Studying Relevant Socio-technical Aspects of Requirements-Driven Collaboration in Agile Teams

Irum Inayat
Faculty of Computer Science and Information Technology, University of Malaya
Kuala Lumpur, Malaysia
irum@siswa.um.edu.my

Sabrina Marczak
Computer Science School, Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS)
Porto Alegre, Brazil
sabrina.marczak@pucrs.br

Siti Salwah Salim
Faculty of Computer Science and Information Technology, University of Malaya
Kuala Lumpur, Malaysia
salwa@um.edu.my

***Abstract*—Requirements engineering requires intensive collaboration among team members. Agile methods also require constant collaboration among those involved in the project. While working on certain interdependent tasks, team members develop social and technical relationships that instigate socio-technical dependencies. The main goal of our research is to investigate socio-technical aspects that underlie requirements-driven collaboration among agile teams and their influence on project performance. In this paper we present our research approach to achieve such goal and briefly report on preliminary findings. A survey revealed that communication and awareness are the most relevant socio-technical aspects that underlie requirements-driven collaboration in agile teams. Initial findings of a case study aiming to identify requirements-driven collaboration patterns suggest that teams well aware of each other have lesser communication gaps and require lesser rework. Findings will contribute to a better understanding of the relationship between collaboration and performance in agile teams.**

***Index Terms*—Agile teams, requirements-driven collaboration, communication, awareness.**

I. Introduction

Requirements engineering is part of a human-centric social activity. The requirements team is seen as a social organism [1] and a requirement as a social entity [2]. When team members collaborate to develop a certain set of interdependent requirements, they establish socio-technical relations, which create social and technical dependencies in the project. These dependencies eventually lead to the development of socio-technical relationships. Collaboration itself is considered a socio-technical issue because it involves team members' coordination on both social and technical levels. Therefore, while working on a certain interdependent requirement-oriented task, team members maintain social relations with one another through technical artifacts to resolve technical issues. Collaboration that is driven by the software requirements is named *requirements-driven collaboration* (RDC in short from now on) [3] Socio-technical aspects related to such collaboration are referred as socio-technical aspects underlying RDC in our work.

The importance of collaboration in agile methods such as Scrum [4] and XP [5] is supported by their various traits like customer collaboration, game planning, and pairing work. A few studies have focused on socio-technical aspects of RDC among agile teams so far (e.g., [5][6][7][10]). For instance, the effects of communication structures on project quality by considering rework and number of defects have been discussed by Cataldo [6] while Whitworth has studied how socio-psychological factors like motivation and excitement facilitate interactions among agile teams [11].

To the best of our knowledge there is no work related to other socio-technical aspects of RDC for agile methods. Knowledge on what socio-technical aspects are relevant to RDC in agile teams and what are their effects on overall project performance can provide us with insights on how to better organize and manage agile teams and to improve requirements engineering processes to better fit agile principles. In this paper we describe our approach to investigate which are the relevant socio-technical aspects of RDC among agile teams and what are the preliminary RDC patterns based on the identified aspects empirically unveiled in an on-going case study of two software projects.

II. Previous Work

The framework proposed by Damian, Kwan, and Marczak [10] defines a strategy for studying RDC based on social network analysis theory. The framework defines the concept of a requirements-centric team (RCT) as a cross-functional group whose members' work on interdependent requirements, as well as downstream artifacts such as design, code and tests. It also defines the concept of a requirements-centric social network (RCSN) to analyze the collaboration within requirements-centric teams. A RCSN is a social network that represents the members, also called actors, and relationships, also called ties, in a RCT. Members who collaborate with the project but are not part of the RCT are called emergent members. Note that emergent members can be people allocated to the project who collaborate in a requirement without having been allocated to work on it, or can be external people to the project itself. In addition, the framework presents a set of measures from social network analysis as mechanisms to explore collaboration driven by requirements. The application of social network measures allows for a fine-grained description of collaboration and its characteristics for requirements centric teams. Although the framework is generic and can be instantiated to any kind of software team, the two case studies conducted by the authors were based on traditional software development with distributed teams. The framework, however, remains silent on issues related to the relationship of RDC patterns and team performance and iteration quality. As defined in literature, they can be calculated through a number of ways such as (i) the number of tasks planned and left undone, (ii) the difference between the number of allocated and completed man-hours for certain user stories, and (iii) the amount of rework [6]. We are interested in identifying what the application of the framework reveals in rapidly changing environments. Therefore, we proposed an instantiation of the framework to study RDC in agile teams. We present this approach next.

