KEYWORDS product lifecycle lean ambivalence paradigm process

AGILE MANAGEMENT PRACTICES

IN A "LIGHTWEIGHT" ORGANIZATION:

A Case Study Analysis

Carine Khalil
Ecole Centrale Paris
carine.khalil@ecp.fr

Valérie Fernandez
Télécom Paris Tech
Valérie.fernandez@telecom-paristech.fr

ABSTRACT

The transition to an "agile" organization remains a challenge. Despite the increasing number of empirical researches regarding"agile" software development, "agile" management and engineering practices are still difficult to implement within large organizations. To our knowledge, there is no specific framework that organizations can rely on to become "agile". Many contextual factors,such as team size and team distribution, seem to constrain the implementation of these emergent methods. From this perspective, how can a "lightweight"team structure become "agile"? What are the major challenges faced by such project teams? The present research aims at analyzing these questions through a longitudinal case study. The study has been conducted in a French telecommunication company that strives to become agile. We adopted a qualitative approach for collecting and analyzing data. Our findings highlight the reasons that drive organizations to become "agile" and stress on the contingency factors that affect the implementation of "agile"practices and tools in a "lightweight" organizational structure.

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 methods called "agile" methods has emerged and gained popularity within software industries. Literally, "agile" concept refers to "*having a quick resourceful and adaptable character*". In software development, "agile" methods focus on delivering tangible results, enhancing feedback and adaptation to changes. They 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.

Rapidly, "agile" concept has gained a significant interest within software industries and management scholars. Different surveys 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[1], 2008; Version One[2], 2010)*. Agile methods seem to improve time-to-market, team productivity and product quality.

Most of the published studies have reported the effects of using and implementing agile practices. But the results regarding these practices highlight different positions among practitioners. Agile management approaches are more adequate to organizational contexts where communication and coordination mechanisms can be easily achieved. What about large organizations where project managers have little authority on their project team? What are the major challenges encountered during the implementation of agile practices?

We adopted a qualitative research approach in order to address these relevant questions. We focused on a single case study to investigate, in-depth, the context settings *(physical, organizational and technical conditions)* in which agile practices are implemented and to shed the light on the challenges encountered during the implementation process. A sixteenth-month period was spent collecting data *(August 2009-December 2011)*. A set of semi-structured interviews was also conducted 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.

We structured this research work as follow: section *(1)* presents scrum method and lean development. It is also an occasion to distinguish both approaches that are often used interchangeably. Section *(2)*reviews the relevant results that highlight the effect of agile practices used in different contexts. Section *(3)* will portray the case study and the research methodology. Section *(4)* will discuss the findings and the concluding remarks of this research work.

1. Agile project management approaches : principles, practices and tools

Different agile methods exist. In the present work, we will focus on scrum method and lean approach. Our choice is guided by the studied context.

Scrum method: principles and management tools

In 1986, "The New New Product Development Game", Nonaka and Takeuchi used the rugby analogy to describe a new, holistic approach to project management. Scrum employs an iterative and incremental approach for managing projects. Three pillars sustain the development process: transparency, inspection and adaptation *(Schwaber et Sutherland, 2010)*. 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 viewed as a critical factor for scrum teams. **Table 1** summarizes the characteristics of scrum.

Lean development: principles and management tools

The "lean" concept was popularized in 1990 by MIT's global best-seller "*The machine that changed the world: the story of lean production*"*(Womack & Jones, 1990 in Poppendieck, 2003)*. Lean development is a way of thinking, a philosophy that encompasses a system-level perspective and therefore extends beyond development practices *(Poppendieck, 2006)*. It's a set of principles that organizations use in order to adapt tools and methods to their own specific contexts and capabilities.

Agile methods and lean development: a complementary approach?

The distinction between lean approach and agile methods remains until now unclear. If for some practitioners, lean development can be a part of agile methods such as scrum and Extreme Programming *(XP[3])(Jalali & Wohlin, 2010)*, for others, these approaches are situated at different levels: the lean approach denotes a philosophy and a set of management principles where agile methods imply a more pragmatic approaches, characterized by a set of management and engineering practices *(Morien, 2005 ; Poppendieck, 2003 ; 2006)*. From this perspective, lean principles are invariable regardless their implementation context*(Poppendieck, 2006)*. They guide and govern the implementation of agile practices and tools *(Ambler, 2009 ; Ambler & Kroll, 2009 ; Wang, 2011)*. Accordingly, lean development goes beyond agile methods, providing a broader perspective that enables these methods to thrive *(Sutherland in Poppendieck, 2006)*. Lean is considered as a global approach that overpasses development activities. Lean principles concern the different levels of the organization. Furthermore, lean governance facilitates

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

[2] http://www.versionone.com/pdf/3rdAnnualStateOfAgile_FullDataReport.pdf.

