Annals of "Dunarea de Jos" University of Galati
Fascicle I. Economics and Applied Informatics
Years XXII – nº2/2016
ISSN-L 1584-0409 ISSN-Online 2344-441X
www.eia.feaa.ugal.ro

Towards an Agile Approach in Academic Software Development - A Case Study

Logica BANICA*, Magdalena RADULESCU**, Alina HAGIU***

ARTICLE INFO

Article history:
Accepted June 2016
Available online September 2016

JEL Classification
M15, C6, A2

Keywords:
Project management software; Agile methodology; Academic environment

ABSTRACT

The main focus of this paper is to emphasize the advantages of the Agile strategy when applied to a typical Project Management task. As software development has an old tradition, a comparison of the strengths of the new approach will be presented in detail, aiming to prove that the popularity of Agile is not without reason. There are software tools that help IT architects keep their teams and projects on track, and a case study for the academic environment will be discussed. The Conclusions part will summarize the results and shed some light on the future.

1. Introduction

Nowadays, in a dynamic and complex business environment, managing a project is considered a challenge, a test of organizational capacity, knowledge and experience of a company. Many projects fail, some of them being in final stages, but even when it is obvious that the final objective cannot be achieved, there are project managers who hardly admit their fault and refuse to add these failures to the chapter of experiences that mustn't be repeated. It would be preferable for the project manager to give a higher importance to the chosen methodology, which to highlight the team's ability, to allow adjustments of the initial version of the project and to provide clear tools for critical analysis of each stage and sub-phase, which allows rapid return after some deviations from normality.

The aim of this paper is not to demonstrate that these methods are obsolete, but that Agile is a new approach that gives more flexibility and more space for initiatives and original solutions to project management. The impact of Agile methodology was also mentioned by the 2015 CHAOS Report, which announced a success rate about 4 times higher for Agile projects, after studying 50,000 projects (Hastie & Wojewoda, 2015). In the experimental study, we developed a Student Data Management (SDM) system and experienced project management with VersionOne application, which led the market in 2015 (Wilson et al., 2015).

Our paper includes four sections and a Conclusions part. Section 2 presents the traditional methodologies (PMBOK and Prince2) versus the new Agile concept, summarizing the state-of-the-art, and briefly emphasizing the advantages and drawbacks of each methodology. Section 3 specifies the most significant tools and technics used by Agile methodology and section 4 includes our study case for managing a project in the academic environment, able to achieve a software for recording schooling in the University of Pitesti. The main concluding remarks close the paper, and present some directions of improving our future research activity in this domain.

2. Literature review

2. 1 What is Project Management?

According to the Association for Project Management (APM), the concept of project management refers to "the application of processes, methods, knowledge, skills and experience to achieve the project objectives which could be defined in terms of outputs, outcomes or benefits" (Naybour, 2014). Project management can be applied to any type of project and, consequently, to the development of software projects. Over time several methodologies have been launched in order to support and streamline the project management, most notably being: Waterfall, PMBOK, Prince2 and Agile.

Choosing methodology and fulfilling each stage objectives, within time and according to the allocated resources represents the key for a successful project (Project Management Institute, 2016). The next section

*, **,*** University of Pitesti, Pitesti, Romania.

provides a brief overview concerning the reflection in literature of the best known project management methodologies, focusing on Agile process.

2.2 Traditional methodologies: Waterfall, PMBOK and PRINCE2

2.2.1 Waterfall methodology

Waterfall is a benchmark methodology, which assumes that the purpose of the project can be reached by following several sequential stages, having the characteristic that each of them ends with a validation and that team members have phase objectives clearly defined and also timelines. This methodology typically used in software development consists in decomposition in sequential phases of the life cycle, which in turn are divided into activities and sub-activities (Banica et al., 2009). The number of stages differs from one implementation to another, between 5-7, but the best known model includes the following steps: Requirements, Design, Implementation, Verification and Maintenance.

We mention a few benefits of traditional waterfall approach in project management of information systems (McCormick, 2012): a total control over the stages, from the concept and planning to project completion and maintenance, project requirements are defined at the beginning, and their changes are not accepted than in exceptional circumstances, this methodology is used most often for large-scale software development projects where detailed planning and a predictable processing are very important.

