

UDC 005.8(497.7)
Original scientific paper

Neda KALESHOVSKA¹⁾
Sasho JOSIMOVSKI²⁾
Lidija PULEVSKA-IVANOVSKA³⁾
Kiril POSTOLOV⁴⁾
Zoran JANEVSKI⁵⁾

THE CONTRIBUTION OF SCRUM IN MANAGING SUCCESSFUL SOFTWARE DEVELOPMENT PROJECTS

Abstract

The goal of this paper is to acknowledge the contribution of Scrum as one of the agile methods, which helps in overcoming the most common reasons for project failure in software projects, and the benefits from this concept. This paper proposes how the success rate in these projects can be improved by adopting Scrum and analyses the benefits and results of such undertaking. Finally, this paper presents an empirical research on project management in software development in Macedonia, and the management of these projects in regards to Scrum.

Keywords: Software projects; Project management; Scrum; Agile Methods; Macedonia.

JEL classification: M15

¹⁾ MSc, Project Office, Vip Operator, Email: neda_kales@yahoo.com

²⁾ PhD, "University Ss. Cyril and Methodius University-Skopje",
Email: sasojos@eccf.ukim.edu.mk

³⁾ PhD, "University Ss. Cyril and Methodius University-Skopje",
Email: lidijap@eccf.ukim.edu.mk

⁴⁾ PhD, "University Ss. Cyril and Methodius University-Skopje",
Email: kirilp@eccf.ukim.edu.mk

⁵⁾ PhD, "Institute of Economics – Skopje",
E-mail: zoran.janevski@ek-inst.ukim.edu.mk

Introduction

The invisible engine today that moves the economy and the whole society at large is software and the innovations in the information technology (IT), which change the way people work and live. The advancement in the information technology, the numerous innovations, programs, software tools etc. lead to many projects in the IT and software area. However, although project management is the answer that many companies have adopted today as a fast and efficient management practice, there is still a high rate of unsuccessful and challenged software projects (CHAOS Research, 2012) that demand new possibilities, new methodologies and frameworks for improving project management as a practice in this area.

Agile methods have been particularly popular for software projects. This increasing interest is believed to be due to the contribution of agile methods as powerful development alternative, supposedly able to avoid project problems such as low productivity schedule delays, high costs, lack of people's motivation and other problems, which are still significant problems in the software engineering field (Dyba and Dingsoy, 2008).

This paper aims at providing answers to the following working hypotheses and their respective research questions:

WH1: The application of Scrum as project management framework, contributes towards increased number of successful software projects;

RQ1.1: How does Scrum contribute towards increased number of successful software projects?

RQ1.2: What is the adoption of Scrum like in comparison to other project management frameworks and methodologies?

WH2: The companies in Macedonia use project management methodologies in managing their software projects;

RQ2.1: What are the most common project management methods for managing software projects in Macedonia?

RQ2.2: Is Scrum used among IT companies in Macedonia and is this project management framework perceived as suitable for managing software projects?

RQ2.3: Could Scrum be considered to be a driver of success for software projects?

The paper emphasises the outcome of both theoretical and empirical research and offers conclusion points for further research analyses in this topic.

In this paper we use the following research methodology: analysis–synthesis; inductive-deductive method; statistical method; historical method; comparative method; descriptive method; classification method; empirical method; and compilation method.

1. Basics of Project and Project Management

Project activities throughout history have enabled the realization of numerous activities, managing the undertakings in a planned and organized manner. The formal acceptance of project management as a management practice and the development of the methodologies, techniques and tools for planning, managing and controlling projects have enabled true exploitation of the benefits of project management.

The international organizations Project Management Institute, PMI, and International Project Management Association, IPMA, have played major role in the wide acceptance and affirmation of project management, by establishing standards, knowledge, processes, skills, techniques, frameworks, etc. that contribute towards established and organized project management approach.

