AN APPROACH FOR IMPROVING THE SOCIAL ASPECTS OF THE SOFTWARE DEVELOPMENT PROCESS BY USING A GAME THEORETIC PERSPECTIVE

Towards a Theory of Social Productivity of Software Development Teams

Murat Yilmaz[1] and Rory V. O'Connor[2]

[1]*Lero Graduate School in Software Engineering, Dublin City University, Dublin, Ireland*

[2]*Lero, the Irish Software Engineering Research Center, Dublin City University, Dublin, Ireland*

Keywords: Software process improvement, Team dynamics, Game theory, Mechanism design, Research framework, Software engineering economics, Productivity, Social aspects of development.

Abstract: As software development is considered a form of knowledge based social activity, investigating social interactions and behaviors of individuals and teams constitutes a starting point for improving organizational performance and productivity. Therefore, a software development organization is regarded as a form of social network, which may be more efficiently structured based upon participants' skills, roles and capacities to exchange information. This paper aims to propose a research framework for modeling development activities in terms of social interrelationships. It investigates ways for improving the productivity of the software development process to several social issues (e.g. team formations, interpersonal conflicts, social loafing) that affect the group setting during the software development process. An industrial survey will be conducted to validate the proposed framework. This survey will be applied to three medium size software organizations; (i) to measure social aspects of productivity, (ii) to assess the effectiveness of our approach, (iii) and to improve the welfare of the software development organization. Semi-structured interviews with practitioners, and expert reviews with the managers will be used to evaluate the results. Ultimately, based on the principles of our game theoretical approach and collected data, we propose a research framework, which will benefit future research by drawing a road map that establishes a body of knowledge, specifically on software development teams and organizations.

1 INTRODUCTION

This paper aims to establish a research framework based on the importance of understanding software development as a knowledge-based social activity (Dittrich et al., 2002). A knowledge-based viewpoint suggests that the intangible value of an organization is created as a deployable knowledge asset (Grant, 2002), where it is formed from the outcome of the interactions of participants. This value is somehow stored by its teams and individuals in terms of intellectual (Rus and Lindvall, 2002) and social capital (Lin, 2002). Therefore, the structuring of individuals' roles in a software organization with respect to the quality of information exchange among the individuals should be considered more important than several other factors affecting the productivity of a software development organization. However, approaches (e.g. game theoretical) that employ the roles of individuals and analyze the outcomes of their social interactions in team-based activities have not yet been investigated in the field of software development.

An initial exploration of the importance of social aspects of software development (Acuna et al., 2005) highlights the fact that modeling individuals' interactions (Acuna and Juristo, 2005) and their motivations (Beecham et al., 2008) to achieve economic and social outcomes is beneficial for maximizing the opportunities for improving the software development process. There is a body of research that has reinforced the view that social networks (i.e. network of social interactions) can be considered as an important factor for investigating organizational issues, in particular when they are used for considering the structure and dynamics of a software development organization (Ohira et al., 2006; Nielsen and Tjornehoj,

Yilmaz M. and V. O'Connor R..
AN APPROACH FOR IMPROVING THE SOCIAL ASPECTS OF THE SOFTWARE DEVELOPMENT PROCESS BY USING A GAME THEORETIC PERSPECTIVE - Towards a Theory of Social Productivity of Software Development Teams.
DOI: 10.5220/0003492900350040
In *Proceedings of the 6th International Conference on Software and Database Technologies* (ICSOFT-2011), pages 35-40
ISBN: 978-989-8425-76-8

2010). Furthermore, the ability for a software development organization to understand the factors that hinder productivity is essential in improving the competitive advantage for the business success (Chemuturi, 2009), where it is not possible to think about productivity regardless of social issues (i.e. matters that affect individuals and teams) affecting the development process.

The remainder of this paper is organized as follows. In section 2, we provide a brief overview of research methodologies. In section 3 we establish a context for our research framework. Based on the research questions and the context of the research, in section 4, three hypotheses are proposed. Finally, section 5 summarizes the research and section 6 shows future work and expected outcomes.