The methodology has several drawbacks (Mishra & Dubey, 2013): the system is completed by proceeding sequentially through all the stages, which means a large period of time sufficient for the beneficiary to change his requirements; it doesn't realize a dynamic systems approach, which means that there are difficulties in treating the changes that occur during the running of the different phases.

2.2.3 PMBOK methodology

PMBOK (Project Management Body of Knowledge) is a methodology created by the PMI (Project Management Institute), that encapsulates a set of knowledge principles in project management, as well the project planning and the project deliverables (Project Management Institute, 2016). PMBOK methodology is based on general management knowledge and skills in project application environment, requires realistic estimations of the activities, resources and durations and also on a rigorous controlling schedule. The project passes through five stages during the life cycle, starting with the problem analysis formulated through a technical proposal and a schedule of activities chained into a logical sequence leading to the project purpose (Martens at al., 2013):

- Initiation – there are considered the requirements, feasibility studies are conducted and the stakeholders decide whether or not to undertake a project;
- Planning – at this stage the plan is elaborated including scope, cost, time, quality, communication, risk and resources; a Gantt chart being suggestive in representing the activities, resources, deadlines and important deliveries.
- Executing – at this phase the project deliverables are performed and almost simultaneously occurs the project controlling;
- Controlling – deals with the performance and the evolution of the project relative to the project plan; depending on fulfillment of performance indicators, the project manager can take corrective measures to keep the project on track;
- Closing – is the final stage of project design, but not of the project life-cycle; it includes the delivery of the product / service, ensuring the resources needed to use it and the end of the projection team activity.

From the experience of managers who confronted with multiple problems that can arise during the development of a project, appeared the component of risk management, having as goal to identify the risks and their impact level, and to propose solutions to solve them. An updated risk register can reduce the number of risks through closer monitoring of environmental factors.

2.2.3 PRINCE2 methodology

PRINCE (an acronym for PRojects IN Controlled Environments), created in 1989 by CCTA (the Central Computer and Telecommunications Agency), is a standard used by the British Government and widely recognized in the private sector in many countries (Matos & Lopes, 2013). A PRINCE 2 project is based on the business plan orientation towards the final purpose, is regularly reviewed to ensure that the business objectives, that may change during the lifecycle, will be met (Cheah, 2013). According to the official website, PRINCE2 is a "de facto process-based method for effective project management" (Prince2, 2016), having several key features: focus on business justification, defined organisation structure for the project management team, product-based planning approach, emphasis on dividing the project into manageable and controllable stages, flexibility that can be applied at a level appropriate to the project. The methodology divides the project into 7 phases, and each phase is managed separately (Bonnie, 2014) (Scheid, 2015):

- Starting up a project – focuses on defining the business case and building the project team;
- Initiating a project – contains the main activities of project, such as planning the project, including the quality management plan and the risk management plan, and also setting up project controls;

- Directing a project – refers to the activities of authorizing a stage or the entire project, and confirming project closure.
- Controlling a stage – includes the activities of monitoring and controlling the stages and the assessing progress of the project; based on the stage status, the project board could review the status, take corrective action, in order to receive completed stage results.
- Managing product delivery – refers to the activities of delivering a work package, acceptance and its launching;
- Managing stage boundaries – focuses on planning a stage, updating a project plan, a project business case, the risk registers and reporting stage end.
- Closing a project - prepares a benefits review plan and project evaluation.

PRINCE2 involves a clearly specification of the roles of designers and customers, so that each team member, along with project manager, to be responsible for decision-making in its field.

Both methodologies, PMBOK and PRINCE2, provide a framework for developing new projects without enforcing the tools, but recommending techniques that can be used. These standards represent two different approaches for project management and, in many cases, are able to complement each other *(*Matos & Lopes, 2013). But, also there is a big difference, namely, the project definition: PMBOK is a knowledge-based approach to project management, which describes the processes and techniques that can be applied to manage a project, while PRINCE2 is a process-based project management methodology that creates a management environment for developing a project. Another conceptual difference is related to the role of project manager: PMBOK emphasizes the role of project manager, which establishes the team, defines the roles and responsibilities of each team member, makes the project plan and takes the final decision in case of problems, while PRINCE2 provides a model of selecting the project team and a general description of roles in any type of project, the project manager solves daily problems, while key-decisions belongs to senior management.

