Communication Network in an Agile Distributed Software Development Team

Paul T. Robinson
Sony Interactive Entertainment LLC
Amherst, Massachusetts, USA
paul.robinson@sony.com

***Abstract*—Challenges in implementing Agile Distributed Software Development (ADSD) are well documented. A primary challenge lies in achieving the necessary degree of cooperation and collaboration, which are founded on good communication. Agile methodology historically relies on frequent face-to-face communication, which is infeasible in distributed development. Recent work suggests that current communication tools can support practices that accommodate ADSD. We present an experience report of a distributed team and describe its interpersonal communication network, which is notably more limited when connections are across an eight-hour time difference. As a limited study, we provide context to allow better comparisons with other work.**

Keywords—Agile development, global software engineering, virtual teams, distributed teams

I. Introduction

Agile Distributed Software Development (ADSD) is the combination of two widespread trends in software development: global software engineering (GSE), and agile software development. Achieving the necessary degree of communication and collaboration across global sites while still adhering to agile principles [1] poses a variety of challenges. Jha et al. [2], Paasivaara and Lassenius [3], and Paasivaara et al. [4] all report on this. The literature review by Lous et al. [5] examines whether Scrum in particular is *"fit for global software engineering,"* enumerating 45 challenges in 19 categories; they conclude that scaling Scrum is the biggest challenge. In response, companies will naturally adapt their processes to meet the need; Diebold et al. [6] show that Scrum adaptations are common, and Kuhrmann et al. [7] find that this tendency to adaptation is not confined to Scrum. Clearly we should be expecting companies to adapt agile processes to meet GSE needs.

Advances in communications technology and bandwidth make reliance on true face-to-face communication less critical to successful agile processes. A report on distributed pair programming [8] describes a process that is a combination of Scrum and Extreme Programming. Lous et al. [9] describe a distributed team whose process *"is significantly influenced by the agile philosophy, however it is not restricted to a specific method such as Scrum or XP."* We conclude that unmodified Scrum is not suited to GSE, although modified agile processes do appear to succeed.

The case study by Lous et al. [9] examines a development environment intentionally *"crafted to embrace the distributed setup."* That study examined a distributed team with a maximum time difference of one hour; this inspired our more limited investigation in a different context, specifically one with a much larger time difference.

Research question. This research studies the virtual work environment of the compiler team and attempts to replicate some of the investigation done in [9]. Therefore we use a similar research question: *"How well can a work environment support agile distributed software development?"*

Contribution. By building on a previous limited study in a new context, we broaden our understanding of ADSD in a different context, support or contrast with the previous work, and potentially raise new research questions.

II. Context

A. The SIE Toolchain CPU Compiler Team

We report on the Sony Interactive Entertainment (SIE) toolchain CPU compiler development team, which delivers the CPU compiler and some related tools for the Sony PlayStation®4 game-developer toolchain [10]. The compiler team currently has a low-double-digit number of developers distributed across four sites in the US and UK, in time zones up to eight hours apart. (Exact numbers of personnel are not provided, by company request.) These sites are: (i) Bristol, UK; (ii) San Mateo, California; (iii) Campbell, California; and, (iv) one remote developer in Amherst, Massachusetts. The Bristol and Campbell sites are part of SN Systems, a wholly owned subsidiary of SIE that has overall responsibility for the entire toolchain. The San Mateo team is part of SIE R&D; it was instrumental in the adoption of the LLVM[1] open-source compiler for the toolchain and remains tightly involved with its ongoing development. The Amherst developer (and author of this paper) was originally a member of the San Mateo team and still reports there.

The team's development process, called TWINS (This Workflow Is Not Scrum) [11], bears a strong resemblance to Scrumban [12]. All sites share a common task backlog. Although each site has its own development manager, development responsibilities are not organized by site; in fact it is quite common for developers from multiple sites to share an interest in a given area of the compiler, and collaborate on related features.

