

Challenges of Applying Agile Principles and Values to IT Project Management

Carmen Novac¹ , **Raluca-Silvia Ciochină²** 

Abstract

The theoretical part of this paper focuses on a macro-analysis of the Agile system philosophy, one of the most popular project management frameworks from the software industry. The second part of the paper consists of a case study analyzing the practices of the Agile framework within a corporation at the level of software production. As far as methodology is concerned, the qualitative research was conducted through a case study of a team in a distributed context (“cross-team”). Participant observation was used as a tool for collecting the data about the individuals and the processes within the teams. The purpose of the research was to analyze the current state of implementation, the benefits and challenges of using the Agile project management framework and also the needs of software development teams within a multinational company. At the same time, the paper analyzes the redefinition of roles by applying the Agile project management framework within the distributed teamwork organization. The research aims to identify the underlying reasons for the choice of the Agile type of management instead of Lean management within the software development industry, focusing on the level of implementation of the Agile/Scrum methodology and principles in distributed teams.

Keywords: Agile, Scrum, IT, project management

INTRODUCTION

IT industry changes have determined the rise of different approaches in managerial philosophy which are currently applied in Romanian organizations as well. Software development implies consistent efforts undertaken by IT managers to coordinate teams and deliver quality, as well as the pressure to respond to competitive forces that emerge from a highly dynamic and

¹ Carmen Novac, Ph.D., National University of Political Studies and Public Administration/Bvd. Expozitiei, No. 30 A, Bucharest, Romania, e-mail: carmen.novac@comunicare.ro

² Raluca Ciochina, Ph.D., National University of Political Studies and Public Administration/Bvd. Expozitiei, No. 30 A, Bucharest, Romania, e-mail: raluca.ciochina@comunicare.ro

Received 16 February 2018; Revised 1 June 2018, 16 September 2018; Accepted 17 September 2018

uncertain environment. As Cervone (2011) emphasized, the effort invested in the planning phase is significant and involves using considerable resources before development activities begin. This paper focuses on identifying the approaches and implementation of the Agile project management framework within a distributed working team context and also the reasons why this system philosophy and methodology represents a solution for applications and IT projects design. These concepts are also utilized in other fields of activity such as research projects or various change management programs and initiatives, but the main domain in which the Agile philosophy is currently implemented is IT.

The research questions refer to identifying and defining the principles that led to choosing the Agile management option instead of the Lean management option within the software development industry.

RQ1: What is the level of implementation of Agile principles and methodology within distributed teams?

Also, another focus of this paper was to identify and analyze the roles perspective, especially the responsibilities and attributions of the project manager within the Scrum team, and of their members.

RQ2: Is there a stability level as far as redefining responsibilities in the Scrum team is concerned?

The Agile management framework appears to be a pertinent choice considering the adaptability and innovation principles installed within the IT departments. Agile principles and Scrum methodology are preferred by most Romanian multinational organizations which specialize in software production. The paper is focused on analyzing the Agile principles and the Scrum methodology and their applicability in the software industry, as well as identifying how the roles are being redefined within this new management context.

LITERATURE REVIEW

History of the agile principles and philosophy

Agile represents a set of principles developed by 17 software designers in February 2001, in the USA. Unsatisfied by project evolution and processes and by the fact that teams were getting more and more limited by various procedures, which affected their performance, the team of software

developers generated a set of ideas and principles which regularly appeared in successful projects. These ideas were put together in the Agile Manifesto (Martin, 1991).

This manifesto consists of the following ideas: finding new, efficient and inventive ways for software development practices and coordination. The following core principles were stated: (1) individuals and their interactions are more important than processes and tools; (2) functional software is prioritized over complex documentation; (3) client collaboration is more important than inflexible contracts; (4) adapting to change is more important than following fixed planning (agilemanifesto.org). All these elements have their own value and impact project management in their own specific way, adding value to both clients and developers, and also bringing satisfaction to team members (Aubry, 2011). The focus is set on two concepts: minimalizing risk through short iterations of defined deliverables and direct communication with collaborators in the development process (p. 19). This type of management can be seen as opposed to other project management models, specifically to Waterfall methods that represent the typical approach to managing large projects, which focus on specifications and linear management. The waterfall approach is called as such because each stage of the project falls into the next one, without the possibility of reiteration or the possibility of revisiting past steps (Sutherland, 2014), which is opposite to the iterative process of Agile project management. The Agile Manifesto also includes the internal learning of facing and accepting that unpredictable events can happen at any time, which is why it is essential to collaboratively work as a team, where individuals and their interactions are more important than documentation. The general idea behind Agile is that flexibility is primary to everything else: instead of debating documentation requests, new processes should be implemented that allow client-team adaptability and ongoing communication.

From an economic perspective, the Agile movement refers to multiple aspects regarding project management within the current organizational culture. Throughout time, different kinds of implementations in various industries appeared, each organization starting to develop internally their informational library based on the proprietary experience with the projects that were conducted, demonstrating the need to develop their own iteration based on specific needs (Kerr & Hunter, 1993).

