Investigating Strategy, Diversity and Multiplexity in Open and Coopetitive Software Ecosystems

José A. Apolinário Teixeira.

1 Background and significance

1.1 Significance of research project in relation to current knowledge, research-based starting points

As early put on *Wall Street Journal* by Marc Lowell Andreessen (2011), a well respected serial entrepreneur, investor, and software engineer (co-founder of Netscape and Andreessen Horowitz venture capital firm), "software is eating the world" and "there is a constant need for new systems and new software". In a time where so many people work with computers and even carry them along, it is worth understanding how software is being produced.

Even if many take the traditional view of software as developed by distinct and autonomous 'software houses', empirical observation points out that on numerous occasions, software is neither developed in-house nor outsourced in dyadic relationships (1:1). Instead, software is increasingly co-developed by a network of individuals and organizations (n:n), which base their relations to each other dynamically on mutual interest. As such, software development is often embedded in networks compromised of heterogeneous and multidimensional ties that blur across organizational boundaries. For example, the Veterans Information Systems and Technology Architecture (VistA) open-source software that supported the medical care of military veterans in the United States for more than 40 years was co-developed by an extensive network of public administration bodies, civil servants, software suppliers, consultant companies, contractors, volunteers, universities, and standardization organizations. Such a system was not procured from a single organization, but instead co-developed in a network that spanned across multiple organizations.

As software is increasingly developed in networks the traditional concept of a 'software-house' is losing its theoretical relevance to the 'software-ecosystem' concept as more and more software is co-developed with others (e.g., suppliers, customers, partners, competitors, communities, crowds, and users among others participating actors). It is somehow surprising that contemporary software development networks increasingly embed rival and competing firms — for instance, it is known that Apple and Samsung kept collaborating in the development of the WebKit Internet browsing technologies while running at the same time expensive patent wars in the courts worldwide (J. Teixeira and Lin, 2014). Or that BMW, Ford, Honda, and Hyundai joint-develop open-source autonomous driving technologies even if they a keep aggressive competition in overlapping geographical markets (J. A. Teixeira, 2023). Moreover, it also contradictory that participating actors progressively give up intellectual property rights that are automatically granted by law and opt to release software under permissive open-source licenses where authors lessen their copyrights to allow the software to be executed, studied, modified and distributed for any purpose (Weber, 2009; Kilamo et al., 2012).

The past decade has seen tremendous transformation on software ecosystems giving rise to new ways of managing, marketing, producing, releasing and consuming software. Five domains in which these changes have been especially notable are: (i) the increasing complexity of the software being developed, (ii) the recurrence of cooperation among competing firms (i.e., coopetition) in the co-production of software, (iii) the slackening of intellectual property as firms increasingly work in an open-source way, (iv) the accommodation of ad-hoc contributions from an unknown workforce as developers do not need to a specific organizational affiliation, contract or permission to contribute, and (v) the soaring difficulties of coordinating work produced by teams that are increasingly diverse, mobile and span across cultures, geographies, and organizations.

From the perspective of strategic management theory, in a world where software is jointly developed by a network of individuals and organizations that often compete with each other (J. Teixeira, Mian, and Hytti, 2016; Roth et al., 2020),firms are not just strategically positioning products and services in relation to their competitors (see Teece, Pisano, and Shuen, 1997; Martin, 2015), instead they are also strategically managing competitive-cooperative relationships within the software ecosystems they 'inhabit' (see Hannah and Eisenhardt, 2018; Hoffmann et al., 2018). For example, IBM, HP, and Huawei compete with each other on the cloud computing market in general, but they also cooperate in the co-development of OpenStack - a large cloud computing infrastructure similar to the ones powering Amazon's Elastic Compute Cloud and Microsoft's Azure. While on the one hand, those firms collaborate intensively in the development of OpenStack, on the other hand, they fight for revenue on complementary software (e.g., OpenStack plug-ins), specialized hardware, consulting, support services, and public clouds hosting (see J. Teixeira, Robles, and González-Barahona, 2015). As recently acknowledged by strategic management scholars, firms increasingly need to "navigate cooperation and competition in ecosystems" (Hannah and Eisenhardt, 2018).

With this research, I will contribute to a better understanding of why and how software production is moving from 'software-houses' to 'software-ecosystems'. To do so I will explore software ecosystems (currently capturing much attention from both Software Engineering and Information Systems scholars) from the lenses of organization theory and strategic management. Given the scarcity of research bridging strategic management with empirical software ecosystems, I will juxtapose the 'strategy' as announced by firms (in press releases, news, conferences, and summits) with what is actually practiced by the software developers (i.e., strategy as announced vs. strategy as practiced). The more we know about how strategy unfolds with practice within software ecosystems, the better we can strategically manage them.