2 A BRIEF OVERVIEW OF RESEARCH METHODOLOGIES

Although there are several different research methodologies used in software engineering research, all can be classified in two main categories: quantitative research (e.g. survey research, experiments, and simulations) and qualitative research (e.g. grounded theory, ethnography, action research) (Dawson, 2002). Based on the idea of seeing the human as an instrument and considering their experiences, qualitative research aims to investigate participants actions and words by explanatory patterns of meaning (Stake, 2010). It studies mental attitudes and social behaviors frequently recorded as information from the participants own words and definitions, and classify them from their natural work (settings) environment. Qualitative researchers use techniques like interviews, individual experiences, case studies and focus groups to capture data about the values and emotions of people for investigative observations. The collected data can be documented in a contextual framework for conducting a closer observation of words and view of the participants and further inspection (Hesse-Biber and Leavy, 2010).

On the other hand, quantitative research is based on the idea of quantifying the interrelations between different types of variables (i.e. independent, dependent) using numerical techniques. It is corroborative and based on conventional and systematical processes to gather data, which describes the information by cause and effects relations. Quantitative research aims to create theories, hypotheses and mathematical models based on empirical investigations (Balnaves and Caputi, 2001).

A blending of different qualitative and quantitative methods and approaches (or their variants) as a technique to improve the validity of research findings and conclusions is called mixed method research. Triangulation (Cook and Campbell, 1979) is one of the most well-known mixed method design strategies. This approach advocates that multiple types of data and methods can support hypotheses, incident or events. In other words, the name triangulation is a metaphor which describes the use of multiple methods in a research: a combination of two research traditions (Jick, 1979); i.e. theories, data sources, methods or techniques to study a research question or to measure a single construct.

3 CONTEXT OF CURRENT RESEARCH

As it is suitable for empirical or applied software engineering research, our approach is an attempt to investigate the influence of social factors over the software production process. In the hope that we will be able to improve the social structure of a software organization and therefore to maximize the productivity of software teams. With these in mind, our goal is to develop a team composition framework (i.e. a game theoretical team interaction model) based on the details (e.g. interactions and types) of the participants.

To facilitate the visualization of our model further, on the one hand, we need qualitative analysis techniques for gathering scenarios of interaction from participants. On the other hand, we need quantitative methods for establishing a rigorous mathematical formulation for investigating several factors, creating a scenario based team formation model, and use them concurrently for testing our theories and hypotheses. By combining these two approaches systematically, our key idea here is to build a more efficient perspective.

3.1 Game Theoretic Personality Types

Before introducing game theoretic personality types, we first briefly survey some similar concepts used in personality profiling in research.

The theory of personality types was introduced by Carl Jung which is based on the classification of people with distinctive set of characteristics found in individual's personalities. Based on the *theory of psychological type* (Jung et al., 1991), which states that there are four types of personality dimension, a questionnaire called Myers-Briggs Type Indicator (MBTI)

was created. An aim is to operationalize Jung's work in order to understand how people make decision regarding to their personalities and categorize them in 16 different psychological types (i.e. types are considered a combination of four preferences) (Myers et al., 1999). Moreover, Keirsey has conducted empirical investigations on the human psychology based on a variant of Jung's theory (Keirsey et al., 1984). In particular, he worked on a temperament sorter about personality types where he also found out 16 different forms. Unlike Myers who investigates people's feeling, his research is directly focused on observable behaviors of humans.

The correlation between personality of actors and their productivity first indicated by Weinberg and later investigated by Shneiderman for its role among the interactions of software development (Weinberg, 1971; Shneiderman, 1980). Da Cunha and Greathead claim that types of personality and skills of software development personnel can be aligned to advance the productivity (Da Cunha and Greathead, 2007). Beecham *et al.* identify several social factors effecting software engineer characteristics (Beecham et al., 2008). For example, software practitioners prefer challenging tasks and therefore recognition in their work, stability in their organization, and most importantly they prefer to be socially identified with their team or group (Beecham et al., 2008).

Although several research was conducted on personality types most of these works use MBTI. (e.g (Capretz, 2003; Da Cunha and Greathead, 2007; Gorla and Lam, 2004)), there is no body of research for identification of personality types applicable to game theoretical approaches. We suggest that, based on our game theoretic model, these types should be compatible with our model for identification of people's behavior with respect to several situations. Therefore, we propose creation of socio-types regarding to variety of interactions based on a game theoretical setting. Therefore, we start our framework by hypothetical game theoretic personality types (GTPT) as; (i) cooperative, (ii) competitive, (iii) socially cooperative, (iv) socially competitive, (v) individualistic.

