Enhancing the Performance of Software Development Virtual Teams through the Use of Agile Methods: A Pilot Study

Saggi Nevo
University at Albany
snevo@albany.edu

InduShobha Chengalur-Smith
University at Albany
shobha@albany.edu

Abstract

This paper develops a conceptual model that explicates the role of synchronous communication media in enabling – directly and indirectly, via social presence – virtual software development teams to adopt and apply Agile methods. In turn, Agile methods, as well as perceived social presence, are theorized to have a positive impact on communication convergence and transactive memory. Ultimately, these outcomes are formulated as direct antecedents of virtual team performance. A pilot study of 40 Free/Libre Open Source Software (FLOSS) teams provides preliminary supporting evidence for the conceptual model.

1. Introduction

A virtual team is a geographically distributed organization whose members are coherently bound by a shared interest or goal, and who communicate and coordinate their projects through information communication technologies (ICTs) [2; 40]. There are clear benefits to the geographic and cultural diversity of virtual teams, such as facilitating around-the-clock work and increased innovation [58]. However, such arrangements often result in lower levels of commitment and urgency, leading to long periods of latency [20]. Using geographically-dispersed Free/Libre Open Source Software (FLOSS) teams, this paper aims to discover conditions that improve performance in virtual software development teams.

Historically, the use of Agile methods has led to improved performance in manufacturing teams and has been extended to increase the responsiveness of supply chains [9] and even entire organizations [21]. Recently, Agile methods such as XP and Scrum have been applied to software development [1; 31]. Although not all Agile methods are feasible for geographically distributed software development teams, even partial deployment of these methods has been found to increase morale and commitment among team members [48].

Several researchers have argued that to be effective, virtual teams need to meet face-to-face intermittently [e.g., 4]. Yet, for some virtual teams face-to-face meetings may never happen. In this paper, we propose that by leveraging the media synchronicity capabilities of ICT to enable the use of Agile methods in virtual settings, it may be possible to overcome some of the obstacles faced by virtual software development teams.

The remainder of this paper is organized as follows. First, we lay the foundations for our research model by summarizing the literatures on virtual teams and Agile methods. Next, we introduce the theories on which we base our hypotheses, viz. media synchronicity, social presence, and transactive memory. This is followed by a description of our research method and data collection. We conclude with a discussion of our results.

2. Literature review

Unlike traditional co-located teams whose members collaborate mostly by communicating face-to-face, members of virtual teams are faced with the challenges of relying on ICTs to communicate and work with one another. Since they often lack a system of mutual understanding based on a common context derived from a shared location, virtual teams frequently suffer from coordination problems and weak group dynamics [30].

Despite continuous advancements in ICTs (e.g., e-mail, chat rooms, instant messaging, and virtual worlds), virtual teams (including virtual software development teams) continue to struggle with task coordination and project control [14; 31; 34]. For example, among virtual FLOSS teams, previous research has observed loss of motivation, commitment issues, and drop in participation [20].
Our contention is that the performance of virtual software development teams can be enhanced through the use of Agile methods that are enabled through the appropriation of relevant ICT.

2.1. Agile methods

Olson and Olson [52] noted that virtual organizations increase effectiveness as their members adapt themselves to changing socio-technical systems with enhanced ICTs. Accordingly, we submit that certain capabilities of ICT, in particular media synchronicity, may afford team members the ability to capture some of the benefits of face-to-face meetings, while retaining the advantages of virtualness. Specifically, we propose that the use of synchronous ICT would enable the application of Agile methods by virtual software development teams, thereby leading to improved outcomes (e.g., team performance).

