

Wojciech WALCZAK*

Dorota KUCHTA*

RISKS CHARACTERISTIC OF AGILE PROJECT MANAGEMENT METHODOLOGIES AND RESPONSES TO THEM

Agile methods for project management are often treated as methods for mitigating risk. However, there is disagreement as to whether explicit methods of risk management should be used in projects which are executed according to Agile methodology or is the implicit risk management built into Agile methodologies sufficient. To contribute to the discussion, an attempt has been made to identify risks that are either caused by the introduction of an Agile methodology to a project or become more significant when such methodology is in use. If such risks exist, this would be evidence that explicit risk management is required, even in the case of Agile methodologies. The results of this research may be useful for any organization that is in the process of selecting a methodology for project management and is considering Agile methodologies.

Keywords: *risk, risk management, Agile software development, Agile project management*

1. Introduction

Starting from the 1990s, several Agile methodologies were created that became popular, especially in the IT industry. All of them have common characteristics, which are well described by the *Manifesto for Agile Software Development* [14]. These methodologies are designed to address most popular issues in information systems development (ISD), primarily through iterative development and close, intensive communication between various project stakeholders. Because of this, the Agile methodologies of project management are often considered as having built-in methods of risk mitigation, which reduce the probability of project failure, especially if the project is carried out in a fast-changing environment.

*Institute of Organization and Management, Wrocław University of Technology, ul. Smoluchowskiego 25, 50-372 Wrocław, e-mail addresses: wojciech.walczak@pwr.edu.pl, dorota.kuchta@pwr.edu.pl

However, it seems that there is still no clarity as to whether Agile methodologies, with their implicit mechanisms of risk mitigation, sufficiently reduce project risk by themselves, or are explicit methods of risk management still required. This discussion is especially intensive within the communities of practice, which are strongly contributing to the development of Agile methodologies but has also been noticed by researchers [5]. Disagreement between various signatories of the *Manifesto for Agile Software Development* is also visible [14]. Beck, the author of the Extreme Programming methodology (XP), states that XP is not sufficient for risk management in the case of safety- or security-critical projects [1], while Highsmith, the author of the Agile Project Management methodology (APM), suggests that explicit risk management process might be redundant, because the APM methodology was designed in a way to handle high-risk projects [4]. On one hand, the scientific literature proposes methods for risk management inspired by Agile software development methodologies [9], while at the same time other research identifies areas of risk management which are not covered implicitly by Agile methodologies and propose solutions as to how explicit risk management can be combined with Agile methodology [7, 8, 11, 12]. To contribute to this discussion, the authors of this paper investigated whether there are any risks caused directly by the usage of the Agile methodology of project management or which become more significant to a project when Agile methodologies are applied. If such risks can be identified, then this is evidence that an explicit process of risk management is still required, since obviously these risks are not mitigated by the Agile methodology itself.

The results of this research may be especially meaningful for any organization that is about to take a decision as to whether to introduce Agile methodologies – by either entirely replacing the methodology currently used in the organization with an Agile one or utilizing Agile methodologies in just a single project. Furthermore, the authors believe that this paper will help any organization that is using or plans to use Agile methodologies, by providing input that will allow to construct better methodology and achieve a faster and less problematic process of transitioning to Agile methodologies.

The aim of this research is to create a list of risks characteristic of Agile methodologies of project management in the context of ISD. Following the ISO 31000 standard [13], the authors define risk as the “effect of uncertainty on objectives”, where “an effect is a deviation from the expected – positive and/or negative” and a risk is “characterized by reference to potential events and consequences, or a combination of these”. All of the risks listed in this paper have their source in the Agile methodology itself or their probability/consequences are greater when these methodologies are in use. This research is not aimed at identifying the most important risks in information system development in general, thus the outcome is not a complete list of risks in Agile ISD. Since “it is widely recognised that Agile methods themselves were introduced to combat well-known risks associated with ISD project failures such as scope creep, cost overruns and schedule pressures” [2], the list of risks presented may be perceived

as a list of secondary risks to be considered when applying Agile methods. This list is not generic, since the research was done based on a limited number of projects, which were carried out in the same company – Agile projects in another company may have risks that were not captured during this research and, at the same time, some risks, which were very visible and important in the projects examined, might be irrelevant for other organizations.

2. Research methodology and background

Six projects were examined during the research. All of the projects were conducted in the same, large company operating in the telecommunications industry. The goal of each project was to customize a complex telecommunications/software product to the needs of an external customer and deploy it at the customer's site. In each of the projects, major development and testing efforts were required. The product to be customized and technology used were the same in all the projects. However, the requirements for the customization of the product and the environment into which the product was to be introduced were unique. Although the project teams were geographically distributed, the members of the development team working on a given project were always located in the same center. In the case of three projects, the development team was located in Poland, in another two projects the development team was located in South Africa and one project had a development team located in Germany. The customers were large mobile network operators from Europe, Africa and Latin America. In all of the projects considered, the customer was located in another country than the project development team.