[3] The extreme programming method relies on a set of engineering practices and collaborative tools that ensure knowledge capitalization and enhance communication and code quality (Beck& Andres, 2004).

Scrum (Schwaber, 2004)	
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

TABLE 1. *Components of scrum method (Scrum practices, tools and roles are defined in Annex I).*

Lean (Poppendieck, 2006)	
Principles	Eliminate wastes, build quality, create knowledge, defer commitment, deliver fast, respect people, optimize the whole
Management practices	Pull system, kaizen, iterative development, test-driven development
Management tools	Andon, value stream mapping, pareto cause analysis, 5 whys, A3 problem solving

TABLE 2. *Components of lean development (Lean principles, practices and tools are defined in Annex I).*

the scalability of agile practices and consequently their integration in large organizations*(Smits, 2007)*.

2. Agile practices: advantages and limitations

The number of empirical researches on agile development has gradually increased throughout the last ten years. In this section, we will review the relevant results that highlight the effect of agile practices used in different contexts. Various types of agile practices and tools have been investigated. We will stress on agile collaborative practices such as daily meetings and on-site customer and on supportive management tools such as story-cards, product backlog and root cause analysis tools.

Daily meetings strengthen the communication between team members *(Svensson & Host, 2005; Chong, 2005)*, improve information sharing *(Melnik & Maurer, 2002)* and collective problems solving *(Robinson & Sharp, 2004, Sharp & Robinson, 2008)*. This type of meetings clarifies the status of the on-going operations and sheds light on the difficulties encountered throughout the project *(Paasivaara, Durasiewicz & Lassenius, 2009)*. It also ensures a better control of the project and provides a coordination mechanism for everyone in the project *(Paasivaara, Durasiewicz & Lassenius, 2009)*. However, in distributed environments, frequent and informal communication is difficult to achieve impacting the collaboration between team members *(Simons, 2002)* and the pursuit of a common goal *(Paasivaara, Durasiewicz & Lassenius, 2008)*. Furthermore, daily meetings seem difficult to realize across long distances and geodistributed teams *(Cockburn & Highsmith, 2001; Begel & Nagappan, 2007, Kircher, Jain, Corsaro & Levine, 2001; Yap, 2005)*. In such environments, team members should be equipped with different communication media and information technologies in order to facilitate their direct communication and documents sharing *(Kircher, Jain, Corsaro & Levine, 2001; Jensen & Zilmer, 2003; Yap, 2005; Braithwaite & Joyce, 2005; Fowler, 2006)*.

Another collaboration practice that has been studied is the on-site customer. It enables continuous feedback *(Karlström & Runeson, 2005)* and improves client satisfaction regarding the developed product *(Mann & Maurer, 2005)*. Involving the customer to frequently collaborate with the team enables a shared vision between the counterparts *(Poole, 2007.)*. Developers perceive this practice as useful because it allows them to contact the client at any time *(Koskela& Abrahamsson, 2004; Svensson& Host, 2005; Mann & Maurer, 2005, Karlström&Runeson, 2005)*. Nevertheless, in a distributed environment, the proximity between the client and the team is not feasible. Some practitioners evoked the term of virtual client, readily accessible and showing interest in the project, so as to play the role of the customer *(Simons, 2002; Laymann, Williams, Damia & Bures, 2006)*. Other difficulties that have been also reported concern the unavailability of the client *(Laymann, Williams, Damia &. Bure, 2006)* and the stress experienced by this latter *(Martin, Biddle & Noble, 2004; Koskela & Abrahamsson, 2004)*. Moreover, a client should have good technical skills that enable him to take the good technical decisions *(Tessem, 2003)*.

