http://jates.org

Journal of Applied Technical and Educational Sciences jATES

ISSN 2560-5429

JATES Journal of Applied Technical and Educational Sciences

Comparative analysis of agile and traditional methodologies in IT project management

Maja Gaborov[a], Dijana Karuović[b], Mila Kavalić[c], Dragica Radosav[č], Dragana Milosavljev[ć], Sanja Stanisaljev[d], Jozef Bushati[dž]

[a]*University of Novi Sad, Technical faculty "Mihajlo Pupin" Zrenjanin, Serbia,maja.gaborov@tfzr.rs*

[b]*University of Novi Sad, Technical faculty "Mihajlo Pupin" Zrenjanin, Serbia,dijanakaruovic@gmail.com*

[b]*University of Novi Sad, Faculty of Technical Science, Novi Sad, Serbia,milakavalic@gmail.com (Corresponding)*

[č]*University of Novi Sad, Technical faculty "Mihajlo Pupin" Zrenjanin, Serbia, aronradosav@gmail.com*

[ć]*University of Novi Sad, Technical faculty "Mihajlo Pupin" Zrenjanin, Serbia, dragana.milosavljev@tfzr.rs*

[d]*University of Novi Sad, Technical faculty "Mihajlo Pupin" Zrenjanin, Serbia, sanja84stanisavljev@tfzr.rs*

[dž]*University of Shkodra "Luigj Gurakuqi", Albania, jozefbushati@gmail.com*

Abstract

It is known that initially, during software development, the traditional methodology was used for all IT projects and projects were often unsuccessful due to the rapid growth of the IT industry, and then agile methodologies began to be developed. Traditional methods have some advantages over agile methodologies, and mostly the most common traditional methodology is Waterfall. Given that it has limitations in handling problems, such as unstructured code, team morale, poor visibility, lack of communication between stakeholders and frequent prioritization of user requirements, it would not be bad to use agile methodologies, which focus on working with users, continuous testing, refactoring and incremental development. This paper deals with the comparison of methodologies in IT project management based on other scientific research. Some of the mentioned methodologies are Scrum, Kanban, Waterfall, etc. It was concluded that the agile Scrum methodology is mostly used in IT companies, with the combination of several methodologies often appearing, due to the need for projects, in order to eliminate the shortcomings of each methodology.

Keywords: agile; traditional; IT project; management; Scrum; Waterfall; Kanban;

1. Introduction

In order for projects to be successful in software engineering, good organization is needed. Projects are more complex, so it is necessary to know some formal project management process. It is important to manage the project in order to use resources efficiently, to have an even division of labor, to form a plan that is implemented on time, with a predictable way of execution and good and expected results.

© jATES: Journal of Applied Technical and Educational Sciences

Projects are used in various industries, for example in construction, technology, materials and the like. Project management is important in the information technology industry where new user requirements change very quickly. In order for projects to succeed and to be managed in the right way, the most important thing is to take care that the projects are as efficient as possible, that they are implemented in a certain period of time with available resources and within a certain budget. Methodologies in IT project management can be traditional and agile, and the choice of some of them depends on the characteristics and nature of a particular project (Vresk, 2020). Agile project management methods have been used for several years, mostly in the IT profession. When deciding which project management method to use, the needs of the stakeholders, the risks associated with the project, the size of the project, the costs and, of course, the complexity of the project must be taken into account (Vladimirovich Orlov, et al., 2021). This paper deals with the comparison of traditional and agile approaches based on scientific literature and other studies and also comes to the conclusion which methodology is most used in practice.

2. Research of methodology

2.1 Subject and problem of research

Since methodologies are important in project management, it is very important to decide on the appropriate methodology and therefore the subject of research is to consider methodologies in IT project management and select the most appropriate. The problem in this research is the choice of methodology that would best suit the nature of a particular project. There are many reasons to keep in mind when choosing, such as project duration, client requirements, budget, etc. Managers must know the methodologies and know the benefits of each methodology. The choice of methodology ensures the success of the project. A systematic review of the literature can help project managers make this choice.

2.*2 Research questions and Research goal*

The aim of this research is to determine which methodology is most applied in practice, what are the reasons and which combinations of existing methodologies are used. It can be presented by the following research questions:

- RQ1: Which are the reasons for choosing agile and traditional methodologies?
- RQ2: Which methodology is most used?

- RQ3: If combinations of multiple methodologies are used, which are those combinations and what are the reasons for that?
- RQ4: What are the differences between Scrum, XP and Kanban?

3. Theoretical Consideration

"Project management is the application of tools, skills, techniques and knowledge in project activities to meet project objectives or client requirements and their default scope" (Vresk, 2020).There are several project management methodologies and some of them will be presented in this section.

3.1 Traditional methodology

Traditional methods are applied in organizations in which fully defined projects are implemented and plans are prepared in advance that aim to meet time, budget and project goals (Vresk, 2020; Shenhar, & Dvir, 2007). Management itself is based on commands and controls, communication is formal, the development model is based on a life cycle model. It is a matter of complete planning, and the examination of the results comes at the end (Vresk, 2020; Engelhardt, 2019). In the traditional method, user requirements are clearly defined at the beginning of the project, and clients state their requirements at the very beginning of the project. The teams are big. Projects take a long time (Vresk, 2020).

3.2 Waterfall method

The waterfall model was introduced by Royce in 1970 (Cocco, et al., 2011). It is the most common method of traditional methods. The phases are as follows: requirements specification, design, implementation / development, testing and maintenance (McCormick, 2012). The project IT team spends a lot of time on the requirements specification phase, i.e planning and design, and when the implementation / development phase starts, there are no additional requirements, doubts or undefined parts. project or final result. For this reason, long-term projects are suitable for access to the waterfall (Vresk, 2020; McCormick, 2012; Bhavsar, et al., 2020;Andrei, et al., 2019).

3.3 Prince2

Prince2 is a traditional method called "Process driven" method that answers 'what' questions and ‘why’, and in small quantities to the question of ‘how’. PRINCE2 developed by the UK

government and more recently are the rights to the methodology in a public-private company (Vresk, 2020).

3.4 Critical Path Method (CPM)

"The critical path method (CPM) is based on a logical, mathematical model of the project, the basis of which lies in the optimal time required for an individual process in the project and the simultaneous exploitation of the most economical available resources" (Antill & Woodhead, 1991). Its greatest advantage is in determining the sequence that does not have a time line reserve (critical path) and in recognizing those sequences of activities that have them (Vresk, 2020).

