CHALMERS
UNIVERSITY OF TECHNOLOGY

UNIVERSITY OF GOTHENBURG

Constructing Social Networks from OSLC-Based Data for Improving Communication in Software Development

Master's thesis in Software Engineering

Wissam Alfreijat

Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG
Gothenburg, Sweden 2017

Master's thesis 2017:NN

Constructing Social Networks from OSLC-Based Data for Improving Communication in Software Development

Wissam Alfreijat

CHALMERS
UNIVERSITY OF TECHNOLOGY

UNIVERSITY OF GOTHENBURG

Software Engineering
Department of Computer Science and Engineering
Chalmers University of Technology | University of Gothenburg
Gothenburg, Sweden 2017

Constructing Social Networks from OSLC-Based Data for Improving Communication in Software Development
Wissam Alfreijat

Supervisor: Grischa Liebel, Department of Computer Science and Engineering
Examiner: Robert Feldt, Department of Computer Science and Engineering

Master's Thesis 2017:NN
Department of Computer Science and Engineering
Software Engineering
Chalmers University of Technology
University of Gothenburg

Typeset in LaTeX
Gothenburg, Sweden 2017

Constructing Social Networks from OSLC-Based Data for Improving Communication in Software Development

Wissam Alfreijat
Department of Computer Science and Engineering
Chalmers University of Technology
University of Gothenburg

Abstract

Background: Communication is a highly important factor in the success of software engineering efforts. However, it is a challenging part of the software development process where insufficient communication is common, especially in large organizations and projects.

Objective: To help with communication insufficiency in software engineering, we investigate the automatic construction of networks visualizing people and their connections based on engineering data, by mining data based on OSLC, which is a specification for integrating software development tools using the principles of linked data. The result networks contribute in addressing those issues and provide a base for further analyzing and identifying them.

Method: We conducted a design science study of two iterations, during which we developed a tool capable of automatically constructing these networks out of existing OSLC-based data. The evaluation was based on applying the approach to data published by the IBM's Jazz initiative. A survey was conducted to evaluate the usefulness of the approach and the developed tool, in addition to the accuracy of the construction process of the network.

Results: Our results indicate that the approach is feasible, albeit it needs customization to the way OSLC is adopted in the target environment by the organization. Additionally, the results indicate usefulness of the approach and tool in helping to tackle communication issues in software development environments.

Conclusion: The approach, albeit promising in tackling communication issues, needs further investigation regarding accuracy of the result networks. Additionally, the approach faces the challenge of the flexibility in implementing OSLC. Thus, further investigation is needed in how to reduce the effort of applying the approach in different organizations.

Keywords: Communication; Software Engineering; OSLC; Social Network Analysis.

Acknowledgements

I would like to thank my supervisor Grischa Liebel for all his support and guidance throughout the project. I would also like to thank my parents for always believing in me, my wife Mariam and my son Michael for standing by me through all the difficult and stressful times, and my friends and colleagues for all their love and support. Additionally, I would like to thank the Swedish Institute for granting me a scholarship to complete my master's studies.

Wissam Alfreijat, Gothenburg, 2017

Contents

List of Figures

1

Introduction

In software engineering, communication is a challenging part of the overall software development process, especially in the requirements engineering (RE) part of it, which significantly affects project success [2, 3]. Insufficient communication during the software development life cycle might have a negative effect where requirements "slip through the gaps" [4], which leads to them being misunderstood or even left out. This results in failure to fulfill the customer's expectations on functional and quality levels [4].

While developing software, many groups of people must coordinate their work, or at least share and exchange information on a regular basis [5]. This is especially true in large projects where there are multiple sources for requirements, and where documentation does not reduce the need for extensive communication between the different groups working on the different phases in the development project [5]. Curtis et al. [5] argue that project members refer to different networks of people to gather different types of information, and that these sources could help reduce learning times for project members when used effectively. Liebel et al. [6] discuss several issues regarding communication in requirements engineering, such as lack of sharing the product and contextual knowledge, in addition to insufficient communication and feedback channels, and their effects on understanding requirements throughout the software development process. Furthermore, communication issues appear vividly in globally distributed software engineering, where communication and collaboration are difficult due to differences in time-zones, cultures and working environments [7].

To tackle communication problems in software engineering, the use of social networks analysis (SNA) was studied in literature and in practice to help investigate and uncover these issues within software development teams [8, 9]. Different types of social networks were built to support different areas of analysis, such as building task-based social networks for exploring collaboration in software teams [8], and the study of socio-technical congruence in social networks and its effect on the probability of build success [9]. However, these networks were not built using fully automated methods, but instead using a partial or fully manual process. Building such networks automatically based on data extracted from different tools is not a trivial job; it involves studying the different meta-models to which data produced by these different tools adheres. This becomes particularly troublesome when one

of these tools is replaced by another which uses a different meta-model [10]. It also requires certain traceability information to be present on different levels in the data, which is not always the case in real-world situations as pointed by Wohlrab et al. [11]. Previous work in this area [12] showed that the approach of using SNA is promising, but the use of multiple tools has only been addressed initially, which presents the need to investigate further into this approach using more than one tool. Our approach aims at providing a tool which builds a social network showing how people are connected as per existing data regarding how the resources they work on are related. This network helps people find the most relevant person to talk to within a certain context or to include in a meeting regarding a certain component for instance. The network also acts a basis for further analysis to aid in discovering gaps in communication once compared with actual communication patters between people for example.

One approach to address the challenge of integration between tools with different data schemes is the use of specifications defined by Open Services for Lifecycle Collaboration (OSLC) [13]. OSLC is an open community creating specifications for integrating tools. By following these specifications, independent tools can be integrated with other tools in the development life cycle process. Therefore, in a software development company where different tools are used in the development process, it becomes possible to integrate data with different formats produced by these tools to serve a purpose.

1.1 Purpose of the Study

The purpose of this study is to investigate the possibility of automatically constructing a social network automatically based on data extracted from different tools used in a company by the different teams working in different parts of the software development process. It aims to define a model for representing nodes and connections within the network, and define how to transform data which conforms to different OSLC specifications, and is produced by different tools, into this model.

A study by Mohamad et al. [12] showed that a social network built on industrial system engineering data has the potential to serve in the process of tackling communication and collaboration issues among people working in a team and across teams. The issue of extracting data produced by more than one tool was not investigated thoroughly. This study aims to build on that by investigating and evaluating how that approach can be applied to multiple tools using the OSLC standard. The resulting networks would then be used in industry to help with these communication issues and aid in tackling collaboration problems in software development.

1.2 Research Questions

In this study, the aim is to answer the following research questions (RQ).

RQ1. How can social networks be built based on data which conforms to the OSLC specifications and is generated by different tools used in the software development process?

RQ1.1. Which OSLC specifications have data properties that may be used to relate systems engineering data to employees to help build the network?

RQ1.2. Is it possible to apply this approach in a real development environment?

RQ2. To what extent can these networks aid in identifying gaps in communication among team members and across teams in industry?

RQ2.1. How valuable do practitioners believe that SNA is in tackling lack of effective communication among project members?

RQ2.2. What features do practitioners expect to see in the tool generating and displaying the social network?

To answer RQ1, we consider the different specifications defined by OSLC to determine which properties could be used to effectively construct the desired network. To accomplish that, we first need to answer RQ1.1 and RQ1.2. In RQ1.1, we investigate these OSLC specifications to find out which of them have properties recording the person related to the resources, which are represented using the specification metadata model. Then to show feasibility of the approach, we use data produced by the Jazz team while developing their products [14] to construct the social network, and hence addressing RQ1.2.

RQ2 is answered by addressing its two sub-questions. In RQ2.1 we ask practitioners what information they would like to extract from the network, and how effective it would be to analyze the constructed networks for identifying communication gaps among people working together on projects. To answer RQ2.2, we also rely on asking practitioners for their opinions on how the tool presenting the network could effectively help reading it and extracting the most information out of it.

1.3 Significance of the Study

With this study, we expect to have both an industrial and an academic impact. In terms of academic impact, we address challenges which have been identified by previous studies regarding communication issues in software development. Additionally, by targeting an existing standard such as OSLC, we aim to generalize the findings to a broader context, independent of domain or company.

As the targeted challenges are highly relevant to industry, successfully addressing these would automatically have an impact on industry. Additionally, previous work [12] found that SNA has the potential and significance for tackling these problems in software development, which makes this work highly interesting in industry.

2

Background and Review Of Literature

In this chapter, we examine previous research regarding communication challenges in software engineering, and especially in RE. Then, we examine related research regarding the use of SNA in studying communication challenges in software engineering. We finally present OSLC and its role in tools integration within software development, along with identified problems while using it for this purpose.

2.1 Communication Challenges in Software Engineering

Communication and collaboration challenges in software engineering and development have been widely studied in literature [5, 4]. Curtis et al. [5] found that although people and their ability to work together and collaborate are the most important assets in the development environment, this factor might not be as strong as it should be. They found that the longer the path which information had to traverse for communication channels to be established, the less likely communication was to happen. Bjarnason et al. [4] studied communication gaps in large-scale software development. During their root cause analysis, they discovered several causes for these communication gaps. These causes include the size of the organization and complexity of its products, which might lead to geographic, organizational and social boundaries that hinder communication; the degree of understanding of other people's roles; the lack of continuous communication between people regarding non-static and constantly changing information, such as requirements; and the insufficient vision of the overall goal within the development team [4].

Requirements and recognition of user needs are key factors in the success of software development projects [2, 3]. Hofmann and Lehner [2] conducted a study based on 15 RE teams, where they identified the RE practices which affect project success. They argue that for a project to be successful, the team needs to adopt several best practices related to knowledge of the application domain, resources, and the RE process. In [3], Voss investigated innovation success factors in software development by examining multiple studies which identified characteristics of successful innovations. Voss

then presented several success factors in software development including good communication, a committed product champion and good marketing and management practices; but considered "recognition of user needs" to be a necessary condition for software development to be successful [3]. However, the role of requirements in software development does not end with specifying them. Implementation of tractability capabilities in the software development life cycle is crucial for requirements to fully perform its role in the process. Bjarnason et al. [15] identify several challenges related to aligning requirements with verification and validation, such as cooperation and communication between people and weak requirements engineering practices among others. Tackling this lack of traceablility has been studied in research, such as in the work of Unterkalmsteiner et al. [16], who provide a practical assessment tool, "REST-bench", which assesses requirements engineering (RE) and software test (ST) alignment to help identify concrete improvement opportunities in this area.

The RE process, albeit an important part of the whole software development process, faces many challenges [6, 17]. Liebel et al. [6] argue that one of the main contributors to these challenges is that RE requires efficient communication and collaboration between the many different stakeholders. The authors investigate challenges facing RE in the automotive industry. Some of the challenges identified were the difficulty in establishing communication and feedback channels within and across organizations and the lack of knowledge regarding the product and its requirements. They concluded that it is important in RE to have correct channels for communication and feedback, especially when requirements in safety critical development environments are too complex to document, such as the case in the automotive industry. Al-Rawas and Easterbrook [17] study communication problems in RE in large software projects. They argue that these projects suffer breakdowns in communication throughout their different development phases, and they study several causes for that. These causes include one way communication channels; different notations used by the different communities involved in the specification; organizational barriers, which could restrain communication between groups working on different parts of the software and within different phases of the development process; and missing traceability links between requirements and the people responsible for them.

Communication issues during RE in particular and during software development in general pose a great challenge to the success of software development projects. We set out to tackle these issues with our approach, which was highly inspired by the work done by Mohamad et al. [12], by providing a tool that helps analyze them as well as overcome them.

2.2 Social Network Analysis

SNA is a promising tool for tackling communication challenges in software engineering. Examples of studies in this area include the work done by Wolf et al. [8] regarding mining task-based social networks to investigate collaboration in software teams. They apply their study on social networks built using data generated during

previous software builds to predict the probability of future broken builds by mining these networks. Kwan et al. [9], in contrast, investigate whether socio-technical congruence influences the success of software builds, by examining the alignment between a project's technical dependencies and social coordination. They carry out a case study with 151 developers over 7 geographically distributed sites to examine the relationship between the level of congruence and the probability of successful builds. They found that, depending on the build type, it either increases or decreases the build success probability. Another example is the work of Damian et al. [18] who construct a requirement-centric social network representing relationships among members working on a requirement and its associated downstream artifacts. In their approach, they study collaboration aspects such as communication, awareness and technical dependencies, all using SNA techniques. In [12], Mohamad et al. investigate constructing social networks by automatically extracting data from system engineering tools including requirements management tools. The aim in that study is to use the constructed networks to facilitate communication and find other possible use cases. The study applies the approach to two tools in a company. We aim to build upon that and study the possibility of applying the approach to a multiple tools scenario, targeting data structured based on the OSLC specifications.

