

## USAGE OF SCRUM METHODOLOGY FOR IMPLEMENTING SOFTWARE PROJECT

Bernard Ciurariu

Technical University Gheorghe Asachi from Iasi, Romania  
ciurariub@yahoo.ca

Costache Risu

Technical University Gheorghe Asachi from Iasi, Romania  
rusucostache@yahoo.com

### **Abstract:**

Purpose: It analyzes the efficiency of the Scrum methodology for the management of software applications. The model used: It is based on the specific literature, the experience gained in implementing IT projects, case studies, and on existing certifications for this methodology. The Results: Highlighted the positive and negative aspect in software project management using Scrum method. Future research: Analysis of other methodologies used in software domain or other industries that lead to improving the methods used to implement IT projects. Practical implications: The paper offers indications about how this method is applied in the software industry. The novelty of this work: Consists in a practical analysis, from which it was detected what type of projects could use this method for management, and to draw some conclusions on how to improve the application of this method.

*Keywords: Scrum, Agile, Software, Management, Development.*

# 1. THE THEORETICAL BACKGROUND

## 1.1. Agile Methodology

In 2001, from the desire to improve the traditional method of software development, Agile Manifesto is written by 17 authors (Kent, et al., 2001):

“We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

- **Individuals and interactions** over processes and tools
- **Working software** over comprehensive documentation
- **Customer collaboration** over contract negotiation
- **Responding to change** over following a plan

That is, while there is value in the items on the right, we value the items on the left more”.

After long discussions, the signers of the document, realized that it should be a better solution to manage the implementation of software projects than the traditional one, and they come up with that manifesto where they described four main values which are important even today.

## 1.2. Scrum method

This methodology is an agile method, that involves an iterative and incremental process using sprints that lasts between 2 weeks and a month. This method can be applied to the projects that do not have specifications well defined and suffer many modifications in the development process. At the end of each sprint, it is performed a retrospective of what was implemented during the sprint and it is prepared the new sprint. Scrum allows a better communication and cooperation inside the team and also a good visibility of project status.

Scrum methodology has three important roles (James, 2015):

1. **Product Owner** – is a person with vision, authority and availability. He is responsible for communication with the development team, manage the backlog and prioritize the tasks. He must to respond and clarify all questions that come from development team.
2. **ScrumMaster** – acts as a bridge between Product Owner and team. His role is not only to manage the team but to remove any impediment that could hinder the team from achievement the sprint objectives. Scrum Master works closely with the Product Owner presenting the progress of the sprint.
3. **Team** – is responsible for its own organization in order to achieve the desired result. A development team can contain three, up to 9 members. Ideally, the team is located in the same room, protected from possible distractions from the outside, where developers can freely talk about different topics of the project. The team usually includes: programmers, software architects, analysts, testers and graphic designers. The team has autonomy and responsibility to achieve the objectives of the sprint.

The usual steps used by agile methodology are described very well by Kelly Waters (Waters, How To Implement Scrum in 10 Easy Steps, 2007) and (Waters, 10 Key Principles of Agile development, 2007):

- Create the backlog list with all tasks that have to be implemented.
- This is a very important step because it contains an analyses part that has to be performed. The responsible for creating of this list is the Product Owner, who will assign a priority for each task.
- Backlog estimation.

In this step it is performed an estimation in relative points (Ken & Mike, 2001) for all tasks included in the backlog. This estimation is important because, it allows Product Owner to understand how much effort it is required for each task, to prioritize the tasks from backlog and to settle the size of the development

team required to implement the functionalities. The relative points will represent the complexity of the task and not the time required to be implemented. The estimation will be given by the development team, because the developers will implement the task and they have the experience to know, how much will take to implement a task or other.

#### *Sprint planning. Requirements clarifications.*

Sprint planning is performed in a meeting where take part all team, including all roles on the project (developers, testers, analysts, Product Owner). During the meeting, it is settled the sprint duration, decision that is taken by the team.