3.5 Agile methodology

The Methods of agile software project management are guides for planning and control thereof (Parker & del Monte, 2014). Some of the characteristics of IT projects have conditioned the need to create new methodologies for efficient work on programming and software development. Traditional approaches have not always been appropriate mostly because these are projects that usually start without firm and unchanging specifications because frequent changes are usually required, which also requires a different and more flexible approach. The client is often unsure of what exactly he expects as a result, and through requests for change he goes towards his goal. The client sometimes wants the process to go back to the previous stage and certain changes to be made. The team working on software development should be ready for constant changes and close cooperation with the client (Islam, 2013). Team cooperation, good communication between team members as well as good communication with the client are important in order to harmonize the wishes of the clients and the possibilities for achieving results (Ahmad, et al., 2016). Agile software development methodologies emerged in the late twentieth century where teamwork is key (Ahmad, et al., 2016; Milošević, 2018; Islam, 2013). The term "agile" means the ability of a method to respond to frequent changes in requirements. As the client has an insight into the prototype, he is able to define the requirements and explain to the team what his requirements really are. With the agile method, the emphasis is on changes that can be constant and thus increase the success of the project. It is better to make changes periodically as needed because it is cheaper, than to make changes when the project is already completed (Ahmad, et al., 2016; Milošević, 2018; Islam, 2013; Vresk, 2020). In agile methodologies, the project can move on the basis of one idea, and in the end result in something different from the original idea (Vresk, 2020; Islam, 2013). Agile/Scrum and Extreme

Programming (XP) are being widely used in companies to accomplish software development projects (Barash, 2013; Garcia, et al., 2020; Venkatachalam, et al., 2017)

3.5.1 Scrum

Scrum is an iterative-incremental process and Scrum is the most common agile (Mahnič, 2015; Schwaber, 2004; Cocco, et al., 2011), approach (Stellman & Greene, 2017;Lei, et al., 2017; Granulo, & Tanović, 2019; Ingason, et al., 2013). The process contains a set of managerial recommendations, but does not define the activities of the development process itself. It is often used in combination with other software development processes. Scrum measures the output of the future system after each iteration.

Roles in the Scrum methodology are product owner, development team, scrum master, manager, client (Pantelić, et al., 2020; Schwaber, & Beedle 2001; Laanti, 2013; Albarqi & Qureshi, 2018; Fustik, 2017). The product owner (Terlecka, 2012; Venkatachalam, et al., 2017), collects inputs from customers, end users and development team members then convert them into requirements and evaluates them in terms of development priorities. He is responsible for product development and delivery according to customer requirements. The development team usually consists of five to ten members, some of whom are analysts, developers, designers and testers. The team has autonomy in decision-making, as well as the freedom to provide the product owner with ideas for product improvement. The Scrum Master should establish mediations between product owners and members of the development team. Responsible for the successful development of the final product and is responsible for the successful implementation of the Scrum method on the project, providing continuous assistance and support to members of the development team. The manager is responsible for the final decision and participates in the process of setting goals and defining requirements. The client participates in the process of generating requests and defining the functionality that the future system should have and participates in the process of checking the obtained results and functionality, providing feedback to the development team.

Software development work takes place in shorter cycles called sprints. After that, it is necessary to continuously consult with the client, and after a certain cycle, an analysis and review is performed, as well as any desired and necessary changes. Meetings are mandatory before and after each sprint to determine if everything has been done as the client requests and to determine if anything needs to change. The sprint can last up to a month (Ferrão & Canedo, 2015; Brezočnik & Majer, 2016), and the result is an executive product (Pantelić, et al., 2020;

Schwaber & Sutherland, 2020; Fustik, 2017). When planning a sprint, it is specified what needs to be done within the sprint (Bhavsar, et al., 2020), i.e. what will be the result of the product and how that result will be achieved. Daily meetings need to be held and they can last about fifteen minutes. Each member of the team is obliged to submit a report by tomorrow's daily meeting on the problems he noticed, on the work done the previous day and on the results he plans to achieve (Pantelić, et al., 2020; Abrahamsson, et al., 2002). The term Backlog is an indispensable part of Scrum. There are Product Backlog and Sprint Backlog. The Product Backlog is a list of all the characteristics of the end result that the team needs to develop in order for the result to be satisfactory. The Sprint Backlog is a set of all the features provided by the product owner along with the team that decided to develop in the Sprint that follows. The characteristics themselves are taken from the product backlog, from which the functions are taken over by priority (Vresk, 2020; Stellman & Greene, 2017). After a couple of years, the second author (R&D Operations Manager in SI) and third author (CTO) felt that Scrum was too rigid, did not scale, and was unsuitable for maintenance tasks (Sjøberg, et al., 2012).

3.5.2 Kanban

Kanban entered software development in 2004. Kanban is the second most famous agile method (Moonden, 2011). Kanban is a method based on the term ‘just-in-time’ (Ahmad, et al., 2013). The idea for this method came from three Toyota engineers (Ahmad, et al., 2018; Andrei, et al., 2019), Kichiro Toyota, Eiji Toyoda and Taichi Ohno in the 1940s (Womack, et al., 1990; Stellman & Greene, 2017; Moonden, 2011). "Kanban has five basic principles: visualize workflow, limit workflow, measure and manage workflow, make process policies explicit, and use models to identify improvements and opportunities" (Ahmad, et al., 2016). Sjøberg, Johnsen and Solberg (2012) compared the use of Scrum and Kanban in a medium-sized firm over a two-year period. Their findings show that with Kanban the number of weighted errors was reduced by 10% and 11%, respectively, and productivity was improved by 21% for implemented functions. Kanban focuses on communication, collaboration, and integration between software developers, testers, and support teams, resulting in rapid software development and continuous delivery to the client. Kanban in software development brings visibility to work and improves work efficiency and throughput. Industrial practice has shown that Kanban can reduce maintenance and development costs (Ahmad, et al., 2016). The basic idea is that each organization implements the basic principles of Kanban in its own way, through continuous learning from its own experiences. For this reason, there are no well-founded guidelines for their effective implementation. Kanban philosophy is focused on reducing