In [8], Wolf et al. mentioned several applications of SNA. Developers joining a project might appreciate information regarding who was working on which task and who was responsible for architectural decisions for example. Another application is predicting project build results by analyzing previous networks constructed from data gathered by mining previous task related communication data. This in turn helps managers discover communication gaps within teams to mitigate risk of failed builds. In another study of applications of SNA, Cross et al. [19] showed how conducting SNA helped in promoting effective collaboration in a fragmented group of people, who had common work-related interests in terms of what they read, the conferences they attend, and the working groups which they are part of within the organization. They also discussed how SNA helped discover points where collaborative activity is not occurring due to boundaries within the organization on the functional, hierarchical, or geographic levels. Another interesting finding they had was that SNA is a powerful tool for individuals to actively shape their personal networks, by assessing its effectiveness along two dimensions; the first focuses on the diversity within each person's network, and the second focuses on the content and what kinds of resources people obtain from their relationships. Viewing the network from these perspectives helps people discover which relationships they should focus their investment on. In their findings, Mohamad et al. [12] reported several use cases for SNA in this context, and these were presented to practitioners to rate. The practitioners showed interest in being able to easily find who they should talk to or include in their emails, or where to direct their questions regarding specific components, in addition to knowing who would be affected by a change performed on a particular component for example.

In contrast to many of the studies mentioned here, our approach does not result in a social network representing actual communication between people. The networks we build represent how people are connected based on what existing engineering

data shows. Another difference is that in many of these studies, the authors did not adopt a fully automatic approach for constructing their social networks [8, 9, 18]. We, however, aim to use an automatic approach to the matter, which is inspired by the work done by Mohamad et al. [12]. While some of these studies analyze and use data such as emails to construct the social network [8, 18], we build our network around well structured and defined data with attributes and connections that have specific meanings. This does not require the use of natural language processing to extract related emails and exclude non-related ones for example. Other studies use code commits to connect people in the social network [8], which is limited to connecting people who work in the same domain. This means that if there was a misunderstanding in a requirement for example, the network needs to show communication links between people working on the code base and people working on the requirement, which would be useful in identifying this type of issue [12]. Restricting the network to resources used in one part of the software development process would not serve our purpose of solving communication issues in software teams mentioned in [5, 4, 6].

2.3 OSLC

Open Services for Lifecycle Collaboration (OSLC) [13] is an open community creating specifications for integrating tools. It is an approach based on the linked data principles of the web. This means that a Uniform Resource Identifier (URI) is used as the name of a resource, and that an HTTP URI is used to look up the resource. Additionally, the result of a HTTP URI lookup is a file with the Resource Description Framework (RDF) format containing useful information about the resource as well as links to other related resources. OSLC does not provide specifications for standardizing tools or their behaviour. Instead, it provides specifications for the data produced by these tools. This removes reliance on a specific tool, and introduces the ability to substitute one tool for another which conforms to the same specification. With these specifications, the goal is to define only properties that are related to integration, not to define the whole structure of data produced by a tool. OSLC provides specifications for tools in different phases of the software development life cycle. These include requirements management, change management and quality management tools among others. OSLC defines several resources, such as Requirement, TestCase and ChangeRequest. These resources are connected by several relationships, such as implementedBy between Requirement and ChangeRequest, and validatedBy between Requirement and TestCase. Additionally, there are common attributes among all resources, such as creator and contributor, which both point to a target resource of the type Person. Such attributes are considered social attributes, which makes them potential candidates to consider in the process of connecting people.

OSLC has been adopted as one of the main technologies in the large European research project Critical System Engineering Acceleration (Crystal) [20]. As many of the key players in the European embedded systems domain participated in this project, we consider OSLC specification for effectively integrating data. Integrat-

ing with data based on the OSLC specification has been investigated in research [21, 22, 23]. Elaasar and Neal [22] investigate integrating modeling tools with other tools in the development lifecyle through OSLC within a model driven development environment. Aichernig et al. [23] introduce a framework for integrating requirement engineering and testing activities data through OSLC. Their framework provides traceability between requirements, test cases and implementation models. However, relying on OSLC for data integration is not without its problems. For example, Leitner et al. [21] mention different obstacles they had to face and overcome while implementing an OSLC requirement management consumer. A re-occurring problem is that the OSLC specification is intentionally kept open to interpretation [21], such as the use of statements like "should implement" or "may implement" in the specification. This creates the opportunity for different "correct" implementations based on the specification, but at the same time eliminates the ability to effectively predict how data is formatted and what information is included or not in different tools following the same specification.

3

Research Methodology

In this study, we use the design science research methodology (DSR). We follow the iteration model described by Vaishnavi and Kuechler [1], and the guidelines defined by Hevner et al. [24]. We use this research methodology since our goal is to create a new artifact in the form of a tool that generates a social network automatically by mining software development data. Our focus is on creating this artifact, rather than how to incorporate it in its context. Thus, we do not resort to other research methodologies such as action research for example. DSR allows us to start with an initial solution for the identified problem, evaluate it, then build on the results of that evaluation to enhance that solution. This guarantees a high quality and practical result of the study.

3.1 Design Science Research

DSR is a method for problem solving, focusing on learning about the situation under study by developing and evaluating artifacts. These artifacts are intended to solve a problem which has not been solved, or provide a better solution than previous ones. They must also be viable artifacts in the form of a construct, a model, a method, or an instantiation [24]. Evaluation of the artifact must be done thoroughly and carefully, and applied to its utility, quality, and efficacy according to Hevner et al. [24].

3.2 DSR Iterations

The applied DSR model in this study is the one described by Vaishnavi and Kuechler [1]. The iteration cycle of DSR, as illustrated by that model, is shown in Figure 3.1. Each iteration comprises several steps; awareness of the problem, where the problem under study is defined and investigated; suggestion, where a preliminary solution is proposed; development, in which the proposed solution is used to implement an artifact; evaluation, where an assessment is done for the work completed so far in the iteration; and conclusion, where a satisfying output is reached even if it does not represent the final goal yet. During each iteration, knowledge and awareness of the problem are built up and used to start the next iteration if it exists. We completed two iterations in this study. The following sub-sections will explain how the DSR

model was applied throughout the study.

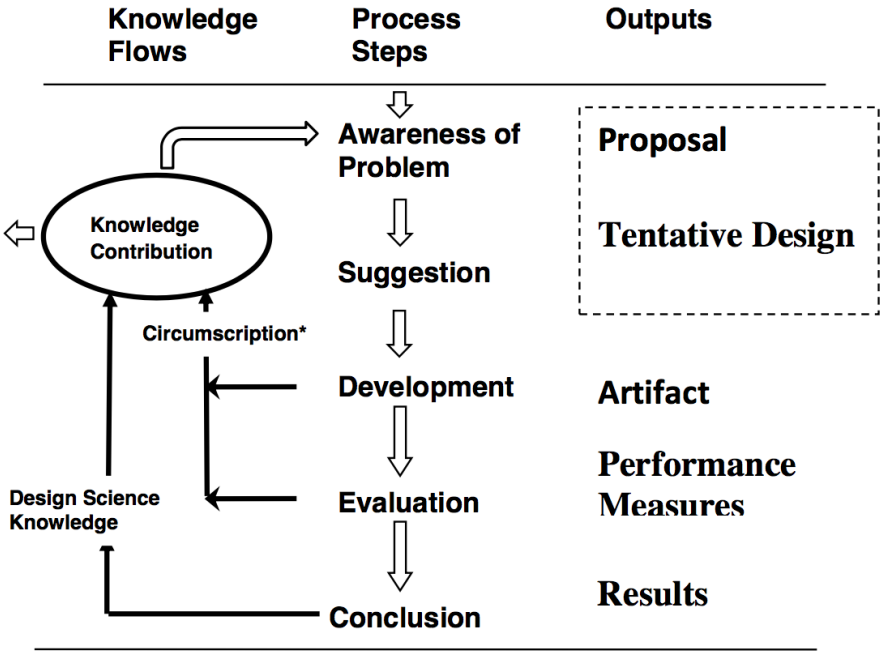


Figure 3.1: DSR iteration cycle [1]

3.2.1 Awareness of the problem

According to the guidelines by Hevner et al. [24], a DSR-based study should target a real-life issue. In our study, we focused on communication challenges through the software development life cycle. During the first iteration, we conducted an exploratory literature review to investigate communication challenges in requirement engineering and throughout the entire software development process. Additionally, we looked into proposed solutions for tackling these challenges using social network analysis, along with the challenges in integrating different tools in the software development life cycle. The network constructed in the first iteration was able to represent only a part of the whole picture of connections between people. This was due to using a small section of the data to construct the network. However, the results of the first iteration served as the source of awareness for the second iteration and helped shape the enhancements made to the approach in the second iteration.

3.2.2 Suggestion

During the suggestion step, based on the information gathered in the awareness of the problem step, an initial solution for the identified issue is proposed. In the first iteration, we based our suggestion on the findings by Mohamad et al. [12]. This included using social network analysis to help tackle communication challenges in software development, and applying the solution on integrating more than one tool used in the development process. Our suggestion in the first iteration was to build a tool that can automatically extract and build a social network based on

data produced by the different tools used in the software development process. The targeted data in our proposed solution would conform to the OSLC specifications for quality management and for change management. For the second iteration, our suggestion was to extend the artifact built in the first iteration to include more resources and relationships in the construction process of the social network.

3.2.3 Development

In this step, the proposed solution in the suggestion step is implemented. For the artifact in our first iteration, we developed a tool that extracts test cases and connects people who worked on these test cases with people who worked on change requests which are related to these test cases. The result network is then drawn using a graphing tool. In the second iteration, we extended the artifact to include requirement resources in addition to the resources included in the first iteration. We also developed two modes for using the tool. The first mode was constructing the social network of a project, starting with the requirements. The second mode was constructing the social network starting from a certain selected resource. We applied our approach using data produced by the Jazz team at [14], who publish their OSLC-compliant data to the community including quality management and change management data.

3.2.4 Evaluation

The evaluation step is very important in DSR according to Hevner et al. [24], as the developed artifact must be thoroughly examined in this step. The goal of this step is to asses the work done up until this point in the iteration, to build a base for decisions regarding future improvements. The intention of our first iteration was to establish feasibility of constructing a social network by automatically extracting data from two tools used in software development. This process was done manually on the data to determine the correctness of the constructed network. In the second iteration, evaluation for constructed networks was done by leveraging the opinions and expertise of practitioners in the field of software development. This was done through a survey (see Section 4.2) which presented the concept and showed some examples to investigate accuracy of the construction process of the network, as well as usefulness of the possible result.

3.2.5 Conclusion

Reaching this final step of the iteration implies reaching a satisfying output, regardless of whether it is the final solution for the problem or just an intermediate milestone in the DSR process. At the end of our first iteration, we had confirmed that the developed artifact produces the expected social network, and we formed an idea of what information to include within the nodes and edges of the constructed social network. After the second iteration, we had extended our approach to get a more realistic picture of the actual connections between people, and provided an additional point of view from which the social network is viewed when it is built

around a specific resource.

3.3 Validity Threats

In this section, we discuss threats of validity in this study. We base our categorization of threats on the classification scheme by Runeson et al. [25].

3.3.1 Construct Validity

In our quest to answer RQ1, we conduct a manual evaluation procedure to determine whether our procedure of automatically extracting data and building a social network out of it is correct. This could introduce bias into the evaluation process since the individuals carrying out the evaluation are the same individuals who developed the evaluated artifact. Additionally, it might not be sufficient to validate that the automatically constructed network is correct.

To reduce misunderstandings and misinterpretations in the survey invitation, we included a small introduction regarding the concept and approach, in addition to our goals and the issues we try to tackle. We also included clarification diagrams in the survey for our questions to increase clarity. However, we cannot guarantee that the participants read or understood all the information we provided in the invitation. Two of the participants of the survey contacted us to mention the difficulty in understanding the concept, and that the abstraction level in the questions was too high. We answered their questions and clarified all ambiguities for them. We believe that if other participants had faced the same issue, they would have raised it with us since there was a direct communication channel with them as we are coworkers. Additionally, we cannot claim that the clarification diagrams we provided are sufficient substitutes for social network diagrams that show real people and their connections, since seeing the real network might have encouraged more engagement and enthusiasm from the participants, which might have resulted in different answers.

3.3.2 Internal Validity

This study had the risk of not being able to validate the findings with practitioners in software development since there was no agreement with a certain company to provide their data or provide time for interviews with practitioners. As a result, we applied our approach to openly distributed data by the Jazz team (see Section 4.2). To validate our approach, we sent out a survey to the Jazz team, but we were unable to obtain any answers from them. Therefore, we used modified questions in terms of asking about the actual relationships between the investigated resources rather than mentioning OSLC, in a second survey which was sent to practitioners who are not in direct contact with the target data. This also resulted in our inability to present the actual constructed network to the people included in it. Hence, missing the opportunity to validate the accuracy of our results. This poses a threat to the validity of our validation process and thus to the research study in general.

Interpreting the results of the survey with the modified questions, then using the interpretation to reach conclusions regarding OSLC specific data, poses the threat of not obtaining accurate results regarding this specific data. This threat was mitigated by preserving the original OSLC-related concepts and relationships, which we intended to investigate in the first survey, while preparing the second survey as explained in Section 4.2.4

3.3.3 External Validity

This study was conducted on data extracted from the tools used by the Jazz team. While we believe that this approach would work in any software development setup where the used tools conform to OSLC specifications, we cannot guarantee it. There might be technical issues in extracting the data, or certain regulations within the environment that would prevent this for example. We try to reduce this threat by following the OSLC specifications, which would make the approach applicable in environments where OSLC is adopted.