As the demand and the complexity of SOFTWARE grows, as ORGANIZATIONS increasingly cooperate with others more openly, and as DEVELOPERS range from a wider array of cultural, geographical, and organizational backgrounds, the stack of infrastructural tools supporting the co-production of open-source software also matured. Among other infrastructural tools widely used by millions of software developers we can note the *Git* version control system, the *Gerrit* code review tool and the *BugZilla* defect tracking system. This means that plenty of relational data on "who collaborates with who" can be collected from those infrastructures that support the production of large open-source software ecosystems. So far most social network studies on software production considered only one collaborative layer (e.g., who codes with who, who reviews the code of who, or who fixes bugs with who) at the time. However, recent advances in multi-level and multi-layer social networks call for the investigation of the multiple data sources tracing different layers of collaboration all at the same time in unison. Furthermore, as multi-layer social network data is rare and difficult to obtain (Kivela et al., 2014; J. Teixeira, Hyrynsalmi, and Leppänen, 2020), we are in a unique position to contribute back to the development of social network theory by exploring multiplexity in software development networks. By this, we mean that we can investigate how software developers establish their connections in source-code editing, code-reviewing, and bug-fixing by taking all the layers in consideration. All towards a more holistic understanding of how software ecosystems evolve. See Figure 1 for an illustrative model of a coopetitive software ecosystems as a multi-level multi-layer networks.

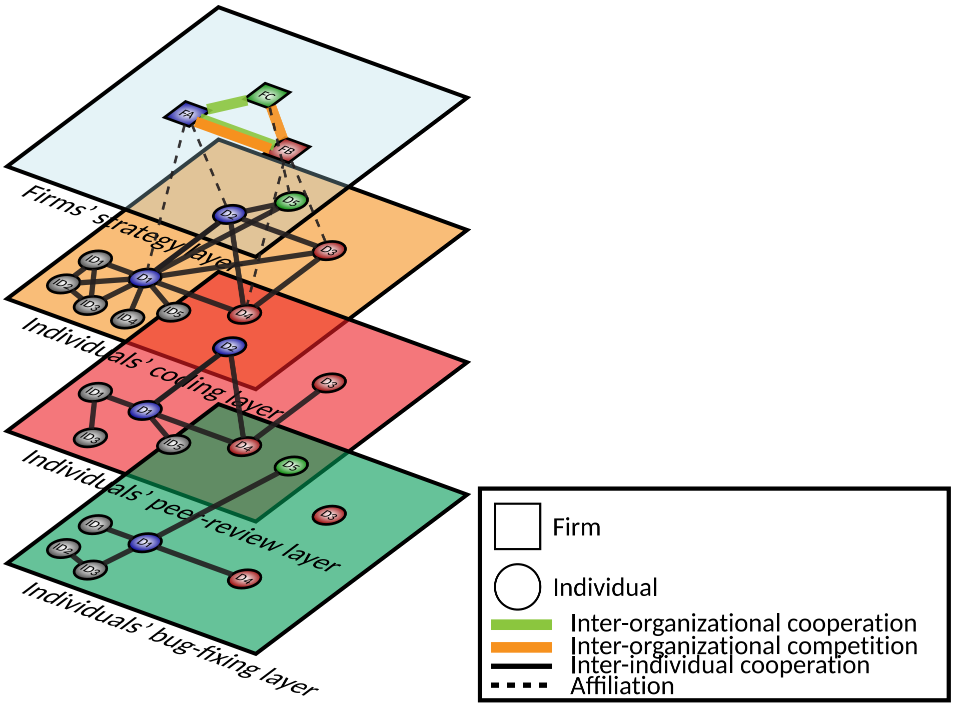

Figure 1: Illustrative cooperative-competitive configuration where a triad of firms simultaneously cooperate and compete. Note that indivuduals' cooperation can be multiplexed into the coding, peer-review, and bug-fixing layers among others.

1.2 Research questions and illustrative hypotheses

After consulting with my collaborators with interests in software ecosystems, coopetition, open-source software and social network analysis (see Section 3.5), we picked three research questions given their impact potential and feasibility within the project resources and duration.

Therefore, this research is framed by three guiding exploratory research questions:

- RQ1:) How does the strategy announced by firms for a given software ecosystem juxtaposes with the actual work of the software developers affiliated with them?
- RQ2:) How does diversity affect the joint production of software ecosystems?
- RQ3:) How do software developers work together in different activities and thus form multiplex social network relationships over time?

By exploring those three questions, we will be in a better position to understand how firms and individuals balance cooperation with competition within the joint-production of open-source software ecosystem – all towards the further theorizing of the 'open-coopetition' phenomena.

Examples of more specific theoretically-driven hypotheses, discussed with academic and industrial collaborators, that can be tested in the context of open and coopetitive software ecosystems by the end of this research could include:

H1 Strategy: Developers affiliated with organizations that announced strategic cooperation for the development of a software ecosystem tend to review the code of each other and put less effort into the review of code produced by other third-party developers across the overall software ecosystem.

H2 Strategy: Even if firms announce their decision to stop contributing to a given software ecosystem, they will still contribute by fixing defects and vulnerabilities jointly with their prior partners.

H3 Diversity: Organizational diverse teams (including developers from many competing firms) introduce a higher number of vulnerabilities in software releases than non-diverse teams (including only developers from a single firm).

H4 Diversity: Gender diverse teams (including developers from different genders) tend to review and reject contributions more often than non-diverse teams (including developers from a single gender).

H5 Multiplexity: Individuals that "code" together have a relational tendency to also "peer review" and "fix-bugs" together as well even if they are affiliated with competing firms (aka competitors).