Finally, supportive management tools such as user-stories, story-cards, and product-backlog seem to create an informative workspace *(Cockburn, 2002; Robinson & Sharp, 2004; Sharp & Robinson, 2007)*. The use of tools such as whiteboards enables team members to have a sharing vision of the project requirements *(Martin, Biddle & Noble, 2004; Sharp & Robinson, 2008; Sharp, Robinson & Petre, 2009)*. Problem solving tools can also help organizations with reducing costs and improving the product quality *(Middleton, Flaxel & Cookson, 2005)*. However, physical distance across teams constrains sharing these tools. As mentioned previously, it is necessary to have a good technology infrastructure that enables exchanges across distributed teams *(Kircher, 2005; Danait, 2005, Berczuk, 2007, Paasivaara, Durasiewicz & Lassenius, 2008)*.

The published results regarding these practices highlight different positions among practitioners. Agile practices are not easily implemented in complex organizations characterized by large and geodistributed project teams. Given this focus, we aim at identifying the challenges encountered in the implementation of agile practices in a “lightweight” organization *(Clark&Wheelwright, 1992)*.

3. Research context

The context of the study

The “Alfa” organization is a French telecommunication company responsible of monitoring and delivering television services on the internet, phone and satellite.The projects generally involve cross-funcional actors attached to a “lightweight” project manager that has little authority on his team *(Clark&Wheelwright, 1992)*. His role is to ensure the coordination of the different functional units of the project and to control goal achievement within the predetermined budget.

“Alfa” 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 change the existing managerial modes and implement new managerial practices based on agile methods. The purpose of the project is to make a better use of the available resources and to deliver on time products that meet customers’ needs. A team of five actors called “L” team has been designated responsible of implementing agile practices and tools. They decided to deploy, within the project teams, a set of scrum and lean practices that emphasize communication and collaboration. This includes daily meetings, virtual whiteboards, kaizen sessions and root causes analysis tools.

The studied organization has a “lightweight”-structure *(Clark& Wheelwright, 1992)* characterized by distributed teams. The project manager coordinates the activities of his team members and facilitates the information exchange between functional units. Projects are generally large, involving approximately thirty five persons each. They combine cross-functional actors that are hierarchically attached to different functional managers. These actors intervene temporarily in the project and work simultaneously on different projects.

Research Methodology

We adopted a single case study *(Yin, 1984)* to investigate, in-depth, the phenomenon within its natural settings and portray the real context in which the implementation of agile practice has occurred. We used a qualitative methodology to shed light on the challenges encountered during the implementation process. A sixteenth-month period was spent *(August 2009-December 2010)* collecting data in the “Alfa” organization. Primarily, we participated at 23 meetings organized by the “L” team members and attended 7 workshops[4] organized between the “lean” team and the pilot teams.We took extensive notes during the observations and, whenever possible, we made audio recordings of the entire session. Each obser-

[4] These workshops aimed at identifying the dysfunctions encountered in the organization. The participants are functional architects, project managers, technical architects, testers, ...).

vation transcript was complemented by key notes in order to get preliminary ideas about the major themes and elements treated during the observation session. Moreover 8 interviews were conducted with different project managers and "L" team members, each, lasting a minimum of one hour. The interviews were semi-structured and the interviewees were selected according to their involvement in the implementation of the agile approach. The interviews were recorded with the approval of the interviewees who were explicitly informed of the study's purpose. Complementary data such as mails and documents, exchanged between "L" team members, was also collected to enhance our understanding of the context and the implementation phenomena.

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 "What are the major challenges encountered in a "lightweight" organizational structure during the implementation of agile practices?" 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 and justified with verbatim*(Brown & Jacobs, 2009, p.136)*. Among these, we cite the following: organizational structure, team composition, team size, inter-individual interactions.

4. Data Analysis

Identified wastes in "Alfa" projects

Many dysfunctions are encountered in the development process of "Alfa" projects. Overproduction is the first reported wastes. Non added value functionalities and unused documentation are produced during the development process leading to additional costs. Due to the lack of communication between actors and the absence of a visual environment *(information are diffused through reported, extensive documentation)* the team cohesion and the creation of a common view of the project are difficult to achieve. Furthermore, the actors tend to anticipate some activities and take initiatives without communicating, leading to the production of unnecessary components.

"The problem is that developers will often correct and implement functionalities more than you ask them for.... They tend to take the decisions instead of the client by producing more than is needed".

