP: Sakai Book and BlogPosts by the same author; GV: GraphQL Keynote videos

Category	Sub-category	Success Factor	Data Source⁷
Process Management	Development	Conducting tests and having quality assurance process	MI1, SB-SBP
Product		Multilingual User Interface	MI1
Product		High quality of code / Functionality	MI1, GV
Product		Being language-agnostic	GV
Product		Self-documenting	GV
Product		Customizability and Flexibility	SI1, SB-SBP, SP, MI2
Product		Having simplified framework	SB-SBP
Industry		Being a solution to a common problem in industry	GV, GBP
Industry		Internationalization	MI1, MI2, SI2, SB-SBP
Industry		Collaboration opportunities	MI1, SB-SBP, SPP
Industry		Having prior collaboration experiences	SB-SBP, SPP
Industry		Power of the driver members /founders	MI1
People		Having passionate people	SI2
People		Having knowledgeable and experienced people	SI2
People		Having creativity	SI2

⁷ *Explanation of Data Source Codes:* MI: openMDM Interview; SI: Sakai Interview; GI: GraphQL Interview; MM: openMDM Meeting Minutes; SB – SBP: Sakai Book and BlogPosts by the same author; GV: GraphQL Keynote videos

2.6.2.2 Sakai Project Success Factors

2.6.2.2.1 Governance

Having an Open Culture and DoOcracy: The community works in an open culture. It is open to everyone. Everyone can attend Sakai events, follow Sakai online communication mediums such as Google Groups, talk about their ideas, or needs openly. They are free to contribute based on their choices and their own priorities [SI1, SB1, SBP5].

Charles Severance explains this situation with the following words: *“If you come to Sakai Camp you will sit next to and talk the leading lights in the Sakai Community, from our QA leadership, support experts, pedagogy experts, UI/UX experts, end users, and core developers. There is no “hierarchy,” there is no “pecking order,” everyone is encouraged to participate, talk, and share your opinions” [SBP5].*

Wilma Hodges explains this situation with the following words: *“I think part of it is really the nature of the community. It is very much a doocracy. And by doocracy, I mean that you know, the pretty much anybody that has work that they want to contribute or, or ideas that they want to promote within the community, if they're willing to do the work, it gets done” [SI1].*

Open culture has been established on the governance level, as well. There is no top-down authority. Instead, there is trust between the members [SPP1, SB1, SBP6, SBP17].

Equal Rights of Member Organizations: The Sakai community has member organizations from research universities, community colleges, and commercial institutions. All of these members have equal rights in the community. All members can vote and can be nominated for board elections. This policy has an influence on attracting organizations to involve in the community [SPP1, SB1].

Bradley Wheeler explains this situation with the following words: *“Sakai's choice of an open BSD-style license, free access to the software, community discussions for anyone and equal rights for any member of the foundation—large or small, educational or commercial—have proven a magnet for rapidly developing both enterprise-scale software and a global community” [SPP1].*

Collaborative Decisions and Planning: Community members have time to time disagreements. The members tell their opinions, discuss the decisions, but in the end, they always find a mutual way to follow [SB1, SPP1, SBP5].

Bradley Wheeler explains this situation with the following words: *“The board rarely exercised its formal power regarding staff assignment, and most disagreements were ultimately resolved through debate. Over time, a meritocracy developed — like in any open source project — that accrued referent power and the ability to effectively influence” [SPP1].*

Members prioritize functionality, requests, and requirements. They have been creating and following roadmaps for future developments [SI2, SBP16, SBP10].

Collective Responsibility: The community members have a collective responsibility to improve the Sakai CLE. At the initial phase, mostly the founder universities worked on the core code development. After the code base reached maturity and inherited by the community, collective responsibilities shared with the community, which increased the speed of code improvements [SB1, SBP8, SBP10].

Charles Severance explains this situation with the following words: *“During 2011-2014 as the founding institutions slowly backed away, patches and bug fixes started to pile up. Now that the community has inherited the code-base and collective responsibility, the outstanding issues are rapidly being addressed” [SBP8].*

On the other hand, commercial affiliates in the community are working collaboratively, although they are competitors [SBP9].

Charles Severance explains this situation with the following words: *“But the essential difference in the Sakai community is that three competitors saw fit to pool their cloud tuning efforts and put their code into the community release. Even while the code was being built and tested, developers from AsahiNet, Longsight, and Unicon were communicating regularly, checking, testing, and fixing code written by one of their “competitors”. And when it was all done the code ends up in the open source trunk of Sakai. There are no secret repositories with the magic sauce”* [SBP9].

Building Social Relations and Ecosystem Around Sakai: From the initial phases of the project, the members gave great importance to build an ecosystem around Sakai. At the first year, in 2004, the first Sakai Conference was organized. Throughout the years, different events are organized, such as Sakai Developers Meeting, Sakai Members Meeting, Sakai Camps, Sakai Virtual Conferences. These events increase interaction between the community members; they provide networking opportunities and keeps people connected. Furthermore, people share their knowledge about Sakai implementations and exchange information [SB1, SI1, SPP1]

Wilma Hodges explains the situation with the following words: *“All of our communication channels definitely are a success factor because there is a lot of communication that happens. And then in particular those, those events throughout the year, I think are a really good way to allow people to network and give people kind of a sense of belonging to a larger group and to their they get to see how other people are using Sakai”* [SI1].

After the conferences and meetings, the Sakai community involve in team building activities such as drinking and dining together or singing karaoke. These activities are essential for building team cohesion, improving relations beyond the professional level [SB1].

Motivating the Contributors: For the Sakai community, it is important to show appreciation to the contributors. During the initial phase of the project, the CIO, Severance was visiting the universities which contribute the Sakai codes voluntarily, such as the University of Lleida in Spain. It was a good way both to build relationships and show appreciation. A further way to thank was Sakaiger toys for the Quality Assurance team [SB1].

Promoting the Project: In the grant-funded phase, the project was promoted by the Board Members by means of giving presentations in the conferences, visiting other universities, organizing events and attending industry-wide events such as Educause. In the first two years, the Sakai program gained almost 120 members which was mostly the result of promoting the software and taking attention of the potential partner universities [SB1, SPP1].

Charles Severance explains this situation with the following words: *“Even while the technical aspects of Sakai seemed to be falling apart, the marketing buzz and hype around Sakai as “the next big thing” was continuing to grow. We had gained a number of partner schools in the first half of 2004. Sometimes schools actually contacted us to join and support our effort out of the blue after reading a magazine article. Other times Brad Wheeler of Indiana University would drop a note to the CIO at a school and encourage them to join which they often did solely based on his recommendation. Still other times, I would go and make a personal visit to a university and walk away with a new membership”* [SB1].

In the current state, Sakai does not have enough Marketing budget, which is considered as a problem by the PMC members. However, Sakai has a new, user-friendly website⁸, which aims to inform both the current members and the decision makers of potential members [SI1, SI2].

⁸ www.sakailms.org

Joshua Wilson explains this situation with the following words: *“The new website needs to be aimed at decision-makers, institutions that already have adopted Sakai but also in institutions that haven't, so that as those as these institutions make their decisions about their LMS platform in the future that they have the information that they need to let them know that Sakai is vibrant and Sakai is moving forward and Sakai is doing great and innovative things”* [SI2].

Adopted Internationally: Sakai is a product which can be used at any university in the world. Adoption of Sakai internationally, provide insights about the perception of the outside world and contributions for the success of the Sakai project [SB1].

