Studying Communication in Agile Software Development

A Research Framework and Pilot Study

Tuomas Niinimäki
Software Business and Engineering Institute
Helsinki University of Technology
taniinim@cc.hut.fi

Arttu Piri
Software Business and Engineering Institute
Helsinki University of Technology
arttu.piri@tkk.fi

Peitsa Hynninen
Software Business and Engineering Institute
Helsinki University of Technology
peitsa.hynninen@tkk.fi

Casper Lassenius
Software Business and Engineering Institute
Helsinki University of Technology
casper.lassenius@tkk.fi

ABSTRACT

Agile software development practices have gained significant importance during last few years as a methodology for implementing software projects. At the same time, software businesses have started to exploit the potential of global software development, most commonly to lower development costs and to access larger pools of competent labor. Global software development has proven to be difficult to implement efficiently, and while several different methods and practices have been proposed, many challenges, most of which are related to communication, still remain.

In this paper, we describe a research design for studying communication in agile software development projects. The goal is to evaluate various data collection methods and analysis methods for agile software project research, and ultimately apply these methods in implementing more comprehensive research project on using agile development practices in global software development projects.

Categories and Subject Descriptors

D.2.8 [**Software Engineering**]: Metrics—*Process metrics, Performance measures*; D.2.9 [**Software Engineering**]: Management—*Software process models, Programming teams*

1. INTRODUCTION

Traditionally, software development organizations have worked under the assumption of a stable business environment producing stable requirements for the products to build. This has been reflected in the development and adoption of rigid, plan-and-control -driven processes for managing and controlling software development work.

ICMI-MLMI'09, Workshop on Multimodal Sensor-Based Systems and Mobile Phones for Social Computing. November 6, 2009 Cambridge MA, USA.

While perhaps never a correct assumption, the contemporary turbulent business environment requires software development projects to work with uncertain and changing requirements and technologies. As the traditional approach to software engineering, exemplified by the waterfall life-cycle model [24], is based on resisting change, it is inherently a poor fit with the modern business climate.

Traditional software engineering approaches focus on up-front planning, including fixing the requirements, modularizing the product to build, and specialized labor, all with the aim of minimizing dependencies between product elements, thus reducing the need for coordination and communication in the software development organization. Given uncertain and changing requirements, this becomes difficult, and modern approaches to software development, such as the so called *agile development* models, focus more on effective communication and collaboration than up-front planning and documentation [2]. In particular, the agile software development models try to make communication as quick and effortless as possible by requiring all team members to share a common open working space and emphasizing face-to-face communication. The adoption of various agile methods, in particular the Scrum process model [25], has been rapid in industry, and their use is becoming common in most fields of software development.

Another main industry trend is the increased use of *global software development* (GSD), mainly as a means to lower software development costs and to access a larger pool of competent labor. While increasingly common, GSD projects often face serious problems. These include issues related to communication between the distributed project members, difficulties in establishing appropriate group relations, cultural issues, and difficulties in managing and coordinating work in distributed projects. In short, intensive collaboration which is needed in software development seems to be challenging in a distributed environment. [12]

As many agile practices rely heavily on face-to-face communication, which is difficult or impossible to arrange in many GSD settings, combining the two at first glance seems difficult. There is, however, some evidence that applying agile methods in distributed projects could make it possible to mitigate some of the most com-

mon problems of distributed software development [19, 21]. In particular, agile methods force team members to communicate openly and frequently, and it seems that this can help mitigate even cultural issues, as well as build a solid foundation for distributed team work.

In this paper we describe a framework and methodology for studying communication, group dynamics, and project success of globally distributed agile software development projects, that we plan to use in a research project scheduled to start in the beginning of 2010. We also describe our currently ongoing pilot study during which the framework has been developed and partly tested.