Agile represents the management paradigm that cannot be applied within traditional organizations. The Agile mind-set is ideal in situations where novelty is present (dynamic requests, evolving priorities, flexible results, etc.). It is worth mentioning that Agile methods are not efficient in industries such as construction or manufacturing. The Agile mind-set can be implemented in

other industries as well, but the human factor remains a challenge and studies still need to address how it can successfully be implemented in this sense.

The Agile principles

“The Agile concepts represent a natural response to the evolution and dynamics of competitive industries, which allow the multitude of concepts such as adaptability and dynamic performance to thrive inside an organization” (Vickoff, 2009).

The Agile philosophy is defined by key values such as *collaboration*, *communication* and *multidirectional feedback*. Implementing a project is difficult without direction parameters and without efficient practices, as unexpected situations emerge all the time. In this sense, repeated errors and inefficient work can affect client feedback, who can become disappointed by the low quality of the delivered products, by delays and increasing budgets (Martin & Martin, 2006). When measured, project success also takes into account the level of client satisfaction and the quality of the end result – the product must apply to what was initially expected and it should be delivered within the established terms (time, costs) and within budget. All organizational projects are influenced by organizational culture, organizational climate and identity, by role and responsibility definition, and by organizational communication. Some organizational cultures favor support, responsibility and ownership within the project team; some do not.

The 12 Agile principles differentiate Agile practices from other complicated processes. The priority of Agile management is to satisfy the client through early delivery of valuable software. The second priority is to deliver results frequently, through a functionality system that allows constant feedback. Thus, clients can choose to place these systems in production if they consider that these are functional or they can test the existent functionality by requesting future changes to be implemented. For implementing successful Agile management, frameworks or management systems have been developed and implemented, such as Scrum and Kanban (Anderson, 2000). Scrum refers to the management process of product development within a changing environment; an iterative process “used to help enable improvement in communication, maximize cooperation, as well as protect the team from disruptions and impediments” (Cervone, 2011, p. 20). Therefore, the Scrum method implies adaptation according to both internal and external factors.

Scrum was first created by Ken Schwaber and Jeff Sutherland, who are the authors of *The Scrum Guide*. This framework for project management can be used in any domain as it is not a predictable process, but a heuristic

and iterative one. Schwaber (2004) insists on the flexibility of Scrum and on transparency, which are most important to managing software production. According to the author, the Scrum Master has the project manager role but undertakes the responsibility of managing the Scrum process and not the tasks. The roles that are most common are Scrum Master, Product Owner and the Scrum team (developers, designers, analysts, testers). In this sense, the process includes practices and terminology and the Scrum Master should know how to apply them in the right manner (Schwaber, 2004).

In a research conducted by França, da Silva, and de Sousa Mariz (2010), analysing the perspective of the software team on the relationship between the adoption of the Agile attributes and project success, 8 attributes were significantly correlated to project success: (1) regular delivery of software, (2) delivering the most important features first, (3) correct integration testing, (4) high competence of team members, (5) following the agile requirement process, (6) following the agile configuration management process, (7) self-organizing teamwork, and (8) good customer relationship. Considering the Scrum main activities, including sprint planning meetings, sprints, daily Scrum and the sprint review meeting (Cervone, 2011) and the focus of these activities on committing to delivering functional products through teamwork, it can be suggested that agile management success is connected to the attitudes and behaviors of team members involved in the project.

In Scrum sprints, commitment and transparency are important as the team is empowered to choose specific tasks and deliver them within the upcoming sprint (Cervone, 2009; Lehnen, 2016). At the end of the sprint, the team “critically discusses current sprint deliveries jointly” (Lehnen, 2016, p. 224), when challenges and setbacks are being discussed, while sprint retrospectives are useful mainly for identifying what could have been done better and what best practices can be developed from the past sprint experience. Constant inspection is required as an essential step in the Scrum methodology, even though it is an on-going process which takes time and effort on behalf of all the members involved (Denning, 2016). Understanding the setbacks, challenges and the successful practices within the project is essential for both project life cycle and team collaboration.

Agile management seeks to develop projects around motivated individuals, as they are one of the most critical factors for project success. Time pressure and deadlines can determine certain productive behaviors, but this is not enough.

By conducting a systematic literature review of motivation in Software Engineering, Beecham et al. (2008) found that earlier models did not take into account cultural and environmental settings, besides factors such as career stages or personality. Amongst the most successful factors described

as motivators were recognition and employee participation and involvement, or, generally speaking, working with others. Additionally, clear goals and tasks and the way they fit into the business and qualitative work were also the most frequently cited motivators in the academic literature (p. 12). As DeMarco and Lister (2013) pointed out, working under time pressure does not necessarily imply success of the overall task: “People under time pressure don’t work better – they just work faster” (p. 18). All the other factors (process, environment, management) are secondary factors and sensitive to change if they have antagonistic effects over individuals.

However, Rasnacs and Berzisa (2015) found that the cooperation motives and the working environment are important to Scrum team members from a Latvian organization, emphasizing that “if the basic needs of group members are not satisfied (utility-pragmatic motives), then group members are not motivated towards more important motives for the development of Agile projects – personal growth, competition, challenge and creative activity” (p. 127). As a result of the case study, the authors suggested introducing the role of project manager when the team did not apply the self-organization principle and the final result was problematic. Therefore, there are many aspects to be analyzed when investigating the success of Agile project management methods, as challenges can emerge especially when group needs are not met.