2.3 Migrating to Agile concept in project management

The emergence of Agile framework occurred in 2001 at a meeting of software developers and IT companies, as a reaction to traditional project management methodologies, which became more restrictive, focusing on planning. The main ideas of this movement were included in the Manifesto for Agile Software Development (also called Agile Manifesto), that defines four values and twelve basic principles to be respected by those who adopt this project management methodology (Manifesto for Agile, 2001).

According to Sanjiv Augustine, Agile Project Management (APM) is "a way of managing projects to deliver customer value via adaptive planning, rapid feedback, continuous improvement and intense human interaction and collaboration" (Sanjiv et al., 2005). APM is typically used in software development for small and medium projects that require a limited upfront planning (Gandomani et al., 2013). Agile approach is based on an iterative and incremental style, that allows the project team to constantly assess project status and make changes while deviations or new requirements occur. Unlike traditional methods which stipulate that requirements definition, project planning and programming the activities for each stage can be established only at the beginning of the project, Agile approach consists of many iterations or sprints, which are themselves sub-projects, including all phases of development (requirements definition, analysis, design, development and testing) (Haas, 2007).

At the end of each iteration is achieved a prototype, then it is assessed and optimized by project team, and finally is transferred to the input of the next iteration. Another new feature brought by APM consists of integrating the customer/final user in the project team, along with the other members having usually well-defined functions: project manager, business analyst and specialists (system developers, subject matter experts, IT architect). Achieving a collaboration based on knowledge appreciation and mutual trust among team members is a key factor for the success of Agile methods. Concerning the tools needed to migrate towards the new technology, the project management should use flexible online tools, that can support incremental evolution, continuous communication and integration (Brockmann & Thaumuller, 2009), such as: progress reports, sprint planning, screening and scoring tools, project success sliders, task boards, bug management and burn-downs charts. Some of these tools will be presented in the next section and used in our experimental study.

Since 1994, every year, Standish Group published Chaos Report, which refers to the global statistics of IT projects successfully completed. The Project Management Institute (PMI) identified the following three critical success factors, also known as the Triple Constraints and the Iron Triangle (Doherty, 2011) (Ebbesen & Hope, 2013): the achievement of specified requirements (onTarget), the compliance with the original estimated budget (onBudget), the compliance with the agreed deadlines of delivery (onTime). The 2015 CHAOS Report studied 50,000 projects, ranging from small to large software development systems and the results show that the share of successfully completed projects is 29%, almost the same as last year, while the percentage of failed projects is 19%, slightly higher than in 2014 (Hastie & Wojewoda, 2015). The report also provides a comparison of project outcomes between Agile and traditional Waterfall projects and the results are summarized in Table 1.

Table 1. A comparison between the Agile and traditional Waterfall projects

Method	Successful	Challenged	Failed
Agile	39%	52%	9%
Waterfall	11%	60%	29%

Source: (Hastie & Wojewoda, 2015)

3. Agile software development - tools and techniques

As we mentioned in the previous section, APM involves sequencing a project into several iterations, each of them having the purpose to deliver a prototype/solution, allowing designers, as well as users, to have a feedback regarding the product evolution and to bring improvements.

The most important advantage of this approach is that the project is no longer accomplished within a rigid planning and doesn't remain unchanged from the beginning until the closing of stages, but allows immediate modifications during the progress of iterations, as partial results occur and other requirements are identified.

Agile projects can be implemented through different frameworks, which have many similarities as the iterative process requires, as well as a series of particularities emerged from the focus on different aspects of the software development lifecycle: improving the software quality (Extreme Programming) or improving the management (Scrum) (Hneif & Ow, 2009):

- *Scrum* is an iterative method which is focused on managing software projects. The Scrum guide states practices that help the project team to find out the tasks for each development iteration and to build a backlog for pending requirements. This tool has many advantages, such as: clearly expressing and ordering the items, optimizing the work of the team to achieve the project goal and mission.
- *Extreme Programming (XP)* is a software development methodology capable of adapting to changing requirements at any point during the project life. There are twelve rules that the project team should depend on when adopting XP referring on planning process (iteration planning), small releases, creating user stories, writing code to agreed standards and continuous integration.

