

Expanding Empirical Studies to Better Understand Requirements-driven Collaboration

Sabrina Marczak¹, Irum Inayat², Siti Salwah Salim²

¹Computer Science School, Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS)
Porto Alegre, Brazil

sabrina.marczak@pucrs.br

²Faculty of Computer Science and Information Technology, University of Malaya
Kuala Lumpur, Malaysia

irum@siswa.um.edu.my; salwa@um.edu.my

Abstract. Requirements engineering involves collaboration among project members. Ineffective collaboration may result in project failure. To study the collaboration of those who need to coordinate work due to interdependencies in requirements, in our previous work we have introduced the concept of requirements-driven collaboration as the collaboration that occurs during requirements engineering and have defined a framework to guide its study. The framework is based on social network theory and provides techniques to study diverse aspects that underlie collaboration driven by requirements. Two case studies were conducted to apply the framework and to reveal empirical insights about requirements-driven collaboration. The investigated projects were globally distributed, of medium-size, and used the waterfall model to guide the development lifecycle. In this paper we posit that additional case studies of projects with distinct characteristics can help us to better understand requirements-driven collaboration. We introduce our intent to study requirements-driven collaboration in agile projects as an example of our plan to further knowledge on the topic. Broader insights can be used by researchers and practitioners to reason about how tools and processes can be improved to better support collaboration throughout the development life-cycle.

Keywords: Requirements-driven Collaboration, Coordination, Social Network Analysis, Empirical Studies, Agile Software Development

1 Introduction

Requirements engineering plays an important role in a project life-cycle, since requirements drive the development of the subsequent project phases. The later in the development life-cycle that a software error is detected, the more expensive it is to repair it. Therefore, it is cost effective to define and specify requirements early on in the project, and to manage them throughout the development cycle.

To develop the project requirements, cross-functional software teams composed of members representing different functional groups need to establish a shared under-

standing or a common ground about the requirements. Effective communication and fleeting knowledge (also known as awareness [1]) are important to foster this common ground [2]. Moreover, communication and awareness are also important for the coordination of work necessary to develop the project requirements since requirements engineering involves highly collaborative activities. For example, requirements analysts often collaborate intensively with customers and end users to gather and to specify requirements.

Coordination in general is the act of managing interdependencies between activities [3]. In order to coordinate properly, team members need to maintain up-to-date knowledge about the requirements, to exchange information about tasks related to a certain requirement, and to propagate information about changes on requirements. In addition, they need to be able to locate expertise when help is necessary to complete their tasks, and to identify who is currently available to help.

In order to study the coordination of those who need to coordinate work due to interdependencies in requirements, in our previous work [4] we have introduced the concept of requirements-driven collaboration (RDC) as the collaboration that occurs during the elicitation, definition, specification, implementation, testing, and management of requirements. In addition, we have also defined a framework to study RDC. Two case studies were conducted to apply the framework and to reveal empirical insights about RDC. Our findings are reported on [5], [6], [7], and [8]. The investigated projects were globally distributed, of medium-size, and used the waterfall model to guide the development life-cycle.

In this paper we posit that *additional case studies of projects with distinct characteristics can help us to better understand RDC*. We believe that specific project characteristics such as number of requirements and development methodology followed might influence team members' behavior towards collaboration with others and therefore should be observed for a better comprehension of RDC.

We present in this paper the RDC framework proposed in our previous work, detail the characteristics we are interested in studying, discuss expected contributions of expanding our initial empirical investigation, and our current intent to study agile distributed software projects as an example of our plan to further obtain insights about RDC to broad knowledge on the topic.

2 Objectives of the research

This paper builds up on our previous research that defined a framework to study RDC and empirically investigated two case studies of a large IT multinational to reveal insights about RDC based on the defined framework. The investigated projects were globally distributed among the headquarter office in the US and the offshore development unit in Brazil, of medium-size (18 and 40 members, respectively), attended internal customers from two distinct business areas of the company (shipping products to the customer and HR applications), of different maturity (new team and a team working together for about 5 years), and mostly used the waterfall model to guide the development life-cycle. Additionally, each project followed a pre-defined organizational structured (one fully centralized in the leaders and another more decentralized due to the members maturity working together) and followed different requirements engineering processes. The number of defined requirements and their level of specification were also distinct in each project: one project consisted of 18 high

level requirements and the other of 120 low level requirements distributed among several legacy applications. Our preliminary findings (e.g., [5][8]) suggested that such characteristics influence to a certain extent collaboration patterns of team members.

Inspired on our initial findings, in this position paper we posit that *additional case studies of projects with distinct characteristics can help us broad the empirical insights and, as a consequence, to better understand RDC in software projects*. Therefore, our long-term goal is *to conduct empirical studies of projects with distinct characteristics*. Larger teams, larger number of requirements dependencies, different types of development methodologies, higher physical distribution, and background are among the characteristics that we aim to investigate when conducting additional studies of RDC. We believe these characteristics might influence team members' behavior towards collaboration with others and therefore should be observed for a better comprehension of RDC.