The most efficient and effective information transfer within a development team is face-to-face communication. Within Agile projects, through Scrum methods, team members discuss and negotiate, the first means of communication being direct face-to-face communication within Scrum meetings. The documentation is created and updated incrementally on the same schedule as the software and only if needed. Working software is the main indicator of progress. Agile projects measure progress by evaluating the quantity of software that satisfies the client’s needs; they do not measure progress based on the volume of documentation or the quantity of the developed code. The result percentage is directly proportional to the percentage of the requested functionality (Stellmann & Greene, 2014). Agile teams appear to have a medium rhythm, working with a speed that allows them to keep the highest standards throughout the entire timeframe of the project.

Agile management encourages continuous attention for technical excellence, focusing on the qualitative factor. Maximum quality represents the scope for Agile projects (Misra, Kumar, & Kumar, 2010). Another specific characteristic refers to the capacity to self-organize: responsibilities are not set for team members individually, they are communicated to the team as a whole, as the team is the one determining the best way in which it can deliver those responsibilities. Thus, Agile teams constantly adjust their structure,

organization, rules, norms, relationships, knowing that the environment is dynamic and changing (Martin & Martin, 2006).

The primary objective of Agile management is to optimize, from a functional perspective, the intermediary management layers of the organization. Intermediary management represents the privileged target of coaching. Agile stands at the basis of the transition program, as well as at the individual and collective levels. The Agile manager needs to be rigorous, open towards new and alternative solutions, available, organized, anticipating, and actively listening to the team. Besides this, leadership, transparency and relational qualities are also outstanding. In a permanently expanding and development environment, the manager needs to deliver fast results by using limited resources and the results need to be of high quality (Rota, 2010).

As a project management professional, the Agile manager needs to know and use these techniques, to be able to explain them and justify the choices, to be able to reproduce practices that worked and ended with positive results into similar contexts or to adapt them in new contexts. Agile managers should also know how to accentuate knowledge and how to inspire trust, for being properly recognized as a professional. Within the project team, the Agile manager should share their vision with the other team members so it can be accepted – a key element for the well-functioning of the project.

Agile management also involves efficiently prioritizing activities, using the Urgency-Important Matrix and taking relevant action if the tasks fit into the first category (Crises Category). For the second category, goals and planning, the manager will plan and delegate tasks, for the third category, interruptions, the manager will delegate efficiently, and for the fourth category, distractions, the manager should abandon all unnecessary activities as these could interrupt optimal working flow.

Using 360 Feedback means integrated feedback on behalf of management, collaborators, clients, with teams using Agile principles at the same time. The main purpose of 360 Feedback is to obtain objective, but diversified feedback, to build trust in the organization, to offer a consistent level of transparency, to reduce the distance between the manager, collaborators and other stakeholders, at a punctual and communicational level, as well as for identifying the strengths and the development axis of the projects.

Management by walking around and listening (MBWAL) (Peters & Waterman, 2006) represents one of the indispensable tools of Agile management as this allows the manager to create and maintain a trustful relationship with the team members and to eliminate the distance between him/her and other collaborators, to promote bidirectional communication and less formality. Another vital tool is one-to-one communication, which can last between 10 minutes to half an hour between team members and the manager. These short

meetings are held weekly with each team member, showing the manager's availability to listen, support and provide feedback (Schwaber, 2004).

For beginner managers, it is tempting to start a Gantt chart or PERT map for coordinating the project, as this can provide project control. Even though these instruments can analyze the progress of individual tasks and help in pinning the ones that are finalized, comparing current completion with planned tasks, they still lack the unpredictability tracking of the project. As the team gathers information about the system and the client obtains information about the team needs, a part of the marked tasks will become irrelevant, while new tasks will be identified and will have to be included. Therefore, the plan will change as far as structure and content are concerned.

More efficient planning strategy includes making detailed plans for the following week, general plans for the next three months and very schematic planning for more than three months. This Agile approach suggests that immediate-task planning is an optimal approach to software development, at least.

RESEARCH METHODS

For the purposes of our exploratory qualitative research, we chose the case study method. We chose an in-depth investigation of a working environment where Agile project management methods are applied. Participant observation was used as a tool for data collection purposes, specifically for analyzing the attitudes, behaviors and processes of a distributed team managed in the Agile framework. A series of declarations regarding the Agile practices were gathered from the software development teams in a distributed context (cross-teams). The organization we selected is an IT organization, focused on delivering IT applications. The two software development teams are situated in Bucharest and Brussels. The teams are newly formed and they implemented Agile practices within the last year as they are still in a transition period. The ownership of the organization is in Brussels. The observations we conducted have the purpose of analyzing the current state of implementation and usage of Agile practices, the needs of Scrum teams within a multinational organization, and understanding the ways in which responsibilities are being redefined.