Agile methods are rooted in concepts of continuous improvement from Lean Manufacturing and Six Sigma approaches [49]. For co-located software development projects, applying Agile methods may be manifested by setting up self-organized teams working in short iteration cycles that result in emergent requirements and incremental functionality [39]. Agile-based software development takes place through a series of sprints, which are short iterative development sub-projects. Agile methods involve teams that work together in close proximity, with frequent reflection and feedback on practices [10]. Teams that practice Agile methods are generally small in size, usually between five and nine members who typically work in a single, large, open space to facilitate communication. This enhanced communication helps to develop tacit knowledge within the team that substitutes for more formal documentation and allows for quicker response to changes. Team members participate in daily stand-ups where they briefly report on their previous progress and their current plans. The emphasis is on short status updates and quick feedback which promote accountability and transparency, characteristics that Agile teams highly value [54].

Agile teams are self-organized in the sense that rather than following the directions of a separate management group, the team members themselves prioritize tasks and make all the decisions pertaining to the coordination and management of the project. Another vital difference between Agile and regular teams is that roles are flexible and members may volunteer to pick up tasks to even out the workload over the duration of a sprint. A sprint typically runs for two to four weeks, but could last anywhere from one week to four months. At the end of a sprint, the team (and in some cases, the customer) reviews the project’s progress and sets goals for the next sprint, allowing the team to be highly adaptive. The idea of continuous improvement is embodied in the retrospective process that takes place at the end of every sprint. The process of reflection and introspection encourages cooperation among the team members and facilitates collaborative decision making [28]. Face-to-face communication, transparency and the sense of a community are key to the success of any Agile team.

Nevertheless, Agile methods are not suitable for all development projects, and using these methods could be a challenge for virtual software development teams since virtual team members are not co-located (e.g., fewer projects are likely to have daily stand-ups). However, it appears that despite these obstacles, an increasing number of global software teams begin to recognize the value in adopting some of these Agile methods and use hybrid development approaches that integrate certain aspects of Agile methods into the development process [31; 24]. We believe that certain ICT can help overcome some of the abovementioned challenges and enable a broader integration of Agile methods into virtual software development teams, thereby increasing the likelihood of project success.

3. Theoretical foundations and model development

Virtual teams are often comprised of individuals with their own idiosyncratic goals and cultures who have only temporary alliances, making the development of a shared understanding a challenging task [26]. Members of virtual software development teams interact primarily via ICT and although there are many accompanying benefits such as improved communication records, opportunities for 24/7 development (i.e., follow-the-sun work schedule), and access to rich skill sets, the reliance on electronic communication media leads to increased communication volume, decreased comprehension, coordination difficulties, delays in project deliverables, and loss of focus [17; 31]. Research on virtual teams has uncovered myriad obstacles that need to be overcome for effective teamwork. These include lack of non-verbal cues, reduced opportunities for serendipitous encounters and informal conversations (e.g., water cooler talks) and hence fewer opportunities for relationships to develop, and difficulties in building consensus and establishing shared meaning [51].
Communication is essential to the success of any team endeavor and this holds true for software development as well. We expect that the use of a synchronous communication media, as part of the virtual work environment, will enable geographically-dispersed teams to better leverage Agile methods and thus improve team performance. We, therefore, propose that adequate appropriation of the communication capabilities of certain ICT can ameliorate some of the obstacles associated with physical distance in virtual software development teams. Specifically, we propose that using synchronous communication technologies such as instant messaging, voice over IP, and videoconferencing will enable such teams to better practice Agile methods, which in turn will enhance process outcomes.

Next, we develop a model that examines virtual team performance in open source software development projects and employs constructs from media synchronicity, social presence, and TM theories (see Figure 1).

3.1. Media synchronicity theory

Media synchronicity theory (MST) examines the fit between the capabilities of communication media and the needs of communication processes and contends that greater fit is associated with improved outcomes [16]. MST proposes five capabilities of communication media, which jointly determine the ability of the medium to support synchronicity – i.e., “a shared pattern of coordinated behavior among individuals as their work together [16; p. 575]”: transmission velocity, symbol sets, parallelism, rehearsability, and reprocessability. Of these five capabilities, the first two are particularly relevant to this study. Transmission velocity refers to the rate at which a communication medium can transmit a message to intended recipients. Symbol sets are the number of ways in which information can be encoded and transmitted by a communication medium. Different media can be characterized in terms of the communication capabilities they possess and the functionalities they enable, which places them along the media synchronicity continuum. For example, videoconferencing is high on transmission velocity and symbol set variety but low on parallelism, rehearsability, and reprocessability whereas e-mail is low on the first two but high on the last three capabilities [43].