[1] LLVM is a widely used OSS compiler project. The term LLVM is not an acronym.

978-1-5386-9196-0/19/$31.00 ©2019 IEEE
DOI 10.1109/ICGSE.2019.00015

Table 1: Communication Tools

Slack	SIE uses the enterprise version of Slack; the CPU compiler team has defined several channels (one-to-many communication) on particular topics, and some developers use it for private messages (one-to-one communication).
Outlook	Microsoft Outlook is used for email (one-to-many and one-to-one communication) and calendar.
Webex	Webex is an online videoconferencing system, used by the team for international conference calls; these include retrospectives, site-leader meetings, and presentations.
Telephone	The Campbell, San Mateo, and Amherst developers hold a daily “stand-up” audio-only conference call.
Confluence	Atlassian's Confluence is used for a project wiki, which has pages for meeting minutes, various project procedures, and other relatively “static” information that project members need for reference.
GitHub Enterprise	GitHub Enterprise (GHE) is used for source control and code reviews (via pull requests).
Bugzilla	Bugzilla is used for tracking bug reports.
Jira	Atlassian's Jira is used for managing developer tasks and backlogs.

B. *Compiler Team Communication Practices*

Because we are examining the team’s communication network, we will briefly review their communication practices.

The U.S. sites (San Mateo, Campbell, and Amherst) have a daily telephone conference call that is functionally similar to Scrum's “stand-up.” This meeting takes place at noon Pacific time, which is 3PM in Amherst, so it is at a reasonable time of day for everyone involved. Although the TWINS process calls for it, the Bristol developers do not hold a “stand-up” meeting.

All sites join in a videoconference call for a monthly retrospective, modeled on the Scrum retrospective but done on a time-based cadence rather than per-sprint.

Twice a year, the majority of the team participates in a week-long, face-to-face “code jam” to investigate new ideas, with a significant goal being for developers to collaborate with people they otherwise do not interact with. These interactions encourage forming new connections and generally improve team cohesion. Activities often include developers from outside the CPU compiler team, such as linker or debugger developers. One code jam is normally held in Bristol in the spring, and the other in California in the fall, to share the travel burden.

Table 2: Empirical Context

Attribute	Value
Year	2018
Empirical focus	Empirically based
Empirical background	Industry
Industry sector	Consumer electronics
Subject of investigation	Practitioners
Study results	Successful practices
Empirical research method	Case study
Source of empirical evidence	Questionnaire
Location	Offshore
Legal entity	Insourcing
Geographical distance	Far
Temporal distance	Large (8 hour max)
# of sites	4 (3 USA, 1 UK)
Team size	Low double-digit number of developers

The team uses a variety of tools for non-face-to-face communication, listed in Table 1; apart from Webex and the telephone, these are all largely asynchronous. Jira, Bugzilla, and GHE are the primary task-oriented tools. Email is the main tool for team-wide announcements and information; one-on-one communication is generally via Slack or email depending on the individual’s preference. Recently some sub-teams have started using topic-specific Slack channels rather than email, as channel membership is opt-in and so does not depend on the sender remembering who the interested parties are. The Confluence wiki is viewed as more archival. Webex video calls are used only for formal cross-site meetings, and telephone primarily for the U.S. stand-up.

Work schedules can vary widely, even within a single site; however these are nearly always an individual preference and not “shifting” to add overlap. In fact several of the California developers generally start their work day at noon local time, which is already 8PM in the U.K. On the other hand, a few of the Bristol developers are “night owls” who also prefer to arrive mid-day and work into the evening hours.

III. Data Collection

A. *Methodology*

Our investigation primarily consisted of a single questionnaire. Table 2 characterizes the empirical context to simplify comparisons with similar work. We adapted the taxonomy from [13] following the example of Lous et al. [9]. While the team is technically spread across two corporate entities, in practice they cooperate fully without reference to the corporate hierarchy, and so the “legal entity” is considered insourcing.