H6 Multiplexity: The "coding", "peer reviewing" and "bug-fixing" social networks cluster by similar means. In other words, we can find the same sub-communities within a software ecosystem no matter what collaborative layer we look at.

2 Impact

2.1 Scientific impact

The first expected contribution regards the further theorizing of 'open-coopetition', a theoretical term coined during my doctoral studies (see Teixeira and Lin, 2014). The term is already being used by practitioners, as I see on news articles, YouTube or Wikipedia (see https://en.wikipedia.org/wiki/Open_coopetition). This research would allow me to further theorize why and how firms cooperate with competitors in an open-source way. After conducting this project, we should be able to tentatively explain boundary conditions in which strategy and diversity do not hinder the collaborative dimension of coopetiton. As recently pointed out by J. A. Teixeira (2023), extant theories on open-source software, coopetition, product development and open innovation often contradict each other (e.g., while some advocate for gate-keeping, information protection and high levels of intellectual property protection, others advocate for transparency, inclusiveness and a more open flow of information). From a practitioner's viewpoint, this is serious since naive assumptions concerning "work with competitors" and "open-source work" can lead in practice to opportunistic behavior, unintended spillover effects, and loss of reputation and trust among partners (see J. Teixeira, Mian, and Hytti, 2016).

Given the exploratory nature of this research, other contributions remain hidden in the large set of digital trace data left by the software developers (objective data not provoked by the researcher but naturally occurring from the behaviour of software developers in their pursuits of developing open-source software). Traditionally the actual behavior of product-developers regarding strategy, diversity, and coordination is often hidden from researchers within the organizational boundaries of the firm. However, given the transparency of open-source communities, we have access to more vast and grained data to pursue contributions grounded on the actual artefacts produced by software developers.

Furthermore, the infrastructural tools and systems supporting the joint production of complex open-source ecosystems are often open as well (i.e., based on popular open-source tools such as Git, Wikis, Gerrit, and BugZilla among others). This allows us to capture cooperation by looking at multiple layers (e.g., coding, documenting, peer-review, and bug-fixing). Traditionally researchers investigate one layer at a time, but recent advances in network science allow us now to model and analyze software ecosystems as multi-level and multi-layer social networks and unveil inter-level (i.e., organizational and individual) and inter-layer dependencies (aka multiplexity). With this research, we will be able to explain how the different social network layers modelling relational behaviour such as "coding", "peer-review", and "bug-fixing" co-evolve with each other over time in the co-production of software ecosystems.

The last contribution will take the form of a "Handbook of metrics for the analysis of digital ecosystems". As as we go along with the project, we will be discovering, cataloguing, and refining metrics that could inform the analysis of ecosystems from both business and technical perspectives. Consequently, we would write down a more utilitarian book alongside our papers. Such a book would inform other scholars interested in the topic and also inform decision-makers within real ecosystems. The book will start as a Wikibook co-developed by the applicant and its collaborators (within academia and software development communities). We will seek to publish it as an edited book from a reputed publisher with recognition among practitioners in the technological sector.

2.2 Effects and impact beyond academia

This research will inform decision-makers involved in software ecosystems. The more we know about how strategy unfolds with practice, the more we know about the effects of diversity on the productivity of teams and product-quality, and the more we learn lessons on coordination from the open-source community, the better we can manage, orchestrate, curate or influence the evolution of software ecosystems with desirable outcomes.

Furthermore, the catalog of measures on business and software ecosystems (i.e., the book) is a delivery of this project that will "speak" directly to practitioners.

3 Implementation

3.1 Work plan and schedule

The project is organized into four working-packages. The first three packages (WP1, WP2, and WP3) will address each of the three research questions (RQ1, RQ2, and RQ3), and a fourth (WP4) will deal with the production of a "Handbook of metrics for the analysis of digital ecosystems". While the first three work packages are to be executed sequentially, the forth spans across the overall project with most efforts to be concentrated on the last project year.

Table 1: Overview of working packages

Package	Objectives	Intitutional cooperation	Time-frame
WP1	Adressing RQ1 (strategy)	NUI Galway (6 months visit)	10 months @ 1st year
WP2	Adressing RQ2 (diversity)	Universidade Rey Juan Carlos (6 months visit)	10 months @ 2nd year
WP3	Adressing RQ3 (coordination)	University of Texas in Austin (6 months visit)	10 months @ 3rd year
WP4	Producing utilitarian book	Cross-instutional alliance	Across all project

Most studies of software ecosystems leveraged social network analysis modelled a single network layer. For example (1) J. Teixeira, Robles, and González-Barahona (2015) on 'who codes with who', (2) J. Teixeira, Hyrynsalmi, and Leppänen (2020) on 'who reviews the code of who', and (3) Aljemabi and Wang (2018) on 'who fixes bugs with who'. In this research project, we plan to go a step further and exploit recent advances in the analysis of complex networks and employ a multi-method approach that encompasses the multi-layers of a software ecosystem social network (see Table 3). I expect that results from the first working-package will add layers to subsequent analytical efforts. Meanwhile, WP4 (i.e., the book) will also get inputs from the executions of the other packages.