The participants of this study include three development teams (between four and nine members), one IT project manager (focusing on the transition to Agile) and Human Resources personnel. The research interval was between January 2017 and March 2017. In terms of data collection, the observations grid was used four hours/day, by three trained researchers. Informal conversations with developers were necessary for clarifying the findings

we found from the observations and for a more complex understanding of whether Scrum methodology and implementation is reflected in the attitudes and perspectives of the software team. The trained researchers added notes on the observation sheet with details reflecting the context and notes from discussions with staff members.

The observations were specifically focused on the Bucharest teams and their level of knowledge, implementation and stability of Agile practices and the coordination practices. The three levels included in this evaluation refer to the following concepts: Value (Level 1), Stability (Level 2), and Speed (Level 3). For the first level, the drive is that of delivering value for both individuals and processes. The business overall, as well as the IT department, collaborate to add value.

For the second level, *stability* was considered: product stability, for instance, or consistent value flow within a specified time interval. Quality is defined in the development process by the business as well as the IT department (which define product value and final value).

The third level includes *speed*: the value needs to be delivered rapidly and efficiently. The developers and the operations team need to share the same purpose, to deliver value on a frequent basis, with low risk. The delivery procedures are automated. The dates are collected and used systematically as product feedback.

The observation grid included a series of dimensions that were established to identify the level of applicability of Agile principles and Scrum practices. At the general flow level, we analysed concepts such as: team, power delegation, autonomy, ownership over processes, technical management, command and control approach, Application Manager visibility and transparency, planning, estimation, design session, decision making, collaboration between development and business teams, understanding of business value, progress and improvement measurement, user experience and acceptance criteria, prioritized backlog, effort estimation, continuous improvement process, client feedback, and team visualizing. The analysis included examining the decision making flow and its specificity, how leadership is implemented, the execution strategy, user experience and characteristics, product feedback and evaluation, end product value, transparency and communication specifications, and business model typology. The aspects that were investigated are relevant for identifying whether Scrum implementation is successful within the analyzed teams. As far as Agile practices are concerned, we concentrated one part of the analysis of visual management and Scrum ceremonies.

For conducting this analysis, we also included a section dedicated to the team platforms, analyzing a series of technical aspects: Cloud, applicative configuration, IDE (Integrated Development Environment), implementation

processes, packaging, software production, system configuration, analysis platforms, performance testing and unit tests.

ANALYSIS OF THE RESEARCH RESULTS ---

A series of observations over general flow were made as a result of this qualitative research approach. We chose to regroup these observations according to the fundamental concepts on which they are based. The analysis shows that Agile project management implementation is not an easy process. For reaching a stability level, the individuals and the teams, in general, need to develop an Agile mind-set. As far as role redefinition is concerned, the analysis shows a somewhat level of conformity of the roles within a distributed context. A part of the Agile responsibilities cannot be undertaken by the working groups as they are in a transition context. The research lot included only the team members from Bucharest, so within the Brussels team, the results could be different.

General flow observations: As far as the team is concerned, we noticed that not all groups form a team – the development teams do not work together to reach a common purpose. They do not represent a team as Agile defines a team to be. Agile suggests that the team members share common values and principles for functioning, while in the analyzed context, the cross teams face this major drawback due to work distribution and geographical separation.

In terms of delegating power, we observed that the teams do not have enough decisional involvement as far as tasks are concerned. When team members do not have the authority to ask for changes, the team is constrained to accept tasks and responsibilities as they are given, even if these have a negative reflection on their daily activity.

Regarding autonomy, in most cases, Bucharest teams are not completely autonomous due to the distributive factor, as they are dependent on the defined processes in Brussels. The activity in distributed teams should follow the same set of rules, principles and values. Otherwise, challenges may appear when a group holds the decisional factor over the other one. Team autonomy is suppressed in this context. If Bucharest teams had the capacity to get involved in the decision making processes and, as such, gain independence in Scrum sprints, these could be useful factors in team self-organization.

Further analysis showed that the Bucharest development teams do not have a stable level of process knowledge and neither do they have a level of autonomy over the property of the process knowledge. Therefore, once again, the Romanian teams depend on the Brussels teams. According to Agile principles, the software team should self-organize to end a task.

In regard to technical management, although all teams have technical capacities and skills, these are not included at a managerial level, as this level involves the business area. Considering the fact that most applications have high complexity from a technical perspective, as well as from a business perspective, by using these, a superior design can be applied, and also information transfer could become efficient enough to gradually coordinate these aspects.

By analyzing the aspects related to the command and control approach, we observed that this dimension does not totally belong to the Bucharest teams. When teams do not have the possibility to decide how to self-organize, due to problematic communication or inappropriate informational exchange or due to artificial dependencies, a series of problems appear as well as duplicate tasks. This context is not compatible with Agile practices either.

Planning visibility and estimation practices were also analyzed. Bucharest development teams are not involved in planning decisions neither in estimation, design, nor feedback sessions. Most of the time, they do not have access to the person handling the Application Manager, as the responsibilities are not clearly set. When the development team from Bucharest is not involved in this set of activities, the knowledge exchange opportunities are wasted. Due to the lack of transparency the team's growth is ceased, creating a context in which the team from Bucharest performs some tasks for the Brussels teams, drastically decreasing the chances to function independently.