In following we will briefly present the meaning of several terms commonly used in the development of a product through Agile methodologies. The *Product Backlog* is a chronological list of items to be achieved by the project team in product development (Sharma et al. 2012). *Sprint* is the set period to achieve a Backlog objective, that finishes with the activity evaluation and the acceptance or rejection thereof. Scrum approach recommends 30 days for a sprint and be steady. At the beginning of each sprint a planning meeting takes place, when the client specifies the requirements (reasonable purpose), and the team determines whether or not they can be completed during the specified period. The requirements must be detailed and written, concept also called *user stories* in XP method. During the sprint, the team has briefly daily meetings, also called *stand-up meetings*, where each member involved in that phase reports the progress or discusses about the encountered problems, in order to find a solution. In Agile methodology, Gantt charts no longer represent the main tool for tracking the project progress. There are others metrics more important, such as *burn-down charts*. The Scrum method provides the burn-down chart as a simple and powerful tool to track progress daily. The project manager together with the design team establishes the model: plotting the graph using story points or using task count. A burn-down chart based on counting tasks is built for each sprint from Sprint Backlog, having estimated timeframe and goals to accomplish (Haas, 2007). First is plotted the linear graph of estimated values and then the daily progress is tracked by drawing the line between two points representing the counter tasks that remain to be fulfilled until the end of sprint. The graph may show some variations, having the following significance: an increase indicates that new features have been added to the backlog, a decrease indicates the progress of the team according to the schedule and a constant level means the team didn't progress and there were no additional requirements.

In Section 4 we will present an example of Agile approach used for project management to achieve a software for recording schooling in the University of Pitesti.

4. Projecting a Student Information System based on Agile methodology

The case study refers to the development of a management information system for academic environment using Agile methodology. It is a medium-sized project, which aims to provide capabilities for registering students at college admission, to publish yearly the Curriculum updated, to track the student attendance to courses, to keep the evidence of grants and fees, to assure a full confidentiality and secure access of each student to its assessment results. We considered the Agile Student Data Management (SDM) system as the core of a university Enterprise Resource Planning (ERP) system, along with the online educational platform and the accounting and budgeting components. At the first stage, the project manager constituted the project team: an analyst, a designer, two developers, a tester and a representative of the university's accounting department and organized the kick-off meeting. The agenda of the meeting included the project goal, deliverables, the deadline and the resources allocated. Also, the PM introduced the project roles and emphasized the reasons for assigning the respective roles. Based on project requirements and information submitted by the PM, the team established a project plan, synthesized in the Sprint Backlog. Thus, we identified 3 iterations in the project, including a total of 15 features (story points), which the team must "burn through" (Table 2).

Table 2. The Sprint Backlog of the project

Sprint no.	Sprint Description	Story point no.	Story point description
#1.	Defining the structure of the scholar register and enrolling new students	1	Handling the admissions process: building the admission forms, accessing the forms from Intranet stations of university
		2	Online enrolling new students: filling out the form and sending to the admission database
		3	Checking of data entered and enrolment conditions
		4	Calculating the average note and setting the student list in descending order of the average note
		5	*Usernames and password* management for accessing the student records; granting students, the credentials of access to their accounts
#2.	Fees and scholarships management	1	First year of study: Establishing the students admitted on budgeted places
		2	Second and third year of study: Calculating the average note of evaluation yearly results and setting the student list in descending order
		3	Tracking fee payments and assuring a private and secure online communication concerning this status with students
		4	Accounting registration and updating of the fees
		5	Accounting registration and updating of the scholarships
#3.	The management of curricular activities, class and teacher schedules, student evaluation results, student promotion conditions	1	Building and updating the curriculum closely with labor market needs and assuring the online publishing
		2	Automatically creating class and teacher schedules
		3	Registering the evaluation assessments and academic progression
		4	Verifying the student promotion based on the total number of credits earned until the end of the academic year
		5	Awarding the student scholarships on the basis of their academic achievements