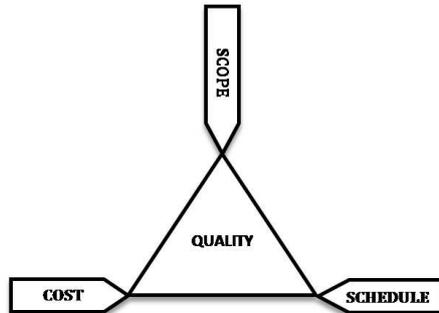
1.1. Defining Project and Project Management

There are numerous definitions in the literature that help in understanding project and project management. A widely accepted and referenced definition is provided from PMI (2013, p.3), which defines the **project** as a “temporary activity, an endeavour, undertaken to create unique product, service or end result”. There are many examples for projects nowadays, such as new product or service development, implementation of new or modification of legacy information systems, building construction, infrastructure etc. On the other hand, PMI (ibid, p.4) defines the **project management** as the “application of knowledge, tools and techniques upon the project activities to realize and achieve the project requirements”.

Projects are *unique*, they are *timely bounded composition of activities* with defined *beginning and end*, well defined *project goal*, and

most often they consist of defined and approved *project budget*. To these regards, a project is bounded by a time constraint, a budget constraint and scope of work constraint, known as project management triangle demonstrated in Figure 1.

Figure 1. Project Management Triangle



Source: Adjusted from Biafore, B. (2011), "*Successful Project Management*", O Reilly Media, Inc., p.6.

The project management triangle demonstrates which aspects a project needs to address to have successful realization. A successful project is the one that will end in the time frame approved, in the budget approved and will deliver the requirements as specified in the project when approved. Deviation from the project baseline would compromise the project success and depending on the constraints set and the proportion of the deviation, the project could end up in partial success or failure.

1.2. Drivers for Project Management Adoption

The drive and incentive for project management as a management practice on one hand derive from the external pressures and competitive rivalry, and on the other hand from the benefits that this practice brings for managing larger activities. External pressures include forces such as advances in technology and increased customer demand. But a major drive for project management as a worldwide accepted management practice are the numerous benefits that project management brings (Kerzner, 2009).

In this paper we also identify several advantages from an empirical point of view and observe these factors as benefits from

adopting project management: focus and goals' accomplishment, identification of client requirements and delivery of these requirements, increased client satisfaction, time planning and faster time to market new products, services, systems, flexibility and opportunities for development of project plan alternatives, effectiveness and efficiency in managing project activities, increased productivity, support and alignment of project goals with the strategic targets of the companies, delivery of the right products at the right time, realization of activities in an organized, planned and coordinated manner, etc.

2. Success in Software Project Realization

One of the most widely used reports when it comes to project success rate in software related projects is the CHAOS Report. The first reporting of the project success/failure rate dates from 1994 (Standish Group Report, 1995), where the Standish Group was collecting data regarding the project success/failure rate throughout the years (Standish Group Report, The CHAOS Manifesto, 2013). The results from these reports take into consideration approximately 365 companies with total of 8380 software systems under development as projects. The results represent a global view of project statistics with heavier concentration in US and Europe. For each reporting period about 60% of them are US based, 25% are Europe based and the remaining 15% are from other parts of the world. The results from this report consolidate the success/failure rate in software projects in three categories:

- > **Project success** - projects completed on time within the budget approved, containing the functionalities according to specification.
- > **Project challenged** - projects completed, but either beyond the budget approved, or longer than planned and/or not fulfilling some of the functionalities initially specified in the scope of the project.
- > **Project impaired** - projects that were cancelled at some point in time after commencement.

The graphical representation of the software development project success/challenged/impaired rate and the progress from 1994-2012 is demonstrated in Table 1.

Table 1: Software Project Success Rate Progress 1994-2012 (Standish Group)

| | 1994 | 1998 | 2000 | 2004 | 2006 | 2008 | 2010 | 2012 |
|---------------------------|------|------|------|------|------|------|------|------|
| Project success | 16% | 26% | 28% | 29% | 35% | 32% | 37% | 39% |
| Project challenged | 53% | 46% | 49% | 53% | 46% | 44% | 42% | 43% |
| Project impaired | 31% | 28% | 23% | 18% | 19% | 24% | 21% | 18% |

Source: According to information from Eveleens, L.,Verhoev C. (2010), “*The Rise and Fall of the CHAOS Report Figures*”, IEEE Software, p.31 and The Standish Group, (2013), “*The CHAOS Manifesto 2013*”, p.1.

Although many authors have disputed the numbers from the Standish group as too extreme and too high (Glass, 2006), (Jorgensen and Molokken, 2006), (Zvegintzov, 1998) claiming that the interpretation and meaning of the formulation for project challenged and project impaired is not as accurate, this report remains as the one with highest reference when the problem of software project success rate is being considered (IT Cortex, 2002).