2. PREVIOUS WORK

A fairly large share of software developers' time is spent on communicating with other developers in the team. Perry, Staudenmayer and Votta report in their study developers spending on average 75 minutes each day in "unplanned interpersonal interaction" [22]. Seminal work by Tom Allen reported that the frequency of communication among engineers drops radically based on distance, and the frequency being almost at the same low level whether engineers are located 30 meters or miles apart [1]. Earlier studies have reported the slower pace of GSD projects mainly to be caused by challenges and delays due to interaction over distance [10, 11, 12].

To overcome the distance, software teams have multiple communication media at their disposal for sharing information and building shared understanding on the task at hand. Media richness theory [3] attributes a property called *richness* to a communication medium. The richness represents a medium's ability to provide additional cues to communicating parties for understanding the message. Media richness theory suggests that tasks with high uncertainty and equivocality are communicated more efficiently via communication media with higher media richness, while certain and unequivocal tasks can be communicated via a leaner media [3].

Agile software development practices claim to solve many issues of software engineering, including long development times, higher costs than anticipated and unmet requirements [14]. Iterative development and short delivery cycles help to mitigate the problem of not meeting user requirements, cost/schedule blow-outs and integration issues [23]. They prevent different sites and partners from doing long periods of independent development, which could lead to modules that are hard or impossible to integrate. Short delivery cycles also bring transparency of work to all partners [21]. Empirical studies on existing agile GSD projects suggest that agile methods can be used on GSD projects [21], and can help avoid some of the most common problems of distributed software development [20].

Previous work has been made in studying communication in various contexts of knowledge work. Earlier studies have reported relation between outcome factors, such as performance and creativity, and communication behavior [9], as well as relation between face-to-face communication and productivity in IT configuration tasks [28]. However, we believe the proposed research to be the first to study the relationship between face-to-face and electronic communication, psychosocial factors and project outcome metrics in distributed agile software development contexts.

3. METHODOLOGY

In this section we describe the research methodology we plan to use in our future research, and that we used in the pilot study.

3.1 Data Collection

In order to understand the relationship between communication and contextual and outcome measure, we need a comprehensive data set collected using a variety of instruments. In following chapters, we will present our data collection methods in more detail.

3.1.1 Face-to-face Communication

We used sociometric badges developed at the Massachusetts Institute of Technology Media lab [18], to automatically collect data on face-to-face communication between collocated members of the teams. A sociometric badge is a device containing a microphone, an infrared sensor, and a radio receiver. When studying communication, the badges are used in two ways. First, each team members wear a badge around his or her neck. This helps us collect data on both physical proximity between team members, as well as their communicational behavior; who talks with whom, how and when. Secondly, badges are attached to physical locations, such as task boards, workstations, and other interesting places, such as cafeterias and meeting rooms. This helps us understand where the communication takes place. By badging the workstations, we can also look into fine-grained issues, such as how much people work together at the same computer (known as pair programming [27]).

In addition to general analyses of the amount and frequency of face-to-face communication and relative proximity of team members, from the badge data we are able to make analyses of communication networks and patterns within the team based on real face-to-face communication.

3.1.2 Electronic Communication

In order to get a comprehensive view on the communication networks and patterns of the team, we have collected the email archives and chat logs of the team. We instructed the teams to keep all project-related email messages, and to log all instant messages and chat discussions related to project. The analysis of electronic communication data allows us to augment our view on the communication in the project, and to compare the volume and patterns between various communication media.

3.1.3 Project Repositories

To get more objective data on project outcome, we plan to collect data from issue trackers, task management tools and version control systems used by the teams. Based on this data, we hope to be able to analyze the relationship between communication behavior, subjective psychosocial factors and project success.

While we acknowledge the difficulty of assessing cross-team performance based on repository data, this data may provide us useful insights at least to support our other findings on the project collaboration and used practices in single case level. While hard project metrics can be difficult to compare between different case projects, identified relations between project outcomes, psychosocial and communication measures can be used to build hypotheses for future research.