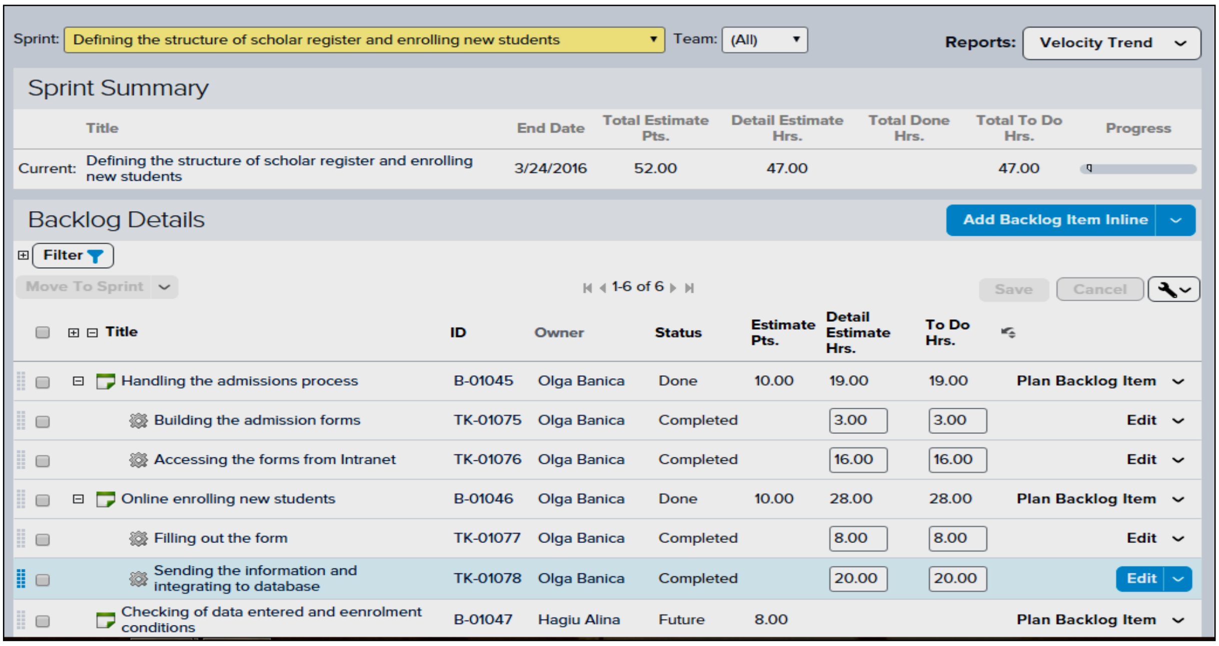
Also, the PM conveyed the message that he/she is open to any recommendations, is able to acknowledge the potential bottlenecks and, consequently, to change schedule and tasks as the project progresses. In this experiment, the project manager plotted an initial burn-down chart of planned, fixed-length iterations (also called sprints) and, as the tasks have been delivered, the progress is represented by a real chart, depending on all features remaining in a sprint backlog. At the end of the first iteration team found it necessary *to add a new task*: ERASMUS agreements for students: admission, registration, university tuition, course exams; following the rent of their accommodation.

The Student Data Management (SDM) project is a software product developed by Agile Scrum Methodology and follows the cycle of schooling of the student from admission to graduation, continuing with his pursuing career after graduation, according to the acquired skills and knowledge.

There are many different types of Agile software implementations, available as web-based services on Linux and Windows operating systems. We choose to use for our project VersionOne software, because it is one of the most popular tools for project management system supporting Agile development, and it is available both as self-hosted tool and as web-based service. VersionOne provides features associated with Scrum methods, quality management, business intelligence and collaboration and has led the market in 2015 (Wilson et al., 2015). In our experiment, we followed several procedures to coordinate the team and activities in a project, to supervise the project, according to Agile approach and VersionOne facilities (VersionOne, 2016).

Initialization of the project is the first step and refers to the title, estimate duration and the team working at this project. The project manager adds new members, specifying their roles. The entire team works and cooperates on web-based platform, password protected, and only the project manager can modify the initial information about schedule and team, and also, may decide a completion and an acceptance of a sprint, after all tests are passed. In this phase is created the Sprint Backlog, a schedule of project that identifies each sprint by the title, duration and team member responsible of its accomplishment.

Product planning is the second phase that includes information and responsibilities more detailed concerning the Items of each Sprint and the Tasks planned (Fig. 1). At the beginning, all Items have the status *Future* and a number of hours estimated, but daily could change the field content, to record the Items (or Tasks) progress. For better tracking the progress, the project manager must establish a responsible for each Item.