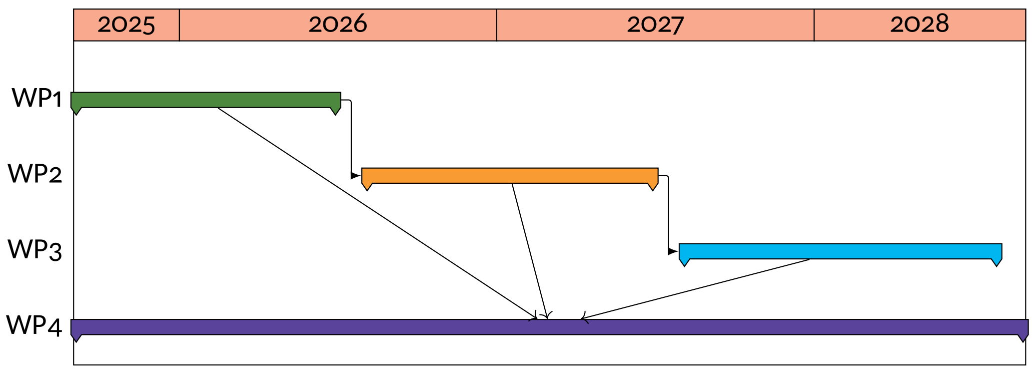

Figure 2: Gantt chart

3.2 Research data and material, methods, and research environment

3.2.1 Data collection

To a large extent, this project will ground on Natural Occurring Digital Trace Data (NODTD) produced by some open-source software projects. Also, practitioners (software developers often with management responsibilities) will act as key informants. The applicant already established good contact with practitioners with responsibilities on the co-production of cloud computing and automotive software ecosystems from prior "engaged scholarship" research efforts (e.g., OpenStack and Automative Grade Linux open-source ecosystems).

I will be collecting very detailed and fine-grained NODTD data from two dyads of rival open-source software ecosystems (see Table 2). This would allow us to account for variation across industrial domains (i.e., cloud computing and automotive) and also for variation across dyads of similar offerings (i.e., the selected software ecosystems are developed by distinct networks of firms but for similar purposes). Such sample selection also reflects prior research from the applicant (J. Teixeira, Robles, and González-Barahona, 2015; J. Teixeira, Mian, and Hytti, 2016; J. A. Teixeira, 2023; J. Teixeira and Karsten, 2019; J. Teixeira, Hyrynsalmi, and Leppänen, 2020) as prior established contacts with practitioners (software developers often with management responsibilities) are worth being explored.

Table 2: Selected open-source software ecosystems.

Project	Domain	Large participants (e.g.)	Finnish participants (e.g.)
OpenStack	cloud computing infrastructure	IBM, HP, Rackspace	Nordcloud, Nokia Bell Labs
CloudStack	cloud computing infrastructure	Citrix, Huawei, SAP	Telia, Elisa
Automative Grade Linux	infotainement systems	Toyota, Suzuki, Honda	Tuxera, Roboride
Genivi alliance	infotainement systems	BMW, Daimler, Honda	Tieto, Reaktor

For addressing RQ1 (strategy as announced vs. strategy as practiced) the data collection will be organized among three subsets: one pertaining strategy announcements, other pertaining competition and another pertaining cooperation.

- **The strategy announcements data subset** is qualitative data on what was strategically announced by firms on the software ecosystem they inhabit. This data can be found to a large extent on the Internet as management often communicates their strategy in official press releases, blogs, tech conferences, trade shows, and summits.
- **The competition data subset** is quantitative data on who competes with who in a given software ecosystem. This requires to systematically assess which firms market similar products and services on the same geographical area. For the particular case of the OpenStack, data is already available from prior research (see https://users.abo.fi/jteixeir/OpenStackSNA/).
- **The cooperative data subset** is social network data on who cooperates with who and it is obtained by mining open-source software repositories. Data from software repositories is objective as it is not provoked by the researchers, instead, it naturally occurs from the behavior of software developers in their pursuits of co-developing software. For the case of the OpenStack, a large and coopetitive open-source project powering thousands of data-center across the world, data is already available from prior research (see https://users.abo.fi/jteixeir/OpenStackSNA/). Note that all of the selected open-source software ecosystem relly on the same infrastructural tools (e.g., Git, Gerrit, BugZilla, and Jenkins) allowing us to re-use data-collection efforts to capture multiple layers of cooperation.