Table 3: Questionnaire

Q1	Do you perceive the developers in Amherst, Bristol, Campbell, and San Mateo as a single team or multiple teams? [Multiple choice: 1 team, multiple teams, free text]
Q2	Please rate your satisfaction with regards to the amount of interaction between sites during working hours. [1: I would prefer less interaction; to 5: I would like to have more interaction than we already have]
Q3	If the entire team would be located in one site: how much would you change the way of doing stand-ups/retrospectives? [1: I would change a lot and make it more co-located; to 5: I would not change anything, I prefer the way it is now]
Q4	Whom do you typically contact directly (individually) regarding work? This includes private messages on Slack or e-mail, but excludes group channels, bugzilla, conference calls, social interactions not regarding work, or contact outside work. [Free text]

Data collection was performed by an email questionnaire sent to the developers in December 2018. Table 3 replicates the questionnaire, which focuses on perceptions by team members of cross-site interactions, and their individual communication network. The first three questions are closely adapted from Lous et al. [9], and the fourth repeats their short interview question which was asked of all team members.

B. *Connections Not Studied*

CPU compiler development has separate Quality Assurance (QA) personnel at the Bristol and San Mateo sites. For better similarity with the inspirational study [9], we are not including QA personnel in this study. Distinct from the previous study, the Product Owner for the toolchain compiler is not considered to be a developer, and is not included in this study.

The compiler team is working on the LLVM open-source software (OSS) project, whereas the previous study [9] looked at a team working on an in-house proprietary product. The LLVM project has hundreds of contributors world-wide, some of whom work for other parts of SIE. We have not examined any of the relationships, practices, tools, or communication network involving contributors outside the team under study, even when they work for SIE.

IV. Results

This section presents the results from the questionnaire and communication network reported by the developers. In order not to reveal exact numbers, all data will be reported as percentages of the total. The questionnaire had a 94% response rate. Regarding the communication network, all developers were mentioned by at least one responder, so we have included all of them (responder or not) in the results and discussion.

Table 4: Communication network percentages

Population	**% interactions reported (SIE team)**	**% interactions reported (Lous)**
All developers	38%	67%
Within Bristol	54%	
Within Campbell	90%	
Within San Mateo	67%	
Avg. within a site	64%	70%
Cross-site	28%	64%
Pacific time zone (two sites)	58%	
Across time zones	25%	
Max time-zone difference (8 hrs)	18%	

A. *One Team or Multiple Teams?*

The developers are divided over whether they constitute a single team or multiple teams. Direct responses were split *(Q1. One team: 53%; multiple teams: 47%),* and free-text comments suggested the answer is not simple.

Two U.S. developers who answered "One team" mentioned that this feeling was a bit stronger for the U.S. sites than when considering Bristol. Another developer suggested that "One team" was at least partly influenced by working on the same OSS project: *"The LLVM community structure ties our work together." --Campbell developer.* One developer qualified "One team" with *"multiple subteams […] but not necessarily with geographical borders." –Bristol developer.*

One developer who answered "Multiple teams" said this was *"Largely due to uncertainty and lack of visibility on what others are working on." --Bristol developer.* Another developer who answered "Multiple teams" qualified this as *"Multiple teams working on the same project." --San Mateo developer.*

In summary, it appears that the developers generally feel they are working together, but not necessarily as a single unified team, and perhaps there are communication issues contributing to that qualification.

B. *Degree of Communication*

For the questions about the level of communication between sites, all developers thought it was sufficient or could be higher *(Q2: Neutral (3): 35%; more interaction (4,5): 65%).* It may be worth noting that nearly half the developers who responded "One team" to Question 1 still thought more communication would be better.

When asked if stand-ups/retrospectives should be done differently, assuming all developers were co-located, the majority thought there should be at least some changes *(Q3: No change (5): 29%; some or a lot of change (1-3): 71%).*

This reinforces the finding from Question 2 that many developers think more communication would be preferable.