The methodology used for software development and project management in the examined projects was based on the most popular Agile methodologies: Scrum [10], Extreme Programming [1], and Agile Project Management [4]. Although in the case of five of the six investigated projects the project team had little or no earlier experience with Agile methodologies, the projects are considered to be Agile, since the project lifecycle was fully in line with Agile methodologies, project roles adapted to or replaced with Agile equivalents, important Agile practices were introduced and all the project team members had been thoroughly trained in Agile methods of development. The Scrum methodology was fully introduced with only minor adaptations, where the main difference to pure Scrum was the introduction of a product owner team, even if there was just one Scrum Development Team. Scrum, with 2- or 3-week Sprints, was the primary methodology used in all six projects. Additionally, the following XP and APM practices were introduced: continuous integration, sit together (the entire development team worked in the same room, only project managers and the people responsible for communication with customers or on site activities usually worked at a dif-

ferent location, often in a different country), Whole Team, Informative Workspace, Stories (not in all projects), Incremental Design (only to a limited extent, since the projects were about customizing an existing product and not building a completely new one), Real Customer Involvement (only to a limited extent, not in all projects), Incremental Deployment (not in all projects) and Team Continuity. The practices from the Agile Project Management methodology that were introduced include: Project Data Sheet, Customer-Development Team Interface (the product owner team was responsible for providing information to the Development Team about the vision for a product, its required features and their priorities), Feature Cards, “Release, Milestone, and Iteration Plan”, Iteration 0 (not in all projects), Low-Cost Change, Daily Team Integration meetings, Participatory Decision Making, Daily Interaction with the Customer Team, “Product, Project, and Team Review and Adaptive Actions”. Please note that different terminology is used in different Agile methodologies, e.g. the “Product, Project, and Team Review” practice from APM is the same process as the Sprint Review and Sprint Retrospective ceremonies from Scrum. In this paper, primarily the Scrum terminology will be used and the names from other methodologies will be used only if there is no equivalent expression in the Scrum terminology.



Fig. 1. Risk breakdown structure in use (based on [6])

One-on-one, semi-structured interviews were conducted with key project stakeholders after the end of the project: project managers, product owners, scrum masters,

line managers and experienced development team members (who had played the role of a project leader in earlier projects). The questions aimed to identify the strengths and weaknesses of the observed projects related to the Agile methodology, as well as threats and opportunities for future Agile projects. The organization in which the projects were carried out had in the past a mature methodology based on the waterfall model and the interviewees mostly provided a comparison of the newly introduced Agile methodology with experiences from projects conducted earlier using the waterfall approach. All the interviews were recorded and a detailed analysis of the input obtained was done based on the recorded material. The interviews were supplemented with an analysis of the project documents (including Agile artifacts) and active observation.

In order to structure the information gathered using the method described above, a Risk Breakdown Structure (RBS) is used, which is created based on the classes and elements of the SEI risk taxonomy [6] (Fig. 1). The following sections of the paper are structured according to the RBS.

3. Development cycle risks

The first class of risks covers all the opportunities and threats that are related to constructing the product for which the project was established. Therefore, these risks are connected with activities that have a direct effect on the product.

3.1. Requirements risks

This section covers risks that are related to the requirements for the product, i.e. risks that may occur during analysis, definition and management of these requirements, as well as risks that have an effect on the degree to which the end product satisfies the needs of the customer and users.

Agile methodologies are designed in such a way to allow very cheap changes to the scope of a project which is especially useful in the case of projects with unpredictable requirements, or in the case of continuously evolving requirements. Easy/cheap changes in the requirements and scope of a project are considered to be an opportunity, since several preconditions need to be met in order to make them happen. From the perspective of the development process, it is sufficient to change the content of the Product Backlog. Although, in the case of some projects (e.g. fixed price), there might still be a need to use some of the traditional techniques of scope management, in these cases the use of Agile methodologies introduces a chance that the introduction of any formal request for a change will be much cheaper and faster.

To maximize this opportunity, the Agile methodology should be introduced completely into a project and the project manager should actively interact with the product

owner (possibly he could take this role) and manage the scope of the project through the Product Backlog. Also, detailed clarifications on the items in the Product Backlog and design activities should be made as late as possible in the project – in the case of an Agile project, this means that they have to be finished just before the iteration in which the item is to be delivered. The customer needs to be educated in Agile methods and willing to take an active role in the project during the Sprints.

3.2. Design risks

The risks related to design that were observed in the projects investigated were not caused by the Agile methodology, or transition to the methodology, neither were there any reasons to believe that their severity or probability would be different if another methodology were used. This is caused primarily by keeping the same approach to design in Agile projects as earlier when using the waterfall methodology – the only difference was the point in time at which the design was prepared: instead of having a single design phase early in the project, the design of features was done in parallel to the implementation of other features. Another reason for this is that the projects investigated were about customizing a product to a customer's needs, therefore the overall architecture and design of the product were known upfront and during the project the design tasks focused only on the features to be customized or added to the product.

3.3. Implementation risks

As in the case of design risks, no implementation risks characteristic of the Agile methodology were found.