Sprint: Defining the structure of scholar register and enrolling new students ▾ Team: (All) ▾ Reports: Velocity Trend ⌄

Sprint Summary

	Title	End Date	Total Estimate Pts.	Detail Estimate Hrs.	Total Done Hrs.	Total To Do Hrs.	Progress
Current:	Defining the structure of scholar register and enrolling new students	3/24/2016	52.00	47.00		47.00	

Backlog Details

	Title	ID	Owner	Status	Estimate Pts.	Detail Estimate Hrs.	To Do Hrs.	
□	⊟ Handling the admissions process	B-01045	Olga Banica	Done	10.00	19.00	19.00	Plan Backlog Item
□	Building the admission forms	TK-01075	Olga Banica	Completed		3.00	3.00	Edit
□	Accessing the forms from Intranet	TK-01076	Olga Banica	Completed		16.00	16.00	Edit
□	⊟ Online enrolling new students	B-01046	Olga Banica	Done	10.00	28.00	28.00	Plan Backlog Item
□	Filling out the form	TK-01077	Olga Banica	Completed		8.00	8.00	Edit
□	Sending the information and integrating to database	TK-01078	Olga Banica	Completed		20.00	20.00	Edit
□	Checking of data entered and eenrolment conditions	B-01047	Hagiu Alina	Future	8.00			Plan Backlog Item

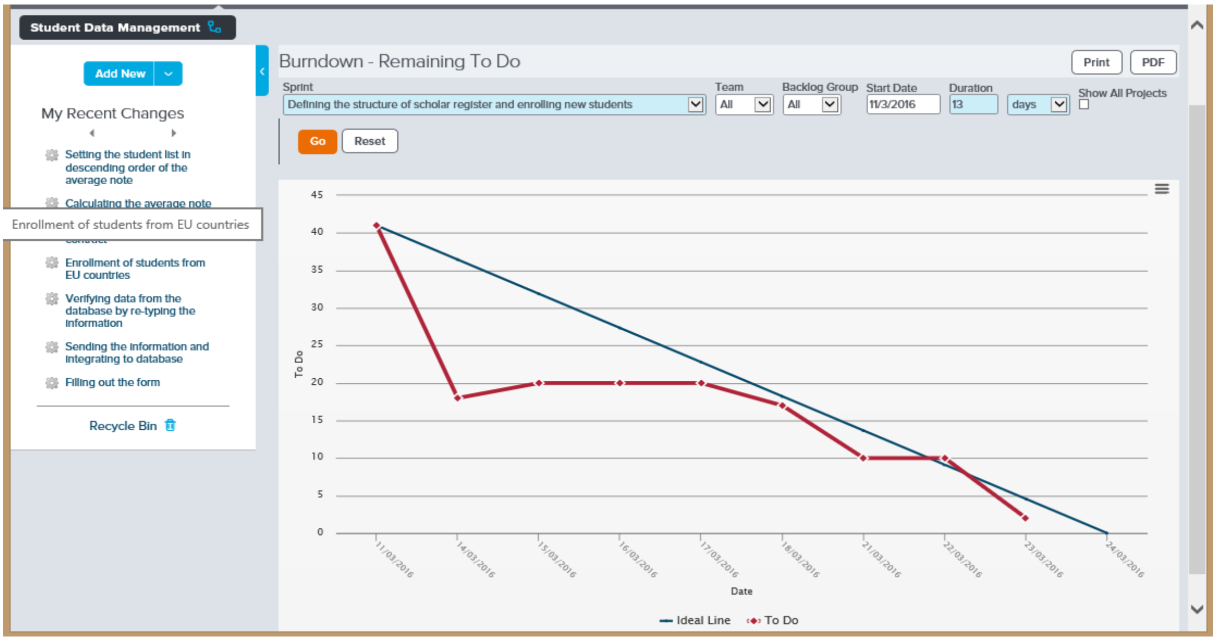
Fig. 1. Building the Backlog for the project and the Tasks for each Sprint