Our decision to use OSLC means that the tool generated from this work will be restricted to integrating with tools producing data which conforms to the OSLC specification. This makes it difficult to verify the results with real company data if the tools used in that company do not conform to this specification. However, there is to our knowledge currently no alternative interoperability standard that would allow us to cover a larger proportion of existing company data. Using this approach in a company to solve the defined communication problems also assumes the use of such tools by software development teams in the company.

In the survey on which we base our results, we obtained results from 15 respondents, which is 31.9% of the 47 people to whom we sent the survey. This also proposes a threat to the validity and generalizability of our results which we cannot eliminate in this study.

3.3.4 Reliability

In our case, the data that was extracted from the tools used by the Jazz team was suitable to the study. The data followed the OSLC specifications to an extent, which allowed us to use social-related properties to construct the social network. This does not work in other environments with tools that do not produce data conforming to the OSLC standard. Furthermore, the OSLC specifications do not mandate using these properties, which means that it might be the case that a tool generates data without social-related properties and still "correctly" follow the OSLC specifications. In this case, our approach would not work.

4

Iterations

In this chapter, we present the iterations conducted in this research study. We discuss our findings in more detail in Chapter 5.

4.1 First Iteration

For the first iteration, we initially planned to start with dummy data conforming to the OSLC requirements management (RM) specification and to the OSLC quality management (QM) specification, but we were able to find publicly available data conforming to the OSLC specifications. This was data published by the Jazz team [14] for their different projects. The goal of this first iteration was to establish technical feasibility as a starting point for next iterations.

4.1.1 Awareness of the problem

At the beginning, we started by conducting a literature review on communication challenges in software engineering. This revealed a number of issues in this regard [5, 4], as well as the existence of gaps in communicating important project related information such as requirements [6, 17]. We then looked at how these issues were addressed using SNA, and what applications were offered by this approach, such as the study by Mohamad et al. [12], where social networks were built automatically for the purpose of helping tackling communication and collaboration issues in requirements and software engineering. This step helped us realize the potential that SNA has in tackling those communication challenges within and across software development teams, in addition to raise our awareness to the difficulties which arise while applying it in practice.

4.1.2 Suggestion

As discussed in Section 4.1, the proposed solution was to automatically construct a social network based on data published by the Jazz team for their projects, and then manually evaluate whether the result network correctly represents the relations in the data, to establish feasibility. We needed to determine which properties were related to people working on resources, and consequently define how connections will be built, in addition to what data to include in the nodes and edges of the

network.

4.1.3 Development

During our investigation of the OSLC specifications, we concluded that the social data that can be used in the construction process of the social network includes two core attributes, which are common in all OSLC resources. These are the dcterms:creator and the dcterms:contributor attributes. One thing to keep in mind is that, as mentioned in Section 2.3, it is difficult to predict how data following OSLC specifications is structured. This is due to the fact that many properties in OSLC specifications are not mandatory, including these two attributes.

According to Barmi et al. [26] and Bjarnason et al. [27], requirements and testing activities should be aligned to enable high product quality and efficient development. This gives the connections between these resources high importance regarding communicating requirements in software development. Therefore, we chose to start with the resources oslc_rm:Requirement and oslc_qm:TestCase. However, after extracting and examining QM data from the tools used by the Jazz team for the project "Rational Team Concert", we were unable to find the necessary properties which are supposed to link QM data to RM data. The missing connections that would be beneficial in our case are:

- A oslc_qm:TestPlan object has a zero-or-many relationship (called oslc_qm: validatesRequirementCollection) to a oslc_rm:RequirementCollection object
- A oslc_qm:TestCase object has a zero-or-many relationship (called oslc_qm: validatesRequirement) to a oslc_rm:Requirement object
- A oslc_qm:TestScript object has a zero-or-many relationship (called oslc_qm: validatesRequirement) to a oslc_rm:Requirement object

Since these relationships are zero-or-many, it is possible for these attributes to be absent in the data structure. This lead us to consider another option, which was to connect QM data with CM data instead of QM to RM data. The connections between test cases and change requests have high importance as well since a change request is essentially an addition to the requirements of the product, and can therefore be considered a requirement by itself. According to the OSLC QM specification, a test case may be connected to related change requests, and may specify which change requests are tested by it. Additionally, a change request may be connected to test cases which test it, and may also state related test cases. These connections are represented in Figure 4.1.

The test case and change request resources have multiple other connections, but we start by using the four connections mentioned in Figure 4.1 and plan to use additional relations and additional resources in future iterations.

The data we gathered for the "Rational Team Concert" project developed by the Jazz team contained values for these connections. Therefore we were able to imple-

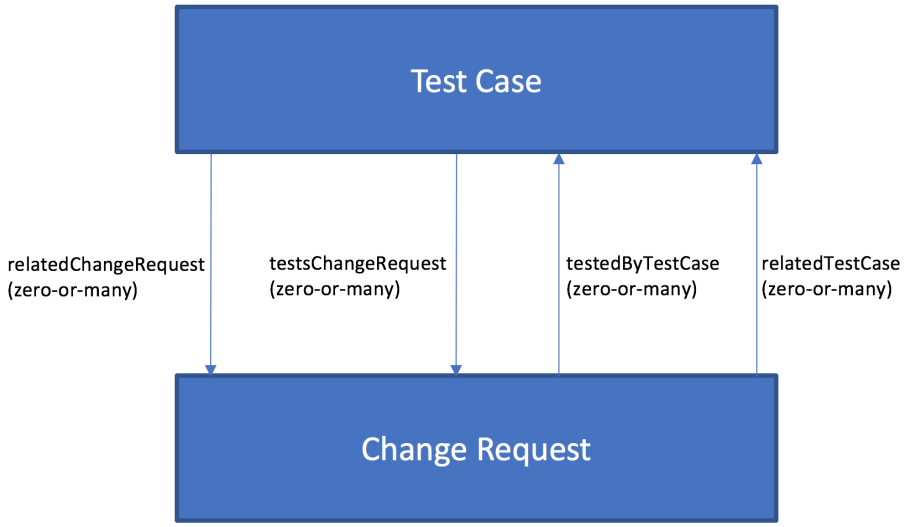


Figure 4.1: The Used Connections of Test Case and Change Request

ment the following procedure, all while remembering already visited nodes to avoid replications:

- Get all test cases in the project
- For each test case get related change requests
  - For each change request get test cases which test this change request, and get other related test cases
  - Link the original test case with the change request and the other test cases
  - Link the change request with the other test cases
- For each test case get change requests tested by it
  - For each change request get other test cases which test this change request, and get related test cases
  - Link the original test case with the change request and the other test cases
  - Link the change request with the other test cases

The result network is illustrated by nodes representing users, and directed edges which carry the following information:

- The length of the shortest path between the two connected nodes
- A list of all paths between the two connected nodes

4.1.4 Evaluation

The evaluation for this iteration was initially planned to be done manually on a subset of the data. However, after examining the targeted data, we observed that

only a subset of the change requests and use cases had data regarding the four relationships we used to construct the network. Therefore, we applied the manual construction of the network to all the available data, and then compared the result to the network that was automatically constructed via the developed tool. The two networks were identical. Hence, we were able to determine correctness of our approach.

4.1.5 Conclusion

At this point, after we established feasibility, we had gathered knowledge on how to automatically construct social networks from OSLC data. This was a solid ground for the next iteration, which would expand on the work in this iteration by including additional resources and properties in the construction procedure.

4.2 Second Iteration

Our goal in this second iteration was to extend the approach applied in the first iteration to portray a more accurate picture of the actual relationships between people. Therefore, we included an additional resource and several additional relationships between resources in the construction process of the social network. After that, we evaluated our approach by sending a survey to professionals in the software development domain.

4.2.1 Awareness of the problem

At the start of this iteration, we analyzed the work done in the previous iteration and concluded that the result network was unable to represent the full image of all connections between people. The reason was that we had included only two types of resources and four relationships in the construction process of the social network. Nevertheless, we had established correctness of the construction method, and therefore, we were able to proceed by extending the approach to include additional resources and relationships.

4.2.2 Suggestion

Our suggestion in this iteration was to expand the artifact built in the first iteration. Based on the work by Mohamad et al. [12], which showed high importance of building such a social network around a certain context, we suggested that the artifact should include two main functionalities: building the social network of a project, and building the social network surrounding a specific resource. This gives users the ability to choose whether they want to get an overall picture of a project or the specific set of people surrounding a certain requirement for example. On the one hand, building the social network for a project could be used to assist in organizing teams or arranging meetings, which would help with issues mentioned in [6] such as lack of sufficient communication channels. On the other hand, building the social

network around a specific resource could be beneficial for individual exploration, which would help with issues such as lack of context knowledge [6].

Since we have established feasibility in the first iteration, we were able to extend the approach to include the oslc_rm:Requirement resource and its relationships with change requests and test cases in the extraction process of the network. The oslc_rm:Requirement resource is vital in the software engineering development domain, which is why we decided to include it in this iteration.

4.2.3 Development

In this iteration, the enhanced tool provides two scenarios for building a social network: a social network around a whole project, and a social network around one specific resource, such as a requirement or a test case.

For the first scenario, we start by fetching all requirements in the system. From each requirement we move to the connected change requests and test cases to connect the people working on them. Then, we loop through all change requests that have not been included in the first step, and move along their connections to cover all resources in the project. We do not loop through the test cases since they must have been visited in the previous steps since each test case must be referenced in at least a requirement or a change request.

For the second scenario, we start by the selected resource and move to connected resources to connect the people working on them. The resources and relationships we included in both scenarios are:

- Resource: oslc_rm:Requirement
  - Relationship: oslc_rm: implementedBy, which points to a oslc_cm: ChangeRequest resource that implements this requirement.
    We utilized this relationship to include requirement traceability up to the implementation phase.
  - Relationship: oslc_rm: validatedBy, which points to a oslc_qm: TestCase resource that tests this requirement.
    This relationship was considered to include requirement traceability up to the testing phase.
- Resource: oslc_cm:ChangeRequest
  - Relationship: oslc_cm: implementsRequirement, which points to the oslc_rm: Requirement resource that this change request implements.
    We added this relationship since it is important to trace a change request back to the requirement it implements.
  - Relationship: oslc_cm: relatedChangeRequest, which points to another oslc_cm: ChangeRequest resource that is related to this change request.
    We included this relationship to ensure that related resources are included in the construction process.
  - Relationship: oslc_cm: testedByTestCase, which points to a oslc_qm:

TestCase resource that tests this change request.
This relationship was included to ensure that traceability between a change request and the use cases testing it is included.

- Resource: oslc_qm:TestCase
  - Relationship: oslc_qm: validatesRequirement, which points to the oslc_rm: Requirement resource that is validated by this test case.
    This relationship is important since it connects the test case to the requirement it is designed to test.
  - Relationship: oslc_qm: testsChangeRequest, which points to the oslc_cm: ChangeRequest resource that is tested by this test case.
    We utilized this relationship since the connection between a test case and the resource it test is important.

The paths between each pair of nodes have a specified maximum length, and no two identical paths are added to the network. A path includes an origin resource, a destination resource, all the resources between them, and all the relationships connecting these resources.

The result network might prove to be too big to read, which limits usefulness. To mitigate this, we implemented a filtering feature in the tool. Using this feature helps the user to set the abstraction level for the constructed social network, and gives a finer control over which parts of the data the user wishes to include in the construction process. We provided filters related to the created date (dcterms:created), the last modified date (dcterms:modified), and the keywords (dcterms:subject) related to the resources. This applies to all categories of resources since the attributes we use for these filters are core attributes, which are common in all OSLC resources. Additionally, we implemented a filter for change requests regarding the state of the change request. The attributes used in this filter are oslc_cm:closed, oslc_cm:inprogress, oslc_cm:fixed, oslc_cm:approved, oslc_cm:reviewed, oslc_cm:verified.

When we tried to apply this on the same data published by the Jazz team, we faced the issue that we were unable to find any requirements in the data. We believe that unlike quality and change management data, the requirement management data is either not exposed to the public, or is simply not used by the Jazz team. Therefore, we did not implement the first scenario which relied heavily on the existence of requirements data. Additionally, we did not implement building the social network around a specific requirement in the second scenario for the same reason. Furthermore, when we tried to build the social network in relation to a test case or a change request, we were faced by widely dispersed resources with minimal relationships to each other. Moreover, the Jazz team uses custom properties to define the relationships that we target instead of using the ones defined by the OSLC specifications, such as the attributes implementsRequirement and testedByTestCase for a change request resource, which are defined under the "http://jazz.net/xmlns/prod/jazz/calm/1.0/" namespace instead of using the same properties defined under the "http://open-services.net/xmlns/cm/1.0/" namespace. Therefore, we were unfortunately unable to apply the modified algorithm to the data and get results.