III. Research Questions and Proposed Approach

As previously mentioned, Damian, Kwan, and Marczak's framework [10] provides a fine-grained and formal way of studying collaboration driven by requirements and identifying collaboration patterns but it does not present a proposal on how to investigate the relationship between such patterns and project performance. The authors indicate this limitation of their work. Their instantiation of the framework in the two case studies conducted also did not contemplate a dynamic development environment based in agile methods. Therefore, we aim to fill in both gaps. The research questions to guide our study are as follows. The social-technical aspects that underlie collaboration in agile teams are still not known in literature. Therefore, we pose our first question:

RQ1. What are the most relevant socio-technical aspects underlying RDC in agile teams?

Following, we aim to better identify characteristics of such social-technical aspects in order to understand the relevance of such indicated aspects for agile teams. For instance, assuming communication is one of the aspects revealed, we aim at learning which communication structures agile teams maintain, what supports communication in such teams, among other characteristics. Hence, our second research question:

RQ2. What are the characteristics of the most relevant socio-technical aspects of RDC in agile teams?

We know from literature that collaboration affects team performance (e.g., [6]). Agile methods proposed constant and effective collaboration among team members as means to reduce misinterpretation of concepts and delays due to communication gaps. Therefore, we aim to find the effects and impact of socio-technical aspects of RDC on the performance of agile teams as defined in terms of task completion and rework after each iteration, and difference between man-hours allocated and utilized. Thus, our third research question is then defined as follows:

RQ3. What is the impact of the patterns of the most relevant socio-technical aspects of RDC identified in real-case agile teams on their performance?

Once we defined our research questions, we moved to better understand how the main concepts used in the framework defined by Damian, Kwan, and Marczak [10], namely requirement, team, and collaboration interactions defined as communication and awareness are defined in agile methods. More specifically, we set to focus our effort in the Scrum method due to its popularity among industry practitioners. This knowledge allowed us to instantiate the framework to serve our purposes. This instantiation mainly guides the design of our empirical study to answer RQ3.

In the Scrum agile method a user story is a high level requirement artifact [11] and a simple description of the desired functionality from the user's perspective [6]. User stories are slimmer than traditional requirements artifacts written by stakeholders [11]. A team is understood as a cross-functional group of people allocated to work on the project. These people often assume one of the following roles: scrum master, project owner, development team, and stakeholder (e.g., customer, vendor, etc). A team in Scrum is self-organizing, meaning that task allocation and responsibilities change as per the group decision as the project progresses. Collaboration is mediated through face-to-face verbal communication and frequent information sharing aiming to promote awareness of current task progress status. Daily stand up meetings, review meetings, and burn down charts are among the main mechanisms defined to promote communication and awareness.

We define the concept of a requirements-centric agile team based on the RCT concept from Damian and colleagues' work [10]. A requirements-centric agile team, in short RCAT, is a group of cross-functional and self-organizing people working on a set of interdependent requirements, broken down into user stories and tasks, as well as on downstream artifacts with participation of the customer (customer representative or proxy customer) Similarly, a RCSN [3] for an RCAT represents cross-functional self-organizing agile team members as actors and their interrelationships as ties. We name such network a requirements-centric agile social network, RCASN in short. For instance a tie in a certain RCASN can represent communication regarding user stories or awareness about the status of interrelated user stories and related tasks. The ties in a RCASN will be further defined based on the findings for RQ1. Each of the socio-technical aspects identified will then represent a relationship of our interest.

Followed by the visualization of the communication and awareness ARCSNs we have aimed to study the impact of the identified patterns on team performance. We aim to study the effects of ARCSNs through social network analysis measures. These measures would be used to find the characteristics of ARCSNs i.e. density, centrality, and how information flows within the networks. This would help in obtaining a better understanding of communication gaps, central people, information brokers, etc.

IV. Research Method

To answer our research questions, we defined a mixed-method research method organized in three major phases. *Phase 1* is empirical and literature-based and aims to answer questions 1 and 2. This phase is mainly organized in two larger tasks: (1) a survey to empirically identify what are the most relevant socio-technical aspects underlying RDC in agile teams (RQ1) and (2) a systematic literature review to identify the characteristics of the identified aspects (RQ2).

Phase 2 consists of our main empirical study designed to identify RDC patterns in agile teams (RQ3) and their impact on team performance. This phase is organized in five major tasks, namely: (1) the instantiation of the RDC original framework to the Scrum-based agile context, (2) the design of the case study itself, (3) the selection of the cases, (4) concomitant data collection and analysis, and (5) reporting of the findings to the participants and to academia. We aim to select multiple cases from diverse large software companies using Scrum. It is also important to note that Task 4 of this phase involves the collection of data about organizational background, processes and tool usage to help us to better comprehend the identified patterns. Onsite observation and interviews will be used for such purposes. Data on the relationships will be collected multiple times using work diaries and a questionnaire aiming to triangulate the data and assure reliability [12].