The consequences from such failure are implicated in content deficiencies and high financial expenses. Some of the well-known software project failures and the connected consequences from this failure are demonstrated in (Charette R., 2005).

The ten most common problems that cause failure in software projects according to the CHAOS report are listed in Table 2.

Table 2: Reasons for Project Failure

| Project impaired factors | % of responses |
|-------------------------------------------|----------------|
| 1) Uncompleted requirements | 13.1% |
| 2) Lack of user involvement | 12.4% |
| 3) Lack of resources | 10.6% |
| 4) Unrealistic expectations | 9.9% |
| 5) Lack of executive support | 9.3% |
| 6) Changing requirements & specifications | 8.7% |
| 7) Lack of planning | 8.1% |
| 8) Didn't need it any longer | 7.5% |
| 9) Lack of IT management | 6.2% |
| 10) Technology illiteracy | 4.3% |
| Other | 9.9% |

Source: Standish Group Report, (1995), “*CHAOS*”, p.4-5,
[<http://www.projectsmart.co.uk/docs/chaos-report.pdf>] (08.09.2014)

Other papers and reports also confirm and correlate with these factors as root causes for failure in software projects, emphasizing “User requirements”, “Unrealistic customer expectations”, “Insufficient planning”, “Transparent communication” etc. as main reasons upon which the success in software projects depends.

3. The Scrum Framework Concept

The increased complexity of software development projects and the dynamic pace of changing of technologies nowadays, have led to increased demand for flexibility and adaptation in the user requirements as a project progresses. This situation has imposed the need to develop methods that offer flexibility in the planning of software projects and project realization.

Incomplete requirements and alteration of requirements have been recognized as the top problems in software development project

management, and at the same time, a reason for failure. According to research by Brewer and Dittman (2009, p.38) in the software development, 25% to 50% of the requirement in these projects change in the process of development. This has led to development of new methods, frameworks and methodologies that will offer flexibility in project management and agility in both project planning, project execution and project change management. There are several agile project management methodologies, such as Extreme Programming – XP, Agile Modelling, Lean development. However, subject of this paper is analysis of one particular agile project methodology – Scrum, which by itself is the most popular and the most used agile project management framework (Forrester, 2009).

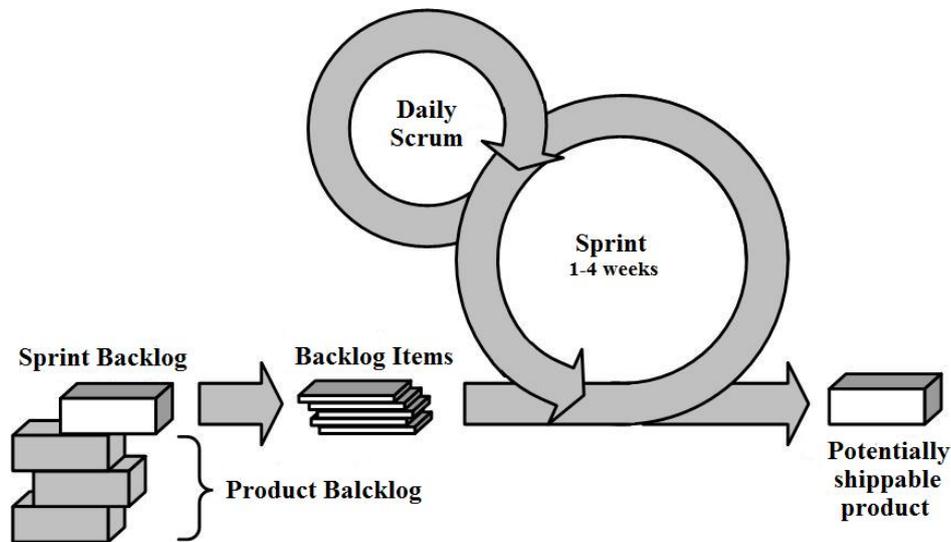
3.1. Scrum Introduction and Development

Scrum has been formally defined as project management framework in 1995 in a workshop on OOPSLA (Object-Oriented Programming, Systems, Languages and Applications) (Schwaber and Sutherland, 1995). In the following years, these authors have continued to develop this method and to contribute to the development and its wide acceptance and affirmation.