overall costs, improving the overall quality and quality of products delivered to customers, shortening product delivery times and increasing customer satisfaction. One of the more important ideas of the Kanban method is to eliminate excess. This is achieved by using Kanban cards and Kanban boards to visualize how resources move through the production cycle. This allows all participants to be fully involved in the process and helps managers to achieve a surplus or deficit in production. Kanban allows organizations to start with their existing workflow, slowly introducing changes over time to reach the desired level and achieve the desired results. This can also be achieved by limiting work in progress (Work In Progress - WIP). The activities encouraged are visualization, work in progress, workflow management, clearly defined rules and feedback. Kanban systems use mechanisms such as the Kanban board to visualize the work and the process that the work goes through. Several indicators of effective visualization have been defined. Some of them are the point of obligation and the point of delivery. The point of obligation arises when the work team agrees to perform a certain work task (Pantelić, et al., 2020). The delivery point is when the team delivers the work item to the customer. Then, effective visualization is reflected in policies that determine which work should be at a particular stage, and also in restrictions on work in progress. When the limits on the amount of work in progress are determined system and use those limits to if you knew when to start new process, workflow can be reduced and time reduced, improve quality and deliver more often. The workflow should maximize value delivery, minimize wasted time, and be as predictable as possible. Teams use empirical control through transparency, inspection and adaptation to balance potentially conflicting goals and how they would not come at all. Clear rules help explain the whole process and the different stages during the process. Rules should be simple, well defined, visible, always applicable and such that can be easily changed. Feedback is an essential element in any a system that seeks evolutionary change (Kniberg & Skarin, 2010; Cocco, et al., 2011; Senapathi & Drury-Grogan, 2021; Santos, et al., 2018).

3.5.3 Differences between Scrum and Kanban

Both Scrum and Kanban belong to the Agile and Lean methods, where the focus is to respond quickly to customer requests. Both are highly adaptable and based on highly collaborative and self-managing teams. Kniberg and Skarin (2010) said that Scrum is more prescribed than Kanban. Scrum prescribes the use of time iterations, Kanban does not. There is a difference between Scrum and Kanban plates. Scrum has two modes: scheduling mode and operating mode. The scrum board is used by teams to plan work before it actually starts. In planning mode, teams share their work in sprints and assign story points to user stories to help plan which

story goes into which sprint. In contrast, Kanban has no plan regime. Kanban uses the same column-based interface as Scrum to track item status without organizing work into sprints (Ahmad, et al., 2016). Sahota explains that Kanban can handle a large number of interruptions in the project, support staff with specialized roles and different sets of skills. Kanban works well for larger teams because the costs of communication and planning are lower. Scrum is better at projects that require deep collaboration and innovation and works best with small cross-functional teams and encourages general experts (Lei, et al., 2017; Ingason, et al., 2013; Ahmad, et al., 2014; Nikitina & Kajko-Mattsson, 2011; Al-Baik & Miller, 2015; Sjøberg, et al., 2012; Ahmad, et al., 2016; Fagarasan, 2021). Many companies previously using Scrum are adopting Kanban due to its traits including transparency and limited work in progress (Ahmad, et al., 2014; Nikitina & Kajko-Mattsson, 2011; Al-Baik, & Miller, 2015; Sjøberg, et al., 2012; Ahmad, et al., 2016; Shafiq & Inayat, 2017).

3.5.4 Extreme programming (XP)

With the Extreme programming method, the focus is on producing better quality software and on more productive work of the development team. It was developed by Kent Beck and Ward Channing in the mid-1980s. Beck added new principles and ways of applying to projects with the application of the most important items, namely communication, simplicity, feedback and courage (Pap, 2008). It is specific because it is used specifically for software development (Beck & Andres, 2004). The principle of this method is that the client is a member of the team and he defines the goals and priorities within the user units, in constant cooperation with the team members in order for the development team to better understand the user requirements. Development goes through short cycles that include the current iteration plan. The success of the development process is measured by observing progress. All details of user units are documented in the form of tests, to facilitate monitoring of progress and implemented requirements. Productive code should be written by two programmers on one development unit. One member writes the code while others check it and thus follow the implementation process. It is desirable that roles change frequently in order to maintain a quality relationship among members. The productive code is written to satisfy the test code. Developed software is the collective property of team members. The essence of planning is in the division of responsibilities between the user and the development team. The user decides which functionalities the software should implement, and how much it costs. The system should be designed as simple as possible with frequent refactoring. The biggest advantage of XP is that it allows software development companies to save costs, frustrations and time by eliminating

unproductive activities. It primarily seeks to reduce the risks associated with project failure, and this allows developers to focus on coding. The construction of the XP software solution process is realized through several development phases: research, planning, iteration implementation, production, maintenance and completion (Marić, 2015; Tadić, 2005; Malik, et al., 2019; Pap, 2008; Subih, et al., 2019).

According to the author (Malik, et al., 2019), the flexibility of agile methods is the most important characteristic, whether it is Scrum, Extreme programming or other methods. They state that one of the problems of agile methods is that complex projects, parts of agile methods, such as meetings or phone calls, are necessary and which can be a problem in international teams that have different time zones. Extreme programming puts more of its focus on software development or best practices for its development, while setting aside, or less priority, best practices in how to implement the entire project within a given budget and within defined deadlines (Javanmard & Alian, 2015). To facilitate this application in larger projects of complexity, it is recommended to use project management software where, according to the authors (Fabac, et al., 2009), divisions into three categories of project management software, process management software and software time tracking are possible (Vresk, 2020).

4. Empirical analysis and discussion

Based on research in one IT company, it was identified that respondents choose traditional methods when they know exactly what needs to be done during the project, while agile methods are more often chosen when only "rough" requirements and goals are known. Traditional methods are chosen even when the project cannot be divided into smaller parts, while agile methodologies are chosen when flexibility is needed. It is important to emphasize that traditional methods are chosen when the project client does not have the technical people to check certain parts and is only interested in the final product, while agile methods are chosen when changes are expected during the project and are common (Vresk, 2020). Livermore's research (Livermore, 2008), shows that there was no significant correlation between team size and the success of the implementation of methods in relation to them. He states that this is an unexpected result, considering that a large number of researches on the mentioned topic conclude that agile methods are less successful over larger teams. These results support the fact that the methods, especially agile, are adaptable to different teams and situations. Their flexibility is not only applicable here in terms of quick response to additional requests, but they are also flexible in adapting to a team or project - team size, team role, sprint length (Vresk,

2020). Agile methodologies are better than traditional ones because they can save money, time and deliver quality products on time (Venkatachalam, et al., 2017). According to the authors (Merzouk, et al., 2018), and their comparison of agile methods, when choosing the correct agile method for a project, it is necessary to observe the size of the team, the project (Vresk, 2020). In agile methodologies, user requirements are not clearly defined, only the end result is defined. Teams are smaller than with traditional methodology. The client is involved in the whole project from start to finish. The key is to have a team that works together. Communication is informal. Agile methodologies are an iterative model (Vresk, 2020). In 2010 Forrester (Womack, et al., 1990), reported results of their Global Developer Technographics Survey, which revealed that 35% of respondents used an agile, 21% an iterative, and 13% a waterfall development process, while 31% did not use a formal process methodology (Mahnič, 2014).