3.4. Test and evaluation risks

It is especially important to tackle risks related to verification or validation before the start of Sprints, since reactive response to the effects of such risks might be very expensive. In this category, the risk of neglecting continuous integration (CI) was identified.

To keep the development efforts at a stable level, a team has to pay great attention to continuous integration and ensure that the product under development passes all the tests based on requirements implemented in earlier Sprints. To achieve this, the team has to be self-disciplined and also see the value of investing in continuous integration (introduction of the continuous integration system and its maintenance in every Sprint require additional efforts from the project team). Neglecting continuous integration

will result in increasing the effort required to make the required progress in each subsequent Sprint, thus the level of advancement of the project becomes more unclear (the end date of the project becomes more unpredictable) and also the possibility of delivering early releases of the working product to the customer is lost.

Possible responses to such risk are training the project team and project management so that they recognize the importance of continuous integration, using a continuous integration and test system that is cheap to introduce and maintain, adding criteria about continuous integration to the Definition of Done and ensuring that they are strictly followed. If a Sprint 0 is introduced into the project, then establishing the CI environment should definitely be included in it.

4. Development environment risks

As noticed in [6], the process/methodology of software development and the environment in which the project is carried out may also be sources of risks. These are the most important risks from the perspective of the goal of this paper.

4.1. Development process risks

The Agile methodology used in the projects investigated is considered to be a methodology for both project management and software development at the same time. Risks that are caused by the Agile methodology itself can be found in this section. The risks discussed here are either inherent to the Agile approach or have their source in the organizational transition towards the Agile methodology.

4.1.1. Inefficient Scrum meetings, ineffective Scrum roles

In an Agile project, many different meetings must be conducted regularly in order to make the methodology and project work. The entire project team takes part in them, thus a significant amount of project time and effort is spent just on these meetings. Also, using the Scrum methodology, new project roles are introduced – the role of Scrum Master and the Product Owner are not similar to any of the roles from traditional methodologies of project management. Although some project roles might disappear with the introduction of the Agile methodology, it was observed that often the cost of the Scrum roles introduced was higher than the cost of the project roles eliminated after adoption of Agile. The roles of Scrum Master, Product Owner, and time spent on meetings are treated here as process overheads, since they do not directly generate value for the customer. Most of the interviewees recognized the importance of the Scrum meetings and roles. However, they pointed out that the overheads were

too great. This issue was noticed in each of the projects investigated but the amount of unproductive hours varied between the projects. Inefficient meetings or ineffective roles may result in project delays and cost overruns.

Two different approaches can be taken to mitigate this risk: the process may be tailored to reduce the amount of time spent on unproductive tasks or meetings and roles can be optimized. The first approach is justified in some cases but brings a secondary risk of dysfunctional/incomplete implementation of the Agile methodology. The second approach might be implemented by establishing fixed rules for the meetings (e.g. fixed agendas for every Scrum meeting and time-boxing), providing the team with strong moderation, or training and coaching about the Agile methodologies. The assignment of the Scrum roles is another possible source of risk – in particular, acquiring an experienced Scrum Master or investing in improving the skills of a novice Scrum Master is important, since an unskilled person in this role not only consumes more effort in his/her own role but also increases the amount of unproductive time spent by the rest of the team.

4.1.2. The whole team working collectively to reach the project goal

The Agile methodologies increase the likelihood of having the entire project team working collectively towards the project goal. Agile practices such as a collocated project team, highly visible Product and Sprint Backlogs, and face to face Scrum meetings with all the team members (especially Daily Scrum meetings) establish effective distribution of information on the progress of a project between team members, remove communication barriers and foster good teamwork. At any point of time, all the team members have the same goal, while in traditional methodologies of software development there are separate sub-teams often only interested in reaching their own milestones. The Agile methodologies also eliminate the risk of conflict between groups of people with different skills who in traditional projects would be in different sub-teams. In the Agile projects investigated, different observations have been made on how the opportunity of the whole team working collectively may materialize. Some team members build a wider set of skills to be able to work on the most important current tasks, instead of limiting themselves to only the portion of tasks in which they were specialized at the beginning of the project. The teams also established ad hoc task forces consisting of team members with different expertise to tackle difficult problems. In all of the projects, team members increased awareness and understanding of the work outside of their own specialization.

To exploit this opportunity, the following approach to personnel management within the organization should be adapted – evaluation of employee performance or setting periodical targets for employees must take into consideration and appreciate the impact of an individual on the overall goals of a project and should not take the job description and official work responsibilities as the primary criteria. Also, a portion of

the team's capacity could be reserved for the team members to learn new skills. Another response would be ensuring that the Agile practices involving an entire team are properly implemented.