Scrum is an iterative framework for managing projects, most common software development projects. The main reason why this framework is suitable for projects with IT background is that it acknowledges and welcomes project flexibility and changes whilst the duration of the project. During the project duration, the clients or users can make adjustment or changes in the requirements, which can be included in later stages of the project, in a following iteration.

The Scrum process in a simplified presentation is depicted in Figure 2.

Figure 2: The Scrum Process



Source: Adjusted from Sutherland J., Schwaber, K. (2011), "*Scrum Papers*"
[http://scruminc.com/tl_files/scrum_inc/documents/ScrumPapers.pdf]
(10.09.2014)

The structure of this framework is based on cyclical circles of work called sprints. These sprints, also known as iterations, are with usual duration of 1-4 weeks and follow one after another, without any breaks and pauses along the way. The sprints are time-bounded and finish on specifically defined date, regardless of whether work under that iteration has been completed and they are never prolonged. At the beginning of every sprint, cross functional team chooses a segment of user requirements from the **product backlog** that has been developed by the **product owner** together with the stakeholders. The product owner is the representative of the stakeholders of that project. The team dedicates itself on working on the chosen segment from the product backlog, known as the **sprint backlog**. The sprint backlog and the set of user requirements it would include is pre-defined during the sprint **planning** on a meeting between the team and product owner. During the sprint, the list of requirements is not a subject to changes. Every day, the team meets for short meetings, known as **daily Scrum meetings** addressing three topics each meeting:

- > What did you do yesterday?
- > What will you do today?
- > Are there any impediments in your way?

This enables daily progress tracking and up-close monitoring and control of the project. The coordination and control of the application of Scrum as a concept within the project is done by a **Scrum Master** that coordinates the Scrum daily meetings and ensures that the methodology is being implemented in the project. At the end of each sprint, the team makes a **sprint review** with the stakeholders and demonstrates what has been done and also receives feedback on matters that the team should focus on in the following sprint. At the end of each sprint, a **sprint retrospective** is also undertaken to evaluate what has been done well and whether there is room for improvement in the following sprint. The sprints go on until the last sprint backlog is implemented, thus fulfilling the whole user requirement list and delivering product backlog. In this manner, this framework offers product development in phases, sprints, enabling adaptability and flexibility that software projects deem as pre-requisite.

To further comprehend Scrum as a project management framework and the suitability of this framework for software projects, we compare Scrum to Waterfall and address areas that distinguish Scrum as appropriate and most fitting project management methodology, specifically for software development projects.

For software projects, the Waterfall methodology pre-assumes that all user requirements are defined in great details in the analysis phase and are not subject to changes during the duration of the project. However, if such changes do occur, which for software projects is quite often, and changes are more practice than exception, the project would result with change of the requests, which dramatically results with increased prices and project cost. Other weak points of the Waterfall framework is its rigidity, its low user involvement as well as the detailed planning in early phases of the project, when the information for the product scope is neither clear nor understood in details. Details of the comparison logic between Waterfall and Scrum are presented in Table 3.

Table 3: Comparison of Scrum and Waterfall Frameworks

| | Waterfall | Scrum |
|---------------------------------------------------|----------------------------------------------|---------------------------------------|
| Process defined | Mandatory at the beginning of the project | Only in planning and close down phase |
| Budget and project finish date | Determined and set in project planning phase | Determined during project duration |
| User involvement | Occasional | During the whole project duration |
| Flexibility and creativity within the team | Limited | Unlimited during the iterations |
| Transfer of knowledge within the team | Prior project starts | During the whole project duration |
| Possibility for success in IT projects | Low | High |

Source: Schwaber, K. (1995), "*Scrum Development Process, Advanced Development Methods*", Inc., p.68.

Although the Waterfall framework is widely recognized and used, it is most suitable for projects that have clear goals and requirements that can be set in the very early stages of the project. However, one of the main reasons why software projects fail is the project requirement definition, incomplete requirements, changing requirements and specification as well as lack of user involvement. Therefore, segmented delivery of the project in increments and iterations offering flexibility and possibility for changes proves more convenient and fitting. Supporting this conclusion, the CHAOS Report (Standish Group 2012), demonstrated the contribution of agile project management that includes Scrum against Waterfall for the realization of software projects from 2002 to 2010. The percentage of projects completed as successful is significantly greater, with an agile project management, than the ones using the Waterfall method – 42% successful software projects against 14% with Waterfall.