4.2.4 Evaluation

We prepared a survey targeted at professionals in the software development domain, such as requirement engineers, software developers, testers, project managers, business analysts, etc. who are in contact with OSLC based data and tools.

The survey started by introducing and explaining the concept of the approach and the goals behind the study. It included both quantitative (asking people to rate using a Likert scale from 1 to 10) and qualitative types of questions, which aimed at investigating the approach used to build the social network with the end goal of tackling communication challenges in the software development environment. We opted for feedback regarding the usefulness of including specific OSLC resources and relationships, in addition to the usefulness of the result network in several use cases and scenarios, which were investigated in a previous study by Mohamad et al. [12]. We also aimed at investigating what information to include in the constructed network along with the nodes and edges. Lastly, as the study by Mohamad et al. [12] showed potential ethical issues with such an approach, we included several questions in this regard in the survey. We were unable to investigate accuracy of the result network through the survey since we were unable to apply the approach during this iteration to actual data to obtain actual results.

The survey was sent through email to nineteen professionals in the Jazz team, who work with OSLC-based data. However, we received no responses for the survey, and since we had no access to them for interviews, we resorted to another approach for getting answers to our questions. We set up a mirroring survey which does not mention the OSLC specification specifically. Rather, we asked the same questions about the actual information, properties and relationships that are represented by specific OSLC resources and attributes. For example, instead of the two questions: "How useful do you think it is to include the oslc_qm:testsChangeRequest relationship between oslc_qm:TestCase and oslc_cm:ChangeRequest in the process of extracting connections between people?" and "How useful do you think it is to include the oslc_qm:relatedChangeRequest relationship between oslc_qm:TestCase and oslc_cm:ChangeRequest in the construction process?", we asked the question: "How useful do you think it is to include relations between test cases and change requests (such as testsChangeRequest and relatedChangeRequest) in the construction process?". By doing this, we were able to reflect the answers to the new questions on the OSLC-based data to answer our original questions.

This second survey, which is presented in Appendix B, was sent to 47 professionals in the software development domain, but who are not in contact with OSLC data and tools. These professionals were employees and consultants, who work with requirements, change requests and tests in the application management team at a company the author has access to. We decided to use this approach in order to be able to calculate a response rate instead of sending the survey out in an uncontrolled way. We received 15 responses for this survey, which is 31.9% of the 47 people to whom we sent the survey, and we were able to obtain the results we needed from this second survey. We present our findings and discuss them in Chapter 5.

4.2.5 Conclusion

At the end of this second iteration, we had expanded our approach to include a wider set of OSLC resources and relationships in the construction process of the social network. Additionally, we had also implemented several additional features in the artifact we built. Lastly, we had received feedback regarding the feasibility of our approach of using SNA to tackle communication issues in the software development environment. As a result, we were able to obtain answers for our research questions, which we present in detail in Chapter 5.

5

Results and Discussions

In this chapter, we present and discuss the results for each iteration. Then we discuss our research questions and their answers.

5.1 First Iteration

As mentioned in Section 4.1, we conducted a manual evaluation of correctness on the data. By analyzing our evaluation result, we concluded that the constructed social network is correct in terms of its building procedure, and that it reflects the connections discovered by extracting the data. This does not mean that the constructed social network reflects the actual relationships between people referenced in the data, since we only used a subset of the available attributes and connections. Figure 5.1 shows the social network constructed by the procedure done in the first iteration on the complete set of data. The nodes represent people with their user names as labels. The thickness of each edge represents its weight. The listing of the network and the GEXF format for it are presented in Appendix A.

The result network has 18 nodes and 31 edges, with an average weight of 1.55. The average number of connections per node was 1.7. It is important to mention that we observed that not all change requests and test cases had the mentioned connections filled. We obtained a total of 559 test cases, of which only 19 had connections to related change requests, and 23 had connections to the change requests which are tested by them. We believe that this low traceability between the test cases and their related change requests affects the accuracy of the built social network in terms of how it realistically reflects the real image.

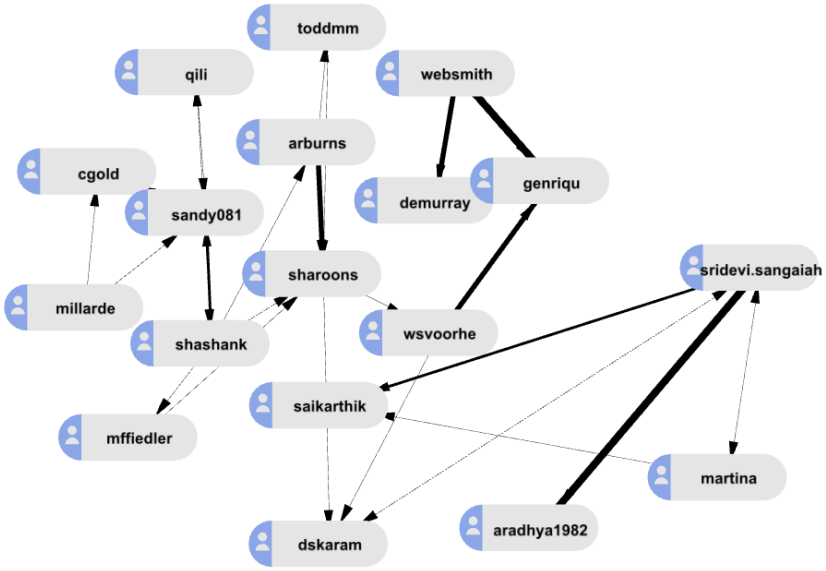


Figure 5.1: Social Network built in the first iteration for the Rational Team Concert project

At the end of this iteration, we had proposed several enhancements on the approach. Having noticed the low traceability between test cases and change requests, we decided to extend the procedure of extracting connections between people by including additional attributes and properties in the operation. We also found great need to validate the accuracy of the approach by presenting it and the result network to practitioners, and gather their input in this regard. The results of this iteration shaped and inspired our work in the second iteration.

5.2 Second Iteration

In this section, we present the tool that we built, as well as the results of the survey mentioned in Section 4.2.

5.2.1 The Result Artifact

The tool we built features two modes. The first mode, shown in Figure 5.2, allows the user to construct the social network for a whole project. The user must enter the URI for the project's requirements, and the URI for the project's change request. The tool then uses these URIs to loop through all requirements and change requests to extract relationships between people and construct the social network. This procedure does not require the URI for the project's test cases since we assume that a test case should not exist without it being connected to at least a requirement or a change request.

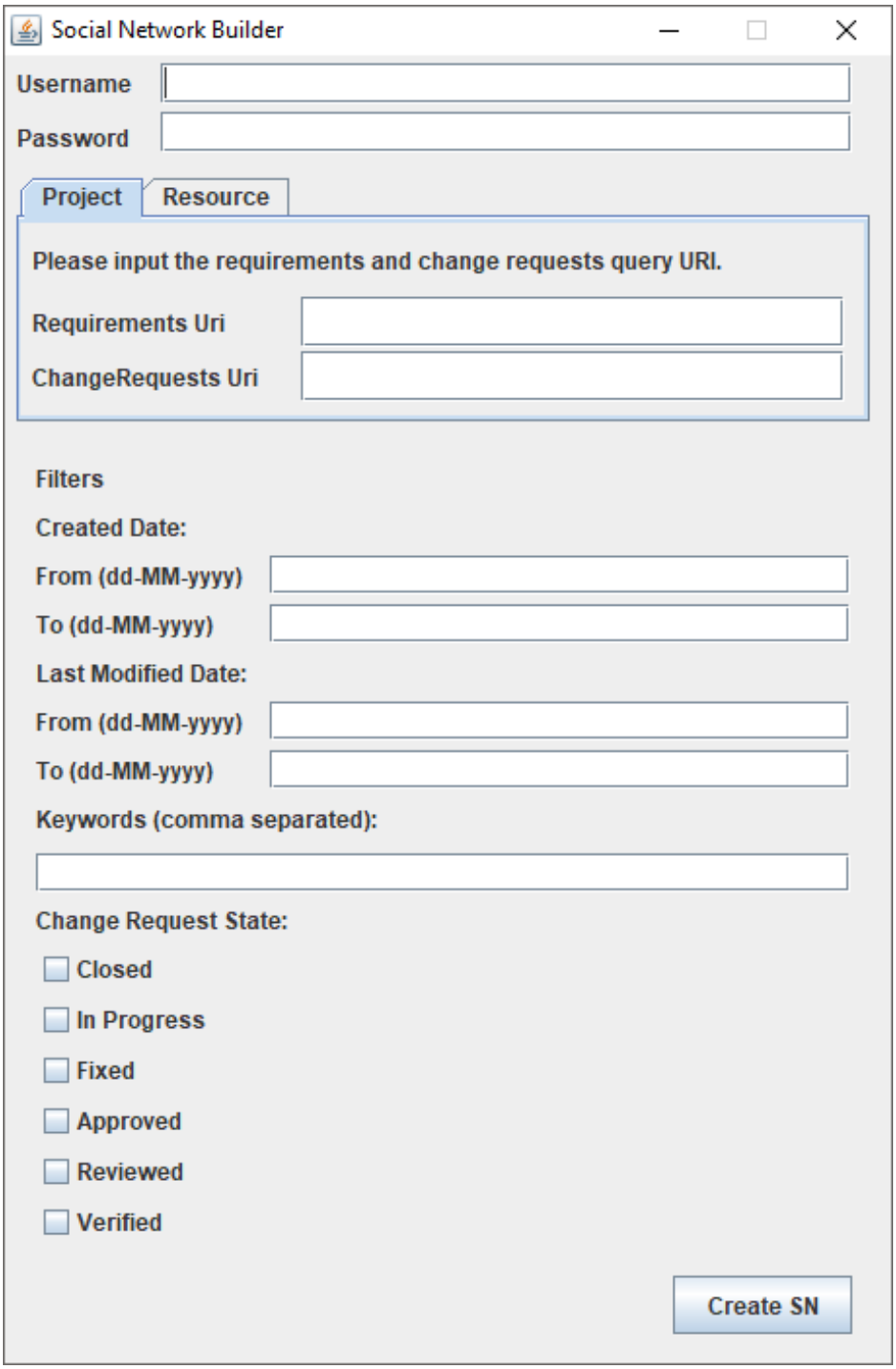


Figure 5.2: Building the social network for a project

The second mode, presented in Figure 5.3, gives the user the possibility to construct a social network surrounding a specific resource. The user selects the type of the resource and enters the URI for the resource. Then the tool constructs the social network related to that resource.

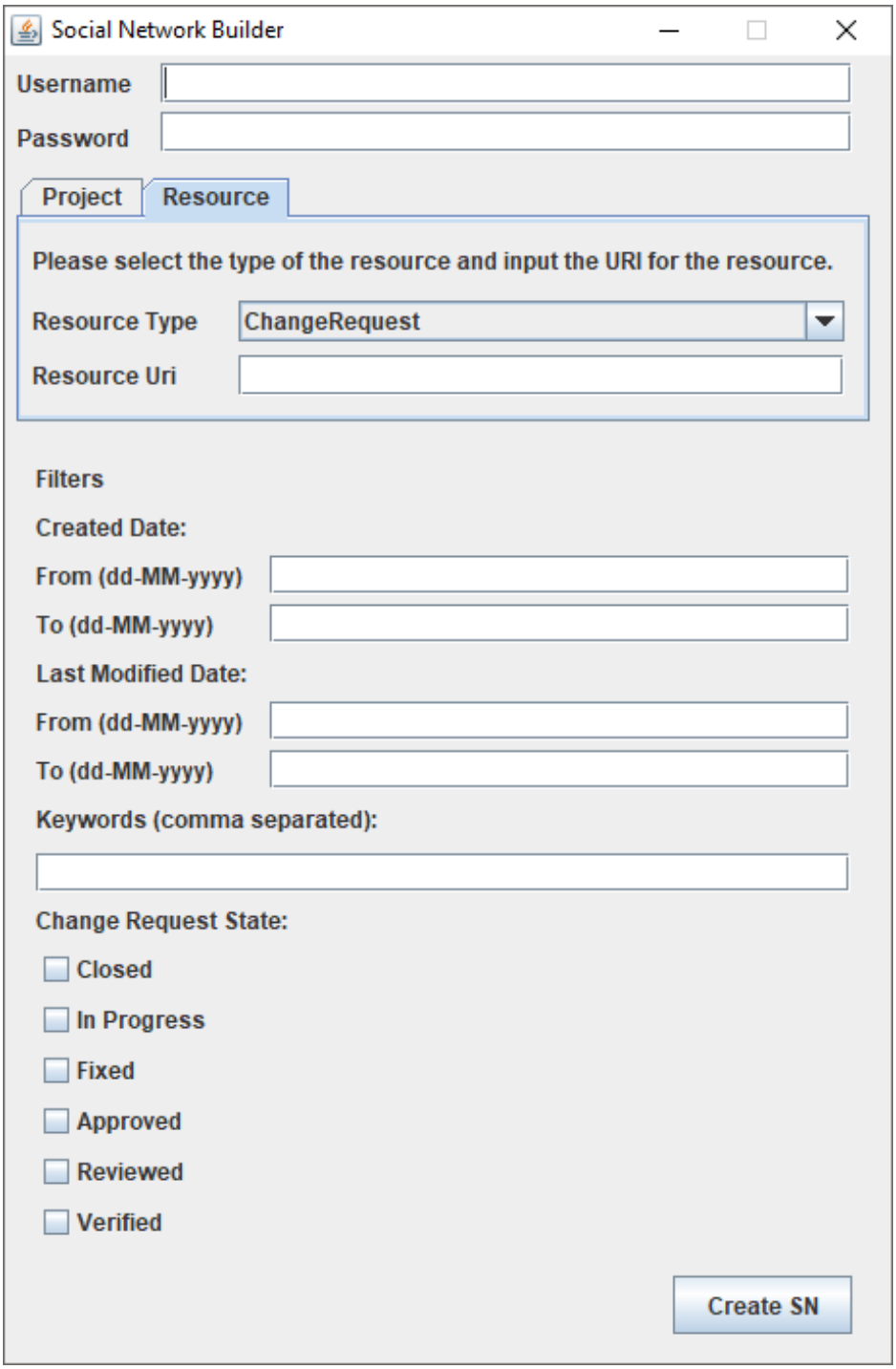


Figure 5.3: Building the social network for a resource

Once a network is constructed, the tool exports it as a file of the GEXF format, which also includes all the extra attributes that we added, such as the original connections between resources on which the social network was built.