#### *Estimation of tasks from sprint.*

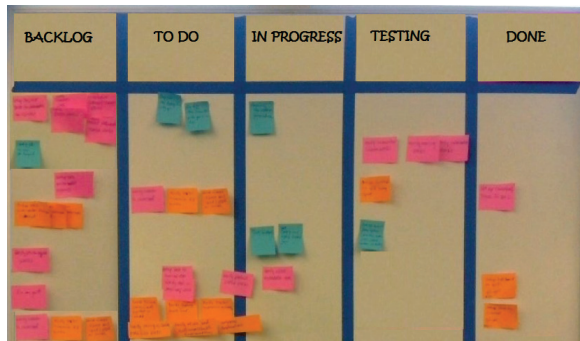
Represents the planning of the sprint in detail. For this, first is defined what features should be implemented, then is taken the list of sprint and divided into smaller tasks for being estimated. As result of the estimates, it is defined the Scope of The Sprint which represents the tasks that can be implemented in the allocated time for sprint.

The tasks that cannot be covered during the current sprint are passed in the backlog for being scheduled for next sprints, or can be kept for being implemented in case the team finish faster, the list of tasks from the sprint.

#### *Create a panel to make visible the status of the sprint.*

The panel contains mainly, the tasks that are arranged according with their status. That panel allows a better visibility of the sprint for entire team and it is the center of all discussions related with the project, the place where takes place the daily stand-ups, the place where can be found information regarding the project. Usually, the panel is represented by a whiteboard (**Picture 1**), where the team sticks post-its with certain information.

**Picture 1:** The Scrum methodology's panel



#### *Implement the sprint.*

In this phase, the team has to implement the assumed tasks in the preceding steps. For this, the team must be empowered with the responsibility of implementing the sprint. Thus, it is very important that no one to make decisions over the development team, otherwise it is possible the team to lose interest in obtaining the objectives of the sprint.

In the meantime, the Scrum Master must show support for the team, to guide and assist the team, when the team asks all these. In case, it is noticed that some features are not possible to be implemented during the sprint, they can be moved in backlog for being prioritize for next sprints, but in case the team finishes the sprint faster, it is possible to include other tasks in the sprint.

### *Daily stand-up*

It is organized daily at a time agreed before, and whole team has to take part. During the reunion, each member of the team has to present what tasks they implemented a day before and what tasks they have in progress and also, if there is any issue for implementing these tasks. The speech should not exceed 1 maximum 2 minutes.

### *Supervising the sprint progress.*

To control the implementation status it is used a tool called Burndown Chart. This is a graph consists of two lines, one representing the natural course of implementation of tasks arising from the estimates and the second represent the tasks that were planned to be finished and completed. By comparing these lines can identify certain delays in initial planning.

This graph is made visible to all, and in this way the whole team is aware of the status in the current sprint.

### *Done means done*

One of the most important principles in agile is "Done means done".

It is very important to respect this principle, because only fully implemented features are delivered to the customer. To get this, each task is tested to verify that what has been implemented is according to customer requirements.

### *Sprint review and retrospective of the sprint.*

At the end of each sprint is important to have a meeting to analyze what happened during the sprint, called "Sprint Review". At this meeting attends the whole team and can be performed a demonstration of the features implemented in the last sprint. This demo allows Product Owner and other persons that are involved in the project, to provide a feedback.

After Sprint Review, it is good to be organized a retrospective meeting to bring out the good things that happened during the sprint and solutions to improve the process where it is needed.

## **2. THE METHOD USED TO STUDY SCRUM**

The study is based mainly on existing literature and the experienced gained in many years of using this method. In addition, we used interviews with experienced managers in using this method for implementing their projects with Scrum, case studies on different projects with more or less success using Scrum as methodology, the presentation given by a company (Institute, n.d.) that organizes certification on this project management method.

## **3. RESULTS**

Analysis was done from two perspectives: studying each individual step required to implement a project with Scrum and a high level analysis.

### **3.1. The analysis over the steps in implementing Scrum methodology, resulted in the following conclusions:**

- Create the backlog list with all tasks that have to be implemented

<b>Pros</b>	<b>Cons</b>
<ul style="list-style-type: none"><li>• It is established and permanent updated, the tasks list that represents the backlog of the project.</li></ul>	<ul style="list-style-type: none"><li>• The backlog is created in haste, after a brief analysis, many features will be implemented</li></ul>

<ul style="list-style-type: none"> <li>Product Owner will assign a priority for each task.</li> </ul>	<p>but will not make part of the final release because the Product Owner will realize in the end that these functions are not suitable for being included in the final version of the product.</p>
---	--

- Backlog estimation.