3.2. Correlating causes of failure in software projects and the contribution of Scrum to bypass this problem

To answer our first research question “How does Scrum contribute towards increased number of successful software projects?”, we correlate the causes of failure in software projects identified in the previous segment, and point out for each one how Scrum contributes in overcoming these problems and how it contributes for successful software project realization.

- 1. Incomplete requirements** – The main idea behind Scrum is the acknowledgement that this concept accepts change of requirements during project duration and management of these changes in upcoming project iterations. The methodology is designed to be flexible all along the project life.
- 2. Lack of user involvement** – User involvement takes great part in Scrum as the users/clients are included in the process during the whole project duration, providing input for User Requirements (UR) and product revision. By using Scrum, a good relationship with the clients is usually developed and also the knowledge of the client about the product/project increases during the time. This relationship sets grounds for stable long term cooperation.
- 3. Lack of resources** – The team working under Scrum are “general specialists”, people with specific field of expertise, however, also with general knowledge from similar areas, thus enabling resource support among the team. A programmer can assist and support a project tester if there is available time allocation on the side of the programmer. The team works in mutual effort to achieve the goal.
- 4. Unrealistic expectations** – In Scrum, the users and the team are in constant communication and there is information spread and progress awareness during the project time. In Scrum, the team size is also small and small teams tend to work more independent and more efficient than the teams in large projects (Rising and Janoff, 2000). In such teams the communication is open, transparent and set on daily basis making it easier for all stakeholders to be up-to-date with the project progress and project delivery.
- 5. Lack of executive support** – Executive support in Scrum concept takes part in several project situations. As such, executive support is included in setting list of prioritization for product backlog, during sprint review and sprint retrospective.

- 6. Changing requirements & specifications** – Iterations in Scrum enable update of product backlog list and inclusion of the modifications and updates in some following sprint. The flexibility of the methodology allows organizations to modify the project and its deliverables at any time, delivering at the end the most appropriate release. In the same time, this methodology offers mechanisms of controlling and improving the project performance, which is essential in the management of changes. Scrum's characteristics make this project management methodology very suitable for software development projects as these projects have many requirements that change rapidly and requirements that are sometimes vague, unclear and not possible to be specific at early stage of the project.
- 7. Lack of planning** – In Scrum, planning phase does not happen only during the project start, but during the whole time of the project. Possibilities for re-planning and adjusting the project baseline are possible to be made after every sprint or iteration.
- 8. Didn't need it any longer** – The risk of terminating the project because of lack of need is mitigated with the Scrum concept, on one hand side as it includes project planning and input regarding the need of the project from all relevant stakeholders, but also on the other hand side as it enables modifications and changes in the requirements that could address the possible need for change in the project.
- 9. Lack of IT Management** – One of the great values in Scrum concept is the self-dedication and empowerment given to the team. Such self-dedication to the tasks in the sprint and daily communication, enables project progress transparency, project tracking and project management, while in the same time avoiding the need for a project manager that will simply dedicate tasks, without the IT knowledge and understanding.
- 10. Technology illiteracy** – The Scrum team (3-9 members) are all IT oriented people whose task is to deliver the user requirements set with mutual cooperation. The team members support each other in the work, share their experience, and undertake tasks that fit his/hers technology knowledge, but can support other areas as well. Furthermore, the team meets daily and shortly discusses impediments on the way allowing other team members to offer their support by suggesting proposal solutions and ideas for improvement.

Taking this in consideration, the Scrum methodology if applied as conceptualized, addresses the needs for software project management and successful software project realization, supporting the first hypothesis of this paper and providing answer to our first research question.