For that purpose, the « L » team decided to implement daily meetings and visual management tools. The objective is not only to improve exchanges between teams but to increase the visibility of the projects as well.

Delay is another waste encountered in the "Alfa" organization. It is manifested through unoccupied actors waiting for information, decisions, results validation or unavailable resources. Team members have difficulty identifying people they need to assist them in their problems and answer their questions. Problems of delay occur in all the phases of the development process. They can be explained by the multiplicity of communication canals, the simultaneous conduct of different projects, the use of common resources, the dependency between the project phases, etc.

"Project managers wait for budget decisions .... The multiple decisions maker constrainsand delays the development process".

In this case, it would have been wise toreconsider the organizational structure and the existing sequential development approach. However, the "L" team members preferred to deal with the problems caused by the lack of communication between project actors. Subsequently, they focused on the implementation of practices and tools which can increase exchanges between teams.

Unnecessary transportation is another waste encountered in "Alfa" organization. Needless mail exchanges and documentation can slow the activities of team members. For the interviewees, useless information exchanges are caused by the sequential development process, the multiplicity of actors involved in it and the lack of clarification of their role inside the organization.

"Sometimes people wonder by whom the project is managed. As I was telling you, there are a lot of actors involved in the project »; « We have difficulty to identify people's names working on this functionality or component".

Motion in software development concerns the redundant activities and the rework of the same activity. The rework of a suspended task leads to a waste of time. It requires supplementary efforts from the responsible actor that should put himself in the previous conditions when he was already involved in other tasks.

"It happens that we stop a work because it is less priority .... We always find ourselves with a useless work..."; "It is difficult to have a global vision... reworks are caused by the lack of this global vision".

These types of wastes are caused by the lack of communication, global vision and the absence of priority strategy. Moreover the risks of rework increase in large projects where teams spend a lot of time in the planning and the study phases of the project.

Another type of waste we find in the project is inventory. In automobile industry, inventory represents unused stock that generates wastes because they necessitate physical space, transportation and maintenance. In software development, this type of wastes concerns the partially done work and the unused documentation and functionality. In this case study, the lack of regular exchanges, clarified requirements and prioritization strategy are at the origin of this type of waste.

"There are changing in priorities in the functionality and project... we spend time doing things and then they ask us to drop them and to focus on other things..."; "Clients often change their requirements and ask us consequently to change our priorities... so we modify the scope of the project in order to integrate other components".

For this reason, the "L" team has decided to implement daily meetings in order to enhance close collaboration between teams and improve the global vision of the project.

A last type of waste that has been reported during the workshops is the defect. In automobile industry, the late identification of defects generates added costs. In this context, actors complain about the bugs appearing at the end of the development processand which seem to affect the quality of the products. In "Alfa" Organization, the anticipation and detection of problems is poor and inefficient.

"We need twelve months to produce something... And when we deliver clients are rarely satisfied because it doesn't fit their requirements...."; "There is no knowledge capitalization in the projects...It is not easy to capitalize when we often change partners".

In order to respond to these difficulties, the "L" team has decided to implement kaizen sessions and a set of root cause analysis tools. The objective is to implement an approach that enables the team to identify rapidly and resolve collectively the problems when they appear.

Projects in "Alfa" organization seem to generate supplementary costs. As the field notes shown, a lot of wasted times and efforts are spent on extensive planning and task documentation. The client doesn't know what he wants and the environment in which he operates is constantly changing. In this respect, the planning activities and the specification of the whole technical solution are not efficient.

A contextual factors analysis

The implementation of agile practices and tools was not that easy. Many challenges were encountered during the implementation process. The following section reports the contextual factors that influence the implementation of agile practices and tools.

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 the implementation of daily meetings and therefore 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 geo-graphically... 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 co-located....The geodistribution can skew the information" (product development manager).

Large scale projects are also viewed as a challenge for implementing collaboration practices such as daily meetings, kaizen sessions... 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 co-located teams, actors can directly deal with their neighbors 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).

The larger the team is, the harder communication becomes. Our field note has shown that close collaboration and tacit knowledge sharing are difficult to achieve. The large size of the team has led to more formalized procedures and a rigorous management system that go further agile organizational characteristics: long decision processes, multiple communication canals, etc.