Sprint Tracking is the third step of the project which refers to the periodically evaluation of the Tasks, Items, Sprints in order to determine the current state of the project. There are many tools to examine the evolution, such as Storyboard that shows the status of Items to a selected Sprint. The fourth phase is *Sprint Review* that allows the project manager to update the schedule with the current state of work. So, based on daily reports of the team about related aspects of change that can occur on a project, and the acceptance of the completed tasks, items or sprints, the PM can change the status from Future – In Progress - Completed – Accepted and finally, closing the planned unit. If a task is delayed, the team cannot afford for the entire project to be delayed by a few days or a week. Instead, the PM must evaluate the resources and options available and get the project back on track, by operating some changes in the previous version of the schedule, including the addition of new tasks or sprints. The fifth step, *Quality and Metrics,* represents a method of evaluation using metrics to indicate different types of progress at the iteration and release levels of SDM project. The *project quality* can be assessed objectively by the PM, since VersionOne offers a lot of tools for understanding the progress of the team and for keeping the stakeholders informed of the project's current status. So, it provides an accurate measurement of the *Open Estimate (*Items not yet completed), *Total Estimate*, and the *Percent Complete* for each task. The *Reports* module is designated for gathering important metrics concerning the entire project using the options: Project Dashboard, Project burndown, Velocity trend and also, about Sprints, using the options: Sprint Dashboard, Sprint burndown and Detail Estimate Trend. This module focuses on evaluation of metrics, can be launched at every moment of project development and depending on the results, could lead to a corrective action or process improvement activities. In Fig. 2 is presented a burndown chart for the first sprint after 4 days of working and after adding a new task: *Enrolling of students from EU countries*.

The graph provides a detailed overview of the first sprint and how the team are burning the user stories:

- the Ideal Trend line indicates the situation in which the team burns down all of the effort that remains at a constant rate by the end of the sprint;
- at the beginning, the team went faster than they planned and added more work;
- generally, the work line is below the ideal line, it means that the project is ahead of schedule;
- the Project Manager added a task in the mid-sprint, which increased the amount of effort required;

- the additional effort and a slower pace of the team conduct to a work line above the ideal work line, meaning the project is behind schedule.


Student Data Management

Add New

My Recent Changes

Setting the student list in descending order of the average note

Calculating the average note

Enrollment of students from EU countries

Enrollment of students from EU countries

Verifying data from the database by re-typing the information

Sending the information and integrating to database

Filling out the form

Recycle Bin

Burndown - Remaining To Do

Print PDF

Sprint: Defining the structure of scholar register and enrolling new students

Team: All

Backlog Group: All

Start Date: 11/3/2016

Duration: 13 days

Show All Projects □

Go Reset

Fig. 2 Sprint burndown report

5. Conclusions

This paper has presented how a project can be efficiently managed and headed for successful completion by using VersionOne planning application, and the Agile principles:

- by reacting quickly to the changes, the product keeps its flexibility and is able to adapt when the customer becomes more demanding;
- many techniques are already supported by VersionOne, such as Kanban, Scrum and XP – though we only focused on Scrum methodology;
- in order to better manage the development process, small iterations are performed and assessed, and problems are solved early in the life of the product; this way, the improvement can occur when needed and the final result will be much more reliable;
- The SMD team used the relevant metrics of VersionOne in order to estimate the project progress and make important decisions concerning the estimated effort and time: daily Sprint burndown charts, Task boards and Sprint boards to provide information about how the team is performing within each Sprint, to predict the evolution and to reduce the risks of project failure.

Concerning the future work, the authors of this paper aim to continue the development of small and medium software projects using Agile methodology and to evaluate other applications available on the IT market, in order to make a comparison among their features and to choose the optimal product.