According to Rising and Janoff, (2000) and Schwaber (2004), the most widespread agile method is Scrum. Pure Scrum is reported to be used by 54%, Scrum/XP Hybrid by 11%, and Scrumban by 7% of respondents. However, one survey also revealed a rapid growth of the number of Kanban users. Compared to 2011, Kanban and Kanban variants (Anderson, 2010; Ladas, 2008), nearly doubled in 2012, mostly due to an uptick in Scrumban use (Mahnič, 2014).

As each method has its own characteristics and disadvantages, a stand-alone frame of Scrum or Kanban or Waterfall cannot provide complete solutions to all challenges. Agile frameworks are very suitable for large or medium range software project, where the project requirement is also his Feasibility often changes over time, while waterfall is suitable for short-term projects and a clear requirement in the initial phase of the project, can be predicts that project requirements may not change during its life cycle until the final delivery of the project. The integration of Scrum and Kanban with Waterfall provides a great strength to Software Engineering Management (SEM) practices in the form Scrumbanfall. Scrumbanfall artifacts are the basis of the framework in customer value form, documentation, workflow management, transparency, augmentation and opportunity for Scrumbanfall team and other stakeholders (Bhavsar, et al., 2020). Cocco et al (2011), developed a system dynamics model for the comparison of means of simulation techniques for prescriptive approach using Scrum and Kanban on Waterfall in 2011, and evaluated that each of them have their own strengths and weaknesses and proposed further research work on the combination of all them to resolve the agile software development issue (Bhavsar, et al., 2020). In 2019, Mohan, Devisree and Kumar combined the rules of Scrum with Kanban flexibility in the form of Scrumban and implemented in government sector software application development that reduced the stress of overhead development and

increased software efficient. Scrum, Kanban and Waterfall into Scrumbanfall which has a great strength compared to stand alone framework and capabilities to answer the challenges of software development and management practices like direct involvement of external stakeholder into project requirement and analysis documentation; project planning, estimation and tracker. Inaccurate estimation invites risks into project and unclear vision about software product, which result into unexpected cost in the project for the software project owners or software development organizations that bear the cost of project development (Bhavsar, et al., 2020). In one paper authors analyzed the dynamic behavior of the adoption of Kanban and Scrum, versus a traditional software development process such as the Waterfall approach. They use a system dynamics model, based on the relationships between system variables, to assess the relative benefits of the studied approaches. Kanban workflow was managed through an effective control mechanism to limit the work in progress and minimize the lead time. One of the advantages of this approach is that the work is better controlled, so that the effects of errors is kept limited. On the contrary, in the Waterfall case often projects may fail to complete due to the difficulty to correct errors, including errors in requirements (Cocco, et al., 2011). In second study, the Kanban Wall Project can facilitate project review and execution and make communication between stakeholders more efficient and effective. The combined use of Scrum and Kanban has been flawless in this project and the new method has been successful from its inception until now (Ingason, et al., 2013). One study showed that the size of the group more adaptable in Kanban and XP than in Scrum, while the WIP (Limiting Work in Progress) size of Scrum is one, two or one month in the sprint, but the assessment of Kanban and XP WIP is relatively low (it can be as little as an hour cluster) (Saleh, et al., 2019; Cohn, 2007). In this regard, Kanban permission organized preconditions from day to day and XP permits organized necessities continuously, while preconditions must be organized based on the length of the run during adoption Scrum . (Kanwal, et al., 2010; Beck & Fowler, 2000). Moreover, in Kanban component the volume is small compared to Scrum and XP but one industry is halved when it moves from Scrum to Kanban in the time frames between proposing the second element and create requests and send them to your user's website (Shalloway, 2011; Raman, 2014). Kanban and XP are also more focused on quality improvement software (Verweij & Maassen, 2011; Sjøberg, et al., 2012; Raman, 2014).

Researches shows that about half of businesses is still using waterfall model, while the other half uses agile and iterative approaches (Holz, 2019). Companies using agile methods, according to data from the tenth annual survey VersionOne, most often opt for Scrum and

Scrum + XP (70%), Scrumban (7%) and Kanban (5%) (Sutherland, 2010). From our selection of agile methods, we removed the Extreme Programming (XP), because its principles are often used in combination with other methods (Scrum, Kanban) (Brezočnik & Majer, 2016). One study showed that Scrum certainly works best in mature companies which have experienced teams that are working on the product or project for more than one year. For companies with continuous production who need a rapid response to changes and product teams that are working in support and maintenance of the product, we suggest using Kanban. Scrumban is best for young, small companies, since it contains flexibility of Kanban and basic characteristics of Scrum. But certainly agile methods include a strong component of flexibility. Teams could, regardless of the method chosen, adapt it in a way that will serve their purpose – effective work organization and development of quality products (Brezočnik, & Majer, 2016).

According to Alaidaros, et al. (2018), Scrum is the most followed method and 58% of the respondents practice it among other Agile methods, while more than 39% of the respondents practice Kanban method within their organization. Scrum is the most popular agile method; daily activities are based on the past experiences and not only on theoretical aspects (Mircea, 2019). The most popular methodology used by the candidates in second research was Scrum, with 62.5% of them using Scrum at least once. 28.1% of them used Waterfall and only 25% used Kanban (Andrei, et al., 2019). Respondents that used Scrum reported that they were highly motivated during the development and liked the fact that they were in good synchronization with the rest of the team. However, some of them did not enjoy the high number of meetings and the fact that they induced a routine. Kanban users enjoyed the simplicity and flexibility of the methodology while reporting issues with task prioritization and the amount of time they spent updating the cards. Waterfall users enjoyed the Plan-Driven Development that this methodology encourages and the strict requirements imposed from the beginning but had issues measuring progress (Andrei, et al., 2019). According to the statistics presented in the Annual State of Agile Report (Engelhardt, 2019), developed by Digital.ai, 58% of the software projects that were implemented by using the Agile methodologies also chose Scrum as an implementation method. In contrast, only 7% of the total software projects were implemented using the pure Kanban development method this statistic should not influence a project managers' decision when choosing the suitable implementation method (Fagarasan, et al., 2021). Various studies reported that Kanban method, currently, is the contender among Agile methods because it has numerous advantages that make it performs better than Scrum and other Agile methods in terms of having experience greater consistency in managing software

engineering (SE) projects (Flora, & Chande, 2014; Lei, et al., 2017; Karunanithi, 2016). However, Kanban method has significant lacking in progress monitoring task during development process of software projects. This problem negatively affects the success of software projects because of lags in projects' scheduling that lead to late delivering (Skinner, et al., 2015; Al-Baik & Miller, 2015; Kirovska & Koceski, 2015).