"There a lot of communication canals which complicate the decision making. For instance, we have different marketing representatives in project… so it would be important to have one decision maker"(Project ManagerD).

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. 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. The coordination of different functional teams has limited the circulation of instruction and information. It was also 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).

The field notes have shown that the management systems on which the teams rely to support their project needs is also challenging. The integration of agile practices is not completely coherent with the existing development process. The products' lifecycle presents difficulties in measuring the productivity of the team at different levels of the project. The organizational culture characterized by a long term project vision also defies the implementation of agile practices.

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. The project manager should have a sufficient authority that enables him to ensure team cohesion and control his team.

5. Conclusion

Thus, the implementation of agile practices in a "lightweight" organizational structure 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.Therefore, we believe that beyond the contextual 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.

The contextual factor analysis, conducted in this study, has important implications for both researchers and practitioners: For researchers, this study is significant in that it allows to combining a range of contextual factors into a framework that can serve as a basis for further investigation in this domain. For practitioners, implementation challenges, as they were discussed in this research, are seen as a mean to help organizations being aware of the various contextual barriers that can disable the deployment of agile methodologies. Therefore, it provides practitioners with useful insights on how to cope with these challenges by taking in consideration different individual and organizational aspects.

This paper highlights obstacles faced by cross-functional teams working in a "lightweight" organization and stresses on the need for thriving towards 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.

ANNEX I

A3 PROBLEM SOLVING: A3 problem solving is a structured approach to resolve problems. Toyota engineers learn the discipline of condensing complex thinking to a single A3 sheet of paper. Different A3 reports have different purposes, but all of them capture critical knowledge in a way that is easy to store in a database, easy to post in a work area, etc.

ANDON: Toyota used the word andon to name the cord that workers could pull to "stop-the-line", since pulling and andon cord usually cause lights to flash, calling attention to the problem area. The idea behind andon is to make problems visible so they can be addressed immediately.

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.

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.

COLLECTIVE CODE OWNERSHIP: It's an extreme programming practice where anyone can change the code, anywhere in the system, at any time.

CONTINUOUS INTEGRATION: It consists on integrating and building the system many times a day, every time a task is completed.

DAILY SCRUM: Its 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: its 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.

PARETO CAUSE ANALYSIS: Its an application of the "vital few trivial many" rule, also called 80/20 rule. The analysis proceeds thus: Divide a problem into categories, find the biggest category, look for the root cause of the problem creating that category and fix it.

PLANNING-GAME: Its a meeting attended by both development and business teams (client representative) in order to identify and prioritize stories of the next release or iteration. It combines business priorities and technical estimates.

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.

PAIR PROGRAMMING: It consists on having two people working side-by-side on the same task. It provides continuous code review in a flow rather than a batch.

PULL SYSTEM: To reduce inventory holding costs and lead times, Toyota developed the pull production method wherein the quantity of work performed at each stage of the process is dictated solely by demand for materials from the immediate next stage.

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: Its 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.

THE 5 WHY'S: It refers to the practice of asking, five times, why the failure has occurred in order to go to the root cause/causes of the problem. There can be one or more cause to a problem as well.

TEST DRIVEN DEVELOPMENT: It consists on writing the tests before writing codes in order to prevent defects. The goal of lean software development is to prevent defect from getting into the code base in the first place and the tool to do this is the test-driven development.

UNIT-TESTS: They are written by developers to test their design intent. Creating a unit test helps a developer to really consider what needs to be done.

authors

Carine Khalil has a PhD in Management science (December 2011) from Telecom ParisTech. She is currently working as a researcher in the field of management information systems at Ecole Centrale Paris. Her research interests focus on agile project management, knowledge management and new organizational forms.

Valérie Fernandez is Professor in Management Sciences at Telecom Paristech, LTCI – CNRS laboratory. She is project leader of international and national research programs (PCRD European programs, FUI Programs, CNRS, French-China collaboration programs, ...) and Head of Advanced Master’s Program in Management of Technical Project with ESSEC Business School. Her research focuses on ICT project management : design, adoption and diffusion of ICT in organizations and in public places.

BIBLIOGRAPHIC references

Abrahamsson P. &Koskela J. *(2004)*, “Extreme Programming: A survey of empirical data from a controlled case study”, *Proceedings of the 2004 International Symposium on Empirical Software Engineering, Computer Society*, Finland, pp.73-82.