MST conceptualizes two communication processes: conveyance and convergence. Conveyance refers to the transmission and processing of new information whereas convergence refers to establishment of the meaning of the information. We argue that the use of ICT to enable conveyance processes is, to a large extent, independent of the team’s arrangement. Specifically, co-located teams and virtual teams alike rely on similar asynchronous ICTs to support conveyance processes such as the transmission of large amounts of new information. Accordingly, this paper focuses on convergence processes where virtual teams aim to develop shared understanding and focus, coordinate and control project activities, and assign roles and responsibilities. Compared with co-located teams, virtual teams face greater difficulties obtaining these goals despite their greater use of a variety of communication media [31].

According to MST, two capabilities of communication media are particularly salient to enabling synchronicity and supporting convergence: transmission velocity and symbol sets [16].
Furthermore, media capabilities alone are not sufficient to affect communication convergence and attention must also be paid to their appropriation by the users of the technologies [15; 16; 23]. The argument for the insufficiency of media capabilities alone to explain various outcomes is consistent with the view that it is how ICT are used in the context in which they are used that will explain the myriad outcomes we observe [e.g., 35]. Hence, we contend that communications media that afford instant messaging, audio channels, and non-verbal cues would allow virtual software development teams to seek and obtain quick feedback in a variety of formats and interact and collaborate in an environment that transcends geographical boundaries. Therefore, synchronous media can be leveraged to improve teamwork, self-organizations, and quick adaptation to change – the hallmarks of Agile methods. Hence,

**H1:** The degree to which a virtual software development team uses synchronous communication media is positively related to the degree to which it uses Agile methods.

Note that co-located software development teams often use a hybrid approach to adopting Agile methods [7]. Thus, rather than envisioning a dichotomy of non-Agile versus purely Agile, we can measure the degree of to which a virtual team uses Agile methods along a continuum.

A key challenge for virtual teams involves coordination of projects. In addition, virtual software development teams need to agree on the list of features and functionalities [20; 55]. Although it is difficult to apply Agile methods to geographically-dispersed software development projects, anecdotal evidence suggests that leveraging the capabilities of the appropriate ICT may help virtual teams overcome these challenges [59]. For example, requirements elicitation techniques that are part of Agile methods can be used to brainstorm functionalities and prioritize tasks.

Past research has identified the importance of shared understanding in the area of software development. For example, Curtis et al. [12] note that a critical issue in software development is the establishment of a common understanding of the software requirements and design elements across the project team. They analyzed the transcripts of team meetings and conclude that designers spend large amounts of time trying to develop a shared mental model of the design. In general, as part of the development process, the software team needs to converge on the project goals and the approach it will take to achieve those goals.

Yet, for virtual software development teams, achieving this convergence is a challenge. Naturally, face-to-face meetings allow richer interactions and are helpful in overcoming organizational and cultural differences [22]. However, face-to-face meetings among virtual software developers are rare [11].

According to MST, convergence is likely to result in improved task performance. Most of the activities in Agile software development environments require open discussion and consensus building. For example, during planning meetings, the team uses elicitation techniques such as brainstorming and prioritization [55]. This suggests that practicing Agile methods, made possible by synchronous communication media, is likely to allow virtual software development teams to convergence on the project’s goals and the means to these goals. Thus,

**H2:** The degree to which a virtual software development team uses Agile methods positively affects the team’s communication convergence.