Regarding RQ2 (the one about diversity in software development teams), there are two exceptional circumstances that make this a rare data collection opportunity. First, the release management team at OpenStack (the one ultimately deciding what and what is not within the official OpenStack release versions) evaluates and tags sub-project repositories with a “team:diverse-affiliation” tag as a sight of healthy collaboration within the sub-project team. This highlights that organizational diversity empirically relevant for OpenStack and it will allows us to statistically compare diverse vs. non-diverse sub-project teams. Another circumstance is the existence of the 'Diversity working group' that sponsors programs that encourage diversity by identifying and removing the barriers that keep OpenStack from having a diverse community. A recent report from Izquierdo et al. (2019) sponsored by the working group and published on IEEE software highlighted several diversity in the community (e.g., gender and startups participation). Still, there is data on performance and quality of software that need to be collected from (1) the source-code repository (orchestrated in git), (2) the bug trackers database (i.e., both on Launchpad bugs and StoryBoard), (3) the code-review tool (i.e., orchestrated in Gerrit) and (4) venerability issues (publicity available via www.cvedetails.com). With all this data, we can then test several propositions regarding diversity in software development teams and its effects on software development (e.g., process and quality). The existence of many layers of data (i.e., information on firms, developers and their interactions in terms of source-code development, peer-review, documenting, and bug-fixing) invites us to model software ecosystems as multi-layer social networks (Kivela et al., 2014; Boccaletti et al., 2014; Lazega and T. Snijders, 2016) - an approach that, to the best of our knowledge, was never employed to research either software ecosystems or business ecosystems.

In order to address RQ3 (i.e., the one about multiplexity) we will consider two levels in a two-mode network where *individuals* (i.e., software developers) affiliate with *organizations* as in prior research (J. Teixeira, Hyrynsalmi, and Leppänen, 2020; J. Teixeira, Robles, and González-Barahona, 2015). Furthermore, we will consider cooperation across the three distinct layers of “coding”, “peer-review”, and “bug-fixing” (see Figure 1 for conceptual illustration).

3.2.2 Methods and tools

This research requires the integration of multiple methods established across different disciplines. All the research questions will force me to retrieve and analyze natural occurring data from the Internet. Here, I will follow established guidelines on the study of the behavior of individuals online (see Kozinets, 2009). In addition, a number of cross-disciplinary methodological notes on how to use archival data in case study research will be followed as well (e.g., Runeson and Höst, 2008).

This research will early force me to mine software repositories. Here, my research will benefit from several guidelines provided by Software Engineering scholars (e.g., González-Barahona and Robles, 2012; Gousios and Spinellis, 2017). By mining software repositories we get relational data on *who works with who*, but I still need to analyze such longitudinal cooperative networks. Here I will turn to established guidelines on social network analysis in general (e.g., Wasserman and Faust, 1994), 2D and 3D social network visualization (e.g., J. Teixeira, Mian, and Hytti, 2016) , and multi-layer social network analysis (e.g., Kivela et al., 2014). In addition, I will also address methodological guidelines on how to model social networks from digital trace data (e.g., Howison, Wiggins, and Crowston, 2011), and the use of mixed-methods on social networks research (e.g., Dominguez and Hollstein, 2014). An overview of methodological notes guiding this research is given on Table 3.

While RQ1 will rely on the visualization of social networks, RQ2 and RQ3 will require more sophisticated social network analysis methods and tools that leverage simulation (e.g., Relational Even Modelling). Practitioners, often with software development responsibilities in the selected software ecosystem, will be engaged in efforts from data collection, data analysis, and interpretation of results. They will act also as ad-hoc evaluators of the practical relevance of this research. After all, this research proposal is shaped by prior interactions with them.

Table 4: Specialized tools for data collection and analysis

RQ	Tool and URL	IN	OUT
1+2	ScrapLogGit2Net	git repository	temporal network (co-edited files)
1+2	git2net	git repository	temporal network (co-edited code blocks)
1	Tulip		
1	BlenderGraphs	temporal network	network 3D model
1	Blender	network 3D model	3D longitudinal visualizations & annimations
1+2	GoldFish	temporal network	statistical tests of social network mechanisms
1+2	Grimoire Lab (www.bitergia.com)	Git, LaunchPad, Gerrit	software development metrics
3	Pymnet	Different static networks	Multilevel and multilayer network metrics and visualizations

Table 3: Multidisciplinary approach

Employed methodology	**Discipline(s)**	**Methodological notes**
Virtual-Ethnography	Marketing	Kozinets (2002) Kozinets (2009)
Case study methodology and the use of archival data	Multidisciplinary	Yin (2011) Eisenhardt (1989) Runeson and Höst (2008)
Mining of software repositories	Software-Engineering	González-Barahona (2012) Hassan (2008) Kagdi et al. (2007)
Network analysis of digital trace data	Software-Engineering Information-Systems	Robles et al. (2004) Hahn et al. (2008) Howison et al.(2012)
Network analysis with emphasis on the visualization of collaborative activities	Multidisciplinary	Lundvall (1992) Cambrosio et al. (2004) Glänzel and Schubert (2005)
Analysis of multi-layer longitudinal network data	Multidisciplinary	Kivela et al. (2014) T. Snijders et al. (2006)
Mixed-methods in the research of social networks	Multidisciplinary	Dominguez and Hollstein (2014) Williams and Shepherd (2017)

read the source code and understand what does the bots do and how they influence coordination. The behavior of bots will be documented and modeled UML activity and sequence diagrams so they can be later shown to software developers (Scanniello et al., 2014). After the audits are conducted, I can then engage with the OpenStack open-source community discuss preliminary results for RQ1 and RQ2 as well as collect data for RQ3. Here we will follow established methods on how to conduct qualitative research (Silverman, 2009; Miles and Huberman, 1994) in general, conduct interviews (Myers and Newman, 2007),building theory from case-studies (Gibbert and Ruigrok, 2010) and design mixed-methods research (Shah and Corley, 2006; Ågerfalk, 2013). All following the principles of engaged scholarship (Van de Ven and Johnson, 2006) and respect for the principles, norms, and values of the open-source community (Stewart and Gosain, 2006; Feller et al., 2008).