Charles Severance explains this situation with the following words: *“The University of Lleida had taken the Beta release of Sakai 1.0, installed it, translated it into Catalan, and used it for teaching and learning on their campus. Later they told me that the software was solid as a rock for them. Again, we see the significant difference of the perspective from the inside versus the outside of the project”* [SB1].

Sakai User Interface (UI) can be translated to more than one language since the Sakai 2.0 version (2004) thanks to the contributions of University of Lleida [SB1].

Having a GPL-Friendly License: The license type for the Sakai software was invented by the Sakai Community at the early in 2004. This license type was named as Educational Community License (ECL). In 2006, a potential adopter university, the Open University of Catalonia, informed that they would not adopt the software due to its license type. ECL was not assumed as a GPL-friendly license, which means it was not possible to fork the code. This situation led the community to take actions about the license structure [SB1]. In October 2006, during the Open Source License Summit, Sakai Project introduced the ECL 2.0 as a modified version of Apache 2.0 license with updated patent clauses. Open Source Initiative (OSI) recognized ECL 2.0 as being compatible with the GPL v3 license [SB1, SWi2].

Having a license which allows forking in the code has a positive effect on the adaption of the software [SI1, SB1].

Being responsive to the needs of users: Users of the Sakai is scholars and students in the universities. Their needs and expectations are investigated by the community members. The focus of Sakai improvements are these expectations. Not only large institutions but also smaller universities are also considered in this process [SI2, SBP1, SB1, SBP25].

Joshua Wilson explains this situation with the following words: *“Sakai evolves, because of what we know, of what our faculty and students need of an LMS going forward. So, members of the Sakai community, they will very actively investigate what their students on their campus need, what their faculty on their campus needs, and they will always be bringing those ideas into the discussions in the Sakai community. So, the responsiveness that Sakai has to those real faculty and student needs, as brought forth fairly directly. It is a real strength of Sakai. And I think that is something that helps us be successful as well”* [SI2].

Being Innovative: Sakai has been offering new features innovatively in the LMS market. Listening end-users and allowing contributions from the community increase the pace of innovation in the Sakai [SBP1, SBP8, SI2].

Providing Sustainability: Sakai is not a commercial product, and the community does not have profit-goals. Commercial vendors may decide to call off their products when their products do not fit the profit expectations. However, this approach would not be seen for Sakai. Sakai has a robust ecosystem with diverse universities and commercial affiliates which are working collaboratively [SBP17, SBP10, SB1, SBP16].

On the other hand, although it is a mature product, for the improvements and sustainability, supporting

the development financially by means of membership fees is essential [SI2].

2.6.2.2.2 Process Management

Benchmarking: From the initial phases of the project, the Sakai Community is following best practices in the industry. The Chief Architect of the project established contact with the creators of related successful projects, tried to learn from their experiences, invited them to conference as keynote speakers, built collaborative relations when possible. Experiences from successful projects and people show different possibilities and aspects for the improvements in the Sakai [SB1, SBP15, SBP16].

Conducting Tests and Having Quality Assurance Process: At the initial period, for Sakai 1.0 pilot experiments were conducted to see the potential of the program in terms of scalability. Conducting quality tests is a further need for the development process of Sakai. For the first release (Sakai 1.0), the quality tests were done by the developer team. Starting from version 1.5 in 2004, Sakai developers have worked with Quality Assurance team to improve the quality approach for the product. When the core code inherited to the community, the pace of code development improved, but the code quality was decreased. After that point quality cycles started to get longer. In 2012, during the Sakai 2.9 development, the Quality Assurance resources were not enough. A number of universities implemented the beta version, and it worked without a problem. Production testing proves the solidity of the software [SB1, SBP16].

In the current phase, the Sakai community has a Quality Assurance Working Group [SI1].

Monitoring the Development: Software development effort is divided into commercial affiliates, universities, and volunteer developers in the current Sakai community. The community has a release manager and community coordinator. Release manager monitors the developments and releases. On the other hand, community coordinator works as a kind of project manager and monitors the coordination between different parties [SI1, SI2].

Using Linked Communication Tools: Developers in the community are using Jira and GitHub coordinately. Using these two tools make them easy to collaborate for tracking issues such as bug fixing or completed features [SI2].

Periodic Communication: From the initial phases of the project, technical and managerial staff are communicating periodically. In the current situation, the core team meets weekly to discuss the issues [SB1, SI1].

2.6.2.2.3 Product

Customizability: Sakai architecture offers flexibility for custom development. Universities can develop tools based on their needs and use these tools integrated into Sakai CLE. The customizability of the software allows the community to fulfill their needs and increase innovation pace in the community [SPP1, SBP17, SWi1, SI1].

Having a Simplified (Less Proscriptive) Framework: Sakai 1.0 version had a proscriptive framework, which challenged the developers to work on. With the Sakai 2.0 version, the framework was simplified, and that allowed the developers to build Sakai tool with a broader variety of approaches [SB1].

Charles Severance explains this situation with the following words: *“In March of 2005, the Sakai 2.0 framework looked really good. It met all of our design goals of being simple and elegant. When we compared Sakai 2.0 to Sakai 1.0, Sakai 1.0 looked like a mess of spaghetti. It was no wonder that all of the developers who were forced to work with Sakai 1.0 had such a painful experience”* [SB1].

2.6.2.2.4 Industry

Having prior collaboration experiences: The founder universities had previous working experiences

together. In 2003, Indiana University, University of Michigan and Stanford University started a project called “Navigo.” This project was integrated into the MIT’s Open Knowledge Initiative (OKI) project and was open sourced, as well [SB1, SPP1, SP4].

The project initiators (core team from the founder universities) were knowing each other from the industry. Knowing each other in individual level and having collaboration experience in institutional level, make it easier and faster to work collaboratively in the Sakai project [SB1].

2.6.2.2.5 People

Sakai project is a university collaboration which provides advantages from the people perspective.

In the Sakai community, the users are the builders at the same time. The people who involved in the project have the motivation to build something, not only consume. They are not motivated by money or profit; they have passion for the project. They have a desire to help other people [SI1, SI2, SB1].

Various universities in different parts of the world interested in the Sakai project which makes it possible to work with a broad and talented people. The best and brightest employees from the universities have involved in the project. Having knowledgeable and experienced people is one of the most significant advantages of the project [SB1, SBP17, SBP5, SBP10, SI2].

2.6.2.3 openMDM EWG Success Factors

2.6.2.3.1 Governance

Equality: One of the most critical factors for success in the collaborations is having equality. The driver members have equal rights in the meaning of sharing resource and having an influence on the Working Group decisions [MI1].

The commitment of driver members: Daimler positions the openMDM as part of their Industry 4.0 vision [MA2] 2018 annual meeting notes show that Daimler and BMW are investing in in-house projects based on openMDM [MAN4, MI1]. On the other hand, Müller-BBM is working on a product based on OpenMDM5 and Müller-BBM’s cloud product [MI1]. These show that the project is important for the driver members.

Clearly defined goals and rules: The Charter document of the Working Group informs about the goals and rules of the community. It is essential to set the goals and rules at the beginning to avoid conflicts in the future [MI1].

