# Can Agile Collaboration Practices Enhance Knowledge Creation Between Cross-Functional Teams?

Carine Khalil[1], Valérie Fernandez[2], and Thomas Houy[2]

[1] Ecole Centrale, Grande Voie des Vignes,
92295 Châtenay-Malabry, France
carine.khalil@ecp.fr

[2] Télécom ParisTech, 46 rue Barrault,
75013, Paris, France
{valerie.fernandez, thomas.houy}@telecom-paristech.fr

**Abstract.** Agile philosophy emphasizes constant interactions and close collaboration between team members. This emergent management philosophy relies on a set of practices that aim at creating an environment in which teams are able to respond rapidly to customer's needs and to deal effectively with changing situations. From this perspective, agile practices can be viewed as a way to enhance knowledge creation and knowledge sharing between team members. However these emergent practices necessitate an environment that facilitates communication and coordination mechanisms. The present paper aims at analyzing how large organizations, characterized by distributed and cross-functional teams, can cultivate an agile environment where inter-individual knowledge exchanges are encouraged. Even though mutual adjustments and face to face interactions are not easily achieved in large and distributed organizations, the contributions of agile practices in such contexts remain significant. These practices can foster knowledge development and collective learning processes and subsequently improve organization's adaptability.

**Keywords:** Agile Methods, Cross-Functional Teams, Learning Organization, Collaboration Practices.

## 1 Introduction

In dynamic and competitive environments, "traditional" software methodologies, based on comprehensive planning, detailed documentation and design have been progressively questioned by a number of practitioners. These methods are considered as unable to deal with rapid technological innovations and changing demands. In this respect, a new style of software methodology called agile methods has emerged and gained popularity within software industries. Literally, the "agile" concept refers to "having a quick resourceful and adaptable character". In software development, agile

methods consist on delivering, at the end of each iterative cycle[1], a functional version of the product. This iterative mode enhances feedback and adaptation to continuous changes and seems to better satisfy customers' demands by delivering a concrete version of the product with the required set of functionalities.

Different empirical studies have been conducted lately assessing the development methods used by agile practitioners. These surveys highlight the growing adoption level of agile practices and tools (Ambler[2], 2008; Version One[3], 2010; Enquête Nationale[4], 2011). These emerging management and development methods seem to improve time-to-market, team productivity and product quality.

Agile methods rely on a set of practices and tools that aim at creating an environment in which development teams are able to respond rapidly to customer's demands and to deal effectively with changing situations. Therefore, agile development advocates constant interactions and exchanges between team members. It is an attempt to bring people together and to enhance collective practices and achievements. From this perspective, agile practices can contribute to knowledge development. They convert new knowledge through shared experiences. However, these collaboration practices are more adequate to organizational contexts where communication and coordination mechanisms can be easily achieved. In other words, in a workplace where team members are collocated, spend time together, often discuss and interact with each other. What about large organizations characterized by transversal project teams? How can agile practices foster knowledge creation processes in such contexts?

Few empirical studies have focused on how agile practices embrace knowledge development in large organizations. We adopted an ethnographic approach in order to address these relevant questions and better understand the context in which the study was taking place. Accordingly, we conducted a set of semi-structured interviews with cross-functional team members that are using agile collaboration practices. A general inductive approach has been adopted for analyzing and interpreting the collected data.

In the following section (2), we will highlight the role of knowledge creation in agile environments and we will review the work related to scrum practices. In section (3), we will present the research methodology and the context of the study. We will portray the results of the research work in section (4). And finally, we will discuss the findings and the limitations of the study in section (5).

## 2 Knowledge Creation in Agile Environments

In today's business environment, software industries require a flexible organization that allows project teams to adapt quickly to inevitable changing demands and re-

[1] The main idea of an iterative approach is to divide the development cycle into simpler and more manageable units. Each unit is analyzed, designed and implemented in order to produce an executable deliverable with a limited set of functionalities. The final executable deliverable encompasses all the functionalities expected from the system.