5.2.2 Survey Results

We obtained results from 15 respondents out of the 47 people to whom we sent the survey. Five of the respondents had less than 20 years of experience (1.5, 3.5, 8, 9 and 16 years). Eight of the respondents had 20 or more years of experience (20, 20, 20, 21, 25, 25, 25 and 29 years). The remaining two respondents chose not to reveal their experience. The survey was divided into four sections, which are shown in the following subsections.

5.2.2.1 Usefulness of the Approach

The survey results show that the overall usefulness of the approach received an average score of 7 out of 10. Additionally, all the specified use cases received a mean value between 6.8 and 8.3 out of 10 for usefulness, which we believe is an indicator of usefulness of our approach. Figures 5.4a to 5.4e show box plots for usefulness of the approach for each use case.

The results in Figure 5.4a indicate usefulness of the approach for finding the most suitable person to talk to regarding a certain issue or requirement. The mean value here is 6.8 out of 10. Participants who gave a higher rating for this case, between 8 and 10, commented that this is good in larger projects involving many people, provided that the data is accurate. The participant who gave this case the lowest rating: 2, commented with the following:

> I'd go to the most connected person, but that might be wrong. It could help if the network is built regarding a certain component for example.

The participant indicated the potential issue of a heavily connected person in a graph related to a whole project for example, and this person would be the go-to person in many situations even if this person is not the most suitable for a certain task. The participant mentioned that showing a graph related to a specific component would help in this case.

Another participant, who gave this use case a rating of 5 out of 10, commented the following:

> For a company, it is not good to always point to the most suitable person. That person will be a risk, for instance if he or she is on vacation, or leaves the company. It is much better to strive to spread the knowledge. By having a team, where everyone in the team can solve the tickets assigned to the team, the knowledge is spread over many people, which will create a more flexible and stable company.

While the participant pinpointed an important issue regarding this use case, he or she indicated another potential use case for our approach. The result social network could help identify centrality of knowledge and information, as opposed to having

the knowledge shared across the company to reduce risks of losing or overwhelming resources.

The results in Figure 5.4b indicate usefulness of the approach for finding out who should work on a certain issue. The mean value here is 7 out of 10. Two participants, who gave a score of 7 and 8 out of 10, expressed the importance of showing people's competences and roles in the network to reduce the chance of it being misleading. Another participant, who gave a score of 5 out of 10, recommended implementing a way to find the most suitable team instead of the most suitable person for working on an issue.

The results in Figure 5.4c indicate usefulness of the approach for finding out who should be included in a certain email or meeting. The answers in this case ranged between 5 and 10 out 10, with a mean value of 7.2 out of 10. A participant who gave a score of 9 out of 10, commented that it is useful provided that the network is build around a specific component. Another participant who gave it a score of 10 out of 10 had the following comment:

> I think this is something that could be a great aid if it suggests recipients. If it could show the reason to why someone has been suggested it would also clarify to the user and thus he/she would trust the tools output.

The participant suggested that the tool shows the reason a person should be included in this case. This goes in hand with one of the features we implemented, which was to show the original relationships between data resources that resulted in two nodes being connected in the constructed social network.

One participant who provided a score of 7 out of 10, stated that this is a good start for a new employee, which goes in line with one application for SNA mentioned by Wolf et al. regarding new employees finding information regarding who works with a certain task and who is responsible for a certain component [8]. However, the participant also stated that this should not be used alone to determine who to include in an email or meeting. Another participant who gave it a score of 6 out of 10, stated that it is not normally a problem to find the right person to include in the email. Therefore the low score they provided was to indicate that the network is less likely to be used in such a case.

The participant who provided the lowest score in this case: 5 out of 10, commented the following:

> It is good to be reminded who could be interested in the area. But certain people tend to receive a lot of emails and a lot of invitations, so it should be used with caution and a good judgment.

This, albeit unexpected, presents the other side of communication issues. We had so far only mentioned lack of communication and communication gaps issues and how our approach aims to help solve those issues. However, using the social network we built might also result in inconveniences for people who are contacted for the wrong

reason. This could happen if the network shows that they are connected somehow to a certain component, while in reality, they could have moved between teams and therefore stopped being in direct connection with that component for example.

The results in Figure 5.4d indicate usefulness of the approach for finding out who will be affected by a certain change. This use case received mostly high scores, between 7 and 10 out of 10, with a mean value of 8.3. However the figure indicates an outlier in the results, which is the value 5 out of 10. The explanation provided by the participant who gave this answer was the following:

> Since the issue perhaps only will be communicated after it is resolved, I think that it will be harder to gather data which will say who will be affected. It is not clear to me how this would work with so little background information as is.

This indicates that the participant did not fully understand how our approach could be used for this purpose, which might be contributed to vagueness in the question, or to lack of explanation of the concept and used approach. This might cause a threat to the validity of the results, which is mentioned in Section 3.3.

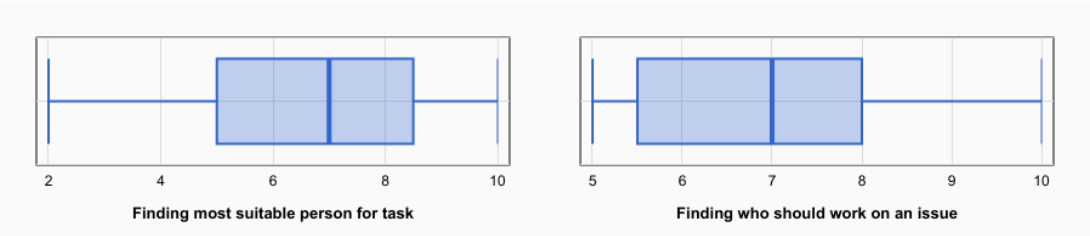
The results in Figure 5.4e indicate usefulness of the approach for identifying potential vulnerability of communication within the team or with stakeholders. This use case received a wide range of scores, between 3 to 10 out of 10, with a mean value of 7. A participant who gave it a score of 6 out of 10 commented the following:

> Maybe I am old fashioned, but I think communication requires certain skills, not just a system support. Or maybe I have completely misunderstood this question...

The participant clearly misunderstood the purpose of using our approach, which is to help identify and tackle communication issues. The participant here assumes that the tool would help in the communication process, which as he or she said requires certain skills. Another participant who gave it a score of 5 out of 10, commented the following:

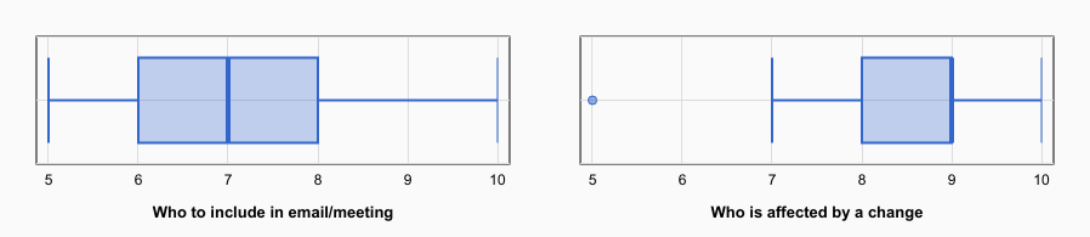
> Unclear to me how this would function. I do not have any answer.

This is another case where vagueness of the question or lack of explanation regarding the approach and its goals, could have affected the answers of the participants.



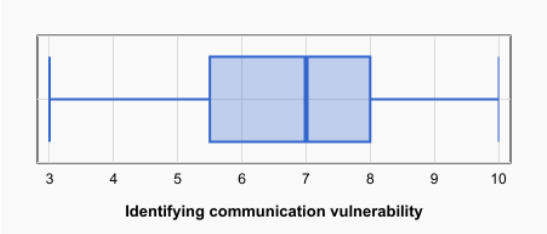
(a) Question: How useful do you think the network would be for finding the most suitable person to talk to regarding a certain issue or requirement?

(b) Question: How useful do you think the network would be for finding out who should work on a certain issue?



(c) Question: How useful do you think the network would be for finding out who you should include in a certain email or meeting?

(d) Question: How useful do you think the network would be for finding out who will be affected by a certain change?



(e) Question: How useful do you think the network would be for identifying potential vulnerability of communication within the team or with stakeholders?

Figure 5.4: Usefulness of the approach

5.2.2.2 Accuracy of the Construction Process

In this section, we investigate the usefulness of including resource connections in the construction process of the network, which gives an indicator of the accuracy of our construction process. The overall usefulness of mining social attributes based on relationships between resources to construct a social network received an average score of 6.9 out of 10. Additionally, all the types of relationships we asked the participants to evaluate received a mean value between 7 and 8 out of 10 for usefulness to include

in the construction process, which we believe is an indicator of high accuracy of our construction process. Figures 5.5a to 5.5d show box plots for usefulness of including each type of relationship.

The results in Figure 5.5a indicate usefulness of including relationships between test cases and change requests in the construction process of the social network. This received a range of scores between 3 and 10 out of 10, with a mean value of 7.

The results in Figure 5.5b indicate usefulness of including relationships between test cases and requirements in the construction process of the social network. This received a range of scores between 4 and 10 out of 10, with a mean value of 8. The results were shifted towards the higher end of the range.

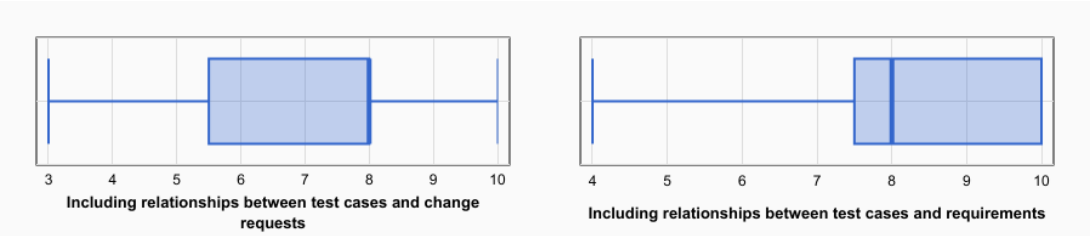
The results in Figure 5.5c indicate usefulness of including relationships between different change requests in the construction process of the social network. This received a range of scores between 5 and 10 out of 10, with a mean value of 7.2. Several participants indicated the importance of taking into consideration how these change requests are related. Additionally, one participant noted that it would be more useful if the change requests belong to different contexts or components. The context can be obtained through the dcterms:subject attribute, which holds keywords related to the resource. Therefore, it is possible to implement a solution using change requests that belong to different contexts to build the network.

The results in Figure 5.5d indicate usefulness of including relationships between change requests and requirements in the construction process of the social network. This received a range of scores between 5 and 10 out of 10, with a mean value of 8. The results were shifted towards the higher end of the range.

5.2.2.3 Tool Features

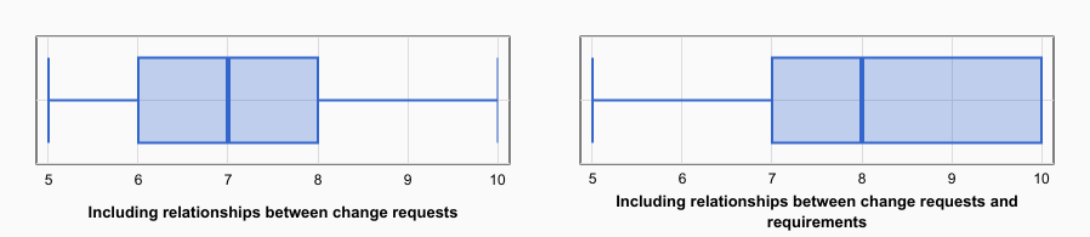
We asked the participants about the information they would like to see in the presented network. A total of 6 out of 15 respondents provided an answer for this question. The area of expertise and competence of a person was mentioned by 5 respondents. The importance of showing responsibilities of a person was also mentioned by 5 respondents. Additionally, organizational information, such as groups and teams a person belongs to was mentioned by 2 respondents. Lastly, the role of a person was mentioned by 2 respondents. This data cannot be obtained from OSLC-based data, which means that a connection with another system, such as an HR system, must be implemented. However, exposing such data in the network might not be acceptable by everyone, which adds to the ethical issues discussed in Section 5.2.2.4.