Collective choice of arrangements: The Working Group members have a common goal, and they are working to reach their goal collaboratively. Although the driver members might have different priorities, for the health of the collaboration, they need to decide together about the Working Group’s priorities since they have limited resources. In order to work effectively and avoid conflicts, the Group follows well-defined decision process and has Steering Committee and Architecture Committee [MI1].

Following best practices in the open source governance: The consortium is working with an existing Open Source Foundation, Eclipse Foundation, instead of creating its own foundation. By this way, the consortium follows the open source governance best practices such as bylaws, IP management policy, and development tools [MI2].

Openness and transparency: Working Group is performing everything in an open manner. The meetings are open to anyone to attend, and the meeting notes are mostly publicly available. The Toolkit Management team is documenting the processes they encounter with problems and put it to their wiki pages. Issues on tracking tools and assignments are open and transparent to all members. Email archives are online reachable. Transparency is significant for effective communication and coordination among the working group members. Furthermore, it is vital for collaboration [MI1].

Ralph Müller explains this situation with the following words: *“As of today, if a guy from Daimler and Audi need, they cannot talk to each other. Because they might break antitrust law, or they are afraid of. When they come to the Eclipse Foundation, they can talk. They can plan together. Because it is open and public and transparent. That is why it is so important that all the information that we have in openMDM is public. It is a lifesaver almost”* [MI1].

2.6.2.3.2 Process Management

Having a project manager at the top and a continuous development team: In 2017, the Working Group changed its project management approach. Instead of splitting the jobs, the Working Group decided to create a shared pool of resources and work with a project manager on the top [MI1]. This approach has a positive influence on the project success due to the efforts such as monitoring the whole process, having a continuous team of developers, using one repository, and making changes on that repository [MI1, MI2].

Following a timebox development approach with milestones: The Toolkit Management Team is following the Timebox development approach, which includes the success factors of having milestone releases, monitoring and measuring the improvements periodically, delivering testable and remarkable releases, and having sanction mechanism [MI1, MI2].

Having a sanction mechanism for qualitative code development: The contract of the developers is prolonged after the end of each timebox period. This approach allows the Stakeholders to assess the performance of the developers, which increase efficiency [MI1].

Following Product Line approach: openMDM Working Group is following the product line approach. This approach allows the driver members to develop in-house solutions based on the core-code (MDM|BL code). By this way, while the members are using the developed code for their interest, Toolkit Management Team of the openMDM Working Group could continue to work on the common interest of the whole members [MI1].

Proper documentation: The Architecture Committee and Toolkit Manager are preparing documents for different groups. They are preparing guidelines, specifications, release notes, and processes. By this way, they aim to provide information about the technical aspects and also avoid the problems they faced previously [MI2].

Angelika Wittek, openMDM Toolkit Manager, explains the importance of documentation with the following words: *“When I started, there was not much technical documentation. So, I set up all the technical documentations, and I am still maintaining it. So, each milestone we do not only do the code updates or publish new code, but we also publish updates from our documentation. Because this is one of the very important things. Software is not only about code, but the software is also about code, and documentation and issue tracking”* [MI2].

2.6.2.3.3 Industry

Transparency: The automobile industry in Germany has a significant role. A number of associations and institutions are supporting this industry such as VDA (Verband Deutsche Automobile Industry), VDAE (Verband Deutsche Automobile Engineering), and Fraunhofer. It is possible to get funds from German Governance or the European Union for the projects related to the automobile industry. There are a number of meetings and organizations that take place every year about the automobile industry. These factors make it easy for company members to meet and get acquaintances. These factors make it easier for industry players to collaborate. On the other hand, antitrust law in Germany makes it difficult to share information between the driver members in the collaboration. Openness and transparency is a solution to benefit from the industry dynamics and ensure success [MI1].

Power of the driver members in the market: The driver members have the power to demand compatible products to their systems from their supplier. This situation makes it easier to set standards in the market. After the software reaches a maturity level, and the driver members make it a part of their core technology for measuring data, it is expected that the software dominates the market with its features [MI1].

Ralph Müller explains this situation with following words: *“This number will go up for a reason within the organizations like Daimler and BMW; they will now not buy anything that is not based on openMDM5. Any system”* [MI1].

2.6.2.3.4 Product

Developing high quality of code: Having a qualitative code base is important for the future of the project. Since the adaptation of the new project management approach, the quality of MBM|BL code is increasing. These have outcomes for driver members, such as using this code in their internal systems or developing products based on this code. A further outcome shows itself for the other market players. Some of the data management systems suppliers are considering replacing their own code with MBM interface [MI1].

Demand for the software in the market will increase the sustainability of the product.

Having a multilingual user interface: The openMDM product has not had a GUI in English until 2019. In April 2019, the development team published the English user interface. With this new interface, Working Group expects to increase the number of openMDM’s users [MI1].

2.6.2.4 GraphQL Success Factors

2.6.2.4.1 Governance

Adopting previously established governance infrastructure: GraphQL Foundation is incorporated under Linux Foundation. This act allows the community to adopt the established governance structure and apply open source best practices instead of trying to create everything from the beginning [GV2].

Adopting previously established legal infrastructure: GraphQL Foundation is collaborating with the Joint Development Foundation (JDF). JDF provides legal infrastructures for collaborative open source projects. JDF offers established documentation and policies, which makes the foundation process easier and faster. GraphQL is benefiting from these established forms. Using previously established legal infrastructure shortens the incorporation phase of the Foundation. By this way, the member organizations do not have to spend time for negotiations on setting the legal standards [GI1].

David Rudin from JDF explains it as follow: *“When you're dealing with 3 to 12 companies, getting agreements on even the most basic terms can be very laborious, very difficult. And especially when you have people coming from different legal systems and different time zones, it makes it a very long-term process, three months of the best-case scenario. It can take up to six to nine months. I've seen some takes 2 years to do. And so, by taking away the ability to negotiate specific details, and choosing from existing terms, it allows us to move much faster”* [GI1].

Having an open source strategy: After deciding about open sourcing, the GraphQL, the core developers thought about how to introduce GraphQL as an open source project and create a community around it [GP1]. They changed the product features according to community abilities. The creators learned from the previously applied methods such as REACT community, and they looked for how other open source programming languages are working [GP1].

Lee Byran is explaining this process with the following words: *“We should really think about what open sourcing GraphQL should look like. I did not think that sharing a PHP library was gonna be successful.*

In fact, I didn't wanna pick any one language to share it in because I was nervous that since GraphQL such a new idea, that we ever tightly whatever language we shipped it as, and everyone who was not using that language would kind of ignore it"[GP1].

Distributing the responsibility: In the community, the workload is distributed to members beyond co-creators. Responsibility for activities such as organizing conferences, coordinating meetups, and preparing newsletters are distributed to different members in the community [GP1].

Periodic Communication and Interaction: GraphQL Working Group is meeting regularly. The group mostly discuss the technical details about the GraphQL and time to time do planning for the events [GP1, GV3].

On the other hand, the community is organizing regular meetups and annual developer conferences in different parts of the world [GBP1].

2.6.2.4.2 Product

Usability and Efficiency: In a query, GraphQL brings exactly what is asked for and follows a single network round-trip. This functionality has a positive influence on the API performance [GV1, GBP7, GBP8]. It is easy to implement without changing the existing systems [GV1]. These properties make the GraphQL as an attractive alternative to REST and attract developers quickly [GBP3, GBP7, GBP8].