[2] http://www.ambysoft.com/surveys/agileFebruary2008.html.

[3] http://www.versionone.com/pdf/2010_State_of_Agile_Development_Survey_Results.pdf

[4] http://fr.slideshare.net/xwarzee/enqute-2011-vous-votre-organisation-et-agile

quirements. In this respect, software practitioners have suggested a set of relatively new management practices that address the challenges faced in unpredictable environments where reactivity and adaptability become fundamental. These practices, called "agile" practices, stress on close collaboration and frequent feedback between team members and the client. They encourage experience and knowledge sharing and subsequently increase the capabilities of team members to cope with uncertain and ambiguous situations.

According to [1], organizational knowledge is created through the continuous social interaction of tacit and explicit knowledge. While explicit knowledge can be codified and transferred through formal communication or documentation (theoretical approaches, problem-solving, manuals and databases), tacit knowledge is more difficult to transfer. It involves both cognitive (mental models of the world) and technical elements (concrete know-how and skills). Dialogue is an important means for elicitating and translating the tacit knowledge into a readily understandable form [2]. Agile methods denote a social approach where groups and individuals constantly interact with each other, collectively construct meaning of the ongoing flow of experience and act accordingly. These methods rely on a set of collaboration practices that promote inter-individual exchanges and enhance knowledge creation and sharing. Thus, agile organizations can thrive and respond better to customer's demands [3-4]. The following section introduces rapidly the scrum method and stresses on how this method helps organizations to achieve sustainable competitive advantages.

### 2.1 Scrum: Principles, Practices and Management Tools

Different agile methods exist. In the present work, we will focus on scrum method. Scrum is an iterative and incremental approach for managing projects. Three pillars sustain the development process: transparency, inspection and adaptation [5]. The main purpose of scrum is to foster team productivity by providing "light" management practices (pre-sprint, daily sprint, retrospective meetings, product backlog, burndown charts, etc.) and creating an environment where teams can easily communicate and adapt to changes. Through continuous feedback and interactions, scrum provides a context where project teams are predisposed to combining and creating knowledge. Hence, communication is considered a critical factor for scrum teams. Table 1 summarizes the characteristics of scrum.

**Table 1.** Components[5] of Scrum method

| **Scrum** | |
|---|---|
| Principles | Transparency, Inspection and Adaptation. |
| Roles | Scrum Master, Product-Owner, Scrum Team |
| Management Practices | Pre-sprint, Sprint Planning Meeting, Sprint, Post-Sprint Meeting and Retrospective Meeting. |
| Management Tools | Product Backlog, Sprint Backlog, Burndown charts |

[5] Scrum practices, tools and roles are defined in Annex I.

### 2.2 Collaboration Practices with Scrum: The Related Work

Different empirical studies have examined the impact of scrum on team collaboration. In this research paper, we stress on three scrum management and collaboration practices in order to explore their impact on knowledge creation: daily scrum, iterative development and scrum whiteboard. Daily meetings such as "daily scrums" strengthen the communication between team members [6-7], improve information sharing [8] and collective problems solving [9-10]. They clarify the status of the on-going operations and sheds light on the difficulties encountered throughout the project. It also ensures a better control of the project and provides a coordination mechanism for everyone in the project [11]. However, in distributed environments, frequent and informal communication is hard to achieve impacting the collaboration between team members [12] and the pursuit of a common goal [13]. Furthermore, daily scrums are difficult to realize across long distances and geodistributed teams [14-16]. In such environments, team members should be equipped with different communication medias and information technologies in order to facilitate their direct communication and documents sharing [17-18]. In addition to the previous collaboration practice, the iterative development encourages collaboration between team members and the client [6], [14]. This practice adds agility to the development process by providing rapid feedback on the implemented functionalities [19]. It facilitates the monitoring of the project progress [15] and improves organizational learning [20] by incorporating feedback into future iterations [8]. Some practitioners stated that frequent iterations combined with client voice can lead to successful results [20]. However, it is hard to manage iterations that run in parallel, on different geographical sites [19], [21]. Least but not last, the whiteboard plays an important role in the organizational learning processes. The use of this tool enables team members to have a sharing vision of the project requirements [22-23] and create an informative workspace [9-10]. However, physical distance across teams constrains the information exchanges. These empirical studies stress on how agile practices promote feedback and collaboration, encourage information exchanges between team members and improve collective problems-solving. An agile organization is therefore considered as a continuous flow of organizing and learning processes where frequent communication and collaboration are fundamental. However, these practices are not easily implemented in large organizations characterized by cross-functional and geodistributed teams. Up till now, few studies have investigated the way agile practices can develop organizational learning in distributed environments. Our research work aims at understanding, from cross-functional teams' perspective, how agile practices can foster the mutual development and the sharing of tacit knowledge within their organization.