The results in Figure 5.6a indicate how important the participants think it is to see the number of paths connecting two people. This received a range of scores between 4 and 10 out of 10, with a mean value of 6.8.



(a) Question: How useful do you think it is to include relations between test cases and change requests (such as testsChangeRequest and relatedChangeRequest) in the construction process?

(b) Question: How useful do you think it is to include relations between test cases and requirements (such as validatesRequirement) in the construction process?



(c) Question: How useful do you think it is to include relations between change requests (such as relatedChangeRequest) in the construction process?

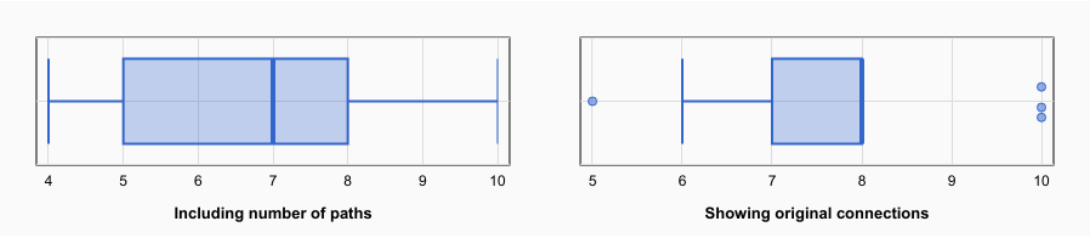
(d) Question: How useful do you think it is to include relations between change requests and requirements (such as implementsRequirement and affectsRequirement) in the construction process?

Figure 5.5: Accuracy of the Construction Process

The results in Figure 5.6b indicate how important the participants think it is to see the original connections between resources to know how two people are connected. This received a range of scores between 5 and 10 out of 10, with a mean value of 7.6.

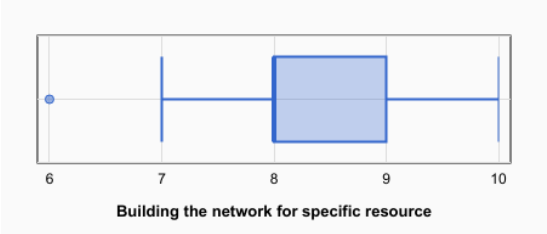
The results in Figure 5.6c indicate how important the participants think it is to view the network of people involved with a certain resource. This received a range of scores between 6 and 10 out of 10, with a mean value of 8.2. We can see from the plot that the answer with value 6 is an outlier, but the participant did not provide any comment for this answer.

These results show that it was important for the participants to have a resource related network where the focus is on a certain context. However, although knowing how heavily people are connected or why they were connected in the first place scored 6.8 and 7.6 out of 10 respectively, they were not as important as per the participants.



(a) Question: How useful do you think it is to see the number of paths connecting two people?

(b) Question: How useful do you think it is to see the original connections between resources to know how two people are connected?



(c) Question: How useful do you think it is to view the network of people involved with a certain resource?

Figure 5.6: Tool Features

5.2.2.4 Ethics Related Issues

The majority of the respondents accept being represented in the social network. Figure 5.7a shows that 86.7% of the participants said yes, while 6.7% said no and 6.7% said maybe.

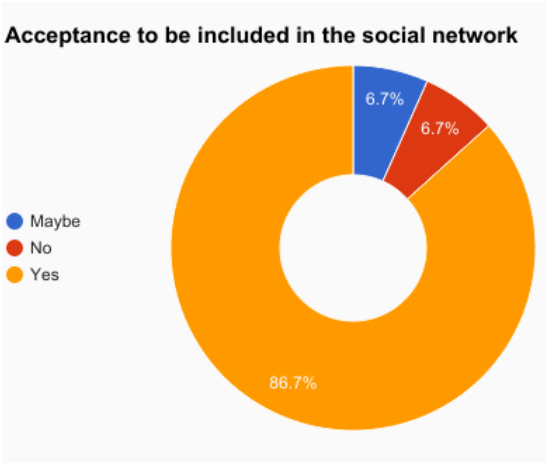
Figure 5.7b shows that 46.7% of the participants said that the social network could be misused in the company, while 53.3% said that it could not. The answers we obtained for the question regarding how it could be misused are:

> It is always easiest to find the answer from the expert. But that is bad for the company, in the long run. It is not a sustainable way of working. It is better let a team have a cooperative responsibility.
>
> Could be used in performance reviews by a manager for example, which is misleading. It could also be misleading if someone does something and the data doesn't allow for recording this type of work.
>
> Could be used to impose who someone should talk to even if there was no need. It should be used to help if someone is not as connected as they should be instead of having them feel bad about it. From experience, I don't think people would misuse it intentionally. It could be different between companies and cultures though.

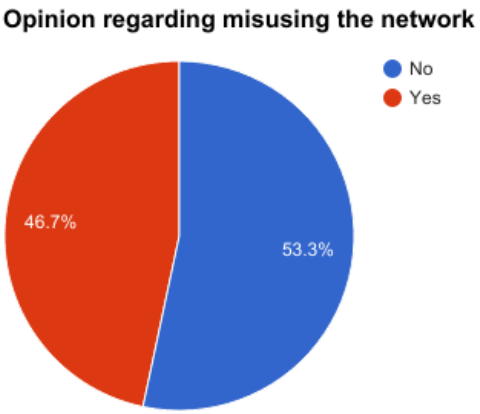
> Assessment by a manager. Could be misleading if one person might be working on big projects and another on small projects.
>
> Knowledge is the base for any kind of misuse. The difference is if the person has intent.

The answers point out the fact that there is a human factor that plays a great role when misusing the social network. This cannot be neglected or mitigated through our approach. A potential case for misusing the network is using it in employee assessments by a manager for example, which could be misleading. Another case where the network could be misused is trying to force communication between people even if there was no actual need, just because they were not as connected as others for example. In addition to that, an issue regarding a side effect of using the network was raised, which was the potential for knowledge to be with one person who is always being asked for example instead of encouraging knowledge transfer to a group of people within the team.

One way to overcome these issues would be to use a query system instead of showing a visual representation of the constructed network. A user can query the system for the person most suitable instead of requesting to view the network and finding the right person through visual representation. This would for example allow the system to propose people who are not the most connected but still suitable for the task, and thus distribute the load.



(a) Question: Would you accept to be represented in such a graph among your colleagues?



(b) Question: Do you think that the presented network and associated information could be misused within your company by different users?

Figure 5.7: Ethics Related Questions

5.3 Answers to Research Questions

In this section, we discuss our findings in relation to our research questions.

RQ1 - How can social networks be built based on data conforming to the OSLC specifications and generated by different tools used in the software development process?

As described in Chapter 3, we used the DSR methodology in our study, which allowed us to create the desired tool and evaluate it in the first iteration, then enhance and extend it in the second iteration. The approach gives the opportunity for evaluation during each iteration to build upon that in following iterations, which provides the potential of creating the desired artifact with high quality.

This question was only partially answered in the first iteration. We investigated which properties in OSLC specifications relate to people, and can therefore be used to build the social network. We concluded that the social properties that are helpful for this purpose are dcterms:creator and dcterms:contributor, which are common among OSLC resources. However, this does not guarantee that these attributes would have values in all resources since they are allowed by the OSLC specification to be empty. Our approach requires these values to be filled. In addition to that, our approach requires that relationships between the different resources are registered in the data, so that our tool can extract connections between people based on these relationships between the resources.

As discussed in Section 4.1, we started in the first iteration to look into what information to include with each node and edge in the constructed network. For the nodes, we showed the name of the person represented by the node. For the edges, we showed the length of the shortest path between the two connected nodes, and a list of all paths between the two connected nodes. Our investigation helped define a starting point for building the social network, in terms of which data to include and which OSLC attributes to consider during the construction procedure.

During the second iteration, we expanded the included set of resources and properties in the social network construction process. We added an additional resource: the oslc_rm:Requirement. Additionally, we included several resource relationships in the construction process. However, as mentioned in Section 4.2, we were unable to apply the modified algorithm to the data related to the projects of the Jazz team. The reason for this was that we were unable to find any requirements in the data. Furthermore, we were unable to construct the social network around a specific test case or change request, since the data was widely dispersed with minimal relationships between resources, in addition to the fact that the Jazz team uses custom properties to define some of the relationships that we target instead of using the ones specified by OSLC. As mentioned in Section 2.3, the OSLC specification is intentionally kept open to interpretation [21]. This leads us to conclude that applying our approach in a real workplace environment requires customization and adaptation to the way OSLC is implemented in that environment.

Another thing to keep in mind is that according to Conway's law, the structure of the data should reflect the structure of the organization [28]. Therefore if an organization has a hierarchical structure, we expect to find a similar structure in

the data, such as having a high level requirement for example, which would be connected to all its sub-requirements and all the resources connected to these sub-requirements. On the contrary, if the organization has a flatter structure, data might not be as heavily or deeply connected. Therefore, the constructed network for a specific resource could be very deep or very shallow, depending on the structure of the organization and the data. This might in turn affect how useful the resulting network is to find communication channels.

Our approach relies on specific OSLC-data being present in the company, which might not be the case in reality according to Wohlrab et al. [11], who argue that implementing traceability between resources in real-life situations faces many challenges; and Bjarnason et al. [15], who study challenges related to aligning requirements with verification and validation. However, obtaining useful results from this approach does not require full traceability information to be present. Depending on availability of traceability information, the approach could serve in resolving certain communication issues mentioned in Section 2.1. For example, having traceability information on different interdisciplinary levels would help with the issue related to lack of common understanding across multiple disciplines [6]. Other examples include tackling the issue of separation between different abstraction levels [6], which requires the availability of links connecting resources between these levels; and tackling the issue related to lack of context information regarding requirements on different abstraction levels [6] , which requires links in the data across these levels. We believe that adapting this approach in a company would encourage addressing these challenges mentioned by Wohlrab et al. and Bjarnason et al. [11, 15], using methods such as "REST-bench" defined by Unterkalmsteiner et al. [16]. Thus, introducing improvements to the way requirements are traced and used in the SDLC, and by that enhancing the chances of succeeding in the software development endeavour as per Hofmann and Lehner [2] and Voss [3].

RQ2 - To what extent can these networks aid in identifying gaps in communication among team members and across teams in industry?

Ideally, an experiment at a company would have been conducted to answer this question. However, since the time we had was limited, and since we had no agreement with a certain company for cooperation, this was not an option for us. Therefore, we opted for a more practical solution in our case, which was to send out a survey as described in Section 4.2.4.

Our survey regarding evaluation of the approach was conducted in the second iteration. We aimed in the survey at answering the sub-questions of RQ2. The survey helped investigate how valuable practitioners think the approach is in tackling communication issues in the software development environment. It also helped explore which features they think would be useful to include in the tool.

In the survey, we asked several questions regarding what information to attach to the nodes and edges in the constructed network. Our results showed that the network should show a person's area of expertise, competence, responsibilities and

role. Additionally, organizational information such as which team and groups to which a person belongs was suggested to be included for each node in the network. As for the edges, viewing the original connections between resources that lead to the result social network was considered a good idea by the participants. Moreover, the participants showed high interest in viewing the network built around a certain resource.

As mentioned in Section 5.2, the survey respondents gave positive feedback regarding the different cases that we presented for using the social network, which goes in line with the findings by Mohamad et al., Cross et al. and Wolf et al. [12, 19, 8] among others mentioned in Section 2.2. These use cases help tackle several of the communication challenges mentioned in Section 2.1. Including information regarding competences, roles and responsibilities in the result social network, as well as constructing the social network around a specific resource, provides the possibility to easily find the proper contact person in a given context. Even though this adds to the technical and ethical challenges as mentioned in Section 5.2.2.3, it helps with several issues uncovered by Liebel et al. [6] regarding lack of context information, unclear responsibilities and borders, and lack of product knowledge. It also helps with communication issues in large projects and organizations mentioned by Al-Rawas and Easterbrook [17], such as weakness in communication which would occur naturally in distributed and large software development setups. Additionally, it helps tackle the issue uncovered by Curtis et al. [5] who argue that it is less likely for people to participate in communication activities if the path, which information has to travel for communication channels to be established, is long. An example case is that if a tester needs clarification of a certain requirement, the social network would help find related people in this context and their responsibilities and roles to help the tester find the person to talk to. The network does not indicate that one person should communicate with another. Rather, it shows possibilities in terms of communication paths within specified context.