3.1.4 Surveys

To be able to map the team members' personal views on communication satisfaction, as well as their group dynamic and psychological factors like trust, commitment, identification, and to be able to analyze the relations between these factors, communication patterns, and project performance, we will use a number of survey

instruments drawing on social psychological and organizational behavior theories.

All instruments have been validated in previous studies, excluding the instruments on individual daily performance, project success, and software engineering processes designed by the authors. The instruments are summarized in Table 1.

Table 1: Survey Instruments

Target	Description	Source
Team identity	6 items	Mael & Ashforth (1992) [15]
Team trust	16 items	Speitzer & Mishra (1999) [26]
Trust on supervisors	10 items	Gillespie (2003) [6]
Job Satisfaction (MSQ)	6 items	
Project commitment	5 items	Hoegl, Weinkauf & Gemunden (2004) [13]
Organizational commitment (OCQ)	15 items	Mowday, Steers & Porter (1979) [17]
Communication Satisfaction (CSQ-II)	8 factors, 40 items	Downs & Hazen (1977) [4]
Team satisfaction	3 items	Gladstein (1984) [7]
Work design Questionnaire (WDQ)	19 factors, 80 items	Morgeson & Humphrey (2006) [16]
Individual performance and creativity	8 items	Designed by the research team
Project success	7 items	Designed by the research team
Process satisfaction	11 items	Designed by the research team

3.1.5 Interviews

By interviewing key personnel (e.g. project managers, scrum masters and product owners) of each project, we hope to get a good understanding of the project goals, processes and practices used, the product being developed, and about the general organizational context of the project. Interviews are also used to validate and triangulate findings based on other data collection methods.

3.1.6 Observations

We have also used non-participative observation in selected situations, such as meetings and planning sessions, to gain qualitative information and understanding about the communicational and group dynamic structures of the team, and about the agile process used in the project. Observations are also used to discover possible contradictions between articulated and actual team behavior.

3.2 Data Analysis

As we are applying several methods of data collection in this study, we aim at using multiple methods and disciplines to analyze the data.

3.2.1 Qualitative Analysis

Qualitative analysis methods are used in the analysis of interview and observation data. After transcribing the interviews, we code the transcripts based on the concepts and phenomena discussed during the interviews. The collected code set is used to identify core concepts for further analysis. A second pass of coding is used to focus in more detail on some of the identified concepts and the related phenomena.

3.2.2 Quantitative Analysis

The survey data and communication data collected from the case projects are analyzed quantitatively. For survey data, we use statistical methods to analyze individual metrics and relations between them. For communication data, we extract communication frequency (e.g. how often people send email to each other, or talk face-to-face), amount and length of discussions and meeting characteristics (e.g. length and number of participants in a meeting). Furthermore, we will use Condor software [8] for social network analysis to extract both network metrics (such as network density, centrality and hierarchy measures [5]) and communication patterns from communication networks collected from both face-to-face interaction and electronic communication.

As the sample size from the pilot study is rather small, and thus larger scale quantitative analysis can be difficult, some indicative correlations within the data set may be possible. Based on the preliminary findings from qualitative analysis and the indicative correlations, we are able to formulate and refine hypotheses for the second round of data collection.

4. PILOT STUDY

The currently ongoing pilot research uses both quantitative and qualitative methods. We are studying three industrial software development projects in two multinational software companies, see Table 2.

All three case projects use agile software development methods based on Scrum [25]. Case A is a collocated project, with all team members working in designated area of open office area. In case B, the project has a dedicated project room, but some team members occasionally work remotely for the project, as well as one person was constantly working remotely from another site. Case C is globally distributed project between Norway, Czech Republic and Finland, with all sites having designated working area. For this study, we were able to only collect data from team members at the Norwegian and Czech sites.