Pros	Cons
<ul style="list-style-type: none"> <li>Development team gives the estimation of the tasks from the backlog and the Product Owner prioritize the features according with this estimation.</li> </ul>	<ul style="list-style-type: none"> <li>At this stage the team will most likely not fully understand what has to be implemented, and even if they manage to gather this knowledge, it is difficult to give some accurate estimates even in relative points.</li> <li>The backlog estimation takes a long time and whole team take part, even those with less experience that results into a high cost for the project.</li> </ul>

- Sprint planning. Requirements clarifications.

Pros	Cons
<ul style="list-style-type: none"> <li>It is created a list of tasks for being implemented during the sprint. The list is called "Scope of the sprint"</li> <li>Functionalities are implemented in the order given by its priority.</li> <li>During this meeting it is clarified all the things that are unclear.</li> <li>Because the development team will make the estimates, the team has to assume the responsibility for achieving the objectives of the sprint.</li> <li>Once it was settled the sprint length, it should be kept the same length for the rest of the sprints.</li> </ul>	<ul style="list-style-type: none"> <li>The clarifications are provided ad-hoc and are not the result of a deep analysis. As a result the explanations are not complete.</li> <li>The clarifications are not laid down on the paper as specifications, which leaves room for later interpretation.</li> </ul>

- Estimation of tasks from sprint.

Pros	Cons
<ul style="list-style-type: none"> <li>Estimates are made by the team, in this way the team is motivated in achieving the sprint.</li> <li>At this stage, it shows up the technical discussions in order to find solutions for implementing tasks.</li> </ul>	<ul style="list-style-type: none"> <li>At this phase, the team is most likely not to have enough information for providing an accurate estimation.</li> <li>Estimates are given by the entire team regardless of individual experience.</li> </ul>

- Create a panel where to make visible the status of the sprint.

Pros	Cons
<ul style="list-style-type: none"> <li>Shows the sprint progress and the status of each task.</li> </ul>	<ul style="list-style-type: none"> <li>A software tool can be used instead the physical panel.</li> <li>Maintenance of the panel up to date can be sometimes difficult.</li> <li>The physical panel obviously, cannot generate some metric very useful in project management and it is required the usage of additional software tool.</li> </ul>

- Implement the sprint.

Pros	Cons
<ul style="list-style-type: none"> <li>• The perimeter of the sprint is known.</li> <li>• Scrum Master and Product Owner show support to the team.</li> <li>• Functionalities that cannot be achieved during the sprint can be moved back to backlog</li> </ul>	<ul style="list-style-type: none"> <li>• Many times the sprint suffers multiple changes, requiring additional adjustments over the Scope of sprint</li> <li>• The process does not follow a natural logic: analysis, implementation and testing, but it goes directly to implementation phase, which generates numerous changes and a disturbance of the normal process.</li> </ul>

- Daily stand-up.

Pros	Cons
<ul style="list-style-type: none"> <li>• It is a good opportunity for communication and synchronization inside the team.</li> <li>• It is highlighted the tasks in progress, some technical solutions for complex tasks and other things that could show interest for the rest of team.</li> </ul>	<ul style="list-style-type: none"> <li>• Costs related with the participation of the whole team.</li> <li>• Although it is estimated at 15 minutes, the actual time is much higher and it is proportional with the size of the team, and some of the team members need to interrupt their work to attend this meeting.</li> <li>• Although, it should be just a synchronization within the team, this meeting often turns into opportunity for other discussion topics.</li> </ul>

- Supervising the sprint progress.

Pros	Cons
<ul style="list-style-type: none"> <li>• It can be done by tracking the situation from Scum panel or from Daily Scrum.</li> <li>• It is used Burndown Chart.</li> </ul>	<ul style="list-style-type: none"> <li>• Cannot rely on the panel because the panel may not be up to date.</li> <li>• There are other software tools that provide many complex reports for understanding the development progress and can replace easily the panel or Burndown Chat used in Scrum</li> </ul>

- Done means done.