## 3 The Research Context

### 3.1 Context of the Study

The case study was carried out in a division of a French telecommunication company.

The organizational structure of the studied entity can be categorized as a lightweight[6] structure [24] characterized by distributed teams. The project manager coordinates the activities of his team members and facilitates the information exchange between parts of the organization. Projects are generally large, involving approximately thirty five persons each. They combine cross-functional actors that are hierarchically attached to different functional managers and they intervene temporarily on different projects in parallel. This French entity operates in a dynamic and competitive technological environment necessitating highly adaptive project management systems. In such a turbulent context, the top management decided to implement new management practices in order to improve the team cohesion in addition to the project transparency. The lack of communication between cross-functional teams increased the need for tighter collaboration. In this respect, top management decided to implement, within the project teams, a set of scrum practices that emphasize communication and collaboration. This includes daily meetings, virtual whiteboards and kaizen[7] sessions.

### 3.2 Methodology

We adopted a qualitative approach based on an instrumental case study [25]. We focused on a single case study to investigate, in-depth, the context settings (physical, organizational and technical conditions) in which project teams operate. 15 interviews were conducted with project team members. The interviewees were selected based on their will to participate in the study. The interviews were semi-structured and each lasted one hour. The interviews were recorded with the approval of the interviewees who were explicitly informed about the purpose of the study. These interviews aimed at identifying how project teams perceive the implementation of agile practices within their organization and how such collaboration practices influence knowledge creation and capitalization between the teams. We also carried out informal discussions with the participants throughout the study period. Complementary data such as e-mails and documents were also collected to enhance our understanding of the context. For data analysis, we adopted an interpretive approach. We began with multiple readings of our field notes to better understand the context in which the project was taking place. The research question "How can agile practices foster knowledge creation processes in such contexts?" has guided us in identifying the key concepts in each sentence or/and paragraph. These meaningful segments were classified into categories, where, each refers to a particular meaning. A set of inductive categories were subsequently defined

[6] The lightweight team structure is an organizational form typically found in large, mature firms. The team members physically reside in their functional areas where each functional organization has a representative in the project team. The project manager, who is typically chosen out of the function that is most vested in the development process, is responsible for coordinating the activities of the different functions. However, the project manager has little authority and influence. The project team members remain under the control of their respective functional managers.

[7] It's a Japanese term that means continuous improvement. Kaizen events consist of gathering operators, managers, owner of a process in one place, mapping the existing process in order to improve it.

and justified with verbatim. Among these, we cite the following: management issues, organizational structure, team composition, inter-individual interactions.

## 4 Results

The following section describes the way the interviewees make sense of agile practices and their impact on the software development process. Even though project team members are aware of the benefits of these collaboration practices, they express scepticism towards their implementation within large organizations and distributed teams. A set of contextual factors has been identified as barriers to knowledge sharing and development.