Table 2: Case Projects

Case project	Team members	Distribution	Sites
A	11	Collocated	Finland
B	10	Collocated + remote work	Finland
C	13 (6 + 4 + 3)	Distributed at 3 sites	Norway, Czech Republic, Finland

For this pilot research project, we have focused on establishing data collection and analysis methods to empirically evaluate and verify the claims made by agile software development practices. Another goal for the pilot study is to improve our research framework and methods for the main study starting early 2010.

In order to understand the mechanics of agile software practices, we collected the total communication records (both face-to-face, electronic communication and project repositories) for three software development projects. In addition to communication data, we collected metrics for project outcomes and for group dynamical factors in the teams by conducting multiple surveys within the teams. Furthermore, we interviewed the key personnel in the studied projects to gain deeper understanding on the studied projects, the practices applied in the project and to verify our own observations on the

projects.

5. EXPECTED RESULTS

By this research we expect to create understanding and find answers to several questions related to interconnectedness of communication patterns in agile teams, group dynamics, and eventual project success. In general we will explore whether the frequency and patterns of communication in agile team are positively related to project success and the role of group dynamics in this relation.

This question will be studied by focusing on several sub-questions, including but not limited to:

- What is the amount and frequency of communication agile teams compared to traditional teams?
- What are the communication patterns in the studied agile teams?
- What are the effects of media richness on communication in distributed agile teams projects?
- Does certain kind of communicative behavior predict project success?
- What are the relations between communicative behavior and group dynamical factors in the studied teams?
- Is it possible to distinguish agile teams from traditional teams based on some group dynamical factors?
- Does some group dynamical factors predict project success of agile projects?
- What is the "nature" of work in agile software teams?
- Why agile software development methods help distributed projects succeed better?

6. ACKNOWLEDGEMENTS

We would like to thank Human Dynamics Group at Massachusetts Institute of Technology Media Lab (Sandy Pentland, Daniel Olguin Olguin, Ben Waber and Taemie Kim) for collaboration with the data collection, especially by providing us the sociometric badges for this study. This study was partially funded by the Finnish Funding Agency for Technology and Innovation (TEKES) and Academy of Finland.

7. REFERENCES

[1] T. Allen. *Managing the Flow of Technology*. MIT Press, 1977.

[2] K. Beck, M. Beedle, A. van Bennekum, A. Cockburn, W. Cunningham, M. Fowler, J. Grenning, J. Highsmith, A. Hunt, R. Jeffries, J. Kern, B. Marick, R. C. Martin, S. Mellor, K. Schwaber, J. Sutherland, and D. Thomas. Manifesto for agile software development, May 2001.

[3] R. L. Daft and R. H. Lengel. Information richness: a new approach to managerial behavior and organizational design. In B. Staw and L. Cummings, editors, *Research in Organizational Behavior*, pages 191–233. JAI Press, Greenwich, Connecticut, 1984.

[4] C. W. Downs and M. Hazen. A factor analysis of communication satisfaction. *Journal of Business Communication*, 14:63–74, 1977.

[5] L. Freeman. Centrality in social networks: Conceptual clarification. *Social networks*, 1(3):215–239, 1979.

[6] N. Gillespie. Measuring trust in working relationships: the behavioural trust inventory. paper presented in the Academy of Management Meeting, Seattle, Aug 2003.

[7] D. Gladstein. Groups in context: A model of task group effectiveness. *Administrative Science Quarterly*, 29:499–517, 1984.

[8] P. Gloor and Z. Yan. Tecflow - a temporal communication flow visualizer for social network analysis. In *ACM CSCW Workshop on Social Networks. ACM CSCW Conference*, 2004.

[9] P. A. Gloor, D. Oster, J. Putzke, K. Fischback, D. Schoder, K. Ara, T. J. Kim, R. Laubacher, A. Mohan, D. O. Olguin, A. Pentland, and B. N. Waber. Studying microscopic peer-to-peer communication patterns. In *Proceedings of the 2007 Americas Conference on Information Systems*, 2007.

