INFORMATION
SYSTEMS IN
MANAGEMENT

Information Systems in Management (2018) Vol. 7 (4) 269−285
DOI: 10.22630/ISIM.2018.7.4.24

ENHANCEMENT OF SCRUM-BASED PROJECT MANAGEMENT LEARNING EXPERIENCE BY USING WEB APPLICATION

WALDEMAR SOBIECKI, ANNA KURZYDŁOWSKA

Faculty of Mathematics and Natural Sciences, Cardinal Stefan Wyszyński University in Warsaw

In recent years project management has become a complex and diverse field of scientific research. It embodies a number of specific tools and techniques, choice of which to particular project is far from being obvious, especially for the beginners. Such situation rationalizes efforts aiming at the development of methods of training, which are more efficient than the “trial-and-error” approach. In this context the paper presents a web application that simulates real life scenarios and allows for developing skills, among others, in identifying the needed/available resources against the dynamically changing goals. The platform teaches the agile approach to project management with Scrum and gives an opportunity to “materialize” the progress made by users with an original piece of software. The prototype of the web application has been successfully tested at Cardinal Stefan Wyszynski University.

Key words: Traditional Project Management (TPM), Agile Project Management (APM), Scrum, Project Management Life Cycle (PMLC), Software Development Life Cycle (SDLC), E-learning

1. Introduction

Today’s society is often described as “Information Society”. We are surrounded by unprecedented and ever increasing variety of information and technology. This leads to the question of how to utilize available resources in the most efficient way.

This article gives a glimpse of how to combine theoretical knowledge with implementing real solutions in IT projects. In order to better visualize the idea, author created an application that has two main objectives – to teach and to provide an environment for developing working it solutions. The application was specifically crafted for arguably one of the most popular agile framework called Scrum.

1.1. Definition of a project

There are multiple definitions of a project, some can be found in international standards such as: Projects in Controlled Environments (PRINCE2), Association for Project Management Body of Knowledge (PMBOK) or ISO standards (i.e. ISO 21500:2012 standard). Some definitions have been thoroughly formulated by researches, while some by practitioners. Nevertheless most of them define project as a temporary endeavor undertaken to create a unique product, service, or result.

In addition to this, projects should meet additional criteria and take place under some specific circumstances. The most important criterion is the goal of a project, usually best reflected by the scope of project. Another important feature is the interplay between the needs to be served, the time and cost of implementation, often referenced as a project triangle. On top of that, projects are executed in limited multi-resources environments (people, knowledge, technology). As a result they carry specific risks. According to the Project Management Body of Knowledge Guide all these aspects render projects their temporary nature [22].

1.2. Project management

Over the years, it has turned out that multiple factors influence projects; thus, initial expectations of the customers and the outcome of the projects finally accepted by them differ significantly. Projects can be finished with satisfaction of the customers only by applying some specific rules that make successful completion of the projects possible. The first and foremost important concept in this context is concept of project management.

Depending on circumstances different project management strategies can be used. The goal is to impose processes and procedures that do not limit creativity of the people involved, by providing them with the tools to efficiently meet project requirements.

A person that is responsible for selection of proper tools and techniques that will be used during the project is called a project manager. Since each project is different, the essence of project management is not in the routinely repeated solutions that have worked in the past. The inherent nature of the project implies that project managers need to be adaptive, creative and flexible. At the same time, however, there are some general rules that help them identify the right paths.

PMBOK defines project management as an application of knowledge, skills, tools, and techniques to project activities to meet the project requirements [22]. It involves client to meet sponsor's needs and deliver a new business value. PMBOK marks out five process groups which can serve as checkpoints during project realization:

Initiating - This is a phase when project manager is assigned to the project. His/her task is to evaluate the current situation. He/she learns the business needs that are presented by the sponsor, who represents the customer. As a result a vision of the project is created. More detailed requirements are documented by the project manager, as they will be needed later.

Planning - In this phase a general vision is described in detail (defining a scope of the project) and project plan is created. This plan covers topics such as budget, milestones and risks.

Executing - Project manager executes project according to the plan. In traditional approaches to project management, unforeseen changes that are reported by stakeholders should be first agreed with the central board. When approved, they can be added to the scope of the project.

Monitoring and Controlling - This is actually the only phase that is not linear. It is a perpetual exercise that lasts as long as the project is not finished. It consists of tracking, reviewing and regulating applied to all actions that are taking place in the project.

