International Conference on Applied Internet and Information Technologies, 2016

DOI:10.20544/AIIT2016.05

Agile Management of IT Projects by Modified Scrum Processes

Zoran Ćirić[1], Stojan Ivanišević[2], Otilija Sedlak[1], Ivana Ćirić[2] and Bogdan Laban[3]

[1]University of Novi Sad, Faculty of Economics Subotica,
24000 Subotica, Segedinski put 9-11, Serbia,
{zoran.ciric, otilijas}@ef.uns.ac.rs
[2]PhD student, Faculty of Economics Subotica, Subotica, Serbia
stojanivanisevic@yahoo.com
civana87@hotmail.com
[3]EPS Distribucija, Beograd, Serbia
bogdan.laban@epsdistribucija.rs

Abstract Scrum is the result of evolution in product development originating from the Japanese management system and process improvement (kaizen) and the Toyota production system from middle of the 20th century. However, Scrum is not the end result but a constant evolution. The purpose of this paper is an attempt to add new knowledge to this evolution. There are numerous alternatives to Scrum process and rules, commonly called hybrids. It is important to be aware that one approach or process cannot be a solution for every software development project. The key is to find a balance between the processes and the people, policies and the principles, as well as focus on long-term or short-term goals, etc. This problem is one of the biggest challenges widely discussed in present-day scientific and industrial communities.

Keywords: project management, agile, Scrum, ROI.

1. Introduction

The area of information and communication technologies is extremely large. IT projects cover wide array of projects in the field of hardware (purchase of computers and / or computer components, installing computer networks, etc.) and software (design and / or implementation of software applications, information systems, databases, web portals, etc.), as well as projects that include a combination of hardware and software. However, because of their frequency, their sensitivity to a variety of changes, their specificity and their creativity, this paper will focus on IT software projects.

The emergence of software development has a very short history - only about half a century long. Despite its short history, the number of approaches, activities, techniques and methods which are used in software development has increased drastically. From the holistic point of view, any practice of software product development is so variable, diverse and complex that any future project management is under pressure and at risk of not being successful. Being successful for a project means to carry out the planned framework, with the specific resources and within the given timeframe. A large number of failed projects points to the lack of real development methodologies. Software crisis of the seventies is a particularly well known result of a complete lack of methodology. However if we look at the meaning of the word crisis (Greek Crisis - make a judgment, evaluation, make a difficult decision) we can say that the choice of the right approach to software development has always been and will remain for a long time in the crisis.

Term ‘methodology’ is mentioned less often and is replaced by the term ‘approach’ in contemporary literature. Concept of an activity in a specific process is being replaced by an activity in a particular framework. The reason for this change should be seen in the fact that the complex challenges are best answered by complex approaches. In other words, one rather simple model can not present response to a complex and turbulent reality. Precisely because of the constant change the emphasis is on the flexibility and the adaptability of the approach, which is often associated with agility and agile methodologies (approach).

In some communities Scrum has a dominant share in the total use of agile methodologies. Scrum is also recognized as the most faithful manifestation of the agile approach. Within the range of discussions and different opinions on this particular topic it is difficult to distinguish between the fans and the fanatics. On the other hand, there are many critics who are disregarded due to mass popularity and application of the Scrum process. The theme of this paper is an attempt to analyze the agility and work experience in order to set a Scrum alternative to such strictly defined framework.

Changes include:

- Situational approach based on the principles of project management rather than a process based on predefined rules,
- Quasi-indefinite duration "time-boxes" instead of explicitly defined time duration,
- Elimination of Scrum master as a separate role in the Scrum team,
- Possible shift from incremental to the continual development,
- Not giving higher priority to development activities based solely on short-term benefits.

2. Project Management

Traditional project management is based on managing people [10]. It includes a variety of techniques for planning, forecasting and control of activities in order to achieve the desired results according to pre-defined rules strictly within a certain time frame and with certain budget [1]. According to PMI, the traditional approach is taken as the basis for the division of the stages of management [8]:

1. Initiating - definition and approval of the project;
2. Planning - the action plan to achieve the goal;
3. Executing - coordinate human and other resources in order to implement the defined project plan;
4. Monitoring and Controlling - monitoring and controlling the progress of the project to detect deviations from the plan and take the necessary corrective action;
5. Closing - acceptance of products, services, or results and completion of the project.