Being language-agnostic: GraphQL was initially developed as a PHP library. When the inventors of the GraphQL decided to open source the project, they wrote specifications and implemented a reference version in JavaScript. JavaScript is a popular language, and in this way, GraphQL could attract a considerable number of developers [GP1]. This approach provided an opportunity to the community for implementing the GraphQL in which language they want. By the end of its first year, GraphQL was implemented in 12 different languages by different community members [GV2, GP1, GBP7, GV1]. Python, Java, C#, Node.js are some of the languages which support the GraphQL [GBP9].

Self-documenting: In GraphQL, API documents are automatically generated [GBP7, GV1, GV3], since the GraphQL queries contain attribute names, data types, and descriptions [GBP8]. It provides the documents to be always up to date [GV3]. This function is helpful to the developers because they need documentation [GBP8].

Proper Documentation: GraphQL has a wide variety of documents. The community produces educational materials in different formats such as guidelines, tutorial documents, YouTube videos, case studies provided by the user organizations [GBP7]. These documents are essential for developers to learn how to use GraphQL. Furthermore, keynotes from the conferences are available online.

2.6.2.4.3 Industry

A solution to a common problem in the industry: GraphQL was created by Facebook as a solution for the deficiencies of RESTful API design. The problems were related to the functions of Facebook mobile application, and GraphQL appeared as a complementary product to REST [GP1].

When Facebook open-sourced GraphQL, other companies and developers started to use it to handle the same problems they were facing with REST and their mobile applications [GBP7].

The application area influenced the recognition of the GraphQL and increased its user numbers.

2.7 Results Discussion

This preliminary research provides early insight into the underexplored phenomenon of user consortia. This research is focused on two areas about user consortia: problems and success factors. In order to understand these factors, the multiple-case case study approach is followed by following the Eisenhardt's theory building framework (Eisenhardt, 1989). This study shows that industry dynamics lead organizations to build open source user consortia due to different reasons. However, confronted problems and the success factors are similar in most cases. For example, for the automotive industry, it is important to ensure transparency and openness for collaborative working. Due to this fact, the principles of open source user consortia strategy suit to organizations in this industry. On the other hand, the organizations in the higher education have different motives. Sharing knowledge with other individuals and organizations around the world and learning from them is the main motivation. Open source user consortia ("community source") strategy is started in this industry. In the IT industry, setting standards seem to be an important motive for building open source user consortia. Even the projects initiated by a single vendor, after being open sourced, the community requires to grow under vendor-neutral structure, which leads to open source user consortia strategy. By choosing samples from different industries with different characteristics, generalizability of the results is aimed. By using a wide variety of data sources and data triangulation, reaching internal validity is aimed. Since this study is a master's thesis, the research is performed by single investigator, and investigator triangulation would not be possible.

This study provides two contributions to the literature. Firstly, the problems which are encountered in three different open source user consortia are investigated and by the end of the examination, results of the cross-case analysis are explained. Secondly, the same approach is followed for the factors which lead to success in the same user consortia. This preliminary research provides insights about these two aspects.

2.8 Conclusion

Although open source user consortia strategy is not a new concept, studies about this phenomenon are limited. On the other hand, the attention on open source technologies and collaboration opportunities show an increase.

The goal of the research is to provide an overview about the factors which lead to problems or success in open source user consortia. Both problems and success factors are highly related to governance policies. The results show that wrong finance politics, such as not monitoring the financial situation and spending more than income; not having accurate number of member organizations for financial stability, not promoting the project, and legal issues such as IP management and license choice are the main problems from the governance perspective. The most important problems related to the process management seem to be splitting the development responsibility to different parties and not monitoring the development process as a whole. Using different frameworks and code repositories are further problems.

On the other hand, the success factors are grouped into five categories: governance, process management, product, industry, and people. Governance related success factors are equality of the members, providing sustainability, transparency, openness, setting boundaries at the beginning of the process, using established governance and legal standards, collective responsibility, commitment of the members, promoting the project and building ecosystem around it. Being innovative and responsive not only to the members but also user's need are other factors. From the process management perspective, monitoring the whole process and having roadmap and milestone releases, proper documentation, periodic communication, using linked tracking tools are the most important factors. Usability and customizability are the most important product related factors. Industry has influence on the collaboration opportunities. Working with experienced, passionate, talented people has positive influence on the project success.

2.9 Acknowledgement

First of all, I would like to thank Professor Dr. Dirk Riehle and Dr. Ann Barcomb for all the guidance and supervision they provided to me during this research.

Furthermore, I would like to thank Ralph Müller, Angelika Wittek, Wilma Hodges, Joshua Wilson, and David Rudin for accepting to interview with me and sharing their expertise.

Finally, I would like to thank Professor Dr. Joachim Henkel (TU Munich) for his pointers at the beginning of my research process.

3 ELABORATION CHAPTER

3.1.1 User Consortia Structure and Characteristics

3.1.1.1 Sakai

3.1.1.1.1 Governance Structure

At the grant-funded phase, Principal Investigator (PI) was responsible for the success or failure of the project, and the PI has the right to make decisions about the grant team. Principal Investigator was the chair of the Sakai Advisory Board [SB1]. The initial members of the Advisory Board were the lead representatives of the four founding universities and uPortal and OKI projects. In June 2004, two more members were selected to the Advisory board who were from the University of California, Berkeley and Foothill Community College. These universities contributed with their development resources to the project.

In addition to the Advisory Board, the project also had the Architecture Team, Tools Team, and Technical Team. These teams were working on planning the structure and user features of the project.

Technical Team and Tools Team were reporting to the Chief Architect of the Sakai Project, and the Chief Architect was reporting to the Advisory Board and PI [SB1].

After the first two years, the Sakai Foundation has been incorporated. Sakai Advisory Board was evolved to Sakai Board of Directors. Initial Board members were the same people with the Advisory Board [SB1]. After the expire of these members term, the next board members were elected by Sakai Partners for three yearly terms. The board had ten members, and they had the power to make decisions about hiring and firing staff, choosing the Executive Director, and deciding about expenditures of the project.

In 2009 Product Council was formed. In June 2010, the Sakai Technical Coordination Committee (TCC) was created. Only the long-term committed contributors to the Sakai Collaboration and Learning Environment (CLE) had the opportunity to be a member of the committee.

In 2012, the Sakai Foundation merged with Jasig Foundation and formed the Apereo Foundation. In 2019, Apereo Foundation is hosting ten more projects other than Sakai. Apereo Foundation has a board with 13 members and one executive director. On the other hand, the Sakai Project has a wide variety of Working Groups such as Quality Assurance, Marketing, Documentation, Core Group, and a Project Management Committee (PMC). PMC is responsible for the planning of the upcoming releases and monitoring the expenditures. PMC has 13 members who are from the active participants of Sakai Community Working Groups.

3.1.1.1.2 Membership Structure and Members

Research universities, community colleges, and commercial affiliates are the members of the Sakai Community. Commercial Affiliates are offering services to the universities which seek for help about hosting, implementation, adaptation or customization of the Sakai Software [SI1]. Universities do not need to be a member of the community to use the software, but in order to provide sustainability of the software, Sakai needs membership fees [SI2].