4.1.3. Team not able to self-organize and make group decisions

In traditional project management, it is not required for the project team to self-organize. Therefore, any risk related to self-organization is characteristic of Agile projects. The team might not be able to self-organize due to internal conflicts within the project team which prevent the team from reaching a consensus. Other reasons for such problems might be the team being overwhelmed with the amount of information to be processed, and the number of decisions to be taken or the very narrow specializations of team members which make it impossible/difficult for individuals to understand the big picture. It is worth noticing that the probability and impact of conflict between team members are greater when the project team adopts the Agile methodology, since in traditional project management there is always a single person responsible for decisions and the progress of the project does not get blocked if the project team is not able to reach a consensus in their discussions. This risk is more likely in organizations where the project team is established for a single project and disbanded after it, because people of different backgrounds, habits and beliefs are expected to make decisions related to a common mode of operation, estimate requirements, and plan a Sprint right from the beginning of the project.

If such a risk materializes, then the problem might be difficult to solve, since the development team might reject suggestions from outside the team using self-organization as an excuse. If the team is not able to set common rules of working on its own, or not able to enforce that team members conform to these rules, the reaction could be to replace some of the development team members or take the privilege of self-organization away from the team (i.e. establish a formal leader). However, since in the mid or long term the benefits from self-organization (e.g. increase in team performance) might outweigh the problems described earlier (which are most likely to occur at the beginning of a project), the best reactive response to such a risk might be helping the team to go through the storming stage of team dynamics.

Possible proactive responses to such a risk are: team building activities at the beginning of the project, providing thorough training in Agile methods to all team members, acquisition of an experienced Scrum Master, Agile coaching for the team during at least the first several Sprints of the project and limiting the self-organization of the team (e.g. by identifying ground rules, which cannot be changed without the approval of the project manager or Scrum Master). The last response reduces not only the negative risk described in this paragraph but also the expected benefits originating from self-organization.

4.1.4. Wrong team decisions

In Agile methodologies, it is assumed that the group decisions made by a team are better than the decisions of individuals. However, there is a risk that the opposite case will occur. In Agile methodologies, the development team takes over some of the managerial responsibilities – they do not only execute the project tasks but also control task execution, organize their work and make key decisions on which the future of the project depends. In traditional project management, decisions in these areas were reserved to project managers, project leaders and team managers, who were trained and experienced in managerial work. In an Agile project, wrong decisions might be made when the team does not have a full overview of the problem while making a decision, or when a decision is not taken by the team members based on merit but on other criteria (e.g. the popularity or determination of individual team members). Teams often use the method of democratic voting to speed up the decision-making process, which introduces the risk that the options preferred by a minority which is most knowledgeable on a subject will lose to those favored by the majority.

As a response to such a risk, it has to be ensured that the team consists of people who have not only technical competence but also an end-to-end understanding of the entire project. This can be achieved by including former formal leaders into the development team (many of the formal project or technical leadership roles are expected to disappear with the introduction of the Agile methodology). Furthermore, it is required that all the team members participate in the discussions and are motivated to actively contribute to them (engineers are not always keen to take part in the discussions and former project leaders might feel demoted because of losing their formal leadership role). Other responses to such a risk include training the team on Agile methods (to achieve an understanding of the additional responsibilities of a team when following the Agile methodologies). The moderation of discussions is equally important, so that the merits of an opinion are not lost and that every team member has equal opportunity to present his/her arguments – this can be achieved by hiring an experienced Scrum master/meeting facilitator.

4.1.5. Misuse of self-organization to stop/revert the adoption of the Agile methodology

Self-organization, being an integral concept in any of the Agile methodologies, may be misused by the project team and used against the adoption of Agile methodologies, especially if the project team members are new to the concept of Agile software development. In the course of an Agile project, the team meets at the end of every Sprint to discuss the way in which they work and introduce improvements. The first such meeting occurs after the first Sprint, i.e. not more than one month after the beginning of the project. During this very short time, the project team often has no chance to observe the benefits of using the Agile methodologies in practice but they may already be exposed to some of the elements of the methodology that may bring

arguments against it (e.g. long, ineffective meetings – a risk discussed separately). Resistance to change, the first, possibly negative, impressions about Agile processes, and/or little experience and understanding of Agile methods may result in the team deciding to reverse adoption of Agile methodologies or counteract the methodology unintentionally (e.g. by removing crucial elements of the methodology while trying to improve the process).

Such risks can be mitigated by providing solid training to project team members about Agile methodologies or ensuring that the role of Scrum Master will be taken by an experienced person with authority in the project team. Limiting the self-organization of the project team (as mentioned earlier in the paper) is another possible response. Elimination of risk by resigning from Agile methodologies (before they are introduced to the team) is also an option worth considering. Instead of complete adoption of the Agile methodology, a smaller change might be introduced which would cause less resistance. Further steps in the adoption of the Agile methodology might be taken in later projects. The last response to such risk may eliminate many of the benefits that the project would gain by complete adoption of the Agile methodology but it also reduces the risk associated with the next project handled by the team.

4.2. Development system risks

There is a group of risks related to the tools used in a project, which includes both software tools, as well as hardware (also the testing infrastructure). With the introduction of Agile methodology, there are different requirements regarding these tools – the risks mentioned below cover scenarios where some of these requirements are not fulfilled.