Beck K. *(2004)* & Andres C, *Extreme Programming Explained :Embrace Change,* Addison-Wesley Professional, 2nd edition, 224 p.

Begel A. &Nagappan N. *(2007)*, “Usage and Perceptions of Agile Software Development in an Industrial Context: An Exploratory Study”, *Proceedings of the First International Symposium on Empirical Software Engineering and Measurement, IEEE Computer Society*, Madrid, pp. 255-264.

Berczuk S. *(2007)*, “Back to Basics: The role of agile principles in success with an distributed Scrum team”, *Proceedings of Agile Conference, IEEE Computer Society*, Washington, DC, pp. 382-388.

Braithwaite K. & Joyce T. *(2005)*, “XP expanded: Distributed Extreme Programming”, *6th International Conference on Extreme Programming and Agile Processes in Software Engineering, LNCS*, n°3556, pp. 180-188.

Chong J. *(2005)*, “Social Behaviors on XP and non XP: A comparative Study”, *Proceedings of Agile Development Conference,* Denver, Colorado, pp. 39-48.

Cockburn A. *(2002)*, “Agile software development joins the “would be” crowd”, Cutter IT Journal, vol 15, n°1, pp. 6-12.

Cockburn A. & Williams L. *(2003)*, “Agile Software Development: It’s about Feedback and Change”, *IEEE Computer*, vol 36, n°6, pp. 39-43.

Clark K.B. &Wheelwright S.C. *(1992)*, “Organizing and leading heavyweight development teams”,*California Management Review*, vol. 34, n°3, Spring, pp. 9-28.

Danait A. *(2005)*, “Agile offshore techniques - a case study”, *Proceedings of Agile Development Conference*, Denver, Colorado, pp. 214-217.

David A. *(2000)*, “Logique, méthodologie et épistémologie en sciences de gestion : trois hypothèses revisitées”, in David A., Hatchuel A. et Laufer R., *Les nouvelles fondations des sciences de gestion,*Vuibert, collection FNEGE.

Fowler M. “Using an agile software process with offshore development”, http://www.martinfowler.com/articles/agileOffshore.html.

Garel G., Giard V. & Midler C. *(2001)*, “Management de projet et gestion des ressources humaines”, Gregor, IAE de Paris.

Herbsleb J.D. &Grinter R.E. *(1999)*, “Splitting the organization and integrating the code : Conway’s law revisited”*, Proceedings of the 21st International Conference on Software Engineering*, CA, USA, pp. 85-95.

Highsmith J. & Cockburn A. *(2001)*, “Agile Software Development: The business of Innovation », *Computer,* vol 34, n°9, pp. 120-122.

Ilieva S., Ivanov P. &Stefanova E.*(2004)*, “Analysis of an agile methodology implementation”,*Proceedings of the 30th Euromicro conference, IEEE Computer Society*, Sofia University, 31 Aout- 3 Septembre, pp. 326-333.

Jensen B. &Zilmer A. *(2003)*, “Cross-continent development using Scrum and XP”, *Lecture Notes in Computer Science*, n°2675, pp.146-153.

Karlström D. &Runeson P. *(2005)*, “Combining agile methods with stage-gate project management”, *IEEE Computer Society*, vol 22, n°3, pp. 43-49.

Kircher M., Jain P., Corsaro A & Levine D. *(2001)*, “Distributed extreme programming”,*Proceedings 2nd Conference on Extreme Programming and Flexible Processes in Software Engineering*, Sardinia, Italy.

Koenig G. *(2003), l’organisation dans une perspective interactionniste,*inVidaillet B., le sens de l’action, Vuibert, pp. 15-34.

Koskela J. & Abrahamsson P. *(2004)*, “On site customer in an XP project: Empirical results from a case study”, *Software Process Improvement 11th European Conference, LNCS*, n°3281, Norway, pp. 1-11.

Layman L., Williams L., Damia D. &Bures H. *(2006)*, “Essential communication practices for extreme programming in a global software development team”,*Information and Software Technology*, Vol 48, n°9, pp. 781-794.

Lejeune N.F. *(2006)*, “Teaching software engineering practices with extreme programming”, *Journal of Computing Sciences in Colleges*, vol 21, n°3, pp. 107-117.