### 3.2. Social presence theory

Social presence is defined as the extent to which a communication medium facilitates awareness of others involved in the communication process while rendering itself ‘invisible’ to the process participants [61]. Social presence theory proposes that during interactions, individuals prefer to perceive others as co-present [41]. Since individuals interact in a variety of ways (e.g., physical, visual, verbal [13; 61]), their ability to communicate realistically in a computer-mediated communication environment depends on two media capabilities – i.e., transmission velocity and symbol sets. At high rates of transmission velocity, computer-mediated communication resembles natural conversation and numerous symbol sets can be used to convey messages in a variety of ways [16]. In addition, these media capabilities can help relay the communicators’ social presence and can provide a variety of important social cues [36; 53], which are important for establishing and maintaining social relationships.

Since many virtual teams never get to meet face-to-face before or during the project, alternative mechanisms may be required to create a sense of social presence. Accordingly, we propose that synchronous communication media may be able to compensate, at least partially, for the lack of face-to-face interactions and create a sense of social presence among members of virtual software development teams. Thus, we foresee a positive link between
media synchronicity and the extent of social presence experienced by members of virtual teams. Hence,

\[ H_3: \text{The degree to which a virtual software development team uses synchronous communication media has a positive impact on the perceived social presence of its members.} \]

While social presence is important for the establishment and support of social relationships [63], past research did not find a relationship between social presence and task performance [e.g., 66]. This lack of evidence may be attributed to the fact that social presence is a media characteristic, albeit a socially derived one [16], which, in and of itself, is insufficient to impact performance outcomes. Ultimately, like other communication media characteristics, social presence must be appropriated and used to leverage communication processes [15; 18]. Accordingly, we propose that interactions among members of virtual software development teams, who use synchronous communication media, which are expected to generate strong social presence, can enable and support the use of Agile methods by virtual teams. Specifically, with greater social presence, which infuses realism into virtual interactions, Agile methods are likely to be seen as a feasible software development methodology for virtual teams. Accordingly,

\[ H_4: \text{The degree of social presence experienced by a virtual software development has a positive effect on the degree to which it uses Agile methods} \]

3.3. Transactive memory theory

Transactive memory (TM) is the notion that, within teams, one’s knowledge is not limited to the information he or she has been able to store, retain, and recall but one can also access other team members’ knowledge [37]. In that sense, other individuals within the same team may serve as external memory repositories [65]. Thus, team members may benefit from one another’s expertise and knowledge if they develop a shared understanding of who knows what in the team. According to the theory, team members can exchange knowledge and leverage each other’s expertise by interacting with one another. Research has shown that TM enables teams to reach higher levels of performance [e.g., 45; 38; 56]. TM was also found to play a positive role in virtual teams [3; 25; 33].

TM develops when team members learn about each other’s knowledge and expertise and mentally encode this information. When the team encounters new knowledge it ‘stores’ it within the domain expert. Later, when this knowledge becomes central to a particular problem, team members can retrieve it by locating the owner of the knowledge [57; 65].

Researchers observed that the absence of certain institutional enablers makes it a challenging task for virtual teams to develop TM [3; 50]. Despite these challenges, some TM was shown to develop in virtual teams, albeit after long periods of time [33]. In the context of software development, we propose that in order to assign roles and responsibilities to a team, high levels of TM are required. A close inspection of the components of TM and how it is developed suggests that the key activities of Agile methods can help virtual software development teams using highly synchronous communication media to develop TM. Specifically, Agile methods can be applied throughout the software development lifecycle, from requirements gathering to product testing. The use of some Agile methods has been shown to reduce the problems associated with temporal, spatial and socio-cultural differences in global software development [31]. As Agile methods de-emphasize documentation, stand-ups are crucial for sharing tacit knowledge with team members. Such activities offer important opportunities to develop TM memory among virtual teams. Therefore,

\[ H_5: \text{The degree to which a virtual software development team uses Agile methods has a positive effect on its ability to develop transactive memory.} \]

Virtual team members are less effective at disseminating information and locating and engaging experts [29]. Prior research has proposed that team members’ co-presence is conducive to the development of TM since it enables confirmation or modification of mental structures via multiple information gathering methods [6]. Accordingly, we propose that virtual software development teams with strong social presence – enabled by synchronous communication media – may be able to compensate for lack of co-presence as far as TM development is concerned. Hence,

\[ H_6: \text{The degree of social presence experienced by a virtual software development has a positive effect on the team’s ability to develop transactive memory.} \]