4.2.1. A lack of or limited compatibility of tools with Agile practices

Development and testing tools that work well in the case of traditional methodologies of project management may have limited or no compatibility with Agile engineering practices, such as test driven development (TDD) or continuous integration. Test tools may not have the flexibility to write test cases before implementation is started (TDD) or in parallel to the implementation of the relevant functionalities (short iterations force this approach to development). Moreover, the run time of the testing process might not provide quick enough feedback to the development process. Finally, the efforts to run regression tests every Sprint might be too high, e.g. in the case where test cases or a test suite require time consuming modifications after every change in the implementation of the system. The problems mentioned above have been observed in the projects investigated, because some of the proprietary tools (crucial to all of the projects) had serious limitations in terms of their compatibility with Agile practices

and also there were no plans in the organization to make significant investments in such tools.

All the risks related to technology and development/testing tools should be considered before the start of the first Sprint in the project. If it is not possible to run regression tests at low cost in each Sprint, automate the execution of test cases or implement and test a single feature in a single Sprint (not longer than one month), then the technology and tools used are not compatible with Agile methodologies and it has to be considered whether a change in technology/tools is possible – if not then the project should not be driven according to the Agile approach.

4.2.2. Missing infrastructure at the customer's site

Any infrastructure which would enable receiving early increment releases of the product from the vendor is required at the customer's site to fully benefit from Agile development processes and also to provide valuable feedback to the development team. However, such infrastructure might be missing if the software being implemented in the project requires dedicated hardware and/or complex integration of hardware and network components – in such cases installation of the infrastructure at the customer's site, together with the time required for shipment of the hardware from the HW vendor, may be a separate project planned to last even several months. To mitigate this risk, delivery of the test platform to a customer's premises should be planned in close connection to the software development project.

4.3. Management process risks

As mentioned earlier, the Agile methodology is also treated as a methodology of project management. The source of the risks described below is the methodology itself.

4.3.1. Better project monitoring and control

Improved control over a project is possible thanks to the high visibility of the state of the project at any moment of time, so that any discrepancy with the plan can be immediately detected. Therefore, the project manager is informed very early on in a project where there is a high risk that one or some of the project constraints cannot be achieved, leaving enough time for the project manager to react. The project manager can use reliable forecasts (the accuracy increases with each Sprint) of the project's end date based on metrics like team velocity (measured empirically), size of the backlog (always expressed by up to date estimates).

To maximize this opportunity, it has to be ensured that artifacts of the Agile methodology like burn-down charts or backlogs are properly implemented and are based on

reliable information. As a precondition, the Product Backlog needs to be up to date at all times and also the estimates of requirements need to be accurate. To keep the Product Backlog up to date, the team needs to reserve a portion of their time to clarify, refine and estimate (or re-estimate) the requirements. Introduction of the Definition of Done can help with obtaining the current state of the Product Backlog, since there is no doubt how much work in the project has been already done and how much is pending.

4.3.2. Shorter project duration from a customer's perspective

Looking at a project from the perspective of the customer, the project does not end when the product is finished and accepted – the product requires integration into the customer's infrastructure (and also with other systems), configuration, inputting the product with data (migration from replaced products) and training users of the system. Agile methodologies allow early pre-releases of working product increments, which may be shipped to the customer by the vendor and enable a very early start for some of these activities at a customer's site. Also, final acceptance of the product at the end of the project lasts a much shorter time, since the vast majority of faults have been addressed earlier and also disagreements about requirements/functionality may well have been clarified at earlier stages of the project. Thus, there is an opportunity that the overall project, as it is seen by the customer, will be finished much faster (even if the vendor delivers the product at the same time as it would have done using a traditional methodology).

To enhance this opportunity, the customer needs to be educated about these possibilities and also a good communication plan has to be established which fosters frequent, direct and honest communication between project teams on the vendor's and customer's sides. Also, the efforts of various specialists from the customer organization should be distributed differently over time and it should be ensured that the proper level of resources on the customer's side are available at any point of time in the project.

4.3.3. The methodology of project management weighs against individual career plans

The Agile methodologies follow the concept of self-organizing and self-managing teams, which formally have flat structures. The traditional roles of project leader or team leader do not exist in them, which limits the possibilities for the future career paths of team members. Employees with plans to build their own career in the direction of managerial positions, might find it more difficult to realize their own ambitions while in an Agile project, since the structure of a project team lacks formal positions of middle management. In the projects investigated this problem was very noticeable, since some of the project leaders lost their formal position of leadership and others had personal development plans, which became obsolete after the introduction of Agile methodology, although they had been earlier agreed with managers. Such individuals

perceive the adoption of Agile as being against their own personal interest and might lose motivation to work.

All the people joining such a team should be informed from the very beginning about the structure of the team in an Agile project. If an employee was hired before the transition to Agile, then a considerable amount of time should be spent on discussing new career possibilities, otherwise the organization might lose some of its most experienced and dedicated team members. During the adoption of Agile, a transition period should be considered, while people formerly having project leadership roles transfer to new Agile roles.

4.4. Work environment risks

Risks to an Agile project may also come from the work environment in which the project is executed – if it is not adapted to Agile methodologies, then it is a source of major risks to the project.