Closing - It is a time of a formal closure of the project. Before project is officially declared as "done", few things need to take place. First, the outcome should be accepted by the customer (sponsor). It is also a good practice to have some sessions on lessons learned during the project. Such sessions allow to formulate conclusions and make improvements in future.

There are also different approaches to project management life cycle (PMLC). The choice of PMLC should be done based on analysis of project goal vs. solution. The solution describes ways in which the project goal can be achieved. When both the goal and solution have been analyzed, then one of the four approaches to PMLC can be adopted [3].

Traditional Project Management (TPM) - used when both goal and the solution are known. It is a plan-driven approach which means it needs a detailed plan preceded by WBS (Work Breakdown Structure). Traditional Project Management works well for all the well-known project subjects, such as infrastructure projects. This approach should be taken into consideration if there is very little chance of unforeseen situations.

Agile Project Management (APM) - used in the case of the projects that are “change-driven”. Such projects usually have a general goal, but the way how to achieve it is not pre-determined. Thus actual solutions are discovered as the project continues. An immanent attribute of agile projects is that the scope and resources might change over time, which makes it difficult to forecast the end result.

Extreme Project Management (XPM) - Extreme approach suits best projects that are high-paced and with high failure rates. Such projects often have neither known solution nor specified goal. A typical example are Research & Development (R&D) projects.

Emertxe Project Management (MPX) - the name comes from “E-x-t-r-e-m-e” read from right to left. MPX projects are defined as reversed R&D projects. The difference is that in R&D projects although the solution still needs to be figured out, there is usually some desired end state. In Emertxe the goal is not defined. In such projects there are only tools that are available and can be utilized, and it is a major challenge for the project manager to find good use for these tools and create an added value.

1.3. Software Development Process

In projects that are strictly set on developing a software, the concept is very similar and is called a Software Development Life Cycle (SDLC). The difference between them is that PMLC is more suitable for establishing life cycle cognizance first and SDLC is more for adapting development models for delivering the technical content.

All above concepts focus on the common problem: how to minimize the difference between the wants and the needs. The traditional approach is not always sufficient to meet the constantly changing business needs. Requirements are in the center of the client interests and are often redefined. That imposes additional responsibility on project managers as to the proper choice of SDLC model.

There are two main approaches to SDLCs. The first one is called a traditional project management (TPM). It has a long history of successful implementations. It mostly relies on linear and incremental build-up of a software. Some of the most popular systems used in this context have been listed below:

- Waterfall
- V-model
- RUP (Rational Unified Process)

The other group is the Agile (APM) approaches. Despite their diversity, they all share an idea of working in an iterative (incremental) way. Such methodology allows for a quicker response to customers' needs. The Agile methods are relatively new and are gaining on popularity at the expense of the traditional methods. There are numerous implementations of the Agile approach, some of them have been listed below:

- Crystal
- Dynamic Software Development
- Feature Driven Development
- Lean Software Development
- Extreme Programming
- Scrum

2. Scrum

Uncertainty and constant change of business requirements are inseparable parts of IT projects. In order to achieve goals one needs to be flexible and react quickly. Scrum has been used since 1990s for actions such as: developing, enhancing, releasing and sustaining new products [13]. Over the years it has proven over and over again in complex projects being run in an uneasy market conditions. These successes have built its reputation of a flexible and reliable tool for solving complex problems. Nowadays, Scrum is one of the most commonly used agile approaches when delivering innovative products or services [11, p. 34]. Official Scrum guide highlights that it is neither a process nor a definitive method, but rather a process framework, that delivers a structure and allows to employ various processes and techniques [13, p. 3]. It distinguishes three major components (dimensions of the projects), that are: people, artifacts and ceremonies. These three, if combined in the right order, allow for reliable and efficient problem solving.

2.1. People in Scrum

The core of each Scrum endeavor is a team. The teams involved into project implementation were originally divided into two funny sounding groups, based on metaphor of pigs & chickens. However, controversy built around that segmentation has been solved by replacing the original classification in the official Scrum Guide between 2010 and 2011 [14]. Today Scrum teams are defined in a more structured way, i.e. below.