The first task which is placed in front of the project manager and the project team is to define the tasks. The definition of the tasks is needed to be done even before the formal start of the project in some circumstances.

When using traditional approach the plan is the foundation of everything because it represents not only the description of the tasks and the time needed for their implementation, but also the tool for decision making. During the planning phase tasks are determined and assigned to the most competent team members. Planning reduces uncertainty because it provides the ability to correct erroneous steps in achieving the desired goal and it increases the understanding of the objectives and tasks in the project. Planning therefore increases the effect of taking into account the possible distribution of work in relation to the timetable and the availability of resources.

The next step, executing, is basically permission to perform assigned tasks. During the execution it is essential to monitor and control whether everything is going according to plan. Time and resource loss should be minimized by the quick correction of the identified irregularities in order to complete the project on time and within the planned budget. In addition to time and cost successfully completed task is a perquisite to successful project completion.

In the final phase the project has to be formally completed and closed. This indicates that all the project`s scheduled tasks have been successfully completed. The finished project is submitted to the client and used resources are now free to be assigned to a new project. During the closure of the project, evaluation is carried out. In certain situations evaluation can serve as a valuable source of knowledge, especially if something unexpected has happened that is then successfully overcome. This designates completion of the project as an important phase which is often overlooked, mostly because of the pressure to begin work on the next project immediately.

Although the traditional approach to project management is very robust it can be applied to the simplest as well as the most complex projects, because the same steps are always applied during the project.

Striving for continuous innovation and the struggle to reduce costs is present in all branches of industry. This has prompted a group of authors to develop a new approach to software development. Although considered new this approach is based on several already-known principles: continuous innovation, adapting products, reducing delivery times, adjustment of people and processes, reliable results [4]. All of these principles are included in the concept of agility which represents attitude rather than process. Agility is the ability to create and respond to change in order to obtain profit in a turbulent business environment [4]. In addition, agility is the ability to balance between flexibility and stability.

To address these issues in the year 2001st. the Manifesto for Agile Software Development [6] was created containing four most prized values of agile development:

- the members of the development team and the relationships between them must be in higher regard then the processes and tools,
- software development is more important than the extensive documentation,
- cooperation and constant communication with customers is placed ahead of the long, slow negotiations and writing contracts,
- prompt and effective response to changes in many ways are more useful than rigid and inflexible respect for the set plan.

Agile project management is based on the traditional approach and it has five phases, but these phases are modified to make them more suitable for use in practice [4]:

- Envision - determine the vision of the project and the project organization (instead of the initial phase in order to emphasize the importance of vision);
- Speculate - develop a model which is determined by the characteristics, functionality, timelines, iteration plan for achieving the vision (instead of planning, to emphasize the uncertainty because a plan is associated with determination);
- Explore - tested to deliver parts for short periods and constantly look for ways to reduce the risk and uncertainty of the project (rather than management, because research iteratively or non-linear);
- Adapt - inspect the delivered results, the current situation and the behavior of the project team and all that is necessary to adjust adequately;
- Close - complete the project and highlight the key things that the team learned during the work on the project.

3. Agile Principles and Scrum Rules

Agile management stresses iterative approach to the project and is therefore applicable to a variety of projects. If a very large project team is involved, one can expect additional coordination. According to the author of this paper it is better to apply agile project management methodology to the projects with small or medium sized teams in order to achieve the best results and the highest level of productivity while preserving stability.

Scrum (a term taken from rugby, marks the moment when the opposing teams are grouped together fighting for the possession of the ball; related to agile software project management) as a development approach includes several important features of the environment and technical characteristics that have a significant ability to change during the course of the project. Emphasis is placed on the organization and motivation of the project team. Life cycle stages are pre game, game development and operations (Figure 1).