4.4.1. Organization does not follow the rules of Agile

When Agile methodologies are introduced for the first time into an organization, then there will be organizational obstacles to the adaptation of Agile. The organization might expect similar behavior from the project team and individual employees as in the case of traditional projects. Such expectations may be harmful to the adoption of Agile and, as a result, also to the project – e.g. sharing employees between projects, a single person's responsibility for an entire team, progress reports not being adapted to Agile processes, a preference for vertical communication across the organization (escalations), the possibility of removing or replacing project team members at any time (e.g. when a project with a higher priority starts), distracting team members with tasks not related to the ongoing Sprint, use of project management/reporting tools for an Agile project without any adaptations. Also, the organizational structure may be problematic for the project. If the project team members have different line managers or the organizational structure changes a lot, then the integrity of a project team is endangered, while good team work is a key element of the Agile methodologies and is a precondition for the benefits and opportunities resulting from this way of working.

The risks related to the environment around the project may be reduced in two ways: the first one is to mock the typical communication interfaces of the traditional mode of development between the project team and the rest of the organization (fulfill all of the formal requirements of traditional project monitoring and reporting). This response slows down the progress of a project and also does not allow the organization to fully benefit from the introduction of Agile methodology in this pro-

ject. Another reaction to such risks is to adopt Agile not only at a project level but also across the entire origination. Since the transformation of the organization might take longer than the project, the first changes to the organization should focus on communication with the project team, the assignment of people to projects (stabilizing the project teams) and also establishing support for the project teams in adopting Agile methods or in solving impediments that the project team encounters during project work.

4.4.2. Dependence on other teams

Cross-functional project teams, which can implement any of the customer requirements independently on any other team, are promoted by Agile methodologies and these methodologies are built on the assumption that such a project team has full control over any of the project's deliverables. Nevertheless, in reality such a situation may happen only when a project involves building a new product from scratch. Nowadays, during the development of a new product, very often pre-fabricated components are used to save time and costs, and also the team uses various software tools, which might be developed by an external company or in house. In the organization carrying out the projects investigated, some of the components of the end solution but also proprietary tools were developed by other teams and the responsibility for these components/tools stayed with them. This dependence brought different kinds of issues to all of the six projects investigated and similar risks may be relevant to any projects, where the project team is dependent on deliverables provided by a different team.

Such risks include: faulty deliverables provided to the project team, resulting in unexpected delays in the project due to the time required for correcting faults, inaccurate estimation of the project because of inappropriate specification of deliverables to be provided by other teams, lack of the possibility for the project team to implement some of the customer's requirements on its own if a deliverable that has to be modified is the responsibility of another team, and risk of misunderstanding a customer's requirements by the team which provides the deliverables to the project (due to a lack of direct contact with the customer). The responses to these risks should focus on the hosting organization of the project and should aim to reduce or eliminate the dependences between the project team and any other teams or establishing close cooperation and effective communication with any relevant teams.

5. Programmatic risks

This section contains mostly extrinsic risks – the project manager typically has no direct control over these risks.

5.1. Resource risks

All of the six projects investigated were fixed price projects with a fixed deadline. The project team, including the project manager, had no or limited influence on a number of resources, such as amount of time, number of people available, quality of team (adequacy of skills and experience), budget or available facilities (including those on the customer's side). However, it turned out that with the introduction of Agile methodologies, there are several opportunities, which can be enhanced even during the project, while using the waterfall model these opportunities are not present to the same extent.

5.1.1. Increasing team effectiveness

The Agile methodologies give the opportunity that the performance of a team will improve during a project. This improvement may be much greater than would be expected in the case of a traditional methodology. The Agile methodology provides opportunities for the project team to discuss the way a project is conducted (e.g. during a Sprint Retrospective) and often the project team finds areas for improvement. It has been noticed that while in some of the projects investigated a big improvement in the team's effectiveness was observed, other project teams performed at a similar level from the first Sprint till the last one.

To make the most of this opportunity, the Scrum development team should be given freedom to decide and experiment about their ways of working. This process of improvement should be driven from inside the Scrum Development Team but it can be facilitated by measuring team performance and making the progress of a project and results visible to each individual in the team. Large emphasis should be placed on the Sprint Retrospective meeting – the project team needs to be aware of the importance of the meeting and conduct it effectively. Some percentage of the team's capacity and project budget should be reserved for improvement-related activities during a Sprint. Attempts to save time and money in the short term (e.g. by resigning from a retrospective meeting or abandoning agreed means of improvement) may be more expensive in the mid and long term, since the project team will not increase its efficiency.

5.1.2. Utilization of full team capacity

Balancing the workload between team members using a traditional methodology of project management is done through the project schedule and requires the engagement of project managers/project leaders. In the case of an Agile project this is done by the project team itself – a well-informed team is able to react immediately to abnormalities in the execution of a task and adapt the assignment of all the team members to tasks so that none of them are ever idle during a project. Therefore, project tasks and their distribution across project team members might be optimized to a much

higher degree than would be possible in the traditional, plan-driven approach. Also, low-performing team members become visible when the Agile practice of Daily Scrum meetings/Daily Stand-up meetings is in place – this provides an opportunity to reduce the wasted capacity of the team.