Developers - it is a group of people that are directly involved in creating a final product. Teams should be self-organizing and cross-functional to ensure comprehensive, interdisciplinary approach to the problems arising. Each member of the team should possess a wide range of skills and be flexible enough to replace his/her colleague if necessary (e.g. in case of illness). A team should not exceed nine people, as experience shows that the most effective teams have from three to six members [2, p. 90]. It does not imply that Scrum can only be used for solving "small problems". For more complex projects, there are usually multiple developer teams.

Product Owner - one of the most common needs during project realization is to manage the changes, in particular changes of the business requirements. In order to minimize turbulence, client in Scrum is represented by one person only, who plays the role of a Product Owner and is responsible for negotiating possible project adjustments. Assigning just one person to the role of Product Owner reduces the time required for negotiating requests for changes. Having multiple Product Owners would make the whole process practically impossible to manage. Obviously, the Product Owner might look for advice of his/her colleagues.

Scrum Master - the primary goal for the Master is to help the team remove any obstacles that arise. Scrum Master manages internal relations between members of a team and takes care of organizing meetings and controlling deadlines.

2.2. Artifacts

Artifacts are all the widely understood "tools", that Scrum delivers to people that are engaged in the project. The most crucial ones have been briefly described below:

- tasks - elementary units of work that a developer can produce. Tasks are combined into stories, which in turn, create a final product. In practice, tasks are usually defined by decomposing stories into smaller chunks (method known as "principle from general to specific"). Product owner should take care of setting priorities and updating sets of stories, so that they match changing needs of client.
- story points – points are assigned to each story. They should reflect time needed for the story to be completed.
- scrum board (task board) - stores stories, this is usually a physical board with post-its stuck to it. It should be situated in a central point of office so that each team member has an access to it. The simplest board consists of categories: to do, in progress, completed [17, p. 19].
- product backlog - that is list, that contains all stories. Stories should be prioritized (from the most important to the least ones). The most important

stories should be placed at the top of the board. As the stories go up on a board, they should be explained more in detail (extended) so that they are understood by each of the team developers. Here it is important to define list of conditions that have to be met to declare that the story is "done". This term is used to indicate that the current piece of project as potentially shippable. When the definition of done is met, then story can be moved to completed section on the board [11, p. 422].

- sprint backlog - these are stories chosen from product backlog that will be created during current sprint. The time estimated for the stories should be aligned with the length of the sprint. In other words, stories should be realized during an ongoing sprint (without a need to move them into next one).
- burndown charts - are showing how fast the team is moving forward. X-axis shows time (or next sprints) and y-axis shows number of story points that still have to be completed. Burndown charts are also taking into account abnormal situations or pauses such as employee illnesses or appearance of new stories.

2.3. Ceremonies

Ceremonies are the events that occur during project development. In order to perform these events, teams need to use previously presented artifacts in a very precise order. They combine theory of Scrum with all the tools it delivers into subprocesses that in the end lead to the final product.

Sprint planning - takes part before each sprint [11, p. 359]. Its purpose is to prioritize stories from the product backlog, so that the progress in product development is possibly the highest [2, p. 97]. It is a time when team agrees on what will be built in upcoming sprint. Product Owner does not have to take part in the whole ceremony. However, if he/her wants to point out some additional functionalities/features he/she is allowed to do so. It is worth noting though, that it is up to the developers to decide if and which stories can be successfully realized during the sprint. Once stories are chosen they are usually split into smaller tasks. The outcome of sprint planning should be a sprint board with all the stories nicely divided into tasks.

- daily Scrum - short meeting (up to 15 minutes) of Scrum Master and the developers, there is no need for Product Owner or stakeholders to participate in it. It is a brief communication about actual status and the progress made so far [2, p. 97]. There are three main questions that every participant of this meeting should ask: "What have I done yesterday?", "What am I planning to do Today?", "What are the obstacles that are blocking me?" Role of Scrum Master is to actively eliminate obstacles and solve the issues that developers are facing.