The geographical distribution of the project teams is considered to be a challenge for gathering the whole team and fostering experiences exchanges between team members. Thus, the lack of face-to-face interactions constrains tacit knowledge capitalization. Even though information and communication technologies enable real-time communication and document exchanges, they cannot replace direct contact where tacit knowledge can be converted to explicit knowledge and transmitted. In addition, the creation of a virtual information environment is very demanding. Storyboards must be often updated and controlled in order to enable smooth coordination between distributed teams.

"*We are geographically distributed and there are a lot of cross-functional animations… It's not possible to share our daily experiences if we are distributed geographically… especially if we meet once a week. It doesn't promote close collaboration*" (project manager); "*Information and communication technologies can help us share documents and communicate over distance but it's different when the team is colocated….The geodistribution can skew the information*" (product development manager)

Large scale projects are also viewed as a challenge for implementing collaboration practices. The implementation of daily meetings that encompasses all the project team members was challenging for the project managers. The project managers aim to reduce the number of participants in order to respect the fifteen minutes time-boxed meeting. Consequently problem sharing and capitalization could not be done properly due to the non-participation of some key members. The interviewees were not encouraged to implement additional meetings. The big number of existing meetings and their long duration discourage the participants to attend supplementary ones. Moreover, kaizen sessions necessitate the involvement of the whole team to make them successful. According to the interviewees these sessions are not efficient if they are done in an isolated way.

"*Managing large teams is so challenging… I can't see how I can include all the project team in one meeting… it's not possible unless the meeting lasts for several hours… In small colocated teams, actors can directly deal with their neighbours if any problem occurs ... In large projects it's different"* (project manager); *"The way we run kaizen sessions cannot optimize the continuous improvement… it is absolutely necessary to involve the entire team"* (project manager).

Another contextual factor we identified is the team's composition. It seems to influence the knowledge development and transfer between project team members. The involvement of project members in different projects simultaneously constrains frequent exchanges and knowledge creation. The lack of time resources disables their participation in collective activities that promote knowledge sharing. Teams couldn't attend the daily meetings organized by the project manager. Furthermore, the updating of the virtual storyboard and its sharing between the teams were also difficult. The creation of a common and a well structured database requires a constant control that guarantees high data quality. Yet, in the studied context, the common database was not well managed and organized. There were missing reports and documents.

*"It is difficult to promote knowledge sharing since each team manages its own planning .... There is a movement within the team... people intervene at some point and then they move out which leads to a loss of information... "* (project manager); *"we have some teams that externalize some of their work which decrease the project visibility"* (project manager); *"the existing database is not reliable…We don't have a system that verifies the data entry"* (project manager).

In addition to the cited contextual factors, the lack of authority of the project manager was also perceived as a challenge for encouraging collective learning processes. The coordination of different functional teams has limited the circulation of instruction and information. It was difficult to foster collaboration and interactions between different functional teams.

*"The role of the project manager is limited to an orchestra leader …we don't control the activities of our project teams…each one of them has its own constraints and priorities"* (project manager).

Thus, the implementation of agile practices within transverse teams and geodistributed environment is challenging. Agile practices necessitate a structuring context, frequent communication and continuous feedback. Furthermore, these practices necessitate an organization where project teams work under the authority of a project manager. The studied actors highlighted their preoccupations regarding the context in which they operate. The project size, the organizational structure and the team composition seem to influence the efficiency of agile practices and their impact on collective learning processes. The perceived usefulness of agile practices is not sufficient to successfully integrate these practices within project teams. The creation of a collaborative learning environment is very demanding when the teams are distributed and involved in different projects at the same time.

Our interpreted data underlined the challenges faced by actors in a lightweight organizational structure. In this respect, we can imagine a reorganization of the studied context that optimizes the use of agile tools and practices. All key members must participate to daily meetings and kaizen sessions in order to ensure knowledge capitalization and collective problem-solving.