While in Figure 1, I provided a conceptual visualization of open and coopetitive software ecosystems as multi-layer networks, in the following Table 4, I provide an overview of the different tools and statistical packages that will support my research execution.

3.3 Risk assessment and alternative implementation strategies

As we are already familiar with the OpenStack and the Automotive Grade Linux communities, we do not foresee many data-collection issues from the infrastructure supporting their software production efforts. The projects gave a quite strict process for accepting contributions from software developers, resulting in quite good digital trace data for analytical purposes.

There are always risks pertaining the access to practitioners (i.e., software developers). To succeed we will:

- Leverage personal relationships from prior research within the ecosystems.
- Attend the community conferences and summits and if possible meet people face to face. Here it is important, to pay special consideration to practitioners who helped us in prior research efforts.
- Conduct the first preliminary research based on public domain open-source data so that we have something to show before engaging with practitioners at a later stage. From our experience, social network visualizations obtained from the project's data can capture the interest of its developers.
- Study the workload and the release management calendar before engaging with developers. We will try to avoid engaging with software developers (1) two weeks before the code gets released (once every 3 months in the case of OpenStack and once every 6 months in the case of Automotive Grade Linux) and (2) when we note a lot of bug fixing activity on the bug trackers. In other words, avoid engaging with developers during periods associated with a heavy workload.
- Be transparent about what kind of data we collect and obtain practitioners' consent if necessary.

A critical factor shaping our research strategy is the complexity of the data that will emerge. Particularly, when mining the software source code with social network analysis for RQ1 (Strategy) we will attempt to "visualize" the dynamics of software ecosystems over time. However, if the data turns out to be too complex for the current state-of-the-art visualization tools, we need to turn to more quantitative approaches for analyzing complex and multi-level longitudinal networks (e.g., Relational Even Models). A recent research visit to ETH Zurich for a workshop on Relational Event Modelling with the goldFish tool will be fruitful in case the data is too complex to unveil relevant findings with visualization alone.

3.4 Project personnel and their project-relevant key merits

Given my dual technical and management background (i.e., I studied in both computer science and management departments and worked also in the industry both as a software developer and as a project/line manager), I perceive to be in a good position to unveil contributions that could be relevant both to the more managerial or the more technical perspectives of the phenomena under investigation.

In prior projects, I evidenced that I can analyze networks modelling *who codes with who* as well as *who reviews who* among individuals and organizations in open and coopetitive software ecosystems (e.g., J. Teixeira, Hyrynsalmi, and Leppänen, 2020; J. Teixeira, Mian, and Hytti, 2016). With this project, I would advance my skills to analyze multi-level networks (i.e., by modelling both organizations and individuals) and multi-layer networks (i.e., by modelling the "coding", "peer-review" and defect-fixing in unison).

Furthermore, this project will allow me to further theorize my prior theoretical work on 'open-coopetition' in the context of the software ecosystems. As I coined the 'open-coopetition' term during my doctoral studies and as I been following recent research from others that followed my ideas on 'open-coopetition', I am in a good position to be on the front line of 'open-coopetition' research that so far gathered interest in Strategic Management, Information Systems, Software Engineering,and Innovation Studies (see Jacobides, Cennamo, and Gawer, 2018; Nguyen-Duc et al., 2019; Pekovic, Grolleau, and Mzoughi, 2020; Roy, Chesbrough, and Bez, 2018).

3.5 Collaborators

Table 5: Collaborators

Colaborator	Justification
Sami Hyrynsalmi LUT University Finland	With who I co-author before some award-winning papers (i.e., ICSOB and ISPMA best paper award). He will offer theoretical insights on software ecosystems.
Annika Laine-Kronberg Åbo Akademi University Finland	Started her Professorship at 2023 in Åbo Akademi after decades of research on coopetition as a management strategy.
Gregorio Robles Universidad Rey Juan Carlos Spain	With who I co-authored before. Will bring theoretical insights on software ecosystems, software development metrics, and methodological expertise on the mining of software repositories with social network analysis.
Sirkka Jarvenppa University of Texas United States	Who I recurrently met at conferences (e.g., ICIS, ECIS) and Inforte seminars in Finland. She will offer theoretical insights into governance and coordination in online communities.
Lorraine Morgan LERO - The Irish Software Engineering Research Center Ireland	With who I also co-authored. I first meat at LERO in the University of Limerick during a scholar research visit during my doctoral studies to work with Professor Brian Fitzgerald (now president of the Association for Information Systems Research.)With much knowledge on networks of open-source service providers. She will bring theoretical insights on open-source software ecosystems, strategy, strategy networks, and strategy as practice.
Kim Holmberg University of Turku Finland	He is Senior Research Fellow and Head of Unit at Economic Sociology at the University of Turku, Finland. To best of my knowledge, he is the scholars with most expertise in social network analysis in Finland.
Peter Holme Aalto University Finland.	Professor of Computer Science interested in understanding the world around us through networks describing how things are connected. We cooperated before in an application for large-scale consortium research.
Mikko Kivelä Aalto University Finland.	Assistant Professor of Computer Science in the fields of network science and complex networks. He maintains several popular tools for modelling multilayer social network such as pymnet. We cooperated before in an application for large-scale consortium research.
N: research assistant Åbo Akademi University Finland	A research assistant (e.g., Master thesis workers) will be hired to develop, test and release new features to the ScrapLogGit2Net open-source tool that so far was developed mostly by the applicant (see https://github.com/jaateixeira/ScrapLogGit2Net) for own research purposes.