Regarding the concept of value from a business perspective, we noticed that a full understanding of this concept is still lacking. As far as measuring progress and ongoing improvements are concerned, although the cross teams engage in video-conference meetings on a daily basis, these do not have a specific and stable way of measuring progress. Decisions and improvements are often difficult to achieve as they are based on a series of false assumptions.

We also analyzed the user experience and the acceptance criteria and in almost all cases the dialogue between the development and business teams is not guided by user experience. On the other hand, developers are being given technical tasks which do not have clear acceptance criteria. Working directly on technical tasks reduces the involvement levels in designing the software and also in decision making. Without value based user experiences, the chances of understanding a whole perspective decreases. Also, without the acceptance criteria, defining and visualizing work becomes deficient: the lack of automated acceptance testing, error protection, and unwanted items in the production environment can emerge.

As far as task prioritization is concerned, we found that the Romanian teams do not have full visibility over this, suggesting that a series of problems can appear. Understanding the greater perspective can help developers

apply the existent information on current platforms. Also, these priorities are essential for developers and the entire team in order to focus on the most important aspects of the project.

Analyzing the team estimation efforts, we noticed that not all the development teams are involved. Most of the times, the estimations are not conducted by the individuals working on the product or the project. Not incorporating user experience factors can cause many problems including developers not being able to adopt pertinent solutions in the functionality implementation. The estimation process should provide the capacity to guide any team member in adequately leveraging tasks.

Moreover, as far as continuously optimizing processes are concerned, most teams are not efficient with the activities and functional improvement initiatives. This aspect is due to the lack of continuous optimization processes which should be defined – such as retrospective analysis (within meetings). Without these processes, teams do not have the capacity to repair autonomous problems appearing at a local level, and several evolving opportunities can be missed. This can lead to low morale at a collective level due to frustrations and functional blockages (Sutherland, 2014).

As far as feedback is concerned, an analysis of the results showed that this was not made visible to all team members. So, in terms of feedback transparency, the teams are still at a beginner level and their ability to self-regulate and learn from their own mistakes becomes difficult. Feedback and dialogue help in building trustworthy relationships between the client and the development teams, offering a sense of belonging when the work is appreciated by others.

With regard to the level of propriety of the knowledge process flow and that of team value, we observed that a Jira table was used. Even so, not all the teams have access to it and, as such, knowledge over process is deficient. The teams cannot visualize the entire process on the Jira table. When the ownership over processes is not complete, introducing and implementing improvements become difficult. A series of opportunities can be missed and wasted, as longer tasks, blocked items and dependences interfere. In the meantime, this challenging context is exposing the lack of understanding roles: who is responsible for what in the process. This can lead to decisional issues over the degree of work efficiency according to necessities.

Some observations were made over continuous delivery as well, where we noticed that there is no visibility or collaboration between the Operational-Development departments. In most teams, there is no visibility amongst the developers and the operational department and this is a problem mainly because, without this collaboration, enhancement opportunities

over delivery process, software quality and non-functional requests such as performance and implementation rapidity are lost.

Observations about the Agile management and principles. We noticed that, at a company level, Agile managing principles have a fundamental basis of implementation, but at the Scrum methodology level, as far as the roles are concerned, there is still a consistent difference between the desirable agile framework and the actual application. Perhaps this is due to the transition period or is due to the fact that the teams are newly developed, but this still needs to be investigated. Within the Scrum teams, there are only three main roles (development team, Scrum Master and Product Owner), and their understanding of these self-organizing roles is fundamental to the success of the entire project. The development team represents a significant investment of the business line, being essential to maximizing winnings (ROI).

On an executive level, Agile principles reached a somewhat level of stability, as the managers use all the available tools for creating a comfortable environment for development. At the Scrum team level, the principles cannot be applied thoroughly, making team responsibilities and role definition hard to define. As a result of our analysis, we concluded that, at the development team level, there is no homogeneity for standard roles, each of them having a more precise role. Also, the lack of self-regulated capacity at a regional level could lead to undermining trust and ceasing cross-team cooperation.

Throughout our analysis, the lack of proper disclosure of information regarding task setting and project evolution affects transparency that should be present at all the levels of team engagement. Unfortunately, due to the Scrum management processes being in the early stages of team adaptation, communication and collaboration for achieving common objectives appear to be challenging.

Regarding the Scrum sprint activities, challenges emerge during actual implementation. Only a part of Scrum activities can be successfully implemented considering the distribution of the teams' authority: the development teams from Bucharest depend to a large extent on the Brussels teams. Scrum team members are not involved in the planning, estimation, and decision-making sessions. Even though sprint planning appears to be present, with the Brussels team describing specific functionalities required for the final product, full access to the backlog is not possible for the Bucharest team. Moreover, the Scrum team (the developers) from Romania does not have the authority to choose subsets of tasks for the next sprint, as these are chosen for them by the Brussels team. Even though Scrum meetings are conducted with the Brussels team through video conferences, these are not necessarily conducted on a daily basis, unless needed when unexpected requirements interfere. Considering the lack of the decisional factor at the

team level, critical or constructive retrospective sessions are absent, at least at a local level. As there is no clearly defined method for measuring progress, optimization criteria cannot be established. As far as the developer's role is concerned, Scrum methodology involves a high degree of authority. What we noticed is that most of the tasks are set by others and not by the developer. Not having ownership over the set of tasks impedes the developer to perform properly, not being able to have a position over the equity and distribution of the task set, and withholding the possibility to engage in continuous learning.