C. *Communication Network*

We asked developers to list those other developers with whom they communicated directly on work-related topics. We used the information to build Table 4, summarizing all these interactions. We find that 38% of all possible inter-personal interactions were reported. Subsetting the data in the table reveals several interesting characteristics.

Same-site connections. The "same-site" network for just the co-located developers in Campbell had 90% of the possible local connections reported, suggesting they form a tight-knit group. The San Mateo site is also very well connected with 67% of local connections reported. The Bristol site has a somewhat less dense network at 54% of possible local connections.

Same-time-zone connections. As the time-zone differences between some sites are significant (up to eight hours), it is worth comparing connections within a time zone to those across time zones. The Campbell and San Mateo sites are both in the Pacific time zone in California. If we treat these two as a single site, we see 58% of the possible Pacific time zone (same or cross-site) connections are reported.

Cross-time-zone connections. The complement of the same-time-zone analysis shows us that only 25% of cross-time-zone connections are reported. Breaking this down further, Amherst to California (3-hour difference) shows 56% of connections reported, while Amherst to Bristol (5-hour difference) has 50% reported. Looking only at the most distant connections, i.e. between the California and Bristol sites (8-hour difference), we find just 18% of possible connections reported.

We can also look at the full set of U.S. sites (up to 3-hour difference), by taking the same-time-zone results for California plus the Amherst connections to those sites, and find 58% of connections reported.

V. Discussion

Cooperative work [14] requires coordination between workers, which inevitably requires communication. While media richness theory [15] tells us face-to-face communication has maximum throughput, this is not necessarily the optimal task/technology fit [16].

Lous et al. [9] provide a case study of a distributed team taking advantage of technology and tools to maintain a good communication network across three sites with up to a one-hour time-zone difference. This prior study did not publish communication network percentages, but did publish the entire network as a table, from which we have computed the values in Table 4 for easier comparison with our work.

The present study follows on to this prior work to examine a larger team with some much larger time-zone differences. It is worth noting that the number of possible inter-personal connections varies with the square of the population, so perhaps percentage of possible connections is not the best metric when comparing teams of different sizes. But, given the constraints on presenting raw data for our study, we are hard pressed to find another metric.

In our study, the percentage of connections varies significantly with time-zone differences. The Pacific time-zone value is 58%, not strikingly dissimilar from the prior study's cross-site value of 64% (with a maximum one-hour time difference), or even the prior study's whole-team value of 67%. This reinforces the conclusions from the previous study that face-to-face communication is not a prerequisite for developer communication. While there is only one Amherst developer, and so the conclusion must be tentative, we still find 56% across a three-hour difference and 50% across a five-hour difference, not far reduced from the same-time-zone value for the California sites.

On the other hand, the communication network across an eight-hour time difference has merely 18% of the possible connections in place, despite the common task backlog and other explicit efforts to avoid a geographical division of labor. This strongly suggests that a large time difference is a significant communication barrier, which is not a surprising result given prior research on time separation and coordination [17]. The developers are mindful that more communication would be beneficial, based on the questionnaire results reported in Section IV.B. Anecdotally, the relatively recent (less than two years) addition of Slack to the communication toolset lowers the cost of short messages compared to email, and so appears to be facilitating more casual conversations.

VI. Conclusions and Future Work

Agile Distributed Software Development (ADSD) is not simple, but is a perhaps inevitable and certainly ever more widespread convergence of the long-standing agile and global software engineering trends. Accommodating the needs of both aspects requires careful attention from organizations in order to succeed at this convergence.

We found that communication within an ADSD team does not require much face-to-face contact, when other appropriate modes of communication are available and time-zone differences are small; this supports some of the findings made by Lous et al. [9]. However, there is a distinct difference in the completeness of communication networks within a time zone versus those across the most distant time zones (eight hours apart). We do not have sufficient data to characterize networks at intermediate distances.