As can be seen in Table 1, agile methodologies are used only when there are rough requirements and a goal, when changes are necessary during the project and where changes are possible throughout the project. This helps save money and deliver a quality product on time. Traditional methodologies are used when all client requirements are known exactly from the beginning. Used by the project client knows exactly what he wants to be his final product. According to the scientific literature, the most agile methodology used is SCRUM, and the traditional one is Waterfall.

Table 1. A review of the reasons for the use of agile and traditional methodologies and a review of the most commonly used methodology

Methodologies in IT Project Management	AGILE METHODOLOGY	TRADITIONAL METHODOLOGY
Selection of methodology	They are used when only rough requirements and goals are known and when the necessary flexibility and changes during the project are expected as they can save money, time and deliver products on time.	There are used when it is known exactly what needs to be done throughout the project and when the project cannot be divided into smaller parts and when the project client is only interested in the final product and the communication in the team is informal.
Mostly applied	Scrum (Mircea, 2019)	Waterfall (Mircea, 2019; (Andrei, et al., 2019).)

Given that each methodology has some shortcomings, it is necessary to combine several methodologies in some cases when the nature of the project requires. Scrum and Kanban are suitable for large or medium scale software projects where requirements change over the life of the project. The waterfall is suitable for short-term projects and the requirements are known from the beginning of the project. Scrum certainly works best in mature companies which have experienced teams that are working on the product or project for more than one year. For companies with continuous production who need a rapid response to changes and product teams that are working in support and maintenance of the product, it necessary using Kanban. Scrumban is best for young, small companies, since it contains flexibility of Kanban and basic

characteristics of Scrum. Scrumban can be used as a combination of Scrum and Kanban and the advantage is that the work is better controlled. It can be better than using Waterfall because at Waterfalls often projects cannot be completed due to the difficulty of correcting mistakes. The combination of Scrum, Kanban and Waterfall can make communication between stakeholders more efficient and effective. The combination of these three mentioned methodologies is called Scrumbanfall. Scrumbanfall has great strength to respond to software development challenges in management practices such as direct involvement of external stakeholders in project requirements and inclusion of analysis documentation as well as project planning, evaluation and monitoring. Given that agile methodologies are current today, it would not be bad to see when it is necessary to choose the appropriate one. According to the scientific literature, it can be seen that Scrum is most used, however group size is more adaptable in Kanban and XP than in Scrum, while WIP size of Scrum is one, two or one month in the sprint, but the estimate of Kanban and XP WIP is relatively low. Kanban permission organized preconditions from day to day and XP permits organized necessities continuously, while preconditions must be organized based on the length of the run during adoption Scrum. Kanban and XP are also more focused on quality improvement software.

5. Conclusion

In software development, the role of the client and constant communication with the client is very important, in order to meet his requirements and came to the desired end result. Presenting and comparing these methods is never done works for a reason to find the best method, but to the differences between them were explored and potential reasons found which talk about why and in what situations to choose a particular method or a combination thereof. There is no single recipe for project management and leadership due to the wide range of problems and requirements that arise. There is no best methodology to use. Depending on the nature of the problem, the appropriate methodology will be elected. If the project has clearly defined requirements and goals, some of the traditional methodologies will be used, but for projects with unstable requirements, some will always be used of agile methodologies or combinations, because they are adaptable.

References

Abrahamsson, P., Salo, O., Ronkainen, J., & Warsta, J. (2002). Agile software development methods: Review and analysis. VTT Technical Research Centre of Finland. VTT Publications No. 478. http://www.vtt.fi/inf/pdf/publications/2002/P478.pdf

Ahmad, M. O., Dennehy, D., Conboy, K., & Oivo, M. (2018). Kanban in software engineering: A systematic mapping study. Journal of Systems and Software, 137, 96-113. https://doi.org/10.1016/j.jss.2017.11.045

Ahmad, M. O., Markkula, J., & Oivo, M. (2013). Kanban in Software Development: A Systematic Literature Review. Proceedings of the2013 39th Euromicro Conference Series on Software Engineering and Advanced Applications, 9-16. https://doi.org/10.1109/SEAA.2013.28

Ahmad, M.O., Kuvaja, P., Oivo, M., & Markkula, J. (2016). Transition of Software Maintenance Teams from Scrum to Kanban. 2016 49th Hawaii International Conference on System Sciences (HICSS), 5427-5436. https://10.1109/HICSS.2016.670

Ahmad. M. O., Markkula, J., Oivo, M., & Kuvaja, P. (2014). Usage of Kanban in Software Companies: An empirical study on motivation, benefits and challenges. Proceedings of the Ninth International Conference on Software Engineering Advances (ICSEA 2014), 150-155. ISBN: 978-1-61208-367-4

Alaidaros, H., Omar, M., & Romli, R. (2018). Towards an Improved Software Project Monitoring Task Model of Agile Kanban Method. International Journal of Supply Chain Management (IJSCM), 7(3), 118-125. https://core.ac.uk/download/pdf/230745993.pdf

Al-Baik, O., & Miller, J. (2015). The kanban approach, between agility and leanness: a systematic review. Empirical Software Engineering, 20(6), 1861-1897. https://doi.org/10.1007/s10664-014-9340-x

Albarqi, A. A., & Qureshi, R. (2018). The Proposed L-Scrumban Methodology to Improve the Efficiency of Agile Software Development. International Journal of Information Engineering and Electronic Business, 3, 23-35. https://10.5815/ijieeb.2018.03.04

Anderson, D. J. (2010). Kanban: Successful Evolutionary Change for Your Technology Business. Blue Hole Press. ISBN: 0984521402

Andrei, B-A., Casu-Pop, A-C., Gheorghe, S-C., & Boiangiu, C-A. (2019). A study on using waterfall and agile methods in software project management. Journal Of Information Systems

& Operations Management (JISOM), 13(1), 125-135. http://jisom.rau.ro/Vol.13%20No.1%20-%202019/JISOM-SU19-A12.pdf

Antill, M. J., & Woodhead, W.R. (1991). Critical Path Methods in Construction Practice (4th Edition). John Wiley & Sons. ISBN: 978-0-471-62057-0