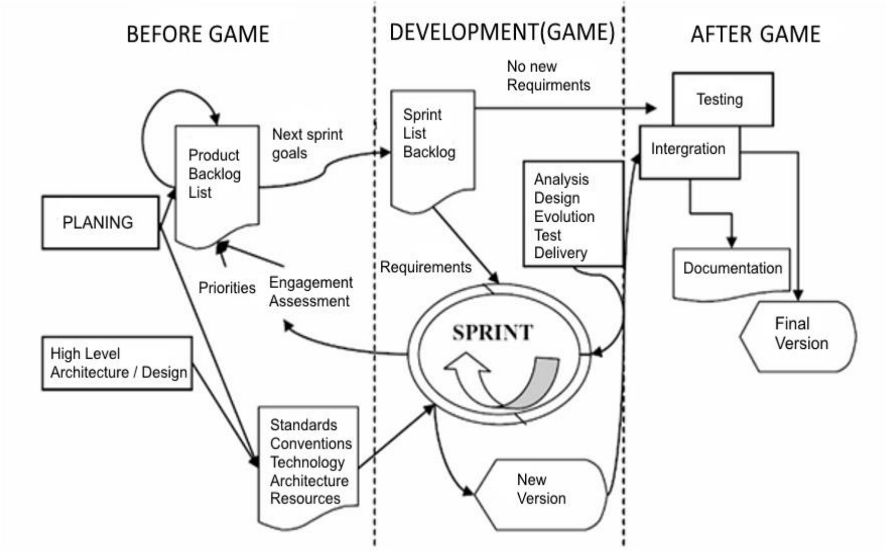

Fig. 1. Scrum life cycle stages and corresponding activities

It is expected for Scrum dogma to put an equal sign between the two terms from the title. However, I believe that the first thought when reading the titles are about the similarities and the differences in approaches between the agile approach and Scrum. This will be addressed later in the article, but the key to this kind of guidance is actually the relationship between the concept of principles and the concept of rules. We can safely say that the principle is more of an abstraction and is more general in relation to the rule. The principle is more often observed through one or more rules than vice versa. The rules are derived from principles. The most important characteristic of all principles comes from the inner mental state, state of mind and motivation of the individual to do the right thing, within externally imposed rules. It is important to note that because of the subjectivity of the principles there may be conflicts between different interpretations of the principles. This conflict does not necessarily mean that the result will be something bad. While translating the principles to the rules and vice versa contradictions may occur, such as those with agile principles and Scrum rules means

Scrum follows strict rules, such as the duration of the event (meeting) which are: Sprint Planning (8 hours for a one-month sprint), daily Scrum (15 minutes), the audit sprint (4 hours for a one-month Sprint) Sprint Retrospective (3 hours for a one-month sprint). Take for instance the daily Scrum meeting, which is the "15-minute restricted event for developers to synchronize activities and a plan for the next 24 hours" [9]. This can be seen as rule concretization of the twelfth principle, the last in the list of principles on which the agile manifesto is based: "At regular intervals, the team reflects on how to become more effective and then adapt their behavior accordingly" [2]. But on the basis of which principles or scientific research has it been determined that the meeting should last 15 minutes each day? Why not 10, 14, 16 or 20 minutes? Can it be expected that a team of three members and a team of nine members would need the same time to get snchronized, not to mention different situations and circumstances. Why every day? If a team is having a free and open communication, and if every team member is aware what every other member is doing daily meetings are not necessary. In these situations time is wasted contrary to the principle of lean management - one of the root principles in agile approach. There has been some research, but also anyone can do an experiment to determine how much time it takes to regain full concentration on some intellectual work after being interrupted. That period is about 20 minutes. This yields potential total loss of 140 minutes each day for a team of 4 members: 4 members * (15 min + 20 min for return of focus on work). Is it not more rational use of time to initiate this type of meetings only if a member of the Scrum team believes that it is necessary and to have them for a necessary duration rather than a limited time period. This is also true for other events. The role of the team leader would be to monitor the "breathing" of the team and make sure that every event takes place in a rational timeframe. In "The Scrum Guide" it would make a significant difference if the word “around” was placed before duration time for the meetings. This transition to the vague time span relies on the belief that every team member is a professional who strives for most effective realization of the mutual goal. Only one of the principles of agile manifesto refers to this: "Build projects around motivated individuals. Provide them with the environment and the support that they need and trust them that they will do the job "’[2].

With this precondition met, I give priority to the flexible and adaptive situational approach over the processes and to the principles over the rules. If you look at the quality of food at fast food restaurants you can expect the same average quality of the finished product each time. This equals a standard output as the result of processes and rules which clearly defines food (input) and processing of these foods. This organization of labor is imposed on the grounds that the workers who prepare the food are uneducated and incompetent. On the other hand if you take real master chefs for example, you can see that they do not use rules (recipes), but their intuition and the principle to, based on the available ingredients selected on the basis of an assessment of freshness and quality, prepare uniquely exquisite dish.