4 RESEARCH QUESTIONS AND HYPOTHESES

In response to the problems highlighted, in this section we develop a list of research questions and hypothesis so as to conduct our research and we employ techniques that can be used to evaluate our conceptual propositions with respect to the accuracy of data collected. We test our hypotheses and analyze the data gathered from selected software organizations as the study of practical cases.

Research Question 1. *Will assigning individuals to the appropriate roles with respect to their GTPTs improve software team productivity?*

Research Question 2. *Can a correlation between productivity and social productivity be identified for software development?*

The goal of the first research question is to initiate a framework to expand a mapping of technical roles and align it with participants' game theoretic types. This question stems from the fact that when characteristics of individuals and their technical roles are better blended, the productivity of team will increase (Acuna et al., 2006; Capretz, 2003).

Brandenburger and Nalebuf (Nalebuff and Brandenburger, 1996) have argued that there is a visible interconnection between the actors of production processes (e.g. client and vendor) which should be highlighted for developing better responses to the market demands. In the light of this argument, the second research question distinctively seeks possible configuration of intellectual (i.e. human and social) capital of software development organization to improve production rate of software development. Therefore, a particular aim here is to expand the definition of software development productivity to include the concerns regarding to social issues for individuals and teams in a software development organization. In particular, we define social aspects affecting the productivity as *social productivity*, i.e. the production rate of software development increases if we give due consideration to maximizing the social relations. This research is concerned with identifying the relationship between productivity and its social aspects and hence it has been developed to support this endeavor.

To seek answers to these questions, we have established and formalized our first hypothesis guided by our research agenda.

Hypothesis 1. *Improving the social structure of a software organization will improve the productivity of software development teams.*

The second group of research questions seek to uncover the ways to improve the formation of software teams by using GTPTs. We suggest that the information exchange problem among software team members needs to be identified and converted into a game theoretic view by combining these information with game theoretic types of interacting individuals.

Research Question 3. *Can organizing individuals regarding to their personality types*

help to improve the social dynamics of a software organization by designing a mechanism?

Research Question 4. *Can we formulate a model and simulate software team compositions and improve the productivity of software development teams?*

The second hypothesis relies on the argument that it should be possible to use our concept of game theoretical personality types to improve several aspects of software development teams.

Hypothesis 2. *Software development teams that are formed by using GTPTs will perform better than those structured with an ad-hoc method.*

The next group of hypothesis and research questions aim to facilitate hypothesis 2 so as to expand to include social network analysis while seeking ways for maximizing the collective outcome. Therefore, the structure of hypothesis 3 and the subsequent two research questions will be built on hypothesis 2. The expected achievement here is to find techniques to visualize a software organization and behaviors and test the distinctive GTPTs have a possible impact on social network structure. For example, by investigating and visualizing the social structure of an organization; (i) key participants can be identified, (ii) the information flow inside the social network can be analyzed, (iii) the outcome of the interaction of different actors can be highlighted,..etc.

Research Question 5. *Can we collectively use both social network analysis and personality types together to maximize the software development team productivity?*

Research Question 6. *Does revealing and examining the positions of participants in a social structure can help us to improve the team productivity?*

A frequently asked question at social and economic research is the impact of revealing the positions of participants on tiers of a network. Therefore, we suggest it is important to show the influence of individuals and their social circles at a conceptual level.

Hypothesis 3. *A combination of social network analysis together with the individuals personality types leads us to understand the potential requirements for structural improvements of software teams.*

Based on an information exchange economy, we consider that the organizational structure of software development is an intellectual effort based on interconnected networks of individuals, their actions and relationships. Consequently, the third hypothesis states that a social network analysis should be helpful to investigate the impacts of participants interactions, and as a result how these social interactions and the structure affect team formations and the productivity of software development. Therefore, we plan to use SNA with GTPTs together for team optimization and to maximize the outcomes of a production process. In conclusion, our aim is to generalize our theoretical findings with empirical observations from the field.