The role of the Scrum Master can coincide with the role of the line manager or with the role of the IT manager, whose task is to supervise and make sure that the transition finality and implementation of a project are according to plan. This is the case for the newly formed teams which are in a transition process. From a traditional perspective, the role of the project manager is to coordinate the entire project and to maintain its planning within the initial parameters. The project manager is responsible for developing and maintaining the relationships with the project stakeholders, for setting and delegating tasks and for creating a working calendar. The working team, in this context, is not accountable for project failure. This aspect represents one of the main value pillars of the Scrum methodology, as the project manager role is completely eliminated, with the responsibilities being shared by the Product Owner, Scrum Master and the development team (Stellman & Greene, 2014). However, in new teams such as the ones we investigated, four roles were identified: project manager, developers, Scrum Master and Product Owner.

Within IT departments, and within the software production environment, the IT project manager has a similar role to the Application Manager who does not have the possibility of maintaining a status for the KPIs, due to not having access to budget planning. This is why the involvement of a project management cell is useful to maintain this activity. We noticed that a part of these attributions are given to an external team, not to the Scrum team.

Therefore, as far as redefining responsibilities is concerned, within new Scrum teams, we can conclude that the motto is that of delivering value (Level 1). As the maturity level of the Scrum teams is relatively low, a plan of action is necessary for training Scrum teams into the Agile mind-set: for working independently, for self-organization, and for cooperation purposes. In the current context, responsibility management cannot be undertaken by developers or by the Scrum Master or Product Owner, as the adopted management framework is Scrum-Waterfall. Even though the Scrum Master and Product Owner coordination involvement exists, at the technical, as well as at the functional level, the teams are still in a transition period, with global project coordination being conducted by the IT project manager and by

a project management team which has the responsibility for managing tasks and following their development and implementation progress.

It can be concluded that the Agile methodology implemented within the IT project management presented in this paper is still at its infancy level. Even though Agile project management first emerged as a response to the highly competitive and dynamic environment determining organizations to focus on their internal strategic agility to cope with external factors by efficiently managing projects, the case study presented herein showed many challenges with respect to self-organization, decisional factors and collaboration between distributed Scrum teams. Moreover, a deficient collaboration between the distributed developer teams and with the Product Owner lead to misunderstandings in product deliveries and a long-lasting functionality production process, which are contradicting two of the core principles of the Agile philosophy: fast delivery of functional software and collaboration between developers and clients.

DISCUSSION

Agile principles and Scrum methodology were analysed within distributed teams. There are communication risks, production and efficiency risks at stake, and also cultural and leadership elements that need to be considered in future research. Also, quantitative research methods should be included, such as surveys, in order to identify what are the perceptions of team members over these types of principles and how these perceptions affect their attitude and behaviors to any kind of Agile and Scrum efforts. For Agile to work, an *agile mind-set* should be developed at a team level, and this does not refer exclusively to training and keeping Scrum methods active, but more on understanding why openness, cooperation, accountability and ownership represent important factors in reaching project success. As Theocharis, Kuhrmann, Münch and Diebold (2015) emphasized, the Scrum methods used in software development projects are set especially to provide fast reactions to dynamic and changing demands from clients, and proper monitoring is expected in order to make the necessary adjustments. Within our case study, no monitoring of the project progress was made and, as such, proper optimization of processes was lacking.

In the software industry, particularly within distributed teams, the challenge to understand and to implement the methodology seems to become a paradigm. In this context, we state that the human factor and delivering an integrated environment are key elements that still need an attentive and unitary implementation within the analyzed segment. This implementation needs

a qualitative guide, as well as training and a specific period in which information, practices, principles and experiences are understood and integrated.

Also, the Agile project management implementation is challenging from a role definition perspective, as the results showed that there was still a tendency to ask for the help of the IT project manager (traditional implementation) for guiding transitory projects. This manager's roles and attributions are limited considering the lack of homogeneous knowledge within the project. Role definition is dependent on team maturity and on finalizing transition projects that still require specific traditional guidance. When distributed teamwork organization is present, we must consider the characteristics of collaboration and communication between team members. Efficient communication is needed when teams work jointly on projects, but merely setting common objectives is not enough for enhancing collaboration between team members. According to Vlaar, Van Fenema, and Tiwari (2008), the quality of interaction appears as the most crucial factor for successful collaboration and communication between team members working in software development. As far as our research results are concerned, client feedback is not visible for the Bucharest teams and deficient communication is present between the Product Owner and the developers.

From a broader perspective, this methodology may appear to be readily applicable, but its implementation can be considered linear (Kenneth, 2012). Modus operandi is essential, as far as cohesiveness and clarity are concerned, and this should be implemented at a functional and operational level over all the projects, for allowing improvement at the production efficiency level, as well as at the product delivery level and at reducing the time in which these are generated.