A question which definitely arises is what is the expected quality of the software product we aim to build and how capable is the team we have to build it. Unfortunately, reality shows that not everyone is a master of the craft and can not be expected to apply all of the principles in order to perform the most effective and most efficient action. For each individual project the team needs to carefully decide what will be the trade-off between the degrees of implementation of the principles and enforcement of the rules. Rules and processes are meant to solve standard problems, while principles and chaotic process create flexibility.

Table 1. The table below represents the different characteristics of the two approaches which are the subject of this part of the paper in the project management

	Rules	Principles
Problem and solution:	Standard	Non Standard
Productivity:	Average and constant	High and volatile
Creativity:	Low	High
Adaptability to changes:	Rigid	Rapid

Minimal team competence:	Low	High

4. Scrum Team Role in Development of Software Products

Scrum teams consist of the owner of the product (product owner), the developers (development team) and the Scrum master. As can be seen, one of the Scrum ideas is to eliminate the conventional role of the manager in the project. Most importantly the direct influences of all stakeholders on the development team are eliminated. Stakeholders can give their input exclusively through the owner of the product. This prevents unnecessary conflicts and confusion. But then why should anyone create conflict and confusion in the context of the Scrum team? This comes from the relationship between the product owner and the Scrum master. On the one hand, the interest of the product owner is exclusively the result of the product development without any empathy for the team, while the Scrum master focuses on the development process, time and the team. It turns out that the result of the project depends on the negotiation skills of the two sides, where we run a completely unnecessary risk of dominance of one or the other party. It is also difficult to see that the Scrum master actually cares for the team and its productivity, because Scrum master acts like a cop enforcing Scrum process like a law and regulates any deviation. The problem with the "law" is that the law is not always in favor of a better functioning within the team and team results, as discussed in the previous topic.

Responsibility of a classical manager is not only to set or deliver requirements to the team but also to be aware of how the team „breathes” and to have expectations in accordance to realistic team abilities. This responsibility greatly resembles the role of coaches in the sport teams. This is precisely the role that each team requires. Even a team assembled from the best players in the world requires a coach who sets demands and cares for execution at the same time. Separating these two roles is wrong. In addition there is no direct control in Scrum. Scrum master does not have the role of the leader and the coach that he is credited with but the one of a shepherd who directs the team like sheep. On this basis, Scrum master as separatist separate role in the team should be abolished. This opens the possibility that the owner of the product or someone competent and respected within the development team can be elected leader of the team, unless he comes from the client’s side. The only task left to the team management is to set the context and identify and bind the team to achieve the goal of the project [7].

Regarding the development process, agile approach as well as time and Scrum are iterative and incremental development processes. That means that the difference between the two is visible as a result of the development of usable and ready for implementation products, as opposed to the traditional methodology. But one can go a step further. We need to recall that agile approach has its roots in the paradigm of lean management and the essence of this approach in the area of production is reflected in the philosophy of production at the right time (eng. Just in Time - JIT). One of the methods for the implementation of this strategy is kanban. It is known that this concept is already being applied in the IT sphere under the same name, but only in cases where the purpose is the provision of services, such as technical assistance, or possibly in a late stage of the product life cycle. The idea is to start using the concept of JIT in combination with iterative development which enables the fastest possible time use but use it only when there are conditions for it. The main condition is that particular requirements, user stories or functionality, can be implemented separately from other requirements. User stories and functionalities can be implemented without the need to wait for the end of an iteration, or a sprint [3]. On the other hand if there are more such requirements with the highest priority for development, it makes no sense to put them in a time-limited iteration and one increment. Going into the estimation, analysis, development and the implementation of "one by one" system the way they are arranged in order of priority in product Backlog is a far better approach.

5. Return of the Investment in Software Product Development

The great advantage of the processes and the rules is that they allow the prediction of the input (resources), the output, the time, the risk and the like, for each instance of iteration. Stakeholders and managers are reluctant to use Scrum precisely for this reason. Iterative and incremental development is a big shift in relation to the waterfall model. Specific problems of software development can occur if the development is viewed solely in the short term. More specifically investors should be aware that investing in a project to develop software is a long term investment [5].