5 THE RESEARCH FRAMEWORK

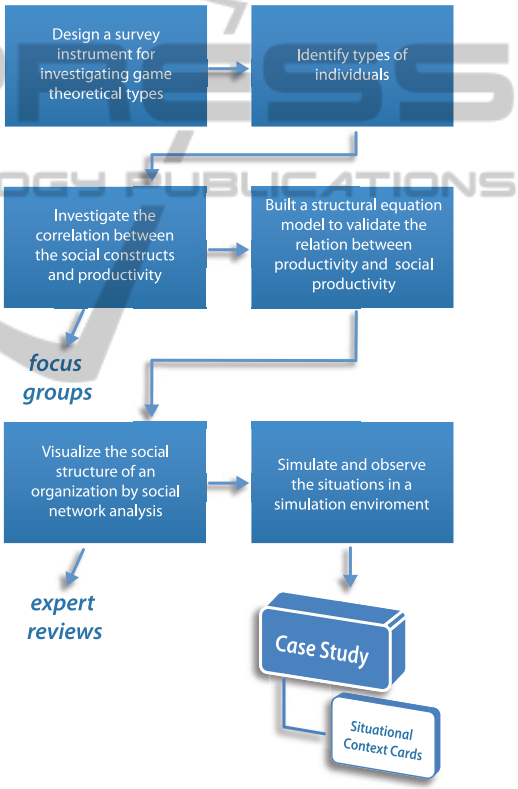

Figure 1: The framework for the proposed research.

Here, we formalize a research framework to conduct our research (see figure 1). As outlined earlier in this paper, the primary goal is to create a team productivity model for software development organizations based on social and economical factors and therefore to analyze the impact of social factors on the productivity of software organizations from a game theoretical perspective. We introduce the concept of social productivity and further we use it to identify the social factors and its relationship with software productivity.

Initially, we begin the first part of the research by creating a survey instrument based on game theoretical personality types of individuals and determine their types and degree of interactions. We are conducting a research on the game theoretical personality types and further we will have them reviewed by experts from the field for evaluation. Secondly, we continue our work by analyzing the correlation between productivity and social productivity by using regression techniques and to construct an empirical relationship model.

Thirdly, we initiate our investigation of several situations and based on this information we evaluate the identification of our game theoretic personality types. To store this information for future use, we plan to create situational context cards. Fourth, we claim that an empirical investigation of the notion of personality types and their variants of compositions need to be identified by their location and role in a social structure of a software company. Therefore, we formalize the social structure of a software organization by using sociometric techniques such as social network analysis for examining relation of personality types and other social aspects such as task roles inside an information exchange network.

Fifth and finally, we aim to simulate the output of this research by creating a *team composer* software prototype for better composition of teams. We argue that a simulation (which has been used frequently to create social and economic modeling (Gilbert, 2008)) of team compositions will be a suitable way to perform several scenarios to investigate the social dynamics of software development. In other words, we suggest simulation is a feasible approach for capturing interactions for this emergent phenomena from the viewpoint of its interacting units. Because it supplies important information on a system in virtual form, we can simulate dynamic scenarios in a virtual landscape. It will, therefore, allow us to organize more productive software teams.

6 CONCLUSIONS AND FUTURE WORK

A significant challenge in the software engineering research is to model software development as a social and economic activity. Based on the defined tasks of development, an organization may have several types of participants with distinctive roles. As we have proposed, one way of improving the social productivity of a software organization is to improve the productivity by exploring people's interactions, behaviors and to design its outcomes.

The focal point of this research is based on game theoretical investigation of a software development projects as it aims to construct a model for software development as a (cooperate) game of incomplete information. As it has been observed that organization of software development can be very sensitive to several types of market forces (Madhavji et al., 2006), organizations can benefit from a model of a software organization for better decision making about their social and organization issues. Further, we will compare our game theoretic personality types with Myers-Briggs Type Indicator (MBTI) (Myers et al., 1999) and Belbin's team roles (Belbin, 2010) so as to identify how our particular part of research intersects with the previous body of knowledge.

We aim to conduct our research on three medium size software companies. Initially, we interview some experts from these companies. However, there are several parameters may hinder our expected results. For example: variations of team size and teams working in different domains. Therefore, our preliminary planing includes testing our approach on three software teams consisting of forty people within the same company for acquiring more homogeneous results. Several social situations and scenarios can be captured at different levels of software development organization. And, these scenarios can be depicted by using use case diagrams. Moreover, the working scenarios can be adapted to a simulation model and rigorously analyze to gather more information about potential alternative cases. This study aims to use computer simulation techniques to perform several hypothetical scenarios based on a team environment. To improve the productivity of software development teams, we compare virtual events with actual outcomes that are exercised by team members on scenario based situations. Furthermore, we suggest that situational outcomes of a computer simulation should also be discussed with management teams of targeted software companies by conducting expert reviews.