Phase 3 aims to evaluate our findings. We have initially defined a set of strategies to conduct such evaluation. For instance, we will triangulate data collected for reliability. This allows comparison of data collected from various sources to confirm similarity. We aim to counter check the perceived data interpretations with the team members and interview them for finding accuracy and usefulness. As we plan for a longitudinal study a prolonged contact with the team members is intended in order to gain deeper understanding of the issues under discussion, over time. We aim to conduct focus group discussions to discuss the interpretation of our findings and receive critical feedback from the participants. Furthermore, we aim to publish papers in peer reviewed conferences and journals in order to validate our contribution to academia.

V. Preliminary Findings

For *Phase 1*, we have identified the relevant socio-technical aspects of RDC in agile teams by conducting an online survey with agile professionals (Task 1), answering RQ1. We used a twofold approach for data collection through (i) emailing targeted worldwide communities and companies that adopt agile and (ii) uploading invitations to related groups at the LinkedIn Professional Network website. We posted our survey questionnaire to specific groups such as Agile, Agile and Lean, Agile CMMI, Agile Project Managers, Group Lean Brasil, Scrum Managers, and Scrum Practitioners. The target population was agile practitioners working in industry using Scrum. The roles to be investigated could be developer(s), tester(s), project manager(s), team lead(s), Scrum Master(s), or any other custom-created role by the company as long as directly involved with software development. Data was collected for 4 entire weeks and a total of 103 responses were collected (3/4 come from LinkedIn).

The survey results showed that communication and awareness are the two most highly recognized socio-technical aspects among agile teams (M = 1.22, SD = 0.88). These findings confirm the fact found by Cataldo that dynamic environments like agile methods are suitable for studying communication [6]. Furthermore, the respondents graded collaboration as crucial and, if missing, it could affect project success (32%), product quality (27%), team performance (24%), and requirements quality (17%). Cataldo [6] has mentioned that by keeping an eye on the communication can improve performance and quality. We have also found that collaboration with distributed teams was affirmed difficult by 68% of the respondents.

Second, we conducted a systematic literature review to find the characteristics of the two main socio-technical aspects of RDC in agile teams—communication and awareness (Task 2), providing answer to RQ2. We found that effective communication media, awareness, team's social climate, interaction of roles, and knowledge brokerage are factors that affect communication patterns, resulting in conflict resolution and better coordination.

About *Phase 2*, based on the preliminary findings of our online survey and of the systematic literature review we have customized the framework (Task 1) as presented in Section 3 and have designed and are conducting our empirical case study (Tasks 2 to 4), which aim to partially answering RQ3. Two cases have been selected from two large IT companies with offices in Malaysia. Presently, we have one researcher on site visiting both companies and closely following the work performed by the teams.

We have collected detailed data from one of the projects (Project 1) using interviews, work diaries, and a questionnaire, and have recently started observations for the second project (Project 2). Project 1 is a 3 months-long Web security project with nine team members including a product manager (PM), 5 software engineers (SE), 2 quality engineers (QE), and a user interface designer (UX). Iteration size is two weeks long. We interviewed all team members, performed document analysis, and deployed a questionnaire to investigate the RDC patterns. The questionnaire was designed to investigate the communication frequency, nature, and media adopted for each task. In addition, it asked about awareness level of team members about each others' presence, work status, professional expertise, and workplace. The team worked on 5 tasks in Iteration 1 for which the questionnaire was deployed.

Preliminary findings revealed that all the team members were overall aware with their teammates' presence, work status and professional expertise. On average the team communicated for all five tasks for more than four times a day through emails and face-to-face interaction. The only people they interacted with apart from the team are from the support unit. The main reasons of communication were change alerts in user stories, user story clarification and negotiation, sprint planning, and discussion on bugs. When asked about the amount of rework done in each task, the teams reported it was found to be minimal. This suggests that teams that are well aware of each other and have frequent communication face lesser rework in agile environment. We look forward to produce more detailed and fruitful results from collected data after the application of the social network measures proposed in [10].

VI. Anticipated Challenges and Limitations

We have used data collection triangulation which aims to provide more robust results. However, each individual method has its own limitation. We used semi-structured interviews and a questionnaire which means that our results are based on respondents' recollection and memory. We tried to minimize this risk by audio recording the interviews and later on matching them with questionnaire responses. In the questionnaire respondents were provided with a set of choices to select from about who they interacted with and why. This might be a shortcoming, limiting the respondents to select only among the provided options. However, this limitation was mitigated by using interviews where respondents were provided an opportunity to discuss in detail according to their own perception. Also, the choices were designed after a first round of interviews so the respondents could relate to their processes and workflows.