## 5 Conclusion

The field notes have shown that the organizational context can be challenging while implementing agile practices and tools. Therefore, we believe that beyond the contex-

tual factors, it seems fundamental to integrate agile methods as structured learning approaches. The organizational agility can be achieved through the capability of its members to rapidly reconfigure their resources and adapt to changes. Agile practices can be viewed as contributors to knowledge development. By integrating these tools in their daily work, development teams become more competent, able to respond to ambiguous situations and subsequently organizations become more agile and develop a customer responsive culture. This paper highlights obstacles faced by cross-functional teams working in a geodistributed environment and stresses on the need for thriving towards a learning organizations by adapting and integrating properly agile tools and practices. Nevertheless, this study presents two major limitations. First, the research results are limited to a single study constraining their generalization to other contexts. Hence, its application to other contexts and teams can constitute a further step of the study. Furthermore, this study has treated a limited number of agile practices. In the future, it would be interesting to consider more agile collaboration and engineering practices.

## References


1. Nonaka, I., & Takeuchi, H. (1995). *The Knowledge-Creating Company*. New York: Oxford University Press.
2. Nonaka, I. (1994). A dynamic theory of organizational knowledge creation. *Organization Science*, 5 (1), 14–37.
3. Beck, K. (1999). *Extreme Programming Explained: Embrace Change*. Addison-Wesley Professional.
4. Poppendieck, M., & Poppendieck, T. (2006). *Implementing lean software development: from concept to cash*. Addison-Wesley Professional.
5. Schwaber, K. (2004). *Agile project management with Scrum*. Microsoft Press.
6. Svensson, H., & Host, M. (2005).Views from an organization on how agile development affects its collaboration with software development team. *Product Focused Software Process improvement, LNCS*, 3547, 487–501.
7. Chong, J. (2005). Social behaviors on XP and non XP: A comparative Study. *The Agile Development Conference, IEEE Computer Society*, 39–48.
8. Melnik, G., & Maurer, F. (2002). Perceptions of agile practices: A student survey. *In: 2nd XP Universe and First Agile Universe Conference on Extreme Programming and Agile Methods*, 241–250.
9. Robinson, H., & Sharp, H. (2004). The characteristics of XP Teams. *In: Extreme Programming and Agile Processes in Software Engineering, LNCS*, 3092, 139–147.
10 Sharp, H., & Robinson, H. (2008). Collaboration and Coordination in mature Extreme Programming teams. *International Journal of Human Computer Studies*, 66 (7), 506–518.
11. Paasivaara, M., Durasiewicz, S., & Lassenius, C. (2009). Using Scrum in Distributed agile development: A multiple case study. *In: 4th International Conference on Global Software Engineering, IEEE*, 195–204.
12. Simons, M. (2002). Internationally Agile. InformIT. http://www.informit.com/articles/article.aspx?p=25929
13. Paasivaara, M., Durasiewicz, S., & Lassenius, C. (2008). Distributed agile development: Using scrum in a large project. *In: 3rd International Conference on Global Software Engineering, IEEE*, 87–95.