We briefly present next the defined framework to study RDC in order to provide contextual knowledge for the understanding of our current research proposal.

2.1 The Defined Framework to Study Requirements-driven Collaboration

The defined framework [4] uses concepts and measures from social network analysis [9] to obtain insights about coordination patterns of those involved in requirementsdriven collaboration. The framework is based on a social structure that focuses on the requirement as the unit of work around which collaboration occurs. We termed this structure a requirements-centric team. A requirements-centric team (RCT) is a crossfunctional group whose members' work activities are related to one or more interrelated requirements, as well as downstream artifacts such as design, code and tests. By 'related to' we considered relationships such as 'assigned to', 'communicating about', 'aware of', among others [4].

We also defined a requirements-centric social network to analyze the collaboration within requirements-centric teams. A requirements-centric social network (RCSN) is a social network that represents the members, also called actors, and relationships, also called ties, in a RCT [4]. The actors in a RCSN are among the members of the RCT, and the ties in the network are representations of different relationships during these members' collaboration. For example, a tie can represent project members' requirements-related communication, assignment to work on the same requirements, or awareness of another's requirements-related work.

Based on the requirements-centric social networks, the framework presents a set of measures from social network analysis as mechanisms to explore collaboration driven by requirements. These measures were selected based on literature review and on the two empirical studies conducted. Each measure can answer one or a set of questions regarding RDC. For instance, one can use the RCSN size measure to identify how many team members are collaborating to get a requirement (or a set of dependent requirements) implemented, or use the RCSN cutpoint measure to identify which members would disrupt communication flow if removed from the team. Yet, one can apply the RCSN reachability measure to find out to what extent information can be shared with everyone involved in the development of a certain requirement or run the ties reciprocity test to learn whether awareness is reciprocal between pair of members.

3 Scientific contributions

We previously provided researchers with a framework that can be used as a mechanism to learn fine-grained details about requirements-driven collaboration. Our approach allows the study of collaboration that spans the entire project life-cycle and the examination of collaboration beyond the relationships established among developers only as traditionally reported in literature (e.g., [10]).

To broaden our understanding of RDC we posit that new projects with distinct characteristics should be investigated. Based on our preliminary findings, we aim to investigate projects that attend to one or multiple of the characteristics listed below. Additional characteristics can be suggested and investigated at any given time.

- (1) Larger teams: to check if the project team size affects RDC behavior;
- (2) Larger number of requirements dependencies: to identify to what extent a larger number of requirements in a same set of dependency and a larger number of dependency sets result in different RDC patterns;
- (3) Different project types (e.g., innovative, new product development): to identify whether the nature of the work to be done has any influence on RDC patterns;
- (4) Different business background: to identify to which extent domain knowledge affects RDC patterns;
- (5) Different types of development methodologies (e.g., agile): to identify whether working practices and processes have any influence on RDC patterns; and
- (6) Higher physical distribution (e.g., more sites involved, no overlapping working hours): to further knowledge of the effect of distance on RDC behavior.

The fine-grained requirements-driven collaboration patterns identified by using the framework in each additional project of distinct characteristics can bring more relevant insights to researchers and tool designers of how to propose better tools and processes to support cross-functional teams' collaboration. For instance, we can inform distinct features to the design of tools to support agile and traditional teams according to the identified needs. We can also provide insights on how to and with which periodicity share information about RDC of teams who have never met face to face or do not have overlapping working hours and need to use asynchronous communication tools to coordinate work. In addition, we can identify how requirements experts are located in small and in large teams.

Our preliminary work ([5][6][7][8]) introduces the research design adopted in our previous case studies. It mainly consists of three phases, namely design, data collection, and analysis. Data collection methods were as follows: on-site observation to observe daily working practices; interviews to gather information about organization and team structures and to ask about requirements engineering processes; document inspection to identify the RCTs and to build the assignment RCSNs aiming to set a baseline for the expected collaboration patterns; and questionnaire and work diaries to collect data on the actual collaboration interactions. This research design can be followed in the additional case studies. Data collected should be analyzed considering the project's background and development context. Once data has been analyzed for each single case study individually, results should be compared among cases aiming to identify similar or specific RDC patterns. Similarities or differences among cases suggest the influence (or lack of) of certain characteristics in RDC patterns.

4 Conclusions

In this paper we argue that expanding empirical studies can be beneficial for a better understanding of requirements-driven collaboration. We presented the framework we have proposed to study RDC and discuss the additional characteristics we would like to investigate to broaden knowledge on the topic. We briefly presented next our ongoing investigation of two agile teams of two large companies hosted Asian offices.