3.3. Team performance

During the process of software development, team members often question, make suggestions, report bugs, provide feedback and discuss project-related issues, all of which complicate the software
development process and can, potentially, adversely affect its outcomes. In general, it is challenging for virtual software development teams to work on tasks that require close coordination [29]. Therefore, teams that perform well are able to resolve conflicts and solve problems, and these efforts predominantly depend on convergence communication [16]. Hence,

Hypothesis 7: The stronger the communication convergence attained by the virtual software development team, the better its performance.

Prior research has studied the relationship between TM and virtual team performance [e.g., 33]. For conceptual completeness, we present a final hypothesis regarding this relationship but provide no further discussion. Instead, we note that our focus is on the enablers of TM in virtual teams, which have been studied to a lesser extent. Accordingly,

Hypothesis 8: Virtual teams with better transactive memory will enjoy greater team performance.

4. Research methodology

4.1. Sampling frame

Over the past decade, Free/Libre Open Source Software (FLOSS) has gained substantial interest from businesses, governments and researchers as an alternative to proprietary software as well as commercial software development practices. Thus, we posit that finding mechanisms that could increase the effectiveness of FLOSS teams is important for academics and practitioners alike.

Since the source code is in the public domain, FLOSS provides a fount of code components that can be reused by any developer [62]. In addition, the process of developing open source code incorporates best practices such as version control, peer review and issue tracking, processes that are characteristics of large, mature software engineering organizations [62]. Thus the advent of open source software has had a positive influence on software development, both in terms of the final product and in terms of the development process. As FLOSS merges into the mainstream, it becomes increasingly vital to identify methods and technologies that will nourish it. Accordingly, this paper studies FLOSS teams.

4.2. Instrument development

For this pilot study, we developed an instrument using validated scales to measure the key constructs: virtual social presence [42], communication convergence [32], TM [37], and team performance [27]. (All items are presented in the appendix.) In addition, we captured the level of media synchronicity by recording the extent to which different types of communication media were used by the teams (i.e., VoIP, Videoconferencing, and IM). Similarly, we assessed the use of Agile methods by measuring the extent to which various activities associated with this software development approach were performed (specifically, Sprints, Stands-Ups, and XP).

4.3. Data collection

We begin with a preliminary exploration of the relationships among the factors in our model. To do this, we conducted a pilot study within a population of developers involved in virtual FLOSS projects.
Accordingly, invitations to participate in a survey were sent to members of an online community for FLOSS developers. There are no costs associated with joining the community and participation is voluntary. Members of the community represent myriad companies in various industries as is evident from their profiles. A personal invitation [19] to participate in the study was sent by email to 577 randomly chosen members of the community (out of more than 2,000 members). In exchange for participation in the study, respondents were offered a report of the findings. 41 completed questionnaires were received in a single wave, representing a 7.1% response rate. To increase the response rate and the absolute number of responses in the full-scale survey we plan to conduct two waves of data collection (initial and follow-up) and offer a small monetary reward. In addition, a leading Agile methods community has agreed to sponsor our study.

4.4. Data analysis

While PLS can handle small datasets [8] and previous studies have conducted a full-scale path analysis using equivalent sample sizes [e.g., 64], we created summated scales for each construct in the model\(^2\). The structural model of the PLS regression was used for a preliminary testing of the hypotheses. A bootstrapping procedure was used to assess the significance of the hypothesized paths and the amount of variance in the dependent variables attributed to the explanatory variables. The results of the analysis – paths coefficients and R\(^2\) values – are presented in Figure 2.