3.1.1.1.3 Financial Resources

At the beginning of the project, the four founder universities committed to the project with services worth \$4.4 million for two years. They made this announcement in the Educause meeting in November 2003. In January 2004, the project received funding from the Mellon Foundation. Seed funding was \$2.4 million for two years. In February 2004, the project received additional funding from the William and

Flora Hewlett Foundation as \$300,000. In March 2004, the Sakai Educational Partners Program (SEPP) was initiated with 19 partner universities additional to four founder universities. These partner universities funded the project with \$450,000 for three years. Membership fees to the SEPP were changing based on the universities number of students [SPP1, SB1].

After the end of the grant-funded phase, the Sakai Foundation was incorporated.

Under the umbrella of the Apereo Foundation, Sakai Community receive fees when the members of the Apereo Foundation choose the Sakai supporting subscription (individual software community subscription). In this case, a minor amount of these membership fees goes to the Apereo Foundation, and the most significant part goes to the Sakai Project. In addition to the institutional membership structure, Apereo Foundation offers a program for individual supporters of the project, which is called “Friends of Apereo” [SI1].

Table 10: Initial Resources of the Project

Resource	Amount	Detail	Start of commitment
Mellon Foundation	\$ 2.4 million*	Seed funding for two years	Jan 2004
William and Flora Hewlett Foundation	\$ 300,000	Seed funding for half of SEPP’s first-year start-up budget	Feb 2004
Sakai Educational Partners Program (SEPP)	\$450,000	Founded with 19 Partners Fee for three years.	March 2004
Founding Institutions (The University of Michigan, Indiana University, MIT, and Stanford University)	\$ 4.4 million in institutional staff (27 FTE) + 5+ developers per institution under the project leadership	For two years	Nov 2003

*Severance (2010) mentions this amount as \$ 2.3 million

3.1.1.1.4 Project Management and Labor

The Sakai Software was created on the seed code of CHEF software, which was the LMS of the University of Michigan. Initial releases of the Sakai Software were created by a closed group of developers. These developers were paid staff from the founder universities. In the next stages, the Sakai Project started to accept voluntary contributions from the Partner Universities. In the current state, the core development is being conducted by commercial affiliates who are the paid developers of the project, by students, and by voluntary contributors. Developers follow milestones internal to their projects. Milestones are defined project by project.

3.1.1.1.5 Documentation

Since the initial phases of the Sakai Project, the community has been working on documents parallel to software development. Specification document, style guide, technical documentation, and tutorial

videos were produced in this phase. With the release of each version, the documents were updated.

Based on the information on the confluence wiki page of the community, Sakai Documentation Working Group is categorizing the Sakai Documents based on the requirements on them. Requirements are listed under the groups for end-users, for user support, for system admins, and developers. Each of the groups has subcategories, which name the related communication channels to reach the information.

3.1.1.1.6 Communication

The Sakai Community uses a wide variety of communication tools. Information and news about the project are provided in the Apereo Foundation website (www.apereo.org), Sakai Project website (www.sakailms.org), and confluence wiki pages of the Sakai Project (www.confluence.sakaiproject.org). On the other hand, the YouTube Channel of the Apereo Foundation and Sakai Projects hosts presentations about the Sakai Project, which were recorded in various conferences and meetings. The community members are communicating with each other by using e-mail lists, google groups, and slack. The members are using Big Blue Button for organizing meetings and giving presentations.

The core team of the Sakai Project is meeting weekly to discuss the recent issues.

3.1.1.1.7 Events

Since the starting year of the Sakai project, the Sakai community has been involving in different events effectively. At the beginning of the project, the community members were attending higher education related meetings such as Educause and giving presentations about the Sakai Project. On the other hand, the SEPP, later the Sakai Foundation, was organizing Sakai Members Meeting, Sakai Developers Meeting, and Sakai Conferences to increase interaction among members. After the conferences, social events have been organized to build bonds between the members.

Under the umbrella of Apereo Foundation, the Sakai Community continue to attend Apereo Events and in addition, organize Sakai related events such as Sakai Camp, Sakai Virtual Conference, and Regional Conferences.

3.1.1.1.8 Marketing and Branding

At the initial phase of the project, members from the Advisory Board, such as Brad Wheeler gave presentations about the Sakai project in various education conferences. Chief Architect Charles Severance visited the universities around the world to build connection and explain the specifications of the software. Sakai Conferences and articles in the Chronical of the Higher Education was another factor which had an influence on creating awareness about the Project.

3.1.1.2 Eclipse openMDM Working Group (WG)

3.1.1.2.1 Governance Structure

The openMDM community has a hierarchical governance structure. The Steering Committee leads the Architecture Committee, Toolkit Management Team, and Quality Committee.

The Steering Committee is responsible for the governance of the strategic decisions of the Working Group. One of the main duties of the Steering Committee is finance management, which includes decisions about membership fees and budget allocation. Other duties are requirements management, finding agreement between the members, splitting the tasks, and allocating the resources. Gaining new members, planning and performing branding and marketing activities, the decision about communication channels and tools are the additional responsibilities of the committee. Each of the driver members has a seat in the Steering Committee, and each of them has three vote rights. On the other hand, service

provider members, application vendor members, strategic user members have at least one representative in the Steering Committee with one vote right. [MC1]

Architecture Committee is responsible for the governance of the openMDM projects, which includes functional, non-functional, and technical aspects. Committee members evaluate the technologies to be applied and establish technical guidelines. In addition, validation of new project proposals and establishing the openMDM architecture compliance service are other duties of this committee. Each driver member and project leaders have a seat in the Architecture Committee [MA1].

Quality Committee was responsible for the qualitative tasks. Defining the WG quality kit and maturity process, establishing the openMDM integration environment and quality assurance service were some of the responsibilities of the Quality Committee. Quality Committee hosted participants from each of the driver members, project leads, and at least one participant from other members besides driver members [MA1]. The Quality Committee was active between November 2016 and July 2017. In the 2017 General Assembly meeting of the openMDM, it was decided that Architecture Committee take over Quality Committee assignments.

3.1.1.2.2 Membership Structure and Members

The openMDM community has five types of membership status. These are driver member, service provider, application vendor, user member, and guest member.

At the establishment stage, the driver members of the community were Audi, BMW, and Daimler. The number of driver members has increased in the first two years. Müller-BBM and Siemens joined into the community as driver members in 2015 and 2016, respectively. On the other hand, the number of service providers decreased in years. While Gigatronik and Peak Solutions stayed in the community from the beginning until 2019, ATOS and Canoo left the collaboration in 2016 and 2017, respectively. Bertrandt, Itemis, Piterion, NorCom, and Bridging IT are the other service provider members who had involved in the collaboration for limited times from one year to three years. One and only application vendor member of the openMDM has been HighQSoft for six years. The community has one user member, TATA, and one guest member, ASAM, since the community's founding year of 2014.

3.1.1.2.3 Financial Resources

The openMDM EWG participation fees change based on the membership class of the members. According to the Charter of the group, the participation fee of the driver members is equivalent to 60 employee service days, and other members ten employee service days [MC2].

Annual meeting minutes shows that driver members are paying \$40,000 and user member, TATA, is paying \$6,700 membership fee annually. On the other hand, service provider members and application vendor member provide employee services in the form of result packages or service packages. Non-profit organizations are not paying any fees for participation [ME1].

The monetary resources are being used for regular expenses such as service costs, web site operation, toolkit management and costs of the openMDM office and in some cases one-time expenses such as purchasing the openMDM trademark [ME1].