14. Highsmith, J., & Cockburn, A. (2001). Agile software development: the business of Innovation. *IEEE Computer*, 34 (9), 120–122.
15. Begel, A., & Nagappan, N. (2007). Usage and perceptions of agile software development in an Industrial Context: An Exploratory Study. *In: 1st International Symposium on Empirical Software Engineering and Measurement, IEEE Computer Society*, 255–264.
16. Yap, M. (2005). Follow the sun: distributed extreme programming development. *Agile Conference*, 218–224.
17. Jensen, B., & Zilmer, A. (2003). Cross-continent development using Scrum and XP. *In: 4th International Conference on Extreme Programming and agile processes in Software engineering*, 146–153.
18. Braithwaite, K., & Joyce, T. (2003). XP expanded: distributed extreme programming. *In: 6th International Conference on Extreme Programming and Agile Processes in Software Engineering*, 180–188.
19. Karlström, D., & Runeson, P. (2005). Combining agile methods with stage-gate project management. *Software IEEE*, 22 (3), 43–49.
20. Middleton, P., Flaxel, A., & Cookson, A. (2005). Lean software management case study: Timberline Inc. *In: 6th International Conference on Extreme Programming and Agile Processes in Software Engineering*, 1–9.
21. Sutherland, J., Viktorov, A., Blount, J., & Puntikov, N. (2007). Distributed scrum: agile project management with outsourced development teams. *In: 40th Annual Hawaii International Conference on System Sciences*, 274–274.
22. Danait, A. (2005). Agile offshore techniques-a case study. *In: Agile Development Conference*, 214–217.
23. Sharp, H., Robinson, H., & Petre, M. (2009). The role of physical artifacts in agile software development: Two complementary perspectives. *Interacting with Computers*, 21 (1-2), 108–116.
24. Wheelwright, S.C, & Clark, K.B. (1992). *Revolutionizing Product Development: Quantum Leaps in Speed, Efficiency and Quality*. New York: Free Press.
25. Stake, R.E. (1995). *The art of case study research*. Thousand Oaks, CA: Sage.

## Appendix I - Glossary of agile terms

Burndown Chart: It shows work remaining over time. Work remaining is the Y axis and time is the X axis. The work remaining should jig up and down and eventually trend downward.
Daily Scrum: It's a fifteen-minute daily meeting for each team member to answer three questions: what have I done since the last scrum meeting? What will I do before the next scrum meeting? And what prevents me from performing my work as efficiently as possible?
Kaizen: it's a Japanese term that means continuous improvement. Kaizen events consist on gathering operators, managers, owner of a process in one place, mapping the existing process in order to improve it.
On-site customer: It consists on having a real, live user that constantly collaborates with the development team. The on-site customer is available full-time to answer questions.

Post-sprint Meeting: At the end of the sprint iteration, a post-sprint meeting is held to review progress, demonstrate features to the customers and review the project from a technical perspective.
Product Backlog: The product backlog is the requirements for a system, expressed as a prioritized list of product backlog items. These included both functional and non-functional customer requirements, as well as technical team-generated requirements. While there are multiple inputs to the product backlog, it is the sole responsibility of the product owner to prioritize the product backlog.
Product Backlog item: In Scrum, a product backlog item ("PBI", "backlog item", or "item") is a unit of work small enough to be completed by a team in one Sprint iteration. Backlog items are decomposed into one or more tasks listed in a sprint backlog.
Product-Owner: In Scrum, a single person must have final authority representing the customer's interest in backlog prioritization and requirements questions. This person must be available at any time especially during the sprint planning meeting and the sprint review meeting.
Retrospective meeting: The sprint retrospective meeting is held at the end of every sprint after the sprint review meeting. The team and Scrum-Master meet to discuss what went well and what to improve in the next sprint.
Scrum-Master: The Scrum-Master is a facilitator for the team and product owner. Rather than managing the team, the Scrum-Master works to assist both the team and product owner.
Scrum team: It consists of seven plus or minus two people. For software development projects, the team members are usually a mix of software engineers, architects, programmers, analysts, QA experts, testers, UI designers, etc.
Sprint: It defines the work for a sprint, represented by the set of tasks that must be completed to realize the sprint's goals, and the selected set of product backlog item.
Sprint Planning Meeting: The Sprint planning meeting is a negotiation between the team and the product owner about what the team will do during the next sprint. The product owner and all team members agree on a set of sprint goals, which is used to determine which product backlog items will be implemented in the next sprint. Then, the Scrum-Master and his team focus on how the selected product items will be implemented. .
Stand-up meeting: It’s a fifteen daily meeting for XP teams. During this meeting, developers share their experiences of the day before, talk about their progress since the last stand-up and the anticipated work until the next stand-up.
Story-cards: They represent brief details of the tasks being actively worked upon.