Pros	Cons
<ul style="list-style-type: none"> <li>• It is an important principle that ensure sustainability of what is implemented and decreasing the time spent for fixing the bugs.</li> </ul>	<ul style="list-style-type: none"> <li>• This principle actually it is a common sense rule that ensures the quality of the code and it is applied to all methodologies not only in Scrum.</li> </ul>

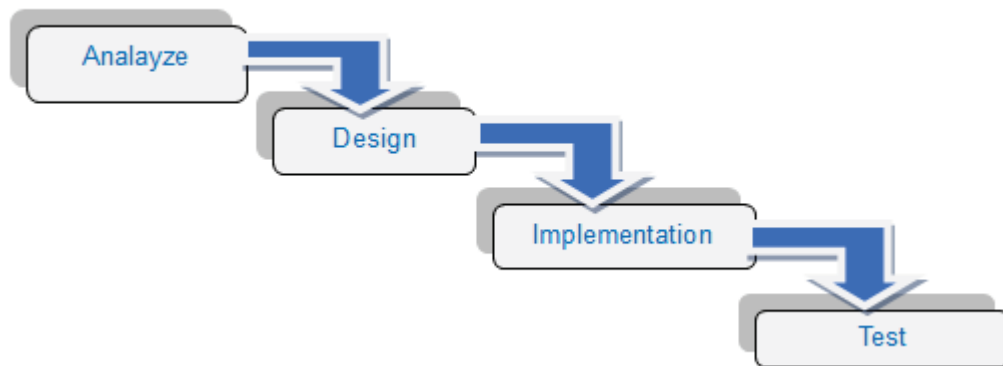
- Sprint review and retrospective of the sprint.

Pros	Cons
<ul style="list-style-type: none"> <li>• It is provided a feedback regarding what was implemented along the sprint.</li> <li>• During the retrospective of the sprint can be found solutions to adjust some things.</li> <li>• A new release can be done for testing the satisfaction of the applicartion users, regarding these new features implemented in the sprint.</li> </ul>	<ul style="list-style-type: none"> <li>• Usually this meeting takes at least half day and whole team participate to this meeting, resulting in a cost on project.</li> <li>• For these meetings could attend only part of the team for instance: Product Owner, Scrum Master and one developer.</li> </ul>

### 3.2. The analysis is done according to the existing methodologies used for implementing software projects.

It is clear that unlike the classical method of implementing software projects (Picture.2), where everything is planned in detail, the agile development relies more on thinking of the moment and vision of those who dictate the course of the project and which attempts certain functionalities that can yield results or not. From this, results a serious of issues that can make from usage of this method a real issue, if the steps are not coordinated properly. The atmosphere seems more like a chaos where the team is struggling to achieve its goal, having with Product Owner meetings over meetings. Lack of analyses before starting implementing the tasks results in very low percentage of features included in the final version of the product. The experience of the team's members plays a very important role in this methodology. The communication inside the time and the capacity of understanding the application are essentials.

**Picture 2:** Clasical method

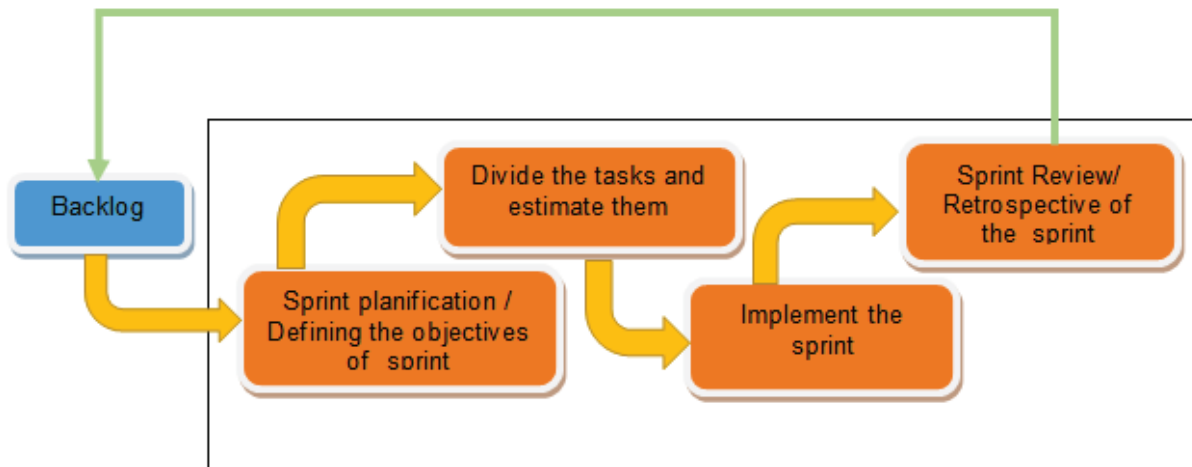


The speed for developing new applications in Scrum (Picture 3), is done against the quality of the code, because it skips the first steps from the traditional method: analysis and design. For those with more experience in development of software products are able to distinguish relative easily, analyzing the written code, whether the application followed Scrum method or the traditional one. This is because the rebate on certain quality principles to reach as soon as possible the end user.