3.1.1.2.4 Project Management and Labor

At the beginning of the collaboration, management of the code development tasks was split between 3 driver members; Daimler, BMW, and AUDI. The initial plan was to end the job split in July 2016 and start to work on developments together. In 2017, the community decided to work with a Toolkit Manager [MI1].

Since April 2017, the projects are managed by Toolkit Management Team. Developers and the Toolkit Manager are following “Timebox” development approach, which is a best practice for the Eclipse Foundation. The team is working in six-weekly timeboxes. At the end of each timebox, the project is reaching a milestone. Toolkit Manager presents the results of the sprints, status of the development team, and next steps to the Steering Committee. Furthermore, in the Toolkit Management presentations difficulties which are faced during the development process are listed with prevention mechanisms to avoid re-encountering the same problems. This approach assures the Steering Committee to monitor and measure the development process and issue the contracts of the developer for the next timebox period [MI1].

In addition to the developer team, committers are working on the projects with duties such as creating bugs reports, contributing code and documentation, reviewing code, and editing the Working Group’s project pages [MI1].

3.1.1.2.5 Communication

The openMDM community uses a wide variety of communication tools. Wiki page of the community and projects and web page are being used to share information to the public and establish transparency. Mailing lists are the tools to share information and proceed with some of the discussions. For issue tracking, both Jira and Bugzilla tools are being used. The community uses the Sonar tool for monitoring code development. Developer team works on Eclipse Git Repositories [MI1].

Steering Committee has a closed mailing list. Committee members had shared some of the meeting minutes publicly in the Working Group’s mailing list; however, not all of the notes are publicly available. Steering Committee discusses the membership fees, financial status, proposals, member requests, updates from the Architecture Committee and the Toolkit Management Team.

Committee members in the openMDM have regular meeting schedules. Architecture Committee has three-weekly conference calls. Members discuss the architecture specifications, technology decisions, job assignments, and project status. For transparency, the meeting notes are being shared regularly on the wiki page of the community and the Working Group’s mailing list.

Besides the regular meetings, all community members are gathering in the annual meeting, which takes place in summer.

3.1.1.2.6 Events

openMDM benefits from hackathons and developer workshops by creating an opportunity to exchange experience between developers and increase efficiency [MI2].

openMDM community considers EclipseCon Europe and EuroForum as essential events for increasing the interaction between current members, communicating and attracting potential members, and getting in touch with other automotive working groups. On the other hand, they are not involving in these activities effectively [MI1].

3.1.1.2.7 Documentation

Architecture Committee and Quality Committee are responsible for the documentation. Architecture specifications and glossary, guidelines for various processes are documented and published on different channels of the community such as the openMDM community’s wiki page, Architecture Committee’s wiki page, and MDM|BL projects Git repository.

Table 11: Document list

Audience	Document Names
----------	----------------

Guideline for developers	<ul style="list-style-type: none"> • Guideline for External Contributions • Introduction to Eclipse IP Processes • Getting Started Guide for MDM BL Project
Guideline for users	<ul style="list-style-type: none"> • Installation Guide for openMDM5 web application
Guideline for Governance and Internal Processes	<ul style="list-style-type: none"> • Eclipse Working Group Charter • Working Group Participation Agreement • Technology Proposal Guideline
General Documents	<ul style="list-style-type: none"> • Architecture specifications document • Glossary

3.1.1.2.8 Marketing and Branding

Decision and performance of marketing activities are established by the Steering Committee. The aim is to create awareness about the Working Group and projects. Due to the fact that the openMDM was a trademark of AUDI, one of the priorities of the Steering Committee was organizing the trademark transfer, which was lasted from 17 July 2014 to 16 June 2017 [MC1].

In order to increase awareness about the community and projects, providing an up-to-date website is another task of the steering community. The website contains a broad range of information from the history of the community to the governance structure and membership fees. News section provides information about the version releases, new members, and some of the events which the community members are attending. Each of the steering community members has rights to post news [MSC8]. On the other hand, user stories are missing on the website. Although, in the Steering Community Meetings, the importance of user stories is mentioned as replying the question: “why openMDM must provide its broad range of methods and solutions” [MSC5], no user stories are available at the openMDM website.

In the Annual Meeting of 2015, workshop organizations for introduction the workgroup to the potential members such as PSA, Ford, Continental, and Bosch were mentioned. Creating presentations about the community, presenting them in the EclipseCon and EuroForum events and sharing them via online channels such as SlideShare are the other decisions of the Steering Committee for outreach.

3.1.1.3 GraphQL

3.1.1.3.1 GraphQL Foundation Governance

GraphQL Foundation is called “Directed Fund” in the Charter document. The Linux Foundation has a guidance role for the foundation. The management of the Directed Fund will be executed by the Governing Board, which will be composed of representatives of the general members [GC1].

Directed Fund will lead the Outreach Committee and other working groups. On the other hand, the technical community will stay independent [GW1].

The meetings of the Governing Board will be conducted privately. The expected participants are Governing Board representatives, the Outreach Committee chair, guests, and the Linux Foundation staff [GC1].

3.1.1.3.2 GraphQL Foundation Membership Structure

The GraphQL Foundation has two membership classes, which are General Members and Associate Members. In order to be a member of the Foundation, the candidate organizations should be a corporate member of the Linux Foundation [GC1].

General Members are divided into two sub-categories, which have different rights in the governing board. One category is “the appointing member,” and the other is the “elective member.” The first ten organizations which are joined to the Foundation are categorized as Appointing Members, and each of these members has the right to appoint a representative in the Governing Board. On the other hand, the rest of the General Members will be included to “Electing Members” category and will be represented by up to five participants who will be elected by the whole electing members as a group.

Associate members are non-profit organizations, open source projects, and government entities. They can attend to governing board meetings, but they do not have rights to vote for the decisions.

Table 12: Membership Structure of GraphQL

Membership Group	Sub-group	Determination	Rights in Governing Board	Rights in Outreach Committee
General Member	Appointing Members	First 10 General Members to join the GraphQL Foundation	Each of the members has the right to appoint a representative in the Governing Board and any Committee	Each of the members has the right to appoint one voting and one non-voting representative in the Outreach Committee
General Member	Electing Members	General Members which are to join the GraphQL Foundation after the first 10.	As a group, annually elects up to five representatives to the Governing Board	Each of the members has the right to appoint one voting and one non-voting representative in the Outreach Committee
Associate Member		Limited to Associate Members of The Linux Foundation & Needs approval from the Governing Board	Can involve to the Directed Fund	

3.1.1.3.3 GraphQL Foundation of Financial Resources

General Members have to pay annual fees based on their consolidated employee headcounts. The range of the annual General Membership fees for 2019 are listed below [GC1, GC2].

The associate members need to be approved by the Government Board to involve into the GraphQL Foundation. They do not pay an annual fee [GC1, GC2].

Table 13: Membership Fees of GraphQL

Consolidated Employees	Annual General Membership Fees
5,000 and above	\$20,000
Between 2,000 and 4,999	\$10,000
Between 50 and 1,999	\$5,000
Up to 49	\$2,000

3.1.1.3.4 Project Management and Labor

The project management is led by GraphQL Working Group. GraphQL Working Group is a meeting of contributors to the GraphQL community and maintainers of the GraphQL OSS projects [GWG0].

In the working group, the participants are discussing the proposals, problems, or improvement requirements about the GraphQL specs. If the participants do not reach a consensus about the discussed topics, they define action items such as benchmarking about how other organizations handle similar technical problems.