To make the most of this opportunity, the Agile practices at Sprint level have to be well implemented. Additionally, the team members should have full freedom in selecting any tasks from the Sprint backlog that they want, even if this results in a much longer duration of a given task, since team members have first to learn new skills. Undertaking actions to expand the area of specialization of each team member can also be treated as a separate response to such a risk.

5.1.3. Insufficient knowledge/understanding of Agile methodology

In each of the projects investigated, an enabling plan was in place, which consisted of training and support from an Agile coach. This enabling was normally provided during the first two months of the project. However, two of the projects did not follow this pattern. In one of the projects, the enabling was delivered with a delay, and in the other, only a reduced scope of enabling was provided to the team – both of these project teams suffered from severe problems because some Agile practices were misinterpreted. The problems observed include conflicts within a team due to different interpretations of the methodology, not following the methodology and falling into a chaotic way of working, low morale in the project team. It is worth mentioning that it has been noticed that some individuals have an aversion to Agile methodologies even before they gain proper knowledge and understanding of this approach – in such cases, the attitude of an individual could be a major impediment in the learning process. It has been observed that even a single team member, who is not sufficiently educated in Agile methodologies, might cause serious disturbance to the whole team, especially if this is a person with a strong position in the team (e.g. due to technical competence and experience) and has a strongly skeptical attitude towards the Agile approach.

These examples illustrate the risks of a lack of knowledge and deep understanding of Agile methodologies by the project team. Responses to such risks include educational activities and support from an experienced Agile coach – a person who can explain the meaning of and reasons for various Agile practices to the team as they are searching for possible improvements to their ways of working and can also provide the team with good advice on how other teams implement the Agile process. Even if a project team receives good training in the Agile methodologies, it has to be kept in mind that the progress of a project will be strongly affected during the first iterations, since the team is discovering its own way of working and adapting Agile practices to their project. If training and expert support cannot be provided, then, as the projects investigated in this research show, introduction of Agile methodology may not bring

any benefits to the project or can even be harmful by introducing chaos, conflict and low morale into the project team.

5.2. Contract risks

Time and material contracts are considered to be optimal for Agile projects. In the case of the projects investigated, all of them were fixed price contracts. There were two primary reasons for this: the customers preferred this type of contract but also the vendor company had limited experience with selling time and material projects. A fixed price contract is an impediment in becoming fully Agile, since it is safer for the vendor to defend the initial assumptions of a project (in order to stay within the project constraints) and also a formal change in the control system is required, even though the actual software development is done according to Agile methodology. With a time and material or cost-based contract, the customer would gain additional benefits from Agile methodology, since the role of Product Owner could be assigned to a customer's representative and in this way the customer would gain the possibility of changing the direction/adjusting the scope of the project at any point of time. In a fixed-price project, assigning the role of Product Owner has to be done within the vendor company, otherwise there is a very high risk of scope creep. To reduce the risks which have their source in a fixed-price contract, it is recommended to introduce flexibility in the scope of a project by making appropriate adaptations to the contractual terms (freezing the definition of the scope, especially at a detailed level, should be avoided) or breaking down the contract into several smaller ones (e.g. ordering groups of product features, instead of a single order for the whole product).

5.3. Program interface risks

Based on the projects investigated, it can be said that interactions between the vendor company and the customer are a major source of risks in the case of Agile projects. In each of the projects investigated, the customer was a company of comparable size to the vendor but each customer behaved differently. According to Agile methodologies, close cooperation with the customer is required at all stages of projects. However, they do not explain how to assure that it is at the right level. Such risks to a project, depending on the approach of the customer to the project, are mentioned below:

5.3.1. Customer interacts in a “traditional” way

Some of the customers treated an Agile project in the same way as any other projects. Such a customer was mostly active at the beginning of the project, when the contract is signed and requirements are specified, and at the end of the project, when

the acceptance of the project deliverables is made. The customer had little or no interest in the middle period of the project. The project team at the vendor's site had to make assumptions during the project, instead of contacting the customer directly and using the feedback from him – this is subject to similar risks as in the case of the waterfall model, i.e. if the assumptions are wrong, then the resulting product will not meet the customer's expectations. Lack of real customer involvement during the iterations introduces overheads into the project (e.g. the necessity of translating the requirement specification document into the Product Backlog) and disables the biggest advantages of Agile methodologies.

To mitigate such risk, the customer should be informed in advance about the desired level of interaction with him, so that work on the customer's side could be organized to meet the expectations of the vendor. Points of interaction during a Sprint, together with the required presence of project stakeholders, might also be explicitly mentioned in the project contract.

5.3.2. Customer not able to provide valuable feedback in time

Because of the size and complexity of the customer organization or limited interest in the middle period of the project, the customer organization might be unable to provide valuable and complete feedback to the project team, e.g. responses to inquiries about requirements and their priorities, feedback about the increments shipped to the customer or demonstrated at the end of a given Sprint. Feedback from the customer or responses to inquiries might not be delivered in time to the vendor. The project team works in very short Sprints and prior to the start of the next Sprint, a significant portion of the requirements might need to be clarified, so that the development team can plan their next Sprint. However, in the case where a large number of stakeholders are involved in projects on the customer's side, gathering the details about these requirements and their priorities might require more time than it takes a well-functioning development team to implement these requirements.