Lindvall M., Basili V., Boehm B., Costa P., Dangle K., Shull F., Tesoriero R., Williams L. &Zelkowitz M.*(2002)*, “Empirical Findings in Agile Methods”, *Proceedings of the Second XP Universe and First Agile Universe Conference on Extreme Programming and Agile Methods, XP/Agile Universe,* n°2418, London, pp. 197-207.

Mann C. & Maurer F. *(2005)*, “A case Study on the Impact of Scrum on Overtime and Customer Satisfaction”, *Proceedings of the Agile Development Conference, IEEEComputer Society*, Washington, pp. 70-79.

Martin A., Biddle R. & Noble J. *(2004)*, “The XP customer role in practice: Three studies”, *Proceedings of the Agile Development Conference, Computer Society*, Salt Lake City, Utah, pp. 42 - 54.

Melnik G. &Maurer F. *(2002)*, “Perceptions of agile practices: A student survey”, *Proceedings of the Second XP Universe and First Agile Universe Conference on Extreme Programming and Agile Methods, Lecture Notes in Computer Science,* n°2418, pp. 241-250.

Midler C. *(1993b)*, “Gestion de projet, l’entreprise en question”, in ECOSIP Pilotages de projet et entreprises : diversités et convergences, sous la direction de Midler C. & Giard V., Economica, pp. 17-31.

Middleton P., Flaxel A. & Cookson A. *(2005)*, “Lean software management case study: Timberline Inc.”, Extreme Programming and Agile Processes in Software Engineering, *Lecture notes in computer science*, vol 3556, n° 1297-1298, pp. 1-9.

Mintzberg H. *(1984)*, le manager au quotidien : les dix rôles du cadre, Organisation.

Petersen K. &Wohlin C. *(2009)*, “A comparison of issues and advantages in agile and incremental development between state of the art and an industrial case”, *Journal of systems and software*, vol 82, n°9, pp. 1479-1490.

Poppendieck M. & Poppendieck T. *(2003),Lean software development: an agile toolkit*, Addison-Wesley.

Poppendieck M. & Poppendieck T. *(2006), Implementing lean software development: from concept to cash*, Addison-Wesley Professional, 304 p.

Robinson H. & Sharp H. *(2004)*, “The characteristics of XP Teams”, *Extreme Programming and Agile Processes in Software Engineering, Lecture Notes in Computer Science*, vol 3092, pp. 139-147.

Sharp H. & Robinson H. *(2007)*; “Collaboration and coordination in mature Extreme Programming teams”, *International Journal of Human Computer Studies*, vol 66, n°7, pp. 506-518.

Sharp H, Robinson H &Petre M. *(2009)*; “The role of physical artifacts in agile software development: Two complementary perspectives”, *Interacting with Computers*, vol 21, n°1-2, pp. 108-116.

Schwaber K. *(2004), Agile project management with Scrum*, Microsoft Press, 192 p.

Simons M. *(2002)*, “Internationally Agile”, InformIT.

Smith G.F. *(1989)*, “Defining managerial problems: A framework for prescriptive theorizing”,*Management science,* Vol 35, n°8, pp. 963-981.

Stake R.E. *(1995), The art of case study research*, Sage Publications.

Sutherland J., Viktorov A., Blount J. &Puntikov N. *(2007)*, “Distributed scrum: agile project management with outsourced development teams”, *40th Annual Hawaii International Conference on System Sciences*, Waikoloa, HI, pp. 274-274.

Svensson H. & Host M. *(2005)*, “Views from an organization on how agile development affects its collaboration with software development team”,*International conference on product focused software process improvement, Lecture Notes in Computer Science*, vol 3547, Finlande, pp. 487-501.

Tessem B. *(2003)*, “Experiences in learning XP practices: A qualitative study”, *Proceedings 4th International conference on Extreme Programming and Agile Processes in Software Engineering*, Genova, Italy, pp. 131-137.

Williams L., Kessler R., Cunningham R.W. & Jeffries R. *(2000)*, “Strengthening the Case for Pair Programming”, *IEEE Software*, vol 17, n°4, July/August, pp. 19-25.

Yap M. *(2005)*, “Follow the sun: distributed extreme Programming development”, *Proceedings of Agile Conference*, Kirkland, pp. 218-224.