The first meeting was held in July 2017, and since then eight more meetings have taken place in-person or virtually. Meetings are open to anyone who accepts the Participation Agreement, signs the Specification Membership Agreement, and wants to contribute the agenda topics [GWG1]. Starting from May 2019, the community plans to conduct regular meetings every six weeks [GWG8].

3.1.1.3.5 Documentation

The legal documents such as GraphQL Charter, Code of Conduct, Participation Agreement and Funding Charter, Specification Membership Agreement, and License are listed on the Github's GraphQL Foundation page [GC3].

GraphQL community is conducting Working Group meetings and publishing the agenda and meeting notes on the Github page of the GraphQL Working Group [GG1].

Official GraphQL web page provides various documents about the query language specifications and how to use it. Documents are in different forms and extend. Besides community interaction resources such as Stack Overflow question page, facebook group, twitter, IRC channel, slack communities, there are also links to different blogs, books, videos, and other related web pages [GW4]. Releases are listed on the spec page of the website [GW3].

3.1.1.3.6 Communication (Community)

Each of the MeetUp events has its own community page, which includes event information, member list, photos, discussions. Furthermore, they have mailing lists and message boards. A YouTube channel for the meetup videos is also available.

The summits and conferences have their own websites which provide detailed information about the events, speakers, and activities.

GraphQL Working Group is conducting virtual or in-person meetings and publishing meeting notes on the Github Page of the GraphQL Working Group.

3.1.1.3.7 Events (Community)

GraphQL organizes regular (bimonthly) meetups in different cities of North America, South America, Europe, Australia, and Asia. These meetings are organized to share the latest developments around the GraphQL environment with the community and create interaction within the community [GW5].

Since 2016, every year, a GraphQL Summit is taking place in San Francisco. In this meeting, best practices, new technologies, and advanced patterns related to GraphQL are being presented [GW6].

On the other hand, different GraphQL events are happening in Europe. GraphQL-Europe, which was organized in 2017 and 2018 in Berlin, has a new name as GraphQL Conf Berlin. A further event is GraphQL Finland workshops, which started in 2018 [GW7].

3.1.2 Success Measures

3.1.2.1 Sakai Project

For the Sakai Project there are not officially defined success measures. At the grant-funded phase, the software should fulfill some expectations to prove its success [SB1].

Success measure becomes subjective when no measures are defined. In the following list, the success measures for different actors in Sakai community are listed.

This list is gathered by the analysis of book and blog posts of Charles Severance (e.g. SB1, SBP16), Kenneth Green's article about Sakai (SPP2), and interview scripts of Wilma Hodges (SI1) and Joshua Wilson (SI2).

- Pace of the adoption [SB1]
- Pace of the development [SB1]
- Market share [SBP16, SB1]
- Changing the market structure [SPP2]
- Gaining respect, being a choice for the universities [SB1]
- Being a solid market product [SB1]
- Having a solid, sustainable funding model [SB1]
- Structure of the conference (increase in the number of attendees, having a more user and adopter conference atmosphere than a developer meeting, having more tracks for end-users) [SB1]

Currently, Sakai Community is reviewing yearly their annual report in the Apereo Foundation Meeting. Report results are assessed in terms of meeting the yearly goals. Meeting the goals is a success measure for the Sakai Community [SI1]. A further measure is that the people's satisfaction with the project and product [SI2].

3.1.2.2 Eclipse openMDM Working Group

For the openMDM project, there are not officially set success measures. However, there are different success indicators for different parties in the collaboration. The presented indicators here are listed based on the interview with Ralph Müller.

The main success indicator for the driver members is meeting their needs. If the software meets their expectations and eases their processes, this means that the project is successful for them. A further measure is establishing industry standards [MI1].

For the service members, the goal is creating business for their self. If the developed software were a medium for them to create value-added services, this would be a success [MI1].

On the other hand, for the Eclipse Foundations, it will be a success if the project influence to extend Eclipse Foundation's technology portfolio, member base, and its user base [MI1].

Appendix A: Mapping Success Factors of Collaborations

Appendix B: Data Sources

Appendix C: More on Methodology

The method of the research is deviated from framework which was presented by Eisenhardt (1989). Suggested steps and deviation on these steps are listed in the Table 14.

Research Step	Activity	Deviation
Getting Started	Definition of research question, Possibly a priori constructs	
Selecting Cases	Neither theory nor hypotheses, Specified population, Theoretical, not random, sampling	Theoretical sampling with 3 cases.
Crafting Instruments and Protocols	Multiple data collection methods, Qualitative and quantitative data combined, Multiple investigators	Data triangulation by collecting publicly available data and conducting interviews. Single investigator
Entering the Field	Overlap data collection and analysis, including field notes, Flexible and opportunistic data collection methods	Iterative, overlap data collection and analysis. No field notes.
Analyzing Data	Within-case analysis, Cross-case pattern search using divergent techniques	Multi-case study and cross-case pattern search
Shaping Hypotheses	Iterative tabulation of evidence for each construct, Replication, not sampling, logic across cases, Search evidence for “why” behind relationships	Iterative tabulation of evidence
Enfolding Literature	Comparison with conflicting literature, Comparison with similar literature	No comparison with conflicting literature
Reaching Closure	Theoretical saturation when possible	

References

- ASAM. (2019). Technology Retrieved from <https://www.asam.net/about-asam/technology/>
- Bailetti T. (2009). Open Source Maturity Curve and Ecosystems. Retrieved from <https://www.slideshare.net/guest239f177/os-maturity-curve-and-ecosystem>
- Baldwin, C. & von Hippel, E. (2011). Modeling a Paradigm Shift: From Producer Innovation to User and Open Collaborative Innovation. *Organization Science*, Vol. 22, No. 6 (November-December 2011), pp. 1399-1417 <https://www.jstor.org/stable/41303133>
- Beecher, K., Boldyreff, C., Capiluppi, A., Rank, S. (2008). Evolutionary Success of Open Source Software: An Investigation into Exogenous Drivers. *Electronic Communications of the EASST*, Volume 8 <http://tfs.cs.tu-berlin.de/gramot/Gramot2006/Papers/Wil.pdf>
- Breeding, M. (2008). Major Open Source ILS Products. *Library Technology Reports*. www.tech-source.ala.org November/December 2008
- Bruce M., Leverick F., Littler D., and Wilson D. (1995). Success Factors for Collaborative Product Development: A Study of Supplier of Information and Communication Technology. *RandD Management* 25, 1, 1995. Pp.33-44
- Chin K-S., Chan B. L., and Lam P-K. (2008). Identifying and Prioritizing Critical Success Factors for Coopetition Strategy, *Industrial Management and Data Systems*, Vol. 108 Issue: 4, pp.437-454, <https://doi.org/10.1108/02635570810868326>
- Comino S., Manenti F. M., and Parisi M. L. (2007). From Planning to Mature: On the Success of Open Source Projects. *Research Policy*. 36(10), pp. 1575-1586.
- Daniel S., Agarwal R., Stewart K.J. (2013). The Effects of Diversity in Global, Distributed Collectives: A Study of Open Source Project Success. *Information Systems Research* 24(2):312-333. <https://doi.org/10.1287/isre.1120.0435>
- Eisenhardt, K. M. (1989). Building Theories from Case Study Research. *The Academy of Management Review*, 14(4), 532. <https://doi.org/10.2307/258557>
- Fitzgerald, B. (2006). The transformation of open source software. *MIS Quarterly*, 30, 3 (September 2006), 587-598 <https://www.jstor.org/stable/25148740>
- Hecht L., Clark L. (2018). The New Stack: Survey: Open Source Programs Area Best Practice Among Large Companies. Retrieved March 8, 2019, from <https://thenewstack.io/survey-open-source-programs-are-a-best-practice-among-large-companies>
- von Hippel E., von Krogh G., (2003). Open Source Software and the “Private-Collective” Innovation Model: Issues for Organization Science. *Organization Science* 14(2):209-223. <https://doi.org/10.1287/orsc.14.2.209.14992>
- Hoffmann W.H., Schlosser R. (2001). Success Factors of Strategic Alliances in Small and Medium-sized Enterprises - An Empirical Survey. *Long Range Planning*, vol. 34, pp.357–381.
- Levy, M., & Germonprez, M. (2015). “Is it Egalitarianism or Enterprise Strategy? Exploring a New Method of Innovation in Open Source. 21st Americas Conference on Information Systems, Puerto Rico, 2015. <http://aisel.aisnet.org/amcis2015/StrategicUse/GeneralPresentations/18/>
- Liu, M., Hansen, S., & Tu, Q. (2014). The community source approach to software development and the Kuali experience. *Communications of the ACM*, 57(5), 88–96. <https://doi.org/10.1145/2593687>