ACKNOWLEDGEMENTS

This work is supported, in part, by Science Foundation Ireland grant number 03/CE2/I303-1 to Lero, the Irish Software Engineering Research Centre (www.lero.ie).

REFERENCES

Acuna, S. and Juristo, N. (2005). *Software process modeling*. Springer Verlag.

Acuna, S., Juristo, N., and Moreno, A. (2006). Emphasizing human capabilities in software development. *Software, IEEE*, 23(2):94–101.

Acuna, S. T., Juristo, N., Moreno, A. M., and Mon, A. (2005). *A Software Process Model Handbook for Incorporating People's Capabilities*. Springer-Verlag New York, Inc.

Balnaves, M. and Caputi, P. (2001). *Introduction to quantitative research methods: An investigative approach*. Sage Publications Ltd.

Beecham, S., Baddoo, N., Hall, T., Robinson, H., and Sharp, H. (2008). Motivation in software engineering: A systematic literature review. *Information and Software Technology*, 50(9-10):860–878.

Belbin, R. (2010). *Management teams: why they succeed or fail*. Butterworth-Heinemann.

Capretz, L. (2003). Personality types in software engineering. *International Journal of Human-Computer Studies*, 58(2):207–214.

Chemuturi, M. (2009). *Software Estimation Best Practices, Tools & Techniques: A Complete Guide for Software Project Estimators*. J. Ross Publishing.

Cook, T. and Campbell, D. (1979). Quasi-experimental design and analysis issues for field settings. *Chicago: Rand Mc-Nally College Publishing Co*.

Da Cunha, A. and Greathead, D. (2007). Does personality matter?: an analysis of code-review ability. *Communications of the ACM*, 50(5):109–112.

Dawson, C. (2002). *Practical research methods: a user-friendly guide to mastering research techniques and projects*. How To Books Ltd.

Dittrich, Y., Floyd, C., and Klischewski, R. (2002). *Social thinking-software practice*. The MIT Press.

Gilbert, G. (2008). *Agent-based models*. Sage Publications, Inc.

Gorla, N. and Lam, Y. (2004). Who should work with whom?: building effective software project teams. *Communications of the ACM*, 47(6):79–82.

Grant, R. (2002). The knowledge-based view of the firm. *The strategic management of intellectual capital and organizational knowledge*, pages 133–148.

Hesse-Biber, S. and Leavy, P. (2010). *The practice of qualitative research*. Sage Publications, Inc.

Jick, T. (1979). Mixing qualitative and quantitative methods: Triangulation in action. *Administrative science quarterly*, pages 602–611.

Jung, C., Baynes, H., and Hull, R. (1991). *Psychological types*. Routledge.

Keirsey, D., Bates, M., and Company, P. N. B. (1984). *Please understand me: Character and temperament types*. Prometheus Nemesis Del Mar, Calif.

Lin, N. (2002). *Social capital: A theory of social structure and action*. Cambridge Univ Pr.

Madhavji, N., Lehman, M., Perry, D., and Ramil, J. (2006). *Software evolution and feedback*. Wiley Online Library.

Myers, I., McCaulley, M., Quenk, N., and Hammer, A. (1999). *MBTI manual*. Consulting Psychologists Press.

Nalebuff, B. and Brandenburger, A. (1996). *Co-opetition: A Revolution Mindset That Combines Competition and Co-operation: The Game Theory Strategy That's Changing the Game of Business*. Doubleday Book, Newy York.

Nielsen, P. and Tjornehoj, G. (2010). Social networks in software process improvement. *Journal of Software Maintenance and Evolution: Research and Practice*, 22(1):33–51.

Ohira, M., Ohoka, T., Kakimoto, T., Ohsugi, N., and Matsumoto, K. (2006). Supporting knowledge collaboration using social networks in a large-scale online community of software development projects. In *Software Engineering Conference, 2005. APSEC'05. 12th Asia-Pacific*, page 6. IEEE.

Rus, I. and Lindvall, M. (2002). Guest editors' introduction: Knowledge management in software engineering. *IEEE software*, pages 26–38.

Shneiderman, B. (1980). *Software psychology: Human factors in computer and information systems (Winthrop computer systems series)*. Winthrop Publishers.

Stake, R. (2010). *Qualitative Research: Studying How Things Work*. The Guilford Press.

Weinberg, G. (1971). *The psychology of computer programming*. Van Nostrand Reinhold New York.