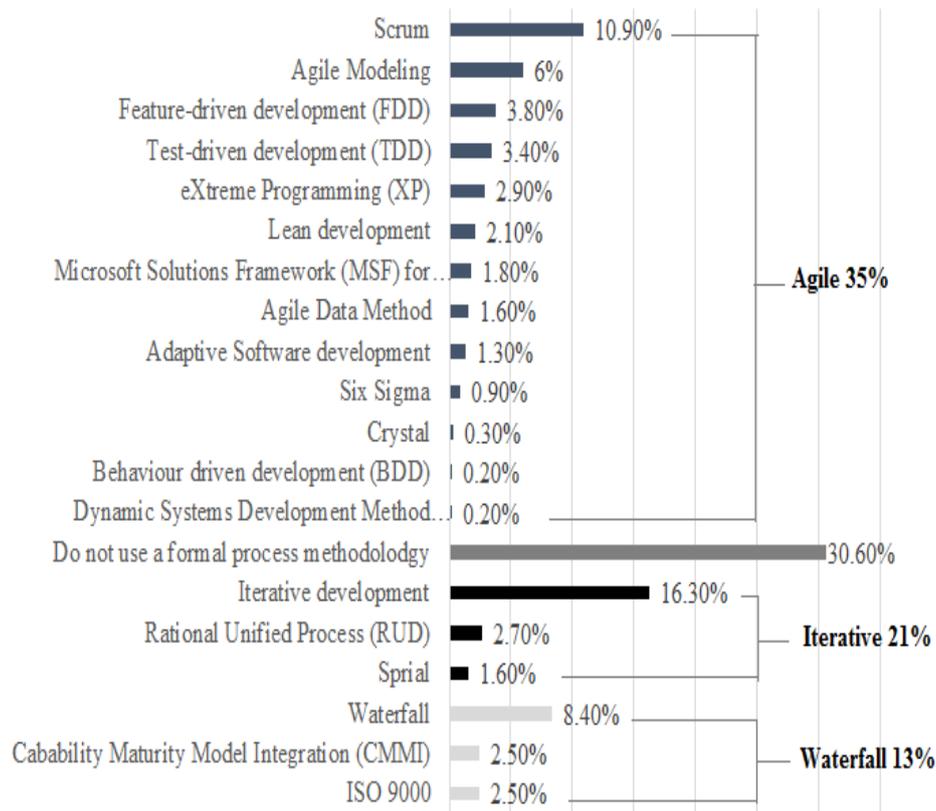
3.3. Adopting Scrum and the Benefits of this Project Management Framework

Scrum has found wide acceptance in well-known and world recognized companies. Google has applied Scrum for developing one of the most profitable software products till now - Google AdWords. Also, the immense project with more than million lines of code, SirsiDynix, has been implemented with Scrum. Furthermore, Yahoo! and many other have done various projects under the Scrum Framework (Sutherland and Schwaber, 2011). But at the same time, Scrum has been adopted in many small entrepreneurial companies from where in the first place Scrum originated.

Coming to our second research question “What is the adoption of Scrum like in comparison to other project management frameworks and methodologies?”, we use the research study conducted in 2009 by Forrester and D-r. Dobb, which included 1298 IT professionals.

As the graphic in Figure 3 demonstrates, Scrum is the method most widely used from agile project management methodologies with 10.9% adoption. Also, agile methodologies have the widest spread adoption of 35% compared to other project management concepts and frameworks, like iterative and waterfall. Furthermore, 30.60% of the respondents do not use a formal process methodology which leads to lack of planning, lack of control points and monitoring, that are essential for project management way of work in meeting the time, budget and scope of work constraints. Thus far, as the research results validate, we can confirm and accept our first hypothesis of this paper in which Scrum if applied, leads to improved IT project management and ultimately to an increased number of IT Project successes.

Figure 3: Agile Adoption Compared to other Methodologies



Source: Forrester, (2009), Dr.Dobb's Global Developer Technographics Survey, [<http://www.softwareconsortium.com/common-sense-agility.html>] (15.09.2014)

4. Software Project Management in the Republic of Macedonia

Several papers discuss the usage of the agile software development methodology in the Macedonian companies (Dalipi et al., 2013; Stankovik et al., 2013). However, the most comprehensive study regarding the common project management methodologies and how they are transferred to project success, as well as the presence of Scrum in the Macedonian IT companies, was presented in a research by Stojkovska (2013).