- story time - it is used to upgrade stories in product backlog, also might involve changing criteria of acceptance [17, p. 29]. This sometimes is considered as additional ceremony, but during this time many important subjects can be discussed such as: criteria of acceptance of stories, conditions it has to meet and how different developers understand those rules - usually criteria might differ depending on point of view of each developer. It is good to have the same understanding of tasks that will be implemented as it will improve the final outcome.
- story sizing estimation - it is an action of time estimation needed for story (tasks) to be completed. Usually determined in time scale (in rare cases it can also by represented by symbols or words). Breaking down the estimation time into points is more useful as it can be later used for sprint planning, where stories need to be selected in such a way that there is a high probability that all the stories will be finished within one sprint. The crucial thing is to relate the estimated time with other stories. One of the most commonly used techniques for story sizing is Planning Poker. Each developer gets special cards (with story points) and shows them at the same time. If the difference between minimum and maximum value is substantial, the “poker players” explain their estimations. After that voting goes through next iteration until all developers agree on timing for discussed story [19, p. 98].
- sprint review - it is a meeting that takes place on each last day of sprint. The main goal is to demonstrate a potentially shippable product (or the outcome of the sprint) It is a good practice to plan it on early morning hours and inviting all stakeholders. Invitations should be sent sufficiently early, so that all can come. Scrum Master and the developers team collate all comments so that they can be taken into account in future sprints. They present stories that have been completed. Ideally all stories that were chosen should be finished [12, p. 98]. End of sprint usually means a beginning of the next one. Usually at the end of the sprint review there is a session called “preplanning” which already involves planning the next sprint.
- sprint retrospective - this is an internal meeting of a developers’ team (without stakeholders). The purpose is to summarize the efforts that have already been made. It is the time to rethink current actions and present new learnings to all team members. Good practice tells that retrospective should take from one to two hours for every week of work on the project [17, p. 33].
- release planning - sometimes the Scrum process is enriched by the process of long term planning (taking into account more than one sprint). In this type of

meeting the product owner should join the team of developers to discuss stories from the backlog, setting schedules, analysis of the workforce capacity and spotting new opportunities as well a identifying potential risks. It is a good practice to set a timeline and time limit for this kind of meetings [20, p. 22].

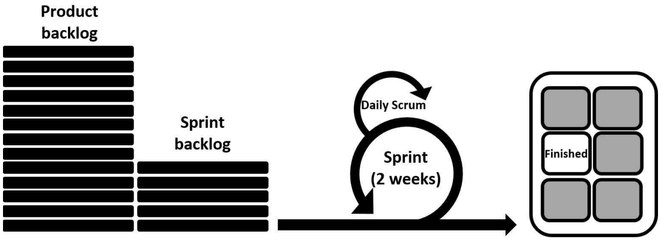


Figure 1. Scrum process

A Fig. 1. shows in a nutshell how Scrum proceeds through next iterations (sprints) in order to deliver a piece of final product. Each of such pieces is later shown during a sprint review and can serve as a reference point in identifying the current status of the project.

3. E-learning system for the Scrum methodology in IT project management

3.1. Existing solutions and challenges for E-learning

There are many available e-learning systems. One of the most complete approaches to E-learning can be realized with the help of computer software in form of Learning Management Systems (LMS). Main goal of LMS is not only to deliver an environment for learning, but also to make process of management easier and more structured. Despite the fact that LMS are usually focused on the administration processes, they are also open for other forms of learning processes such as: blended learning (form of learning that combines traditional approach with distance learning) [8, p. 60]. LMS provide tools that allow to track students' achievements as they progress with the course [9, p. 9-12]. They provide functionalities that can be used while performing daily activities such as: management of educational materials, schedules, reports and individual assessments of participants.

The European LMS Market Dynamics research shows that one of the most commonly used LMS in Europe is Moodle [1]. Moodle stands for Modular Object-

Oriented Dynamic Learning Environment and is distributed on an open source license. It has a lot of features that allow for networking. As creators of Moodle say, the biggest strength of this solution is the switch from just storing educational materials on a server to building a supportive community in the process of gathering knowledge [10, p. 19-20].

The growth of e-learning over the years has led to a concept of MOOC (Massive open online course). It is a concept of open courses that are widely available for a mass receiver. It is called massive because the number of listeners greatly exceeds numbers of participants of normal e-learning courses. They are usually free of charge and make use of the most up-to-date tools and technologies. Companies like: edX, Harvard Open Courses or Coursera often offer courses held by prestigious universities, such as: Harvard, University, Berkley University of California or Massachusetts Institute of Technology. In the United States of America, institutions running such courses are accredited by the American Council of Education. This makes it possible to acclaim results of the students in the traditional learning process [21, p. 58].