To improve the quality, a good solution is to include the analysis and design steps inside the Scrum process. These steps can be performed by persons with high experience, before the sprint or among first steps from the sprint.

Writing specification documents is other aspect that should take care because it is important to track the changes and to be able to include easily other colleagues on project in case it is required this thing.

**Picture 3:** Scrum process



#### 4. CONCLUSIONS

This article was the result of analyzing many projects with Scrum, as methodology and a very strong willingness to share the experience gathered for many years using this method. I have heard many times that Scrum does not help us in our project, the solution to use Scrum is not the proper one.

In this article, it is shown in the Section 3, the positives and negatives parts of using Scrum methodology and also some solutions to improve this method, because it is very important for a software project to use the right management method. There are many cases where projects using a methodology, after a while, managers decide that it is more suitable for these projects to change the process, including parts from other methodology or even to change the entire process. To avoid such cases it is important to make an analysis before choosing a methodology. There are many books, tutorials, certifications or articles that describes Scrum methodology but it is important to present all practical aspects of this methodology because most of the projects will need some refinements of this method, in case it is chosen Scrum for their management.

Along the Results section, it is presented weak parts of this methodology and solutions that could improve the usage of Scrum. It is shown the need to be included the analysis and design parts inside the process, to have specifications at least for the case when it is necessary to include other persons in the team, to adjust the way the meetings are organized, because usually the team contains specialists with different experiences and capacity to understand in the agile way, the application business.

The Scrum method can be used on some range of software projects, but without an experience in using this method all advantages described by this method can result in disadvantage. A very low percentage of applications that use Scrum, respects the principles defined by the agile manifesto. Among those projects who can afford using pure Scrum are startups and small applications where the rapidness to release periodically new versions is very important. It is obvious that this method with some adjustments can be used on a larger area of software projects.

Novelty is represented by the critical analysis and by the solutions proposed to improve the usage of this method in management of software projects.

After this paper, I would like to write an article regarding methodologies used in other industries that can have applicability in software projects. Since most management methods that are used in implementing IT softwares, are taken from other industries which have proven reliability (e.g. Just in Time from Toyota underlying Kanban), I would like to study the techniques used in project management of other industries



that can bring an improvement in the software industry.

## REFERENCE LIST

1. Blackburn, J. D., Lapre, M. A., & Van Wassenhove, L. N. (2002). *Improving Software Productivity by Managing Complexity*.
2. Institute, S. (n.d.). *WHAT IS SCRUM?* Retrieved from International SCRUM Institute: [http://www.scrum-institute.org/What\\_Is\\_Scrum.php](http://www.scrum-institute.org/What_Is_Scrum.php)
3. James, M. (2015). *Scrum Methodology*. Retrieved from <http://scrummethodology.com/>
4. Ken, S., & Mike, B. (2001). *Agile Software Development with Scrum*. Prentice Hall PTR Upper Saddle River, NJ, USA.
5. Kent, B., Mike, B., Arie, v. B., Alistair, C., Ward, C., Martin, F., . . . Dave, T. (2001). *Agile Manifesto*. Retrieved from Manifesto for Agile Software Development: [www.agilemanifesto.org](http://www.agilemanifesto.org)
6. Kniberg, H., & Mattias, S. (2010). *Kanban and Scrum - making the most of both (Enterprise Software Development)*. InfoQ.
7. Lloyd, W. (2014). *Agile Development: A brief History*. Retrieved from <http://www.provenmethod.com/agile-development-brief-history/>
8. *Waterfall Model*. (2015). Retrieved from Waterfall Model: [www.waterfall-model.com](http://www.waterfall-model.com)
9. Waters, K. (2007). *10 Key Principles of Agile development*. Retrieved from <http://www.allaboutagile.com/category/10-key-principles-of-agile/>
10. Waters, K. (2007). *How To Implement Scrum in 10 Easy Steps*. Retrieved 2015, from All about Agile: <http://www.allaboutagile.com/category/how-to-implement-scrum-in-10-easy-steps/>