While this report alone does not allow us to reach a broader conclusion, it leads to questions regarding practices and supporting tools that can ease or improve communication across more widely distributed global development teams.

Acknowledgment

P.T.R. thanks Sarah Beecham for encouragement to write this paper, as well as the anonymous reviewers and Marcelo Cataldo for invaluable suggestions for improvements. PlayStation is a registered trademark of Sony Interactive Entertainment Inc. "PS4", "PS3", and "PS Vita" are trademarks of the same company.

REFERENCES

[1] 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," 2001. [Online]. Available: http://www.agilemanifesto.org. [Accessed 2018].

[2] M. M. Jha, R. M. F. Vilardell and J. Narayan, "Scaling agile Scrum development: Providing agility and quality to platform development by reducing time to market," in *11th International Conference on Software Engineering*, 2016.

[3] M. Paasivaara and C. Lassenius, "Scaling Scrum in a large globally distributed organization: A case study," in *11th International Conference on Global Software Engineering*, 2016.

[4] M. Paasivaara, B. Behm, C. Lassenius and M. Hallikainen, "Large-scale agile transformation at Ericsson: A case study," *Empirical Software Engineering,* vol. 23, no. 5, pp. 2550-2596, 2018.

[5] P. Lous, M. Kuhrmann and P. Tell, "Is Scrum fit for global software engineering?," in *12th International Conference on Global Software Engineering*, 2017.

[6] P. Diebold, J.-P. Ostberg, S. Wagner and U. Zendler, "What do practitioners vary in using Scrum?," in *16th International Conference XP*, 2015.

[7] M. Kuhrmann, P. Diebold, J. Münch, P. Tell, V. Garousi, M. Felderer, K. Trektere, F. McCaffery, C. R. Pause, E. Hanser and O. Linssen, "Hybrid software and system development in practice: Waterfall, Scrum, and beyond," in *Proceedings of the International Conference on Software and Systems Processes*, 2017.

[8] M. Rajpal, "Effective distributed pair programming," in *13th International Conference on Global Software Engineering*, 2018.

[9] P. Lous, P. Tell, C. B. Michelsen, Y. Dittrich, M. Kuhrmann and A. Ebdrup, "Virtual by design: How a work environment can support agile distributed software development," in *13th International Conference on Global Software Engineering*, 2018.

[10] P. T. Robinson, "Developer Toolchain for the PlayStation(r)4," November 2013. [Online]. Available: http://llvm.org/devmtg/2013-11/#talk8. [Accessed 24 August 2018].

[11] P. T. Robinson and S. Beecham, "TWINS--This Workflow is Not Scrum: Adapting agile for open source interaction," in *Proceedings of the International Conference on Software and Systems Processes*, 2019.

[12] A. Reddy, The Scrumban [R]Evolution: Getting the Most Out of Agile, Scrum, and Lean Kanban, Addison-Wesley, 2016.

[13] A. R. Techio, R. Prikladnicki and S. Marczak, "Reporting empirical evidence in distributed software development: An extended taxonomy," in *10th International Conference on Global Software Engineering*, 2015.

[14] K. Schmidt, "Cooperative work and its articulation: Requirements for computer support," *Le Travail Humain,* vol. 57, no. 4, pp. 345-366, 1994.

[15] R. L. Daft and R. H. Lengel, "Organizational information requirements, media richness and structural design," *Management Science,* vol. 32, no. 5, pp. 554-571, 1986.

[16] D. L. Goodhue and R. L. Thompson, "Task-technology fit and individual performance," *MIS Quarterly,* vol. 19, no. 2, pp. 213-236, 1995.

[17] J. A. Spinosa and E. Carmel, "The Impact of Time Separation on Coordination in Global Software Teams: a Conceptual Foundation," *Software Process Improvement and Practice,* vol. 8, pp. 249-266, 2003.