Another use case for these social networks is helping discover communication gaps studied by Bjarnason et al. [4], which are related to the size of the organization and complexity of its products. To achieve this, other forms of communication-related data, such as comments and emails, must be mined and extracted to build the actual picture of communication between people, as in the work by Wolf et al. [8] and Damian et al. [18], keeping in mind that not all forms of communication is recorded as data, such as the case of verbal communication such as face-to-face communication and water-cooler chats. The result is then compared with the result network from applying our approach to discover these gaps in communication. In this sense, our approach provides a foundation for such a use case, since the result networks from our approach do not show actual communication between people. Rather, they provide a view on how the existing data says people are connected based on the connections between resources related to them.

6

Conclusion and Future Work

In this study, we aimed to tackle the communication challenges found by other researchers in the software development environment. Our proposal was to automatically build and visualize social networks, based on existing software development data, to provide a useful tool to be used in social network analysis for identifying communication issues and overcoming them.

We conducted a design science study, in which we conducted two iterations where we developed and evaluated a tool for automatically constructing social networks by mining social attributes and connecting people based on relationships between the resources that they work with. The data from which we extract the social network follows the OSLC specification, which provides great ease of integrating tools and removes the reliance on specific tools.

To evaluate our approach, we built the social network using OSLC-based data published by the Jazz team for their projects, to establish feasibility and applicability in a real development environment. Our investigation lead us to conclude that OSLC-based data is highly customizable in terms of properties and relationships between resources. This imposes the need for specific customization of the approach based on the chosen adaptation of the OSLC specification in the organization.

Additionally, we conducted a survey aimed at professional practitioners in the software development domain to evaluate usefulness and accuracy of the construction process, as well as the ethical side of the approach. The answers we obtained from participating practitioners confirmed usefulness of the approach in identifying and tackling communication issues within the software development environment, and accuracy of the construction process of the social networks. The respondents suggested presenting specific additional data in the visualization of the social network to solidify usefulness of the networks in everyday situations. The survey also revealed some use cases in which the social network could be misused by people.

In our study, we built upon the approach by Mohamad et al. [12]. We generalized the approach to using data from three specifications: the OSLC Requirement Management Specification, the OSLC Change Management Specification, and the OSLC Quality Management Specification. However, we discovered that even while using such specifications, certain customizations must be implemented to apply the ap-

proach in a real-life situation. This is due to the fact that the OSLC specification is kept open intentionally. We also validated the approach in a different company than the company targeted by Mohamad et al. [12], where we obtained proof that such an approach for tackling communication issues is promising. Lastly, we confirmed the ethical issue discovered by Mohamad et al. [12] regarding using the result social network in employee assessments by a manager, and discovered new issues, such as forced communication between people without need and the tendency to overload individuals with tasks when they are more heavily connected than others.

Although we expanded our social network construction algorithm during the second iteration to include more resources and relationships, we still do not cover all available resources and relationships provided by the OSLC specification. This approach should be extended to include these resources and relationships to get an image closer to the true situation. Additionally, we encourage further investigation into the meaning of the directed edges in the graph representing the social network, as well as what the length of a path represents. Lastly, there is great need to investigate whether the constructed social network is accurate by comparing it with actual relationships between people in a real environment.

Bibliography

[1] V. Vaishnavi and W. Kuechler, "Design Research in Information Systems," Jan. 2004.

[2] H. Hofmann and F. Lehner, "Requirements engineering as a success factor in software projects," *IEEE Software*, vol. 18, no. 4, pp. 58–66, 2001.

[3] C. A. Voss, "Determinants of success in the development of applications software," *Journal of Product Innovation Management*, vol. 2, no. 2, pp. 122 – 129, 1985.

[4] E. Bjarnason, K. Wnuk, and B. Regnell, "Requirements are slipping through the gaps - a case study on causes amp; effects of communication gaps in large-scale software development," in *2011 IEEE 19th International Requirements Engineering Conference*, Aug 2011, pp. 37–46.

[5] B. Curtis, H. Krasner, and N. Iscoe, "A field study of the software design process for large systems," *Commun. ACM*, vol. 31, no. 11, pp. 1268–1287, Nov. 1988.

[6] G. Liebel, M. Tichy, E. Knauss, O. Ljungkrantz, and G. Stieglbauer, "Organisation and communication problems in automotive requirements engineering," *Requirements Engineering*, 2016.

[7] K. Ehrlich, G. Valetto, and M. Helander, "Seeing inside: Using social network analysis to understand patterns of collaboration and coordination in global software teams," in *International Conference on Global Software Engineering (ICGSE 2007)*, Aug 2007, pp. 297–298.

[8] T. Wolf, A. Schröter, D. Damian, L. D. Panjer, and T. H. Nguyen, "Mining task-based social networks to explore collaboration in software teams," *IEEE Software*, vol. 26, no. 1, pp. 58–66, 2009.

[9] I. Kwan, A. Schroter, and D. Damian, "Does socio-technical congruence have an effect on software build success? a study of coordination in a software project," *IEEE Transactions on Software Engineering*, vol. 37, no. 3, pp. 307–324, 2011.

[10] W. Zhang, V. Leildé, B. Møller-Pedersen, J. Champeau, and C. Guychard, "Towards tool integration through artifacts and roles," in *2012 19th Asia-Pacific*

Software Engineering Conference, vol. 1, Dec 2012, pp. 603–613.

[11] R. Wohlrab, J. P. Steghöfer, E. Knauss, S. Maro, and A. Anjorin, "Collaborative traceability management: Challenges and opportunities," in *2016 IEEE 24th International Requirements Engineering Conference (RE)*, Sept 2016, pp. 216–225.

[12] M. Mohamad, G. Liebel, and E. Knauss, "Loco coco: Automatically constructing coordination and communication networks from model-based systems engineering data," *Information and Software Technology (to appear)*, 2017.

[13] OSLC. (2017) Open services for lifecycle collaboration. [Online]. Available: http://open-services.net/

[14] Jazz. (2017) Jazz community site. [Online]. Available: https://jazz.net/

[15] E. Bjarnason, P. Runeson, M. Borg, M. Unterkalmsteiner, E. Engström, B. Regnell, G. Sabaliauskaite, A. Loconsole, T. Gorschek, and R. Feldt, "Challenges and practices in aligning requirements with verification and validation: a case study of six companies," *Empirical Software Engineering*, vol. 19, no. 6, pp. 1809–1855, Dec 2014. [Online]. Available: https://doi.org/10.1007/s10664-013-9263-y

[16] M. Unterkalmsteiner, T. Gorschek, R. Feldt, and E. Klotins, "Assessing requirements engineering and software test alignment—five case studies," *Journal of Systems and Software*, vol. 109, pp. 62 – 77, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0164121215001508

[17] A. Al-Rawas and S. Easterbrook, "Communication problems in requirements engineering: a field study," 1996.

[18] D. Damian, I. Kwan, and S. Marczak, *Requirements-Driven Collaboration: Leveraging the Invisible Relationships between Requirements and People.* Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 57–76.

[19] R. Cross, S. P. Borgatti, and A. Parker, "Making invisible work visible: Using social network analysis to support strategic collaboration," *California Management Review*, vol. 44, no. 2, pp. 25–46, 2002.

[20] Crystal. (2017) Crystal. [Online]. Available: http://www.crystal-artemis.eu

[21] A. Leitner, B. Herbst, and R. Mathijssen, "Lessons learned from tool integration with oslc," *Communications in Computer and Information Science*, pp. 242–254, 2016.

[22] M. Elaasar and A. Neal, *Integrating Modeling Tools in the Development Lifecycle with OSLC: A Case Study.* Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 154–169. [Online]. Available: http://dx.doi.org/10.1007/

978-3-642-41533-3_10

[23] B. K. Aichernig, K. Hörmaier, F. Lorber, D. Nickovic, R. Schlick, D. Simoneau, and S. Tiran, "Integration of requirements engineering and test-case generation via oslc," in *2014 14th International Conference on Quality Software*, Oct 2014, pp. 117–126.

[24] A. R. Hevner, S. T. March, J. Park, and S. Ram, "Design science in information systems research," *MIS Quarterly*, vol. 28, no. 1, pp. 75–105, 2004.

[25] P. Runeson, M. Host, A. Rainer, and B. Regnell, *Case study research in software engineering: Guidelines and examples*. John Wiley & Sons, 2012.

[26] Z. A. Barmi, A. H. Ebrahimi, and R. Feldt, "Alignment of requirements specification and testing: A systematic mapping study," *2011 IEEE Fourth International Conference on Software Testing, Verification and Validation Workshops*, 2011.

[27] E. Bjarnason, P. Runeson, M. Borg, M. Unterkalmsteiner, E. Engström, B. Regnell, G. Sabaliauskaite, A. Loconsole, T. Gorschek, and R. Feldt, "Challenges and practices in aligning requirements with verification and validation: a case study of six companies," *Empirical Software Engineering*, vol. 19, no. 6, pp. 1809–1855, 2013.

[28] M. E. Conway, "How do committees invent?" *Datamation*, April 1968. [Online]. Available: http://www.melconway.com/research/committees.html

Appendices

A

Social Network Example

A.1 Listing of nodes and edges in the social network for the Rational Team Concert project

Graph:
Node 13 with label websmith
Edge 20 to genriqu, weight 4
Edge 21 to demurray, weight 3

Node 2 with label saikarthik
Edge 0 to martina, weight 1

Node 9 with label cgold

Node 11 with label dskaram
Edge 17 to sridevi.sangaiah, weight 1

Node 4 with label shashank
Edge 28 to sharoons, weight 1
Edge 30 to arburns, weight 1
Edge 2 to qili, weight 1
Edge 27 to mffiedler, weight 1
Edge 3 to sandy081, weight 2

Node 17 with label mffiedler
Edge 29 to sharoons, weight 1

Node 0 with label wsvoorhe
Edge 19 to genriqu, weight 3
Edge 15 to dskaram, weight 1

Node 14 with label demurray
Edge 23 to genriqu, weight 1

Node 10 with label sharoons
Edge 14 to dskaram, weight 1

Edge 13 to wsvoorhe, weight 1

Node 1 with label genriqu
Edge 22 to demurray, weight 3

Node 7 with label sridevi.sangaiah
Edge 9 to saikarthik, weight 2
Edge 16 to dskaram, weight 1
Edge 26 to aradhya1982, weight 4
Edge 7 to martina, weight 1

Node 12 with label arburns
Edge 18 to sharoons, weight 3
Edge 24 to toddmm, weight 1

Node 15 with label toddmm
Edge 25 to sharoons, weight 1

Node 5 with label qili
Edge 5 to shashank, weight 1
Edge 4 to sandy081, weight 1

Node 16 with label aradhya1982

Node 3 with label martina
Edge 1 to saikarthik, weight 1
Edge 8 to sridevi.sangaiah, weight 1

Node 8 with label millarde
Edge 11 to cgold, weight 1
Edge 10 to sandy081, weight 1

Node 6 with label sandy081
Edge 12 to cgold, weight 1
Edge 6 to shashank, weight 2

A.2 GEXF file representing the social network for the Rational Team Concert project

<?xml version="1.0" encoding="UTF-8" standalone="yes"?> <gexf version="1.2">
<meta lastmodifieddate="2017-04-12">
<creator>Wissam</creator>
<description>graph representing a social network based on test cases and related change requests for the Rational Team Concert project</description>
</meta>
<graph mode="static" defaultedgetype="directed">
<nodes>

```
    <node id="13" label="websmith"/>
    <node id="2" label="saikarthik"/>
    <node id="9" label="cgold"/>
    <node id="11" label="dskaram"/>
    <node id="4" label="shashank"/>
    <node id="17" label="mffiedler"/>
    <node id="0" label="wsvoorhe"/>
    <node id="14" label="demurray"/>
    <node id="10" label="sharoons"/>
    <node id="1" label="genriqu"/>
    <node id="7" label="sridevi.sangaiah"/>
    <node id="12" label="arburns"/>
    <node id="15" label="toddmm"/>
    <node id="5" label="qili"/>
    <node id="16" label="aradhya1982"/>
    <node id="3" label="martina"/>
    <node id="8" label="millarde"/>
    <node id="6" label="sandy081"/>
</nodes>
<edges>
    <edge id="20" source="13" target="1" weight="4"/>
    <edge id="21" source="13" target="14" weight="3"/>
    <edge id="0" source="2" target="3" weight="1"/>
    <edge id="17" source="11" target="7" weight="1"/>
    <edge id="28" source="4" target="10" weight="1"/>
    <edge id="30" source="4" target="12" weight="1"/>
    <edge id="2" source="4" target="5" weight="1"/>
    <edge id="27" source="4" target="17" weight="1"/>
    <edge id="3" source="4" target="6" weight="2"/>
    <edge id="29" source="17" target="10" weight="1"/>
    <edge id="19" source="0" target="1" weight="3"/>
    <edge id="15" source="0" target="11" weight="1"/>
    <edge id="23" source="14" target="1" weight="1"/>
    <edge id="14" source="10" target="11" weight="1"/>
    <edge id="13" source="10" target="0" weight="1"/>
    <edge id="22" source="1" target="14" weight="3"/>
    <edge id="9" source="7" target="2" weight="2"/>
    <edge id="16" source="7" target="11" weight="1"/>
    <edge id="26" source="7" target="16" weight="4"/>
    <edge id="7" source="7" target="3" weight="1"/>
    <edge id="18" source="12" target="10" weight="3"/>
    <edge id="24" source="12" target="15" weight="1"/>
    <edge id="25" source="15" target="10" weight="1"/>
    <edge id="5" source="5" target="4" weight="1"/>
    <edge id="4" source="5" target="6" weight="1"/>
    <edge id="1" source="3" target="2" weight="1"/>
```

```
        <edge id="8" source="3" target="7" weight="1"/>
        <edge id="11" source="8" target="9" weight="1"/>
        <edge id="10" source="8" target="6" weight="1"/>
        <edge id="12" source="6" target="9" weight="1"/>
        <edge id="6" source="6" target="4" weight="2"/>
      </edges>
   </graph>
</gexf>
```