The results from the structural model generated by PLS support our key arguments for the antecedents of virtual software development team performance. We found empirical support for the hypothesized relationship between the extent of use of synchronous communication media and the extent of Agile methods used by the virtual FLOSS team (H1), and between the former and virtual social presence experienced by the virtual team (H3). We also found support for the hypothesized relationship between the extent of Agile methods used and communication convergence (H2), and between the extent of Agile methods used and TM (H5). Support was also found for the relationship between virtual social presence and Agile methods (H4), and between the former and TM (H6). Finally, communication convergence and TM were found to have a positive effect on virtual team performance (H7 and H8, respectively). All path coefficients were significant at \(\alpha = 0.05\) or better.

5. Discussion and conclusions

The goal of the study reported in this paper was to examine the impact of media synchronicity and Agile methods on virtual team performance. The use of Agile methods in software development requires high levels of interaction and communication, which is posing a challenge for virtual teams intending to adopt these methods. However, we demonstrate that by leveraging the communication capabilities of the appropriate ICT not only can virtual software development teams successfully adopt Agile practices, but their overall performance can be improved as well. In particular, we find that by using ICT with high media synchronicity, such as instant messaging, videoconferencing, and voice over IP, team members are likely to increase their perception of virtual social presence as well as their ability to practice Agile methods.

Media synchronicity posits that the alignment of communication media capabilities with process requirements is likely to lead to improved outcomes. In the context of software development, this paper outlines a mechanism via which an enabling configuration could occur. In particular, leveraging ICT with high media synchronicity capabilities has a direct, as well as an indirect, impact on the use of Agile methods, which in turn increases the TM and communication convergence of the virtual team and ultimately its performance. The overall effect is that high media synchronicity combined with virtual social presence enhances the ability of a team to use Agile methods. The perception of virtual social presence has a direct impact on TM, but when combined with the use of Agile methods, it seems to provide an additional boost to the creation of this tacit resource. Thus, adopting and practicing Agile methods appears to act as a catalyst in harnessing virtual co-presence to generate TM.

Our data show that the use of Agile methods integrated with technologies that enable virtual social presence increases the TM and communication convergence of the team. About half of the variation in team performance can be attributed to the forming of TM and the attainment of communication convergence.

The results suggest that the use of Agile methods is not only feasible, but could also be valuable for virtual software development teams. However their successful implementation is contingent upon

---

\(^2\) Confirmatory factor analysis indicated that the scales exhibit adequate convergent and discriminant validities. Additional tests will accompany the full-scale survey.
leveraging the appropriate ICT. Future research should amass larger datasets to assess the robustness of these findings. The results obtained in this pilot study encourage us to continue to explore the relationships outlined in the conceptual model using a larger sample.

References


[23] Fuller, R.N. and Dennis, A.R. “Does fit matter? The impact of task-technology fit and appropriation on


[27] Hardin, A.M., M.A. Fuller, and Davison, R.M. “I know I can, but can we? Culture and Efficacy beliefs in global virtual teams,” Small Group Research, 38, pp. 130-155.


Appendix – survey items

**Transactive Memory (TM)**
1. I am comfortable accepting procedural suggestions from other team members.
2. I trust that other members’ knowledge about the project is credible.
3. I am confident relying on the information that other team members bring to the discussion.
4. When other team members give information, I want to double-check it for myself.
5. I do not have much faith in other members’ “expertise.”
6. The team works together in a well-coordinated fashion.
7. The team has very few misunderstandings about what to do.
8. The team needs to backtrack and start over a lot.
9. We accomplish the task smoothly and efficiently.
10. There is much confusion about how we would accomplish the task.

**Team Performance**
1. I am satisfied with my team members.
2. I am pleased with the way my teammates and I work together.
3. I am very satisfied working with this team.

**Virtual Social Presence**
The technologies used by the team enable me to...
1. ...feel as if I were present in the same location as the other members (even when I wasn’t).
2. ...engage in synchronous interactions with other team members mirroring the way I would have interacted if I were face-to-face.
3. ...easily call for impromptu communication between team members.

**Communication Convergence**
1. Effectively debate [project] issues.
2. Effectively resolve disagreement on [project].