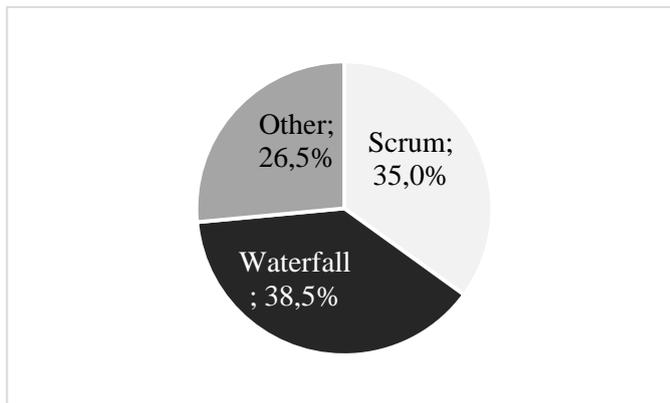
The survey itself was conducted among 63 IT companies in Macedonia out of which 31 responded to the survey. The respondents were partially IT project managers (25 respondents) and partially team

members/programmers (20 respondents). Following the principle in this paper, we firstly identified the success rate of the software projects from the sample in the questionnaire. In this regard, 56 % of the projects were successful projects (project success), 44% were projects that were completed, but either beyond the budget approved, or longer than planned and/or not fulfilling some of the functionalities initially specified in the scope of the project (project challenged), and no project was categorized as project impaired.

In regards to usage of the project management frameworks for software development, the dispersion of the results shows that 76% of the companies use some project management methodology. Also, out of the 56% of the projects realized with project success, 79% use some project management methodology and out of 44% of the project realized with partial success, 64% use some project management methodology, which speaks positively in regards the correlation of project management approach used and project success endured.

From the companies that use project management framework, the dispersion of usage of methodologies is shown in Figure 4. As it can be seen, waterfall is still the most widely used methodology for software development with 38.5%, followed by Scrum with 35% and other methodologies with 26.5%.

Figure 4: Project Management Frameworks for Software Development



Source: Stojkovska Nikcevska, S. (2013), *“The Influence of the Software Development Methodologies on the Efficiency of the IT projects”*, p.101-108.

We can confirm that Scrum is used as a project management framework among the IT companies in Macedonia. However, many companies still use the waterfall method, which is in some respect an obsolete methodology for software project management. However, according to the survey, 19% of the respondents believe that agile methods are more convenient for software development, while only 6% of the respondents prefer the waterfall method in software development. The majority of the respondents think that the type of the project determines the method. According to the opinion of the sample in question, 70% of the respondents have answered that the involvement of project management framework affects positively the project success and is essential contributor to this result whereas 15% believe that it complicates the process and way of work.

According to the results in the survey presented in Stojkovska (2013), we can confirm our second hypothesis in this paper and conclude that it is a common practice for companies in Macedonia to use project management methodologies in managing their software projects as 76% have positively confirmed that they use project management framework in the software project management they undertake. In regards of the research questions RQ2.1 “What are the most common project management methods for managing software projects in Macedonia” both Waterfall and Scrum are used and applied for software development projects, Scrum with 35% of participation, but mostly the answer is that it depends of the project undertaken. In that manner we can confirm our RQ2.2 “Is Scrum used among the IT companies in Macedonia and is this project management framework perceived as suitable for managing software projects?” as the results confirm that Scrum is used among the IT companies in Macedonia as project management methodology. Furthermore, we can also relate this methodology as suitable for software development projects because in this survey agile methodologies were found more convenient for software development than the Waterfall method. Finally, responding to our RQ2.3 “Can Scrum be correlated as a driver of success in IT projects?” we have also identified a positive correlation between companies that have results of project success and that they use project management framework among which Scrum and Agile methods have been proclaimed as more convenient for software development undertakings.

Conclusion

The goal of this research was to acknowledge if and how Scrum as a framework for managing software projects influences improving project success rate. For this reason, we have analyzed the concept of Scrum. We have also correlated the reasons for project failure with how Scrum contributes in their overpowering, which helps support the hypothesis set in this paper that the application of Scrum leads to successful project realization and improvement of the project success rate. Our second hypothesis and research is focused on software development projects in Macedonia. We have looked into the success/failure rate of these projects and the relevance of the project management methodologies and Scrum regarding project successes. According to the survey of Stojkovska (2013) we can conclude that most of the companies use some project management methodology (76% have responded affirmative). In this regard, Scrum (35% have responded that they use Scrum) together with the Waterfall method (38.5% have responded that they use Waterfall) is among the methodologies that are used most widely in this area in the sample taken into consideration. Scrum and the agile methods have been recognized as most suitable for software development projects, due to the flexibility and responsiveness it employs.