References

1. *Banica, L., Rosca, D., Stefan, C., (2009), "A software for project management process", Proceedings of the Romanian Academy, seria A, volume 10, nr.2.*
2. *Bonnie, E., (2014), Project Management Basics: PRINCE2 Explained, https:/www.wrike.com/blog/project-management-basics-prince2-explained/, accessed on March 2016.*
3. *Brockmann, P. S., Thaumuller, T., (2009), Cultural Aspects of Global Requirements Engineering: An Empirical Chinese-German Case Study, 4-th IEEE International Conference on Global Software Engineering, pp. 353–357.*
4. ***Cheah, A., (2013), PRINCE2 Project Management Explained!, https://www.projectsmart.co.uk/prince2-project-management-explained.php, accessed on January 2016.***
5. *Doherty, M. J., (2011), Examining Project Manager Insights of Agile and Traditional Success Factors for Information Technology Projects: A Q-Methodology Study, http://www.pmi.org/~/media/ PDF/ Surveys/_Doherty-summary.ashx, accessed on February 2016.*
6. *Ebbesen, J., B., Hope, A., (2013), Re-imagining the Iron Triangle: Embedding Sustainability into Project Constraints, PM World Journal, volume II, Issue III, March 2013, https://www.researchgate.net/publication/235933029_Re-imagining_the_Iron_Triangle_Embedding_Sustainability_into_Project_Constraints, accessed on February 2016.*
7. *Gandomani, T., J., Zulzalil, H., Ghani, A, Bakar, A., (2013), Obstacles in Moving to Agile Software Development Methods; At a Glance, Journal of Computer Science, volume 9, Issue 5, pp. 620-625, https://www.researchgate.net/publication/ 237077450_Obstacles_in_moving_to_agile_software_development_methods_At_a_Glance, accessed on December 2015.*
8. *Hass, K., (2007), The Blending of Traditional and Agile Project Management, PM World Today, volume IX, Issue V, retrieved from http://cs.anu.edu.au/courses/COMP3120/local_docs/readings/the-blending-of-traditional-and-agile-project-management.*
9. *Hastie, S., Wojewoda, S., (2015), Standish Group 2015 Chaos Report - Q&A with Jennifer Lynch, http://www.infoq.com/articles/standish-chaos-2015, accessed on March 2016.*

10. *Hneif, M., Ow, S.,H., (2009), Review of Agile Methodologies in Software Development, International Journal of Research and Reviews in Applied Sciences, volume 1, Issue 1, pp. 1-8.*
11. *Manifesto for Agile Software Development, (2001), http://www.agilemanifesto.org/, accessed on December 2015.*
12. *Martens, P., Van Riet, M., Santos, R., M., (2013), The Management of International Development Projects: Enhancing Sustainable Outcomes with Improved Project Management Education and Training, Maastricht School of Management Research Conference 2013*, *http://web2.msm.nl/RePEc/msm/wpaper/MSM-WP2013-20.pdf, accessed on April 2016.*
13. *Matos, S, Lopes, E, (2013), Prince2 or PMBOK–A Question of Choice, Elsevier Procedia Technology, volume 9, pp. 787–794.*
14. *McCormick, M., (2012), Waterfall vs. Agile Methodology, http://www.mccormickpcs.com/images /Waterfall_vs_Agile_Methodology.pdf, accessed on January 2016.*
15. *Mishra, A., Dubey, D., (2013), A Comparative Study of Different Software Development Life Cycle Models in Different Scenarios, International Journal of Advance Research in Computer Science and Management, volume 1, Issue 5, 2013, pp. 64-69.*
16. *Naybour, P., (2014), Project management – an introduction, Association for Project Management-APM, https://www.apm.org.uk/WhatIsPM, accessed on February 2016.*
17. *Project Management Institute Official site, (2016), What is Project Management?, http://www.pmi.org/About-Us/About-Us-What-is-Project-Management.aspx, accessed on April 2016.*
18. *Prince2 Official site, (2016), https://www.prince2.com/uk/what-is-prince2, accessed on May 2016.*
19. *Sanjiv. A., Payne, B., Sencindiver, F., Woodcock, S., (2005), Agile Project Management: Steering from the Edges, Communication of the Association for Computing Machinery (ACM), volume 48, No. 12.*
20. *Sharma, S., Sarkar, D., Gupta, D., (2012), Agile Processes and Methodologies: A Conceptual Study, International Journal on Computer Science and Engineering (IJCSE), volume 4*, *No. 5, pp. 892-898.*
21. *Scheid, J., (2015), Project Management Methodologies: How Do They Compare, http://www.brighthubpm.com/methods-strategies/67087-project-management-methodologies-how-do-they-compare/, accessed on March 2016.*
22. *VersionOne Official site - Help Center, (2016), https://community.versionone.com/Help-Center, accessed on April 2016.*
23. *Wilson, W., Duggan, J., Murphy, T., Sobejana, M., Herschmann, J. (2015), Magic Quadrant for Application Development Life Cycle Management, https://www.gartner.com/doc/reprints?id=1-2A61Y68&ct=150218, accessed on May 2016.*