The main metric used to determine the whole purpose of the project is the return on investment - ROI. Responsibility for maximizing ROI falls upon product owner through the requirement schedule, user stories or features in Product Backlog. According to Scrum, scheduling is performed based on the value that each item contributes. Value comes down to the formula of benefit / development costs for each item. By viewing things this way, development is focused on short-term ROI. For example if the goal and the greatest value of building a house is the protection from rainfall, it would mean that you should start building it from the roof down. In addition, ROI depends largely on the context. Priority should not be accompanied by the current ROI. Priority of constructing the roof is much higher in the early fall than in the late spring because of the much higher incidence of rainfall. Therefore it is necessary to analyze and compare the items in Product Backlog in a manner which will take into account the current situation (context) and the expected ROI at the end of the future sprint on the one hand and the future ROI of all future sprints on the other hand. Future sprints ROI will depend on current decisions about which item in Product Backlog will be developed.

Calculating ROI based on future sprints:

$$ROI=ROI(n)+(ROI(n+1)*\alpha+ROI(n+2)*\beta+\cdots+ROI(n+i)*\omega) \qquad (1)$$

Where:

n – Represents number of the sprint for which we are calculating ROI,

i – Last sprint in a row which brings added value related to the n sprint,

Greek alphabet letters – coefficient of shared ROI from future sprints as a result of investing into n sprint.

This formula is the way to calculate the total potential investment return based on the investment in Sprint n. Unlike the current short-term perception that the investment in a sprint delivers ROI equal to ROI at the end of the sprint, the long term ROI can be calculated as the sum of ROI at the end of the current sprint and the amount of shares in ROI of future sprints which are the result of investment in the current sprint. Coefficient share is represented by the symbols of the Greek alphabet and is defined on the basis of analysis and projection of future products and future market trends and business needs.

It is important to keep in mind that there is always a suspicion about the total, short-term and long-term, return of investment. This suspicion is crucial for decision what needs to be developed in the next iteration. If we look at investing as a business activity, we see that it depends on the size of the investment, the size of return, time and risk. It is necessary to keep abreast of all these factors when deciding on future investments. The time dimension of investment in terms of immobilization of the means until their return and the risks are the most common reasons to favor investments in short-term results. On the other hand if the rate of return on investment over the long term is much higher than the short-term, this justifies the opportunity cost and the risk. Because of this higher return rate the focus should be on long-term investment. The conclusion is that the layout of user stories in Product Backlog should be determined by taking into account all relevant variables.

6. Conclusion

Many aspects of software product development process can be found through reviewing the Scrum and its characteristics. Due to the existence of different perspectives, deciding what is best or at least good enough is not easy. This applies to software development and management as well as to each member of the development team all the way to the highest managers. By covering the different roles and responsibilities of the Scrum, this article suggests improvements in some aspects of work. The aspects that were highlighted are related to managers via the planning, organization and project management, through to the team members role and particularly to product owners who need to prioritize development and to Scrum master whose responsibilities should be divided into pre-existing roles in the team.

References

1. Bach, J., & Pressman, R. Enough about process: What we need are heroes. IEEE Software, 12(2), 96–98. http://doi.org/10.1109/52.368273 (1995).
2. Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Thomas, D. Agile Manifesto. Retrieved from http://agilemanifesto.org/ (2001).
3. Gray, L. Gray rebuts Bach: no cowboy programmers! Computer, 31(4), 102–103,105. http://doi.org/10.1109/2.666846 (1998).
4. Highsmith J.: Agile Project Management, Boston, MA: Addison-Wesley, 2004.
5. Lukasiewicz, K., Miler, J. Improving agility and discipline of software development with the Scrum and CMMI, IET Software, Volume 6, Issue 5, pages 416-422, DOI: 10.1049 (2012). (accessed 10.03.2016.)
6. Manifesto for Agile Software Development , http://www.agilemanifesto.org (accessed 03.03. 2016.)
7. Permana, Putu Adi Guna, Scrum Method Implementation in a Software Development Project Management, International Journal of Advanced Computer Science and Applications, Volume 6, Issue 9, pages 198-204 (2015).
8. PMI: A Guide to the Project Management Body of Knowledge, Third Edition (PMBOK Guide), Newtown Square, PE: Project Management Institute, 2004.
9. Schwaber, K., & Sutherland, J. The Scrum Guide - The Definitive Guide to Scrum: The Rules of the Game. Scrum. Org, October, 2(October), 17. http://doi.org/10.1053 (2011). (accessed 10.03. 2016.)
10. Wysocki K. R., McGary R.: Effective Project Management, Third Edition, Indianapolis, In: John Wiley & Sons, Inc, 2003