Barash, I. (2013). Use of Agile with XP and Kanban Methodologies in the Same Project.PM World Journal, II(X). https://pmworldlibrary.net/wp-content/uploads/2013/10/pmwj15-oct2013-barash-use-of-agile-xp-kanban-same-project-SecondEdition.pdf

Beck, K., & Andres, C. (2004). Extreme Programming Explained: Embrace Change (2nd Edition). Addison-Wesley Professional. ISBN: 0321278658. https://ptgmedia.pearsoncmg.com/images/9780321278654/samplepages/9780321278654.pdf

Beck, K., & Fowler, M. (2000). Planning extreme programming. Addison-Wesley Longman Publishing Co., Inc. ISBN:978-0-201-71091-5

Bhavsar, K., Shah, V., & Gopalan, S. (2020). Scrumbanfall: An Agile Integration of Scrum and Kanban with Waterfall in Software Engineering. International Journal of Innovative Technology and Exploring Engineering (IJITEE), 9(4), 2075-2084. https://10.35940/ijitee.D1437.029420

Brezočnik, L., & Majer, Č. (2016). Comparison of agile methods: Scrum, Kanban, and Scrumban. Proceedings of the 19th International Multiconference Information Society - IS 2016: Collaboration, software and services in information society, 30-33. http://library.ijs.si/Stacks/Proceedings/InformationSociety/2016/IS2016_Volume_C%20-%20CSS.pdf

Cocco L., Mannaro K., Concas G., Marchesi M. (2011). Simulating Kanban and Scrum vs. Waterfall with System Dynamics. Processes in Software Engineering and Extreme Programming. XP 2011. Lecture Notes in Business Information Processing, 117-131. https://doi.org/10.1007/978-3-642-20677-1_9

Cohn, M. (2007, April 6). Differences Between Scrum and Extreme Programming. Mountain Goat Software. https://www.mountaingoatsoftware.com/blog/differences-between-scrum-and-extreme-programming/comments

dos Santos, P. S. M., Beltrão, A. C., de Souza, B. P., & Travassos, G. H. (2018). On the benefits and challenges of using kanban in software engineering: a structured synthesis study.

Journal of Software Engineering Research and Development, 6(1). https://doi.org/10.1186/s40411-018-0057-1

Engelhardt, N. (2019). Comparison of Agile and Traditional Project Management: Simulation of Process models. Acta academica karviniensia, 19(2), 15-27. https://10.25142/aak.2019.011

Fabac, R., Pihir, I., & Radošević, D. (2009). The Application of Modern Methodologies and ICT in the Management of Projects in Croatian Companies. Proceedings of the 1st International Conference on Information Society and Information Technologies, ISIT 2009. https://ssrn.com/abstract=2505743

Fagarasan, C., Popa, O., Pisla, A., & Cristea, C. (2021). Agile, waterfall and iterative approach in information technology projects. Proceedings of the IOP Conference Series: Materials Science and Engineering, 1169(1). https://10.1088/1757-899X/1169/1/012025

Ferrão, S. É. R., & Canedo, E. D. (2015). A study of the applicability of an agile methodology scrum allied to the Kanban method. Proceedings of the 10th Iberian Conference on Information Systems and Technologies (CISTI).http://dx.doi.org/10.1109/CISTI.2015.7170382

Flora, H. K., & Chande, S. V. (2014). A Systematic Study on Agile Software Development Methodologies and Practices. International Journal of Computer Science and Information Technologies (IJCSIT), 5(3), 3626-3637. https://www.researchgate.net/publication/266742256_A_Systematic_Study_on_Agile_Software_Development_Methodlogies_and_Practices

Fustik, V. (2017). Scrum Methodology Compared with Other Methodologies Applied in the Software Development Projects. Proceedings of the International Conference on Information Technologies (InfoTech-2017). http://infotech-bg.com/sites/default/files/2017/A-VF.pdf

Garcia, A. L., da Rocha Miguel, I., Eugênio, J. B., da Silva Vilela, M., & Marcondes, G. A. B. (2020). Scrum-Based Application for Agile Project Management. Journal of Software, 15(4), 106-113. https://10.17706/jsw.15.4.106-113

Granulo, A., & Tanović, A. (2019). Comparison of SCRUM and KANBAN in the Learning Management System implementation process. Proceedings of the 27th Telecommunications forum TELFOR 2019, 636-639. https://doi.org/10.1109/TELFOR48224.2019.8971201

Holz, B. (2019). Results Summary: Agile in the Enterprise. Gartner. https://circle.gartner.com/Portals/2/Resources/pdf/Agile%20in%20the%20Enterprise%202019%20-%20Results%20Summary%20(updated).pdfhttps://doi.org/10.1002/hfm.4530040310

Ingason, H. T., Gestsson, E., & Jonasson, H. I. (2013). The Project Kanban Wall: Combining Kanban and Scrum for Coordinating Software Projects. PM World Journal, II(VIII). https://pmworldlibrary.net/wp-content/uploads/2013/08/pmwj13-aug2013-Ingasson-Gestsson-Jonasson-Project-Kanban-Wall-FeaturedPaper.pdf

Islam, K.A. (2013). Agile Methodology for Developing & Measuring Learning: Training Development for Today's World. AuthorHouse. ISBN: 9781491823880

Javanmard, M. & Alıan, M. (2015). Comparison between Agile and Traditional software development methodologies. Cumhuriyet Science Journal, 36(3), 1386-1394. https://dergipark.org.tr/en/pub/cumuscij/issue/45132/564438

Kanwal, F., Junaid, K., & Fehiem, M. A. (2010). A Hybrid Software Architecture Evaluation Method for FDD - An Agile Process Model. Proceedings of theInternational Conference on Computational Intelligence and Software Engineering (CiSE), 1-5. https://doi.org/10.1109/CISE.2010.5676863

Karunanithi, K. (2016). Metrics in Agile and Kanban , Software Measurement Techniques. Project: Software Measurement. California State University. https://www.researchgate.net/publication/305613018_Metrics_in_Agile_and_Kanban_Software_Measurement_Techniques

Kirovska, N., & Koceski, S. (2015). Usage of Kanban methodology at software development teams. Journal of Applied Economics and Business, 3(3), 25-34. http://www.aebjournal.org/articles/0303/030302.pdf

Kniberg, H., & Skarin, M. (2010). Kanban and Scrum - Making the Most of Both. C4Media Inc. http://www.agileinnovation.eu/wordpress/wp-content/uploads/2010/09/KanbanAndScrum_MakingTheMostOfBoth.pdf

Laanti, M. (2013). Agile and Wellbeing -- Stress, Empowerment, and Performance in Scrum and Kanban Teams. Proceedings of the 46th Hawaii International Conference on System Sciences, 4761-4770. https://doi.org/10.1109/HICSS.2013.74

Ladas, C. (2008). Scrumban-Essays on Kanban Systems for Lean Software Development. Modus Cooperandi Press.