The increasing spiral of the process inflation is responsible for at least a part of this failure, even if it is emerging on the basis of good intentions. The Agile principles were created as a means for helping teams to step out of the process inflation and for focusing on easy techniques to reach different objectives (Martin & Martin, 2006). Scrum methodology, in particular, cannot be applied everywhere, but it can be a basis for supporting complex development efforts.

Research limitations

The limits of this research paper include the non-homogeneous character of the teams being evaluated. Half of the Scrum teams are working in Bucharest offices, while half work in Brussels. From this perspective, we analyzed exclusively the Bucharest teams. Therefore our information access was limited: we did not have a holistic perspective of the project management processes and over the decisional factors specifically. Access was restricted to the team from Brussels and, as such, we were only able to gather some level of data

from the Bucharest teams. Also, no access to spontaneous communication between the distributed teams was granted: the teams communicate with each other for solving specific tasks, but the communication flow and the content they share was not available to us for analysis for our research purposes. Work-related tasks were also difficult to follow as the tasks were being distributed regionally and, considering we did not have access to the backlog, we based our notes on what the team members communicated.

Furthermore, the collection of data was difficult because the software developers were not very open, especially at the beginning, and transparency was difficult to attain. Understanding the level of implementation of Agile principles and Scrum methodology is challenging when a proper understanding of the technical language does not exist, which represented another limit of this research. Additional explanations were required by the team of researchers when they were confronted with technical concepts they did not know.

Therefore, the flow of the working tasks was challenging to identify, as they were distributed regionally and full access was not granted in this sense. In this type of research, observing task flow management is very important. The results of this research cannot be extended or generalized outside the context of the case study that was presented in this paper.

CONCLUSION

The primary purpose of developers and development teams in general is to deliver functional software within an optimum timeframe that can bring the highest value to all the stakeholders involved, including employees and clients. However, most projects fail on a general level or do not manage to deliver the desired value.

Our research showed many methodological and practical aspects regarding the implementation of Scrum methodology, including the fact that distributed teams require continuous orientation and involvement on behalf of the management level so as its evolution can become progressive towards homogenous practices. Irrespective of the analyzed environment, the Agile framework brings a series of benefits to projects and general mind-sets, as it is focused on people and on delivering value, being an iterative and incremental process. Agile management promotes receptivity, value and collaboration within and between teams, ensuring the development of a value-added framework in which tasks are set only by urgency and importance factors.

To conclude, the research results showed that teams have a collective mentality built around Agile principles, focused on people and result, which

could result, in the future, in an identical plurality on a mind-set level that could overcome regional and cultural challenges, but this still needs to be investigated. Training practices have the capacity to grow teams from the concept level of value, in which collaboration is focused on the business and information technology lines, where constant feedback flows are present, and transparency is critical. At this stage, Scrum teams can reach the point at which they are focused on frequent delivery of value, with low risk over automation delivery and over using dates systematically as a way of gathering product feedback. As the results showed, ongoing improvements need to be made.

The declarative evaluation that we conducted over Scrum team members from Bucharest did not include measuring the performance of teams within the distributed context we investigated. However, it offered, however, a general view over the evolution of implementing Agile principles in project management and, of Scrum methodology, evaluating the transformative flow of redefining responsibilities in a distributed context.

The methodological foundations of Agile and Scrum have started gaining popularity in the software industry and continuous research over implementation techniques should be conducted. As far as this aspect is concerned, not all software production companies follow the same implementation model, as this is dependent on the organizational culture of the business and local factors as well.

References

- Anderson, D. J. (2000). *KANBAN. Enclenchez le Moteur d'Amélioration Continue de votre IT*. USA: Blue Hole Press.
- Aubry, C. (2011). *SCRUM. Le guide pratique de la méthode Agile la plus populaire* (2nd ed). Malakoff, France: Dunod Editions.
- Beecham, S., Baddoo, N., Hall, T., Robinson, H., & Sharp, H. (2008). Motivation in software engineering: A systematic literature review. *Information Software Technology Journal*, 50(9-10), 860-878.
- Cervone, H. F. (2011). Understanding agile project management methods using Scrum. *OCLC Systems & Services*, 27(1), 18-22.
- DeMarco, T. & Lister, T. (2013). *Peopleware: Productive Projects and Teams* (3rd ed.). New Jersey: Addison-Wesley.
- Denning, S. (2016). How to make the whole organization « Agile». *Strategy & Leadership*, 44(4), 10-17.
- França, A.C.C., da Silva, F.Q.B., & de Sousa Mariz, L.M.R. (2010). An empirical study on the relationship between the use of agile practices and the success of Scrum projects. *Association for Computing Machinery, ESEM'10*, September 16-17, 2010. Italy: Bolzano-Bozen.
- Kenneth, R. S. (2012). *Essential Scrum: A Practical Guide to the Most Popular Agile Process*. (1st ed.). Michigan: Addison-Wesley Professional.