VII. Final Remarks

Requirements engineering is a social- and human-centric activity that involves intense team members' collaboration. The main aim of this research is to identify the relevant socio-technical aspects of RDC in agile teams, to define an approach to study collaboration among them, and to empirically identify RDC patterns of agile teams in practice. We conducted an online survey with industry practitioners. Responses revealed which socio-technical aspects of RDC they find relevant (RQ1). It was concluded that communication among teams and awareness of each other's activities, presence, and work status and knowledge level are the important aspects of RDC for agile methods. We also found that communication can be best studied in dynamic environments like agile methods (RQ2). We have reported the characteristics of communication and awareness and based on those drawn implications for agile teams in our literature review. Preliminary findings from the case study revealed that agile teams are highly aware of each other and this makes communication easier and frequent. (RQ3). We are about to start observing two other teams aiming to gain in depth knowledge and produce more generalized results.

A. Expected Contributions and Implications

The result of our study has important implications for research. The identified aspects should be studied in agile environments for different teams setting i.e. distributed. More empirical studies with real-world cases should be conducted to estimate the interdependency of these socio-technical aspects on each other. The quality of requirements, team relationship and their effect on project success should be studied in context of agile methods. For industry practitioners like project managers the results of this study invites them to look into communication patterns and awareness levels of their teams in order to enhance their performance, identify the people who play the most important part in communication networks, identify communication lapses and avoid rework. Moreover, the results invite managers to make better team selection decisions based on their teams' awareness levels.

Acknowledgments

This research is funded by Bright Sparks Unit, University of Malaya, Malaysia. We would like to thank the experts for their reviews and the respondents for their cooperation.

References

[1] J. Goguen, "Formality and informality in requirements engineering," in Proc. of the Int'l Symp. on Req. Eng., IEEE, Colorado Springs, USA, 1993, pp. 102-108.

[2] J. Goguen, "Social issues in requirements engineering," Proc. of the Int'l Symp. on Requirements Engineering, IEEE, Colorado Springs, US, 1993, pp. 194–195.

[3] M. Cataldo, K. Ehrlich, "The impact of the structure of communication patterns in global software development: An empirical analysis of a project using agile methods," Institute for Software Research, Carnegie Mellon University, No. CMU-ISR-11-103, 2011.

[4] D. Mishra, A. Mishra, S. Ostrovska, "Impact of physical ambiance on communication, collaboration and coordination in agile software development: An empirical evaluation," IST Journal., vol. 54, no. 10, pp. 1067–1078, 2012.

[5] E. Bjarnason, K. Wnuk, B. Regnell, "A case study on benefits and side-effects of agile practices in large-scale requirements eng.," in Proceedings of the 1st Workshop on Agile Requirements Engineering, 2011, pp. 31-35.

[6] E. Bjarnason, K. Wnuk, B. Regnell, "Requirements are slipping through the gaps: A case study on causes & effects of communication gaps in large-scale software development," in Proceedings of the Int'l Requirements Eng. Conf., IEEE, Washington DC, USA, 2011, pp. 37–46.

[7] Y. Yu, H. Sharp, "Bring everyone closer Analysing Requirements in a Case Study of Pairing," in Proc. of Workshop on Agile Requirements Engineering at the European Conf. on OO Prog., Lancaster, UK, 2011, pp. 1-6.

[8] N. Abdullah, S. Honiden, H. Sharp, B. Nuseibeh, D. Notkin, "Communication patterns of agile requirements eng.," in Proc. of Workshop on Agile Req. Eng. at the European Conf. on OO Prog., Lancaster, UK, 2011, pp. 11–14.

[9] E. Whitworth, "Agile experience: Communication and collaboration in agile software development teams," Carleton University, 2006. Carleton Univetsity, Ontario, Canada.

[10] D. Damian, I. Kwan, S. Marczak, "Requirements-driven collaboration: Leveraging the invisible relationships between requirements and people," in Collaborative Software Eng., A. Finkelstein, J. Grundy, A. van der Hoek, I. Mistrik, and J. Whitehead (Eds.), Ch. 3, Springer-Verlag, 2010, pp. 57–76.

[11] R. Yin, Case Study Research: Design and Methods, 2nd ed., Thousand Oaks: Sage, New York, 1994.

[12] A. Strauss, J. Corbin, Basics of qualitative research, 2nd. ed. Sage Publications, Thousand Oaks, USA, 1998.