3.6 Responsible science

3.6.1 Research ethics

Overall this research is to be conducted under the guidance of the Finnish National Board on Research Integrity TENK as appointed by the Finnish Ministry of Education and Culture to promote the responsible conduct of research.

More specifically related to our investigation, we will follow the principles of engaged scholarship (Van de Ven and Johnson, 2006) and respect the principles, norms, and values of the open-source community (Stewart and Gosain, 2006; Feller et al., 2008) while engaging with practitioners. Furthermore, we will address the challenges of close collaboration with industry partners by following Simonsen (2009) recommendations for projects that involve intensive collaboration with industry partners.

Finally, all the developers' entity data will be pseudonymized, even if we do not expect any ethical issues to arise with the study of communities that emphasize openness and transparency in their modus-operandi.

3.6.2 Open science

I will follow the Åbo Akademy Guidelines policy for Open Access. As an open-source researcher, I value the values of openness and all my recent publications are available in open-access. Furthermore, all of my research curated data-sets as well as scientific software tools are publicly available at my website (http://users.abo.fi/jteixeir) from the time articles get accepted.

To publish to an Information System audience, I should aim at the senior basket of eight journals proposed by the Association of Information Systems. Among those, I will prioritize journals that allow green open-access (i.e., *Information Systems Research*, *Information Systems Journal*, and *Journal of Management Information Systems*). All those journals allow self-archiving without restrictions (processing fee or one year embargo) for allowing archive pre-print and post-print. In some cases, I will need to pay an open-access processing charge.

To communicate to Software Engineering audience, it is simpler, as all the leading conferences and journals either allow self-archiving or offer gold open-access mechanisms.

3.6.3 Equality and non-discrimination:

This research will investigate gender issues in open-source software ecosystems by comparing diverse vs. non-diverse teams. From prior empirical engagement with the OpenStack community, there is a perception that female software developers commit fewer bugs into the software being released (in proportion to the small number of women contributing to the project). Furthermore, there were also informants pointing out that female software developers document more carefully their submitted patches and therefore pass peer review more easily. I will now critically and scientifically assess those perceptions. Most probably, this will lead to results that call for more females in the development of software ecosystems.

3.6.4 Sustainable development objectives

This project, by addressing diversity in software development will contribute to a more participatory society for citizen by investigating gender issues within open-source software ecosystems. Overall, this research will lead to an increased understanding and better management of open-source software projects. This is good for sustainable development as open-source project reuse many other open-source projects, therefore leading to the removal of duplicated efforts and duplicated resources.

References

Ågerfalk, P. J. (2013). “Embracing diversity through mixed methods research”. In: *European Journal of Information Systems* 22.3.

Aljemabi, M. A. and Z. Wang (2018). “Empirical Study on the Evolution of Developer Social Networks”. In: *IEEE Access* 6.

Boccaletti, S. et al. (2014). “The structure and dynamics of multilayer networks”. In: *Physics Reports* 544.1. The structure and dynamics of multilayer networks.

Cambrosio, A., P. Keating, and A. Mogoutov (2004). “Mapping Collaborative Work and Innovation in Biomedicine: A Computer-Assisted Analysis of Antibody Reagent Workshops”. In: *Social Studies of Science* 34.3.

Dominguez, S. and B. Hollstein (2014). *Mixed Methods Social Networks Research: Design and Applications*. Structural Analysis in the Social Sciences. Cambridge University Press.

Feller, J. et al. (2008). “From Peer Production to Productization: A Study of Socially Enabled Business Exchanges in Open Source Service Networks”. In: *Information Systems Research* 19.4.

Gibbert, M. and W. Ruigrok (2010). “The “What” and “How” of Case Study Rigor: Three Strategies Based on Published Work”. In: *Organizational Research Methods* 13.4.

Glänzel, W. and A. Schubert (2005). “Analysing scientific networks through co-authorship”. In: *Handbook of quantitative science and technology research*. NYC, USA: Springer, pp. 257–276.

González-Barahona, J. M. and G. Robles (2012). “On the reproducibility of empirical software engineering studies based on data retrieved from development repositories”. In: *Empirical Software Engineering* 17.1-2.

Gousios, G. and D. Spinellis (2017). “Mining Software Engineering Data from GitHub”. In: *IEEE/ACM 39th International Conference on Software Engineering*, pp. 501–502.

Hannah, D. P. and K. M. Eisenhardt (2018). “How firms navigate cooperation and competition in nascent ecosystems”. In: *Strategic Management Journal* 39.12.