One of the biggest difficulties when learning is to find the linkage between academic knowledge and real life. As an example one can refer to the graduates, who have very strong theoretical background but may not yet have professional experience. For such users it might be worth trying a more customizable approach. Something that would be flexible enough to adjust to the skills and needs of participants. A solution that takes into account needs of novice programmers and project managers learning Scrum methodology will be presented in the next section.

3.2. Main idea of the application

The application presented here is a web application written in Python (back-end) and JavaScript (front-end). It offers an environment for a student to place him/her in an imaginary scenario, where he/she is a part of a programming team charged with the goal of manufacturing a piece of software that is cut according to the vision of course creator or Product Owner.

The intent was not to create a fully complete solution (as that would require much more time), but rather to familiarize and stimulate participants to use Scrum approach with their further real life development projects.

The presented solution was inspired by the idea of gamification, which is based on simulating different scenarios that require from participants taking good and accurate decisions [5].

Since accessibility is a key feature of an e-learning idea, a presented system is a web application. Before use, each user is obliged to create an individual account, which can later be registered to one of the existing course templates. Each course starts with a description and some predefined settings and tasks, which are open for modifications. Once user joins the course, he/she has two Scrum sprints to create all the tasks (number of sprints has been limited for the sake of simplicity). As a result the team should create a potentially shippable product that should fulfill the requirements set up in the course. At the end of the course students not only should have a better understanding of agile methodology such as Scrum, but also they should be able to continue their project outside the application. The main idea is to familiarize users with the concept of Scrum, project management and the opportunities of "organized" cooperation.

One of the biggest challenges when creating an application like this is to replace these Scrum elements that need a "true" interactions, such as scrum meetings or the concept of stickers used on a scrum board, by a virtual solutions. Solutions to that problems have been presented in the latter part of the article.

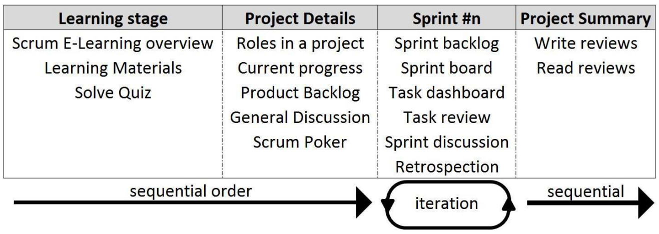
3.3. Who takes part in a process

There are several roles defined in the presented application. The first role is a role of Product Owner, which symbolizes a vision of the final product. Product owner might invent changes or change existing system requirements. He/she is responsible for all actions that are coming from outside of the team. He/she focuses on the product being done on time. His/her vision might change over time. Next, there is a role of Scrum Master - one person responsible for organizing the team. Scrum Master resolves conflicts, makes decisions when needed and tries to keep the whole team motivated to deliver a final product on time. Each sprint ends by showing a potentially shippable product that is created by a team of developers. Although developer role is the most basic one, it is essential for the whole course - this is where most of the job is done. The primary job of developers is to write code that corresponds to what has been agreed on user stories. They need to cooperate to efficiently deliver the final product. Apart from the users, that are directly involved in a realization of the course, there is also a role of an administrator which allows to supervise each course and make system modifications if necessary. Due to simplicity of a current application, some tasks requiring both changes in the application or specified course can be done by administrator.

The software presented here is not a fully complete solution, some simplifications were used, so roles are limited. In real projects there should be a much greater variety of roles, such as: testers, UX designers, graphic designers, software engineers etc.

3.4. Course workflow overview

The goal of this e-learning system was not only to guide students through the course, but also to allow for as much of their own invention as possible.



Learning stage	Project Details	Sprint #n	Project Summary
Scrum E-Learning overview	Roles in a project	Sprint backlog	Write reviews
Learning Materials	Current progress	Sprint board	Read reviews
Solve Quiz	Product Backlog	Task dashboard	
	General Discussion	Task review	
	Scrum Poker	Sprint discussion	
		Retrospection	

Figure 2. Scrum Learning application workflow

Fig. 2. presents main stages of the application. The general direction of work goes from left to right and from top to bottom. The subsequent horizontal stages are only available when Scrum Master enables them (they are locked by default). Sections indicated in Fig. 2 are presented below, sorted according to their order of occurrence in a project.

3.5. Learning stage

The learning stage begins with introduction to the application and describes the goal and mechanics of the system as well as provides some general information about Scrum, team management and programming techniques.