Lei, H., Ganjeizadeh, F., Jayachandran, P. K., & Ozcan, P. (2017). A statistical analysis of the effects of Scrum and Kanban on software development projects. Robotics and Computer-Integrated Manufacturing, 43, 59-67. https://doi.org/10.1016/j.rcim.2015.12.001

Livermore, A. J. (2008). Factors that Significantly Impact the Implementation of an Agile Software Development Methodology. Journal of Software, 3(4), 31-36. http://www.jsoftware.us/vol3/jsw0304-04.pdf

Mahnič, V. (2014). Improving software development through combination of scrum and kanban. Proceedings of the 8th WSEAS International Conference on Computer Engineering and Applications (CEA '14), 281-288. http://www.wseas.us/e-library/conferences/2014/Tenerife/INFORM/INFORM-40.pdf

Mahnič, V. (2015). From Scrum to Kanban: introducing Lean principles to a software engineering capstone course. International Journal of Engineering Education, 31(4), 1106-1116

Malik, R. S., Ahmad, S.S., & Hussain, M.T.H. (2019). A Review of Agile Methodology in IT Projects. Proceedings of the 2nd International Conference on Advanced Computing and Software Engineering (ICACSE-2019), 655-656. http://dx.doi.org/10.2139/ssrn.3351064

Marić, M. (2015). Metodološko-radni okvir za razvoj softverske arhitekture poslovnog softvera u agilnim procesima [A Methodological Framework for Developing the Software Architecture of Business Software in Agile Processes]. [Doctoral dissertation, University of Novi Sad, Faculty of Economy, Subotica]. PQDT Open. https://hdl.handle.net/21.15107/rcub_nardus_5151

McCormick, M. (2012). Watefall vs. Agile Metodology. MPCS, Inc. http://mccormickpcs.com/images/Waterfall_vs_Agile_Methodology.pdf

Merzouk, S., Elhadi, S., Cherkaoui, A., Marzak, A., & Sael, N. (2018). Agile Software Development: Comparative Study. Proceedings of The second International Conference on Smart Application and Data Analysis for Smart Cities (SADASC'18). http://dx.doi.org/10.2139/ssrn.3186323

Milošević, D. (2018 December 2). Agilna metodologija [Agile methodology]. Analitika i vizuelizacija podataka [Data analytics and visualization]. https://dusanmilosevic.com/agilna-metodologija/

Mircea, E. (2019). Project Management using Agile Frameworks. Economy Informatics, 19(1), 34-44. https://10.12948/ei2019.01.04

Mohan, A., Devisree, A. S., & Kumar, C. V. P. (2019). Rationalized Scrumban Based Methodology for Improving the Slackening Pension Scheme of Employee Sectors in India. International Journal of Recent Technology and Engineering (IJRTE), 8(1), 1525-1550. https://www.ijrte.org/wp-content/uploads/papers/v8i1/A9236058119.pdf

Moonden, Y. (2011). Toyota Production System: An Integrated Approach to Just-In-Time (4th Edition). CRC Press, Taylor & Francis Group. https://doi.org/10.1201/b11731

Nikitina, N., & Kajko-Mattsson, M. (2011). Developer-driven big-bang process transition from Scrum to Kanban. Proceedings of the 2nd Workshop on Software Engineering for Sensor Network Applications - SESENA ’11, 159-168. https://doi.org/10.1145/1987875.1987901

Pantelić, O., Nikolić, U., & Krstović, S. (2020). Razvoj softvera za upravljanje projektima zasnovan na različitim agilnim metodologijama [Development of project management software based on diferent agile methodologies]. Časopis za informacionu tehnologiju i multimedijalne sisteme, (71), 38-43 [Journal for information technology and multimedia systems, (71), 38-43].

Pap, J. (2008). Ekstremno programiranje kao metoda agilnog razvoja softvera [Extreme programming as a method of agile software development]. [Graduate thesis, University of Novi Sad, Faculty of Science]. PQDT Open. http://www.robertpapp.info/docs/theses/PappRobert_XP_BSc-thesis.pdf

Parker, M. E. F., & del Monte, Y. F. (2014). The Agile Management of Development Projects of Software Combining Scrum, Kanban and Expert Consultation. Proceedings of the 10th IFIP International Conference on Open Source Systems (OSS), 176-180. https://10.1007/978-3-642-55128-4_25

Raman, S. (2014, April 2). eXtreme Programming The Methodology. InfoQ.https://www.infoq.com/articles/implementing-xp-methodology/

Rising, L., Janoff, N.S. (2000). The Scrum software development process for small teams. IEEE Software, 17(4), 26-32. https://10.1109/52.854065

Saleh, S. M., Huq, S. M., & Rahman, M. A. (2019). Comparative Study within Scrum, Kanban, XP Focused on Their Practices. Proceedings of the International Conference on

Electrical, Computer and Communication Engineering (ECCE), 1-6. https://doi.org/10.1109/ECACE.2019.8679334

Schwaber, K. (2004). Agile Project Management with Scrum. Microsoft Press. ISBN:978-0-7356-1993-7

Schwaber, K., & Sutherland, J. (2020). The Scrum Guide. The Definitive Guide to Scrum: The Rules of the Game. Scrum.org. https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-US.pdf#zoom=100

Schwaber, K., Beedle, M. (2002). Agile Software Development with Scrum. Upper Saddle River, New Jersey: Prentice Hall. ISBN: 0130676349

Senapathi, M., & Drury-Grogan, M. L. (2021). Systems Thinking Approach to Implementing Kanban: A case study. Journal of Software: Evolution and Process, 33(4). https://doi.org/10.1002/smr.2322

Shafiq, S., & Inayat, I. (2017). Towards Studying the Communication Patterns of Kanban Teams: A Research Design. Proceedings of the 2017 IEEE 25th International Requirements Engineering Conference Workshops (REW), 303-306. https://doi.org/10.1109/REW.2017.34

Shalloway, A. (2011). Demystifying Kanban. Cutter IT Journal, 24(3), 12-17.