Hoffmann, W. et al. (2018). “The interplay of competition and cooperation”. In: *Strategic Management Journal* 39.12.

Howison, J., A. Wiggins, and K. Crowston (2011). “Validity Issues in the Use of Social Network Analysis with Digital Trace Data”. In: *Journal of the Association for Information Systems* 12.12.

Izquierdo, D. et al. (2019). “OpenStack Gender Diversity Report”. In: *IEEE Software* 36.1.

Jacobides, M. G., C. Cennamo, and A. Gawer (2018). “Towards a theory of ecosystems”. In: *Strategic Management Journal* 39.8.

Kilamo, T. et al. (2012). “From proprietary to open source—Growing an open source ecosystem”. In: *Journal of Systems and Software* 85.7.

Kivela, M. et al. (2014). “Multilayer networks”. In: *Journal of Complex Networks* 2.3.

Kozinets, R. V. (2009). *Netnography: Doing ethnographic research online*. London, UK: Sage Publications Limited.

Lazega, E. and T. Snijders (2016). *Multilevel Network Analysis for the Social Sciences*. Ed. by E. Lazega and T. Snijders. Methodos Series 12. Springer, p. 375.

Martin, R. (2015). “Strategy is about both resources and positioning”. In: *Harvard Business Review*.

Miles, M. B. and A. M. Huberman (1994). *Qualitative Data Analysis: An Expanded Sourcebook*. Second. SAGE Publication Inc.

Myers, M. D. and M. Newman (2007). “The qualitative interview in IS research: Examining the craft”. In: *Information and Organization* 17.1.

Nguyen-Duc, A. et al. (2019). “Do Software Firms Collaborate or Compete? A Model of Coopetition in Community-initiated OSS Projects”. In: *e-Informatica Software Engineering Journal* 13.1.

Pekovic, S., G. Grolleau, and N. Mzoughi (2020). “Coopetition in innovation activities and firms’ economic performance: An empirical analysis”. In: *Creativity and Innovation Management* 29.1.

Roth, S. et al. (2020). “Open coopetition: when multiple players and rivals team up”. In: *Journal of Business Strategy* 41.6.

Roy, F. L., H. Chesbrough, and M. Bez (2018). “Open coopetition: a research program”. In: European Academy of Management (EURAM).

Runeson, P. and M. Höst (2008). “Guidelines for conducting and reporting case study research in software engineering”. In: *Empirical software engineering* 14.2.

Scanniello, G. et al. (2014). “On the Impact of UML Analysis Models on Source-code Comprehensibility and Modifiability”. In: *ACM Transaction on Software Engineering Methodology* 23.2.

Shah, S. K. and K. G. Corley (2006). “Building Better Theory by Bridging the Quantitative–Qualitative Divide”. In: *Journal of Management Studies* 43.8.

Silverman, D. (2009). *Doing Qualitative Research*. SAGE Publications Limited.

Simonsen, J. (2009). “A Concern for Engaged Scholarship: The challenges for action research projects”. In: *Scandinavian Journal of Information Systems* 21.2.

Snijders, T. A. B. et al. (2006). “New specifications for exponential random graph models”. In: *Sociological Methodology* 36.1.

Stewart, K. J. and S. Gosain (2006). “The Impact of Ideology on Effectiveness in Open Source Software Development Teams”. In: *MIS Quarterly* 30.2.

Teece, D. J., G. Pisano, and A. Shuen (1997). “Dynamic Capabilities and Strategic Management”. In: *Strategic Management Journal* 18.7.

Teixeira, J., S. Hyrynsalmi, and V. Leppänen (2020). “Network Science, Homophily and Who Reviews Who in the Linux Kernel?” In: *Proceedings of the 28th European Conference on Information Systems (ECIS 2020)*. AIS.

Teixeira, J. and H. Karsten (2019). “Managing to release early, often and on time in the OpenStack software ecosystem”. In: *Journal of Internet Services and Applications* 10.1.

Teixeira, J. and T. Lin (2014). “Collaboration in the open-source arena: The WebKit case”. In: *Proceedings of the 52nd ACM SIGMIS Computers and People Research (CPR 2014)*. ACM.

Teixeira, J., S. Mian, and U. Hytti (2016). “Cooperation among competitors in the open-source arena: The case of OpenStack”. In: *Proceedings of the 33th International Conference on Information Systems (ICIS 2016)*. AIS.

Teixeira, J., G. Robles, and J. M. González-Barahona (2015). “Lessons learned from applying social network analysis on an industrial Free/Libre/Open Source Software ecosystem”. en. In: *Journal of Internet Services and Applications* 6.1.

Teixeira, J. A. (2023). “Towards understanding open-coopetition – Lessons from the automotive industry”. In: *Proceedings of the 44th International Conference on Information Systems (ICIS 2023)*. AIS.

Van de Ven, A. H. and P. E. Johnson (2006). “Knowledge for theory and practice”. In: *Academy of Management Review* 31.4.

Wasserman, S. and K. Faust (1994). *Social Network Analysis: Methods and Applications*. Structural Analysis in the Social Sciences. Cambridge University Press.

Weber, S. (2009). *The Success of Open Source*. Harvard University Press.