[10] R. E. Grinter, J. D. Herbsleb, and D. E. Perry. The geography of coordination: dealing with distance in R&D work. In *Proceedings of the international ACM SIGGROUP conference on Supporting group work*, 1999.

[11] J. D. Herbsleb, A. Mockus, T. A. Finholt, and R. E. Grinter. Distance, dependencies, and delay in a global collaboration. In *Proceedings of the 2000 ACM conference on Computer supported cooperative work*, pages 319–328, 2000.

[12] J. D. Herbsleb, A. Mockus, T. A. Finholt, and R. E. Grinter. An empirical study of global software development: distance and speed. *ICSE*, 2001.

[13] M. Hoegl, K. Weinkauf, and H. G. Gemuenden. Interteam coordination, project commitment, and teamwork in multiteam r&d projects: A longitudinal study. *Organization Science*, 15(1):38–55, 2004.

[14] H. Holmstrom, E. O. Conchuir, P. J. Agerfalk, and B. Fitzgerald. Global software development challenges: A case study on temporal, geographical and socio-cultural distance. In *Global Software Engineering, 2006. ICGSE '06. International Conference on*, pages 3–11, Oct. 2006.

[15] F. Mael and B. Ashforth. Alumni and their alma mater: A partial test of the reformulated model of organizational identification. *Journal of Organizational Behavior*, pages 103–123, 1992.

[16] F. P. Morgeson and S. E. Humphrey. The Work Design Questionnaire (WDQ): developing and validating a comprehensive measure for assessing job design and the nature of work. *Journal of Applied Psychology*, 91(6):1321–1339, 2006.

[17] R. T. Mowday, R. M. Steers, and L. W. Porter. The measurement of organizational commitment. *Journal of Vocational Behavior*, 14:224–247, 1979.

[18] D. O. Olguin, B. N. Waber, T. Kim, A. Mohan, K. Ara, and A. S. Pentland. Sensible organizations: Technology and methodology for automatically measuring organizational behavior. *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics*, 39(1), 2009.

[19] M. Paasivaara, S. Durasiewicz, and C. Lassenius. Distributed agile development: Using scrum in a large project. In *IEEE International Conference on Global Software Engineering*, pages 87–95, Aug. 2008.

[20] M. Paasivaara, S. Durasiewicz, and C. Lassenius. Using scrum in distributed agile development: A multiple case study. In *IEEE International Conference on Global Software Engineering*, 2009.

[21] M. Paasivaara and C. Lassenius. Could global software development benefit from agile methods? In *ICGSE '06: Proceedings of the IEEE international conference on Global Software Engineering*, pages 109–113, Washington, DC, USA, 2006. IEEE Computer Society.

[22] D. E. Perry, N. A. Staudenmayer, and L. G. Votta. People, organizations, and process improvement. *IEEE Software*, 11(4):36–45, 1994.

[23] D. Read. Iterative development: Key technique for managing software developments. In *ICT WA 2005 Conference*, 2005.

[24] W. W. Royce. Managing the development of large software systems: Concepts and techniques. *TRW Software Series SS-70-01*, 1970.

[25] K. Schwaber. *Agile Project Management with Scrum (Microsoft Professional)*. Microsoft Press, March 2004.

[26] G. M. Speitzer and A. K. Mishtra. Giving up without losing control: trust and its substitutes' effect on managers' involving employees in decision-making. *Group & Organizational Management*, 24(2):155–187, 1999.

[27] L. A. Williams and R. R. Kessler. All I really need to know about pair programming I learned in kindergarten. *Commun. ACM*, 43(5):108–114, 2000.

[28] L. Wu, B. N. Waber, S. Aral, E. Brynjolfsson, and A. S. Pentland. Mining face-to-face interaction networks using sociometric badges: Predicting productivity in an it configuration task. In *Proceedings of the International Conference on Information Systems*, 2008.