B

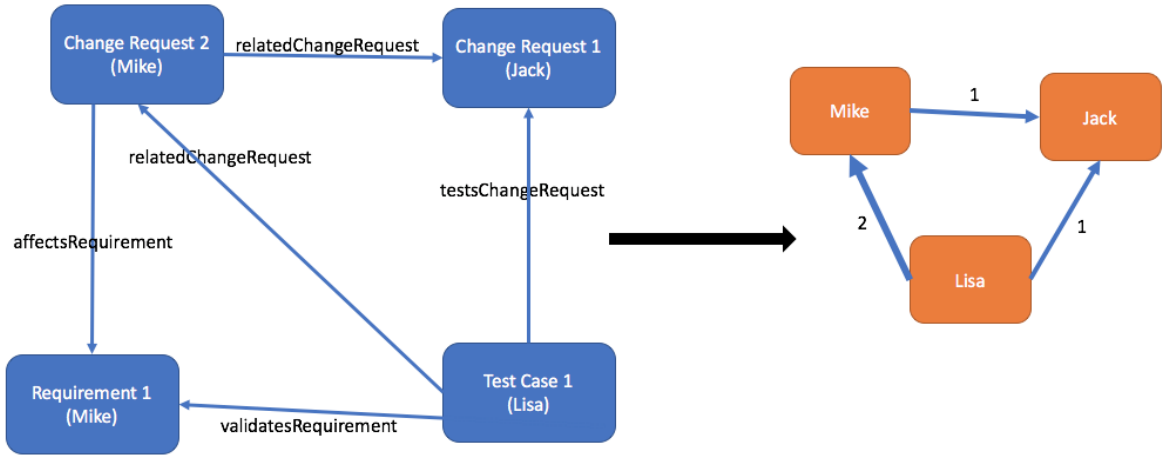
Survey

Constructing Social Networks from Software Development Data

This survey is part of a master's thesis project in software engineering at Chalmers University of Technology. In our study, we try to tackle communication issues in the development environment through social network analysis. To achieve that, we look through data produced by all the different tools used in the environment, such as requirements, change requests and test cases. Then we extract connections between people based on the social attributes, such as creator and contributor, which are available within the data. An example is shown in the image below.
This survey should take between 15 to 20 minutes.

* Required

Example of extracting the social network from the data



The following three questions are optional

1. **Company Name**

2. **Role in Company**

3. **Work Experience in Years**

Skip to question 13.

Accuracy of the Approach

In this section we investigate how accurate our approach is in terms of which connections to use in the network construction process.

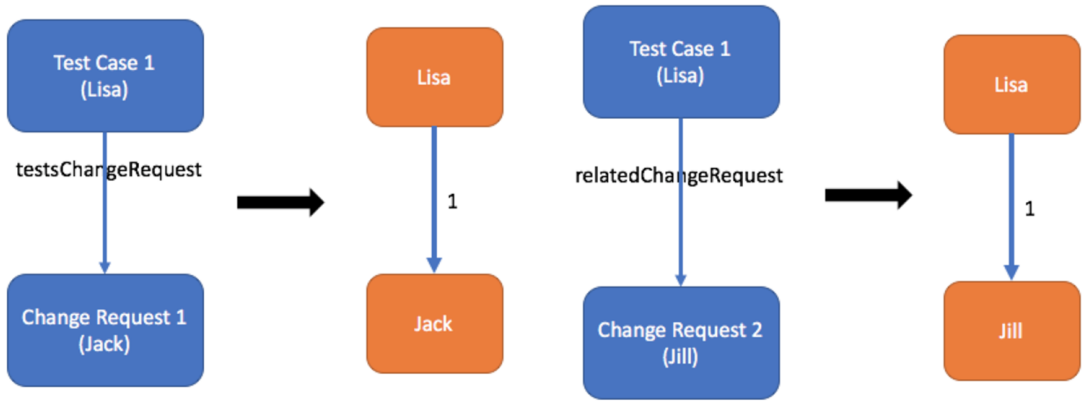
4. **How accurate do you think it is to construct the constructed social network through mining social attributes based on relationships between resources?** *

Mark only one oval.

	1	2	3	4	5	6	7	8	9	10	
Not accurate	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	Very accurate

In the following questions, please indicate the usefulness of including each relationship with the aim of supporting communication between project members.

5. **How useful do you think it is to include relations between test cases and change requests (such as testsChangeRequest and relatedChangeRequest) in the construction process?** *

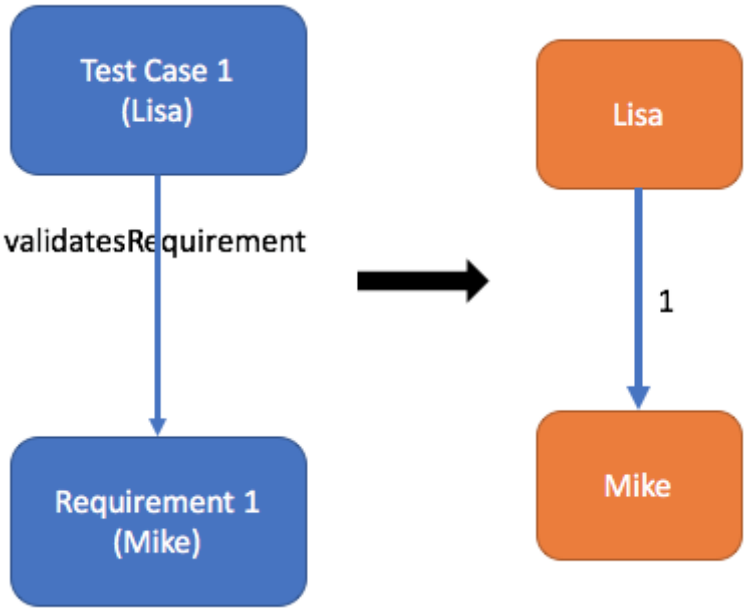


Mark only one oval.

	1	2	3	4	5	6	7	8	9	10	
Not useful	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	Very useful

6. **Please clarify your answer in the previous question**

7. **How useful do you think it is to include relations between test cases and requirements (such as validatesRequirement) in the construction process?** *

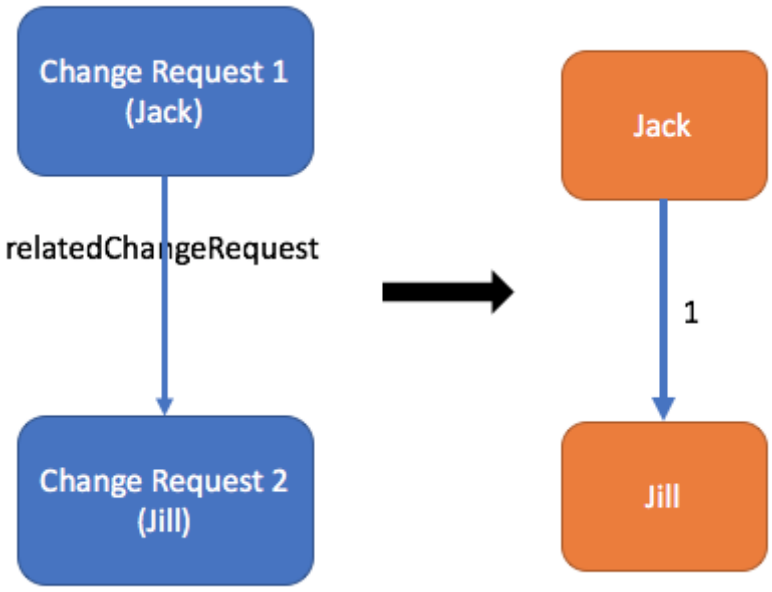


Mark only one oval.

	1	2	3	4	5	6	7	8	9	10	
Not useful	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	Very useful

8. **Please clarify your answer in the previous question**

9. **How useful do you think it is to include relations between change requests (such as relatedChangeRequest) in the construction process?** *

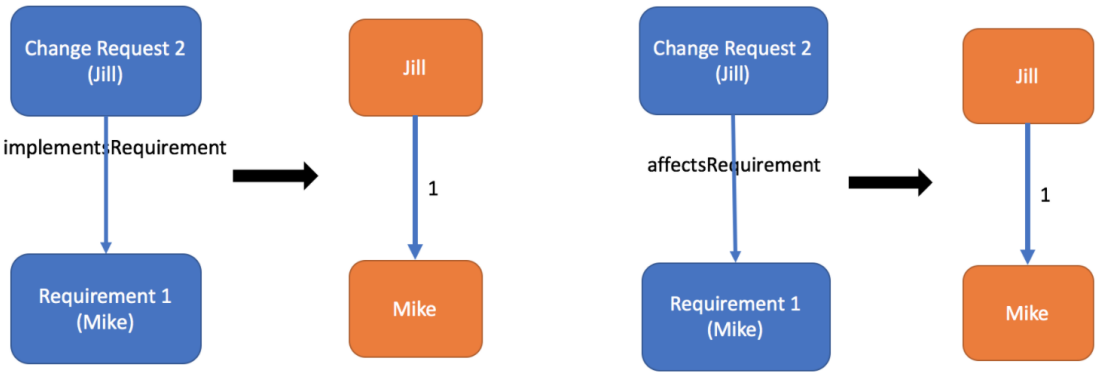


Mark only one oval.

	1	2	3	4	5	6	7	8	9	10	
Not useful	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	Very useful

10. **Please clarify your answer in the previous question**

11. **How useful do you think it is to include relations between change requests and requirements (such as implementsRequirement and affectsRequirement) in the construction process?** *



Mark only one oval.

	1	2	3	4	5	6	7	8	9	10	
Not useful	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	Very useful

12. **Please clarify your answer in the previous question**

Skip to question 24.

Usefulness of the Approach

In this section we investigate the usefulness of the approach in terms of which issues it might help to solve.

13. **How useful do you think it is to have these connections presented to you in such a visual graph?** *

Mark only one oval.

	1	2	3	4	5	6	7	8	9	10	
Not useful	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	Very useful

14. **How useful do you think the network would be for finding the most suitable person to talk to regarding a certain issue or requirement?** *

Mark only one oval.

	1	2	3	4	5	6	7	8	9	10	
Not useful	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	Very useful

15. **Please clarify your answer in the previous question**

16. **How useful do you think the network would be for finding out who should work on a certain issue?** *

Mark only one oval.

	1	2	3	4	5	6	7	8	9	10	
Not useful	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	Very useful

17. **Please clarify your answer in the previous question**

18. **How useful do you think the network would be for finding out who you should include in a certain email or meeting?** *
Mark only one oval.

	1	2	3	4	5	6	7	8	9	10	
Not useful	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	Very useful

19. **Please clarify your answer in the previous question**

20. **How useful do you think the network would be for finding out who will be affected by a certain change?** *
Mark only one oval.

	1	2	3	4	5	6	7	8	9	10	
Not useful	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	Very useful

21. **Please clarify your answer in the previous question**

22. **How useful do you think the network would be for identifying potential vulnerability of communication within the team or with stakeholders?** *
Mark only one oval.

	1	2	3	4	5	6	7	8	9	10	
Not useful	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	Very useful

23. **Please clarify your answer in the previous question**

Skip to question 4.

Tool Features

The idea is to build a tool that would mine software development data to extract the social network, build it, then present it. The following questions regard this tool.

24. **What information would you like to see in the presented network of you and your colleagues?**

25. **How useful do you think it is to see the number of paths connecting two people?** *
Mark only one oval.

	1	2	3	4	5	6	7	8	9	10	
Not useful	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	Very useful

26. **How useful do you think it is to see the original connections between resources to know how two people are connected?** *
Mark only one oval.

	1	2	3	4	5	6	7	8	9	10	
Not useful	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	Very useful

27. **How useful do you think it is to view the network of people involved with a certain resource (requirement, change request, test case, etc.)** *
Mark only one oval.

	1	2	3	4	5	6	7	8	9	10	
Not useful	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	Very useful

Ethics Related Issues

In this section we investigate potential ethical issues related to our approach and the information we present.

28. **Would you accept to be represented in such a graph among your colleagues?** *
Mark only one oval.

- Yes
- No
- Maybe

29. **Do you think that the presented network and associated information could be misused within your company by different users (developer, manager, project manager, etc.)?** *
Mark only one oval.

- Yes
- No

30. **If yes, how do you think it could be misused?**