- Kerr, J. M. & Hunter, R. (1993). *Inside RAD: How to Build a Fully Functional System in 90 Days or Less*. New York: McGraw-Hill.
- Lehnen, J., Schmidt, T.S. & Herstatt, C. (2016). Bringing agile project management into lead user projects. *International Journal of Product Development*, 21(2-3), 212-232.
- Martin, J. (1991). *Rapid Application Development*. Indianapolis, USA: Macmillan Publishing Co.
- Martin, R.C., & Martin, M. (2006). *Agile Principles, Patterns, and Practices in C#*. Upper Saddle River, USA: Prentice Hall.
- Messenger Rota, V. (2010). *Gestion de Projet Agile*. Paris: Groupe Eyrolles.
- Misra, S.C., Kumar, V. & Kumar, U. (2010). Identifying some critical changes required in adopting agile practices in traditional software development projects. *International Journal of Quality & Reliability Management*, 27(4), 451-474.
- Peters, T., & Waterman, R.H. (2006). *In Search of Excellence: Lessons from America's Best Run Companies*. New York, USA: Collins Business Essentials.
- Rasnacis, A., & Berzisa, S. (2015). Adaptation of agile project management methodology for project team. *Information Technology and Management Science*, 18(1), 122-128.
- Saxena, A., & Burmann, J. (2014). Factors affecting team performance in globally distributed setting. *Association for Computing Machinery, SIGMIS-CPR'14*, May 29-31, Singapore, 25-33.
- Schwaber, K. (2004). *Agile Project Management with Scrum*. Redmond, WA: Microsoft Press.
- Stellmann, A., & Greene, J. (2014). *Learning Agile. Understanding Scrum, XP, Lean and Kanban*. New York, USA: O'Reilly Media Inc.
- Sutherland, J. (2014). *The Art of Doing Twice the Work in Half the Time*. New York, USA: Crown Business.
- Theocharis, G., Kuhrmann, M., Münch, J., & Diebold, P. (2015, December 2-4). Is Water-Scrum-Fall reality? On the use of agile and traditional development practices. In P. Abrahamsson, L. Corral, M. Oivo, & B. Russo (Eds.), *Product-Focused Software Process Improvement*. Paper presented at the Proceedings of the 16th International Conference PROFES, Bolzano, Italy (pp. 149-166). Springer. Lecture Notes in Computer Science.
- Vickoff, J. P. (2009). *Méthode AGILE. Les meilleures pratiques. Compréhension et mise en œuvre*. Vanves, Hauts-de-Seine: Editions QI.
- Vlaar, P.W.L., Van Fenema, P.C., & Tiwari, V. (2008). Co-creating understanding and value in distributed work: How members of onsite and offshore vendor teams give, make, demand, or break sense. *MIS Quarterly*, 32(2), 227-255.

Online sources

www.Agilemanifesto.org

www.scrum.org

Abstrakt

Część teoretyczna niniejszego artykułu koncentruje się na analizie makro filozofii systemu Agile, jednej z najpopularniejszych ram zarządzania projektami w branży oprogramowania. Druga część artykułu zawiera studium przypadku analizujące praktyki Agile w ramach korporacji na poziomie produkcji oprogramowania. Jeśli chodzi o metodologię, badania jakościowe przeprowadzono na podstawie studium przypadku zespołu w kontekście rozproszonym ("cross-team"). Obserwacja uczestników była wykorzystywana jako narzędzie do zbierania danych o osobach i procesach w zespołach. Celem badań było przeanalizowanie obecnego stanu wdrożenia, korzyści i wyzwań wynikających z zastosowania platformy zarządzania projektami Agile, a także potrzeb zespołów programistycznych w ramach wielonarodowej firmy. Jednocześnie w artykule przeanalizowano redefinicję ról, stosując ramy zarządzania projektami Agile w ramach organizacji pracy zespołu rozproszonego. Badanie ma na celu wskazanie przyczyn leżących u podstaw wyboru metody zarządzania Agile zamiast Lean Management w branży rozwoju oprogramowania, ze szczególnym uwzględnieniem poziomu wdrożenia metodologii Agile/Scrum i zasad w rozproszonych zespołach. **Słowa kluczowe:** Agile, Scrum, IT, zarządzanie projektami.

Biographical notes

Carmen Novac teaches at the College of Communication and Public Relations at the National University of Political Studies and Public Administration in Bucharest. She has participated as an expert in various national and international projects such as Instruments, Analysis and Screening Techniques for the Regional Policy Institution at the European Integration Office (2005), Developing entrepreneurial competencies through transnational transfer of good practices and professional training of Romanian entrepreneurs – STEPS (2011-2013), HER – Entrepreneurship for Human Resources (2012-2013). She is interested in researching professional performance evaluations, but she also focuses her research on personnel management, project management, employee motivation, and innovative teaching methods.

Raluca Ciochină teaches *Digital Marketing* and *eBusiness* online at the College of Communication and Public Relations at the National University of Political Studies and Public Administration in Bucharest. Her research focuses on online social networks and user behavior, human resources management and online recruitment and selection. Her position as an evaluator within the national project called EHR – Entrepreneurship for Human Resources (2012-2013) raised her interest in researching sustainable entrepreneurship and organizational communication techniques for employee engagement.