Application allows to upload some of the most commonly used formats such as .pdf, .mp3 or .mp4. Course creator role is to put all the files that should be necessary in the process of learning Scrum (such as Manifesto for Agile Software Development) as well as materials that concern a specific programming language (JavaScript is used for solving tasks).

Apart from that, learning stage covers topics that are specific for the project and would be necessary to understand the topic of the course. For instance, if the project was to develop software for a chess game, it would be nice if the rules of the chess were somehow explained in this section.

The last part of this stage is a test, which has to be passed in order to proceed to the next stages of the course. It ensures that every participant of the course has acquired the required knowledge. It resembles a little bit an entrance exams to the universities and is essential for assuring that everyone is “on the same page” with the course reading. Without passing the quiz it is not possible for the students to proceed further.

3.6. Project details

First step for starting a project is to assign roles in a team (forming a team in Scrum is one of the most important tasks). In the "current progress" section users can see details of the course such as: deadlines, participants and their roles, burndown charts, number of user stories etc. Each user can access this site at any given moment of the course.

At the beginning this section remains empty but as the team makes progress with their tasks more and more information is presented. Product backlog stage is one of the most important ones. It is here, where the Scrum team creates stories that will be realized. Some stories were predefined by the course creator. Other might be created by the users itself, if they want to create something extra. This gives the team an independence and makes each course very different from others.

General discussion section is a place where every participant can share their opinion on a public chat. Communication in Scrum is very important. It would be best to communicate in person, but that is one of many simplifications of the Scrum rules in the application, to fit into e-learning criteria.

Scrum poker is a place where team members estimate the time needed for stories to be completed. It is a very effective technique that allows to reach an agreement, even if first estimations are far apart.

3.7. Sprints

Once stories have been estimated and moved to the product backlog the actual work can be done. Each sprint starts with selecting stories from product backlog to the sprint backlog. Users together come to agreement on how many stories can be delivered during the sprint. Once selection is made, developers may proceed to sprint board (as shown in Fig. 3). Each developer can choose a story that he will be working on and move it to Work in progress section on the board.

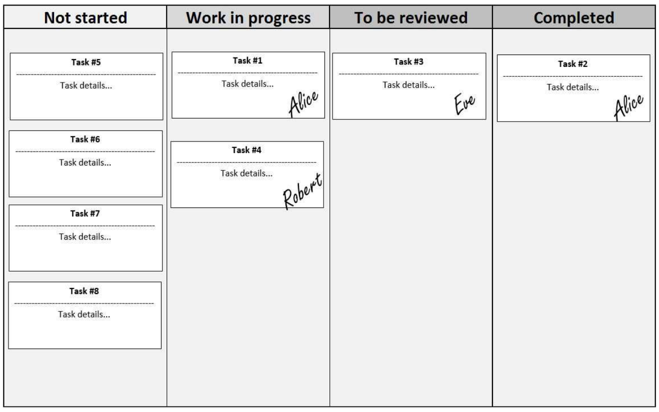


Figure 3. Scrum process

Once the developer decides the story is ready he/she can move the story to “To be reviewed” section. In this stage, automatic tests of the code can be carried out - in addition to any required manual test. After the story passed tests it can be moved to Completed section and story points are accounted to the current project status. The developer can pick another one from Not started section and start working on it. If story has not passed the tests or the person that was doing a manual inspection did not accept it, then such story needs to be rewritten to match the requirements.

At the end of each sprint there is a retrospection. This mechanism of presented application checks which stories have been finished and have met the criteria defined in the tests.

The goal of each sprint is to develop a shippable product increment. When sprint is finished, the system gathers all the stories from Completed section, combines them and presents results on a screen. This allows product owner and each member of the team to see the current progress of the project.

When the course is finished students can keep working on a project on their own by uploading files with code to the version control system of their choice or keep the results as a proof of their competence. They could also create another course and refine the project.

3.8. Project summary

The essence of Write reviews stage is a sincere opinion about each member of the team. Each participant has to fill a review form about other participants. The form is simple and has only two fields. A brief description of a user goes into the first field. The next field stores an overall grade (value from 1 to 5). Reviews are anonymous.

In Read reviews section each user can read his/her reviews without knowing their authors. This encourages all the participants to create honest reviews.