5 Ongoing and future work

Our first step putting in practice expanding empirical studies to better understand RDC focuses in investigating teams that follow a different development methodology than the one we have previously studied. The previous teams adopted the waterfall model. By convenience, we are now investigating two agile teams using Scrum.

Agile methods are based on iterative and incremental development where product scope, software requirements, and the overall technical solution evolve throughout the project through highly collaborative behavior of self-organizing and cross-functional teams. The dynamic nature of such methods promotes and encourages rapid and flexible responses to changes at any given stage of the project life-cycle. We are mainly interested to investigate which requirements-driven collaboration patterns are identified in such teams and how they differ from our previous findings.

The investigation of agile RDC invites us to think about how to conceptualize the definitions involved in our framework starting from the unit of analysis – requirements, defined in the Scrum methodology as a feature and transformed into a user story which is used as reference to derive project tasks, and passing through the concepts of requirements-centric teams (who are the members assigned to work on a certain requirement if there is no previous planning and team members should self-assign themselves to implement tasks?) and requirements-centric social networks (e.g., how to define and measure communication when it is expected that members communicate face to face and constantly?). Thus the Scrum-based agile perspective invites us to reconsider the previous definitions of our proposed framework.

We have so far conducted a literature review to better understand the aspects that underlie RDC in Scrum-based agile teams. We found that communication and awareness are the two main aspects involved in such collaboration. The knowledge acquired has been used to design the empirical investigation study that will collect data about RDC in practice in two Scrum-based agile teams located in Asia. We have a researcher on site observing the team members in their work environment and have just recently deployed a questionnaire that will collect about communication and awareness in RDC. Applying the same measures proposed in the framework and reporting early findings (including those from the literature review) are our next steps in order to contribute to a broader understanding of requirements-driven collaboration.

Acknowledgments

We would like to thank our dear colleagues Prof. Dr. Daniela Damian and Dr. Irwin Kwan from University of Victoria, Canada, for their partnership in defining the concept

of requirements-driven collaboration, developing the framework to investigate it, and discussing the meaning of the empirical insights of the two previous case studies.

References

1. Espinosa, A., Slaughter, S., Kraut, R., Herbsleb, J.: Team Knowledge and Coordination in Geographically Distributed Software Development, *Journal of Management Information Systems*, vol. 24, issue 1, pp. 135-169, 2007.
2. Coughlan, J., Macredie, R.: Effective Communication in Requirements Elicitation: A Comparison of Methodologies, *Requirements Eng.*, vol. 7, issue 2, pp. 47-60, 2002.
3. Malone, T., Crowston, K.: The Interdisciplinary Study of Coordination, *ACM Computing Surveys*, vol. 26, issue 1, pp. 87-119, 1994.
4. Damian, D., Kwan, I., Marczak, S.: Requirements-Driven Collaboration: Leveraging the Invisible Relationships between Requirements and People, *Collaborative Software Engineering*, Mistrik, I., Grundy, J., van der Hoek, A., Whitehead, J. (Eds.), Chapter 3, pp. 577-6, Springer-Verlag, Computer Science Editorial Series, London, England, March 2010.
5. Damian, D., Marczak, S., Kwan, I.: Collaboration Patterns and the Impact of Distance on Awareness in Requirements-Centred Social Networks”, in *Proc. of the Int’l Requirements Engineering Conference*, New Delhi, India, pp. 59-68, IEEE Computer Society, 2007.
6. Marczak, S., Damian, D., Stege, U., Schröter, A., Information Brokers in Requirements Dependency Social Networks, in *Proc. of the Int’l Requirements Engineering Conference*, Barcelona, Spain, pp. 53-62, IEEE Computer Society, 2008.
7. Marczak, S., Kwan, I., Damian, D., Investigating Collaboration Driven by Requirements in Cross-Functional Teams”, in *Proc. of the Collaboration and Intercultural Issues on Requirements Communication, Understanding, and Softskills Workshop*, in conjunction with the Int’l Requirements Eng. Conf., Atlanta, United States, (available online), IEEE, 2009.
8. Marczak, S., Damian, D.: How Interaction between Roles Shapes the Communication Structure in Requirements-driven Collaboration, in *Proc. of the Int’l Requirements Engineering Conference*, Trento, Italy, pp. 47-56, IEEE Computer Society, 2011.
9. Wasserman, S., Faust, K.: *Social Network Analysis: Methods and Applications*, Cambridge University Press, Cambridge, United Kingdom, 1994.
10. Hinds, P., McGrath, C.: Structures that Work: Social Structure, Work Structure and Coordination Ease in Geographically Distributed Teams, in *Proc. of the Conference on Computer Supported Cooperative Work*, Banff, Canada, pp. 343-352, ACM, 2006.