Shenhar, J. A., & Dvir, D. (2007). Reinventing Project Management: The Diamond Approach To Successful Growth And Innovation. Harvard Business Publishing

Sjøberg, D. I. K., Johnsen, A., Solberg, J. (2012). Quantifying the Effect of Using Kanban versus Scrum: A Case Study. IEEE Software, 29(5), 47-53. https://10.1109/MS.2012.110

Skinner, R., Land, L., Chin, W., & Nelson, R. R. (2015). Reviewing the Past for a Better Future: Reevaluating the IT Project Retrospective. Proceedings of the 10th International Research Workshop on Information Technology Project Management (IRWITPM), 110-119. https://aisel.aisnet.org/cgi/viewcontent.cgi?article=1008&context=irwitpm2015

Stellman, A., & Greene, J. (2017). Head First Agile: A Brain-Friendly Guide. O'Reilly Media, Inc. ISBN: 9781449314330

Subih, M. A., Malik, B. H., Mazhar, I., ul-Hassan, I., Sabir, U., Wakeel, T., Ali, W., Yousaf, A., bin-Ijaz, B., & Nawaz, H. (2019). Comparison of Agile Method and Scrum Method with Software Quality Affecting Factors. International Journal of Advanced Computer Science and Applications (IJACSA), 10(5), 531-535.

https://thesai.org/Downloads/Volume10No5/Paper_69 Comparison_of_Agile_Method_and_Scrum_Method.pdf

Sutherland, J. (2010). Jeff Sutherland's Scrum Handbook. The Scrum Training Institute. https://www.researchgate.net/publication/301685699_Jeff_Sutherland's_Scrum_Handbook

Tadić, B. (2005). Ekstremno programiranje i primjena na Balkanu [Extreme programming and application in the Balkans]. Proceedings of the 4. Naučno-stručni skup sa međunarodnim učešćem ”KVALITET 2005” [4th Scientific-professional conference with international participation "QUALITY 2005"], 237-246. http://www.quality.unze.ba/zbornici/QUALITY%202005/034-Q05-028.pdf

Terlecka, K. (2012). Combining Kanban and Scrum -- Lessons from a Team of Sysadmins. Proceedings of the Agile Conference (AGILE), 99-102. http://dx.doi.org/10.1109/Agile.2012.20

Venkatachalam, A., Rajkumar, S., Selvanathan, N., & Karthick, K. (2017). Agile SCRUM – The Revolution in IT Industry. Proceedings of the International Conference on Intelligent Computing Systems (ICICS 2017 – Dec 15th - 16th 2017), 228-238. http://dx.doi.org/10.2139/ssrn.3122169

Verweij, D., & Massen, O. (2011). Kanban at an Insurance Company in the Netherlands. Cutter IT Journal, 24(3), 18-22.

Vladimirovich Orlov, E., Mikhailovna Rogulenko, T., Alexandrovich Smolyakov, O., Vladimirovna Oshovskaya, N., Ivanovna Zvorykina, T., Grigorevich Rostanets. V., & Petrovna Dyundik, E. (2021). Comparative Analysis of the Use of Kanban and Scrum Methodologies in IT Projects. Universal Journal of Accounting and Finance, 9(4), 693-700. https://10.13189/ujaf.2021.090415

Vresk, A. (2020). Agilne metode za upravljanje projektima [Graduate thesis, University of Zagreb, Faculty of Organization and Informatics]. PQDT Open. https://urn.nsk.hr/urn:nbn:hr:211:105099

Womack, J., Jones, D., & Roos, D. (1990). The machine that changed the world. Rawson Associates.

About Authors

Maja GABOROV is currently a PhD candidate at the Faculty of Technical Sciences, University of Novi Sad. She is employed as a teaching assistant in the field of Information Technology at the Technical Faculty “Mihajlo Pupin” in Zrenjanin. She received her MSc degree at the Technical faculty "Mihajlo Pupin", University of Novi Sad in 2020. Her research interests include the field of information technology.

Dijana KARUOVIĆ is a PhD of science in IT area. She obtained her PhD in Technical sciences at University of Novi Sad, Technical faculty “Mihajlo Pupin” in Zrenjanin, Serbia in 2010. Her major field of study is human-computer interaction in learning management system. Since November 2000, she is employed at the University of Novi Sad, Serbia, at Technical faculty “Mihajlo Pupin” in Zrenjanin as an associate professor.

Mila KAVALIĆ defended hers doctoral dissertation at the Technical Faculty “Mihajlo Pupin” in Zrenjanin, University of Novi Sad, Serbia in 2020. She is currently a PhD candidate at the Faculty of Technical Sciences, Novi Sad. She received her MSc degree at the Technical faculty "Mihajlo Pupin" in 2014 and at Faculty of Technical Sciences, University of Novi Sad in 2017. Her research interests include Knowledge management, Human resources, Organizational behavior and Pedagogical Research.

Dragica RADOSAV is full professor (Department of Information Technology) and Dean at the Technical Faculty “Mihajlo Pupin” in Zrenjanin, University of Novi Sad, Serbia. She defended hers doctoral dissertation at the Faculty of Technical Sciences, University of Novi Sad, Serbia, in 1994. Employed at the Technical Faculty in Zrenjanin since 1983. Her research interests include Methodology of Pedagogical Research, Methodology of Scientific Research, Software Engineering, Information System Design andGeographic Information Systems.

Dragana MILOSAVLJEV is Doctoral student at the Technical Faculty “Mihajlo Pupin” in Zrenjanin, University of Novi Sad. Previously, she completed a master's degree at the Technical Faculty “Mihajlo Pupin” in 2019 and the Faculty of Technical Sciences in Novi Sad in 2020. She is employed as an teaching assistant at the Technical Faculty “Mihajlo Pupin”, in the narrow scientific field of management. Her research interests include Knowledge Management, Organizational Culture, Public Relations and Risk Management.

Sanja STANISAVLJEV received her PhD in 2017 at the Faculty of Technical Sciences in Novi Sad. She is employed at the University of Novi Sad, at the Technical Faculty "Mihajlo Pupin" in Zrenjanin as professor at the Department of Management. Her areas of interest are

management, organization of work, production and management, management of technological development.

Jozef **BUSHATI** is full professor at University of Shkodra "Luigj Gurakuqi", Albania. Holds a PhD in Information and Communication Technology and MBA (Faculty of Economy, University of Tirana Albania &University of Nebraska Lincoln United States) and Bsc at High Pedagogical Institute of Shkodra /today University of Shkodra "Luigj Gurakuqi" at the Faculty of Natural Sciences in the branch of Math & Physic. His publications at International Conferences and Journals are in the area of applications of Information and Communication Technology in Education, Economy, and Environment, Digital learning etc.