It is important to note that that Project summary should not be the only stage that allows for assessment as it should also take place during solving tasks and handling inter team communication during Scrum Poker or General Discussion phase. Such an approach to assessment is called formative assessment and is oriented toward individual student. Its effectiveness improves with the number of assessments. For that reason they should be performed throughout for the whole duration of the course and not just at the beginning or at the end. Systematic assessing stimulates students' involvement and systematic work that can boost their motivation and engagement. Finally it results in more thorough understanding of the material [6].

In contrary, summative assessment usually checks degree of subject absorption [7]. The tests are a valuable source of information for teachers as they can identify what parts of material have been well understood by students and which need a more in-depth explanation or revision. However, this approach is often criticized for leaving student on his own. If tests are taken at the end of course, it might be difficult to realize that some students are having problems with specific part of material in time.

4. Summary

The main objective of this paper was to present a new tool for training project managers and developers by offering to them the learning environment imitating the major aspects of the Agile projects. Since the project management skills greatly improve with the growing experience, the hands-on experimenting with various scenarios offered by the web-based system described here is likely to provide a helpful insight into field of project management.

It should be noted that the application at this stage is a prototype open for modifications. There are multiple ways that it could be further developed with one idea being to increase the number of parameters to current Scrum settings e.g. defining length of the sprint stage. Another idea is to build a tool, that would actually not only serve as a learning platform, but could also be used in a real-life projects. Such change would require by-passing the currently existing restrictions on number of iterations and project roles and increasing the role of product owner. These are, however, only technical issues, which can be solved with relative ease.

REFERENCES

[1] MindWires LLC (2016) *European lms market dynamics*.

[2] Goll J., Hommel D. (2015) *Mit Scrum zum gewünschten System.* Springer Vieweg.

[3] Wysocki R. K. (2013) *Effective Project Management Traditional, Agile, Extreme[7 ed.]*. John Wiley & Sons, Inc.

[4] Breitner M. H., Bruns B., Lehner F. (2007) *Neue Trends im E-Learning.Aspekte der Betriebswirtschaftslehre und Informatik*. Physica-Verlag Heidelberg.

[5] Bielecki W. T. (2014) *Gamification - learning by doing and fun?*

[6] Mokwa-Tarnowska I. (2015) *E-learning i blended learning w nauczaniu akademickim. Zagadnienia metodyczne.* Wydawnictwo Politechniki Gdanskiej.

[7] Thomas M. J. W. (2017) *Training and Assessing Non-Technical Skills a Practical Guide.* CRC Press.

[8] Basińska E., Majchrzak O. (red.) (2014) *Blended Learning jako przykład wdrożenia˙ nowoczesnych technologii w proces nauczania języków obcych, [in:] PLEJ3 czyli Psycho-Lingwistyczne Eksploracje Językowe.* Wydawnictwo Uniwersytetu Łódzkiego.

[9] Babo R., Azevedo A. (2011) *Higher Education Institutions and Learning Management Systems: Adoption and Standardization.* IGI Global.

[10] Brzózka P. (2011) *Moodle dla nauczycieli i trenerów. Zaplanuj, stwórz i rozwijaj platformę e-learningową.* Helion.

[11] Rubin K. S. (2013) Scrum. *Praktyczny przewodnik po najpopularniejszej metodyce agile.* Helion.

[12] Schiel J. (2011) *The ScrumMaster Study Guide.* CRC Press.

[13] Schwaber K., Sutherland J. (2017) *The Scrum Guide. The Definiive Guide to Scrum: The Rules of the Game*.

[14] Changes in Scrum Guides: https://www.scrumguides.org/revisions.html

[15] Dyer N. (2016) *Successful ScrumButt: Learn to Modify Scrum Project Management for Student and Virtual Teams.* CRC Press.

[16] Dyer N. (2011) *Scrum Project Management.* CRC Press.

[17] Sims C., Johnson H. L. (2014) *Scrum: a Breathtakingly Brief and Agile Introduction.* Dymaxicon.

[18] Pichler R. (2014) *Zarządzanie projektami ze Scrumem. Twórz projekty, które pokochają klienci*. Helion.

[19] Viscardi S. (2013) *The Professional ScrumMaster's Handbook.* Packt Publishing.

[20] Kuruliszwili S. (2014) *E-learning w kształceniu służb społecznych.* Centrum Rozwoju Zasobów Ludzkich.

[21] Project Management Institute (2017) *A Guide to the Project Management Body of Knowledge (PMBOK® Guide) – Sixth Edition.* Project Management Institute.