To mitigate this risk, the customer needs to be notified in advance about the required level of communication, so that a communication plan can be established within the customer organization. Furthermore, a single person should be identified on the customer's side to serve as a contact person for the clarification of requirements – this should be someone capable of discussing the requirements on a technical level.

6. Summary and conclusions

Based on a series of interviews with project team members from 6 Agile projects, who also had a lot of experience in the waterfall model, it was possible to identify a list of risks that are either directly caused by the Agile methodology or require more

attention when this methodology is used. These risks include threats to the project, as well as opportunities, extrinsic and intrinsic risks. Dealing with some of these risks is considered critical to project success but none of them are appropriately managed by the Agile methodology itself. Therefore, it can be concluded that the implicit risk management, which is built into the Agile methodologies, is not sufficient and explicit risk management processes should also be applied to an Agile project.

A significant portion of the identified risks are related to the introduction of the Agile methodology as a new process for software development in an organization, or to the imperfect implementation of Agile methodology. While such risks are most likely to occur in an organization, which does not have a lot of experience in Agile development, they are still valid for any Agile project. The Agile methodologies are empirical, which means that only a generic framework of the methodology is codified and the methodology actually used differs from team to team and from project to project. Therefore, every time a new project starts, a new project team is established, the team has to go through the same process of discovering the methodology – during this period all of the risks identified are likely to be present. Also, during a project the team and the methodology might be destabilized (in the projects investigated this was usually connected to changes in the project team composition), so these risks are present during the whole duration of the project.

The results described in this article cannot be generalized, because all of the projects investigated were carried out in the same company and all of the projects involved customization of the same product. The research proves that, in this concrete company and with this type of project, there were project risks which were not mitigated by the Agile methodology. The research was done based on projects carried out in different countries, the various customers and project teams came from various cultures but further research is required based on a larger number of projects, carried out in various companies and fields.

References

- [1] BECK K., ANDERES C., *Extreme Programming Explained: Embrace Change*, Addison-Wesley, Boston 2005.
- [2] COYLE S., CONBOY K., *A Case Study of Risk Management in Agile Systems Development*, 17th European Conference on Information Systems, Verona, June 8–10, 2009.
- [3] DEMARCO T., LISTER T., *Waltzing with Bears: Managing Risk on Software Projects*, Dorset House, New York 2003.
- [4] HIGHSMITH J., *Agile Project Management: Creating Innovative Products*, 2nd Ed., Addison-Wesley, Boston 2009.
- [5] HIJAZI H., KHDOUR T., ALARABEYYAT A., *A Review of Risk Management in Different Software Development Methodologies*, International Journal of Computer Applications, 2012, 45 (7), 8–12.

- [6] KENDALL R., POST D., CARVER J., HENDERSON D.B., FISHER D., *A Proposed Taxonomy for Software Development Risks for High-Performance Computing (HPC) Scientific/Engineering Applications*, Technical Notes CMU/SEI-2006-TN-039, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania, 2007.
- [7] NELSON C.R., TARAN G., HINOJOSA L.L., *Explicit Risk Management in Agile Process*, [in:] *Agile Processes in Software Engineering and Extreme Programming*, P. Abrahamsson, R. Baskerville, K. Conboy, B. Fitzgerald, L. Morgan, X. Wang (Eds.), Springer-Verlag, Berlin 2008, 190–201.
- [8] NYFJORD J., KAJKO-MATSSON M., *Commonalities in Risk Management and Agile Process Models*, Proceedings of IEEE International Conference on Software Engineering Applications, IEEE Computer Society Press, Los Alamitos, California, 2007.
- [9] RIBEIRO L., CRISTINE G., FEIJÓ W., BEZERRA V., *A Case Study for the Implementation of an Agile Risk Management Process in Multiple Projects Environments*, PICMET 2009 Proceedings, August 2–6, Portland, Oregon, USA.
- [10] SCHWABER K., BEEDLE M., *Agile Software Development with Scrum*, Prentice Hall, Upper Saddle River, New Jersey, 2002.
- [11] SMITH P.G., PICHLER R., *Agile Risks, Agile Rewards*, *Software Development*, 2005, 13 (4), 50–53.
- [12] WALCZAK W., *Risk Management in Agile Project Management Methodologies*, [in:] *Modeling and Risk Preferences*, T. Trzaskalik (Ed.), Wydawnictwo Uniwersytetu Ekonomicznego w Katowicach, Katowice 2010, 241–256 (in Polish).
- [13] ISO 31000:2009, *Risk management. Principles and guidelines*, International Organization for Standardization.
- [14] Manifesto for Agile Software Development, <Agilemanifesto.org> accessed February 2013.

Received 11 May 2013

Accepted 18 December 2013