- Liu, M., Hull, C. E., & Hung, Y.-T. C. (2013). Antecedents of Community Source Network Formation: The Case of Kuali. In R. H. Sprague (Ed.), 2013 46th Hawaii International Conference on System Sciences (HICSS 2013): Wailea, [Maui], Hawaii, USA, 7 - 10 January 2013; [proceedings] (pp. 4267–4276). Piscataway, NJ: IEEE. <https://doi.org/10.1109/HICSS.2013.95>
- Liu, M., Wang, H., & Zhao, L. (2007). Achieving flexibility via service-centric community source: The case of Kuali. *AMCIS 2007 Proceedings*, 103.
- Liu, M., Wang, H. J., & Zhao, J. L. (2012). Technology flexibility as enabler of robust application development in community source: The case of Kuali and Sakai. *Journal of Systems and Software*, 85(12), 2921–2928. <https://doi.org/10.1016/j.jss.2012.06.026>
- Liu, M., Zeng, D. D., & Zhao, J. L. (2008). A Cooperative Analysis Framework for Investment Decisions in Community Source Partnerships. *AMCIS 2008 Proceedings*, 278.
- Liu, M., Wu, X., Zhao, J. L., & Zhu, L. (2010). Outsourcing of Community Source: Identifying Motivations and Benefits. *Journal of Global Information Management*, 18(4), 36-52, October-December 2010.
- Mattessisch P.W., Monsey B.R. (1992). Collaboration: What Makes It Work. A Review of Research Literature on Factors Influencing Successful Collaboration. Amherst H. Wilder Foundation.
- Midha, V. and Palvia, P. (2012). Factors Affecting the Success of Open Source Software. *Journal of Systems and Software*. Volume 85, Issue 4, April 2012, pp. 895–905. <http://www.sciencedirect.com/science/article/pii/S016412121100286X>
- Open Source Initiative. (2018). *History of the OSI*. Retrieved from <https://opensource.org/history>
- Open Source Initiative. (n.d.). About. Retrieved from <https://opensource.org/about>
- Radtke N. P., Janssen M. A., and Collofello J. S. (2009). What Makes Free/Libre Open source software (FLOSS) Projects Successful? *International Journal of Open Source Software and Processes*, Volume 1, Issue 2
- Rai A., Borah S., and Ramaprasad A. (1996). Critical Success Factors for Strategic Alliances in the Information Technology Industry: An Empirical Study. *Decision Sciences*, Volume 27, Number 1, pp. 141-155.
- Riehle, D. (2014). Open Source User Foundations. Retrieved from: <https://de.slideshare.net/driehle/open-source-user-foundations>
- Riehle, D. (2018). How to Capture Open Source User Consortia. Retrieved from: <https://dirkriehle.com/2018/08/12/how-to-capture-open-source-user-consortia/>
- Riehle, D. (2019). The Innovations of Open Source. *Computer*. 52. 59-63. 10.1109/MC.2019.2898163.
- Rikkiev A., Mäkinen S. (2009). Success Factors for Technology Integration Convergence Collaborations: Empirical Assessment. *PICMET 2009 Proceedings*, August 2-6, Portland, Oregon USA. pp. 374 – 382.
- Sagers G. (2004). The Influence of Network Governance Factors on Success in Open Source Software Development Projects. *ICIS 2004 Proceedings*. 34. <http://aisel.aisnet.org/icis2004/34>
- Schwab, B. (2018). The Economics of Open Source User Foundations. Master's Thesis. Friedrich-Alexander University Erlangen-Nürnberg
- Severance, C. (2011). Sakai: Building an Open Source Community. https://www.dr-chuck.com/sakai-book/Sakai_%20Building%20an%20Open%20Source%20Community%20-%20Charles%20R.%20Severance.pdf

- Singh P.V. (2010). The Small-World Effect: The Influence of Macro-Level Properties of Developer Collaboration Networks on Open-Source Project Success. *ACM Transactions on Software Engineering and Methodology*, Vol. 20, No. 2, Article 6, Pub. date: August 2010.
- Singh P.V., Tan Y., Mookerjee V. (2011). Network Effects: The Influence of Structural Capital on Open Source Project Success. *MIS Quarterly*, Vol. 35, No. 4 (December 2011), pp. 813-829. <https://www.jstor.org/stable/41409962>
- Stewart K. J., Ammeter A. P., Maruping L. M. (2005). A Preliminary Analysis of the Influences of Licensing and Organizational Sponsorship on Success in Open Source Projects. *Proceedings of the 38th Hawaii International Conference on System Sciences – 2005*
- Strauss, A. L., & Corbin, J. M. (1990). *Basics of qualitative research: Grounded theory procedures and techniques*. Newbury Park, Calif: Sage Publications.
- West, J. & O'Mahony, S. (2005). Contrasting Community Building in Sponsored and Community Founded Open Source Projects. *System Sciences (2005)*: 196c. doi:10.1109/HICSS.2005.166
- West, J. & Gallangher, S. (2004). Key Challenges of Open Innovation: Lessons from Open Source Software. <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.465.4497&rep=rep1&type=pdf>
- Wheeler, B. (2007). Open Source 2010: Reflections on 2007 Educause Review, vol. 42, No. 1 (January/February 2007):48–67. <https://er.educause.edu/articles/2007/1/open-source-2010-reflections-on-2007>
- Wheeler, B. (2004). Open Source 2007: How Did This Happen? *Educause Review*, vol. 39, no. 4 (July/August 2004): 12–27, <http://www.educause.edu/er/erm04/erm0440.asp>.
- Winkelmann, R. (2015). *Governance in Collaborative Open Source Software Development Organizations: A Comparative Analysis of two Case Studies*. Master's Thesis. University of Bern
- Wikipedia contributors. (2019). Bruce Perens. In *Wikipedia, The Free Encyclopedia*. Retrieved June 11, 2019, from https://en.wikipedia.org/w/index.php?title=Bruce_Perens&oldid=898960562
- Yin R. K. (2013). *Case study research: Design and methods*. Sage publications.