Literature:

1. Biafore, B. (2011), "*Successful Project Management*", O Reilly Media, Inc., p.6.
2. Brewer, J., Dittman, K.C. (2009), "*Methods of IT Project Management*", Prentice Hall, p.38.
3. Charette R. (2005), "*Why Software Fails*", IEEE Spectrum, [<http://spectrum.ieee.org/computing/software/why-software-fails>] (05.09.2014)
4. Dalipi, F., Rufati, E., Idrizi, F. (2013), "Applying Agile Software Development Methodology in a Dynamic Business Environment", *International Journal of Engineering Research and Development*, Vol.8, Issue 3, pp 20-25.
5. Dyba T., Dingsoyr T. (2008), "Empirical Studies of Agile Software development: A systematic Review", *Information and Software Technology* Vol.50 pp.833-859.

6. Eveleens, L., Verhoev C. (2010), “*The Rise and Fall of the CHAOS Report Figures*”, IEEE Software, p.31.
7. Forrester, Dr.Dobb's, (2009), Global Developer Technographics Survey,
[<http://www.softwareconsortium.com/common-sense-agility.html>] (15.09.2014)
8. Glass, R. (2006), “The Standish Report: Does It Really Describe a Software Crisis?” in *Comm. ACM*, Vol.49, No.8, pp.15–16.
9. IPMA, [<http://ipma.ch>] (01.09.2014)
10. IT Cortex, (2002),
[http://www.it-cortex.com/Stat_Failure_Rate.htm#The%20Chaos%20Report%20%281995%29] (10.09.2014)
11. Jorgensen, M., Molokken, K. J. (2006), “How Large Are Software Cost Overruns? A Review of the 1994 Chaos Report”, in *Information and Software Technology*, Vol.48, No.8, pp.297–301.
12. Kerzner, H. (2009), “*Project Management, A System Approach to Planning, Scheduling and Controlling*”, 17th edition, John Wiley and Sons, p.64.
13. PMI, (2013), “*Project Management Body of Knowledge, PMBOK*,” 5th edition, PMI Inc., p.3-4.
14. PMI, [<http://www.pmi.org>] (01.09.2014)
15. Rising, L., Janoff, N.S. (2000), “*The Scrum software development process for small teams*”, IEEE Software, 26-32.
16. Schwaber, K. (1995), “*Scrum Development Process, Advanced Development Methods*”, Inc., p.68.
17. Shelly, G., Rosenblatt, H. (2010), “*Systems Analysis and Design, Course Technology*”, Cengage Learning, p.20.
18. Standish Group Report, (1995), “*CHAOS*”, p.2,
[<http://www.projectsmart.co.uk/docs/chaos-report.pdf>]
(08.09.2014)
19. Standish Group, (2012), “*The CHAOS Manifesto 2012*”, pp.3-25.
20. Standish Group, (2013), “*The CHAOS Manifesto 2013*”, p.1,
[<http://www.versionone.com/assets/img/files/CHAOSManifesto2013.pdf>] (15.09.2014)

21. Stankovik, D., Nikolik, V., Djordjevik, M. (2013), "A Survey Study of Critical Success Factors in Agile Software Projects in Former Yugoslavia IT Companies", *Journal of Systems and Software*, Vol.86 Issue 6, pp.1663-1678.
22. Stepanek, G. (2005), "*Software Project Secrets, Why Software Projects Fail*", New York, NY:APRESS, p.8.
23. Stojkovska Nikcevska, S. (2013), "*The Influence of the Software Development Methodologies on the Efficiency of the IT projects*", Ss. Cyril and Methodius University in Skopje, Faculty of Economics.
24. Sutherland, J. (2004), "Agile Development:Lessons Learned from the first Scrum", p.1, [<http://www.scrumalliance.org/resources/35>] (18.09.2014)
25. Sutherland J., Schwaber, K. (2011), "*Scrum Papers*", [http://scruminc.com/tl_files/scrum_inc/documents/ScrumPapers.pdf] (10.09.2014)
26. Takeuchi, H., Nonaka I. (1986), "*The New Product Development Game*", *Harvard Business Review*, p.138.
27. Zvegintzov, N. (1998), "Frequently Begged Questions and How to Answer Them", *IEEE Software*, Vol.20, No.2, pp 93–96.