

THE DIFFERENCE BETWEEN SCRUM AND OTHER AGILE AND TRADITIONAL METHODOLOGIES

Maja GABOROV¹, Dijana KARUOVIĆ¹, Dragica RADOSAV¹, Mila KAVALIĆ², Dragana MILOSAVLJEV¹, Sanja STANISAVLJEV¹

¹University of Novi Sad, Technical faculty „Mihajlo Pupin“, Zrenjanin, Republic of Serbia

²University of Novi Sad, Faculty of Technical Science, Novi Sad, Republic of Serbia

Abstract: SCRUM defines the systems development process as a loose set of activities that combines known, workable tools and techniques with the best that a development team can devise to build systems. In the recent years, software development organizations are striving to adopt agile software development methods and techniques. Successful agile adoption leads to producing higher quality software, enhances developers moral at a lower cost than the traditional waterfall model approach. It is necessary to know which methodology is the best for the appropriate project, considering that the projects differ according to the people in the team, the duration of the projects and the like. It is assumed that Scrum is the best methodologies, that it is most applied and should be compared with others in order to see the differences and to know which methodology is best used in the appropriate project because it is assumed that a methodology must be chosen to meet all project needs. This paper presents a review of Scrum Methodology, as well as the extent to which it is applied, and describe difference and advantages between SCRUM and other methodologies. It was concluded that Scrum is the most used, but that it is desirable to use the methodology that is best for the respective project since the projects differ.

Keywords: Scrum, Waterfall, Kanban, agile methodologies

INTRODUCTION

Many new software development approaches were introduced to fit the new cultures of the software development companies. Most companies nowadays aim to produce valuable software in short time period with minimal costs. Customer requirements are frequently changing and making it even more difficult [1]. Traditional software development approaches have a potential to provide straightforward, systematic, and organized process in the software development. The traditional approaches have limitations such as adaptation to rapidly changing business requirements, a tendency to be over budget and behind schedule, a lack of dramatic improvements in productivity, reliability, and simplicity [2].

Software Development Life Cycle (SDLC) consists of few phases like planning, analysis, design and implementation [1][3][4]. Software Development Life Cycle Model is used as a process of creating and altering current existing system. SDLC can be thought of as a concept that used by many software development methodologies, which are currently available in market or software industry [4]. Many number of SDLC models have been created like Waterfall, Spiral and V-model etc. There are many number of new approaches, SCRUM (Agile methodology) is one of them. Agile consists of many methodologies but SCRUM is most famous and powerful methodology which provides benefit to companies. SCRUM is simple for managing difficult projects. It is used at many companies with success when compare to traditional SDLC model [1]. It was included in agile methodology since it contains the same concepts of agile. A SCRUM is a team pack, where everyone in the team acts together. It delivers the project within time and with minimal cost [1]. Scrum starts with the premise that software development is too complex and unpredictable to be planned exactly in advance [5].

Some of important development models are: Waterfall model, V model (verification and validation), Incremental model, RAD model (Rapid Application Development), Agile model, Iterative model, Spiral model [6]. In our comparative study, some agile and traditional methodologies will be discussed and the differences of the Scrum methodology in relation to other methodologies will be stated.

THEORETICAL CONSIDERATION

The difference between traditional methodologies and agile software development methodologies is that traditional methodologies are complicated, constant, sequential, oddest and highly mature level but agile methodologies had been adopted in many projects due to its ability to better cope with frequent changes in requirements. Agile methodologies are suitable when requirements change during each process. Keen observation of analysis shows that agile methodologies help us to acquire better outcome with high quality, more client satisfaction, efficiency and management within the specified time and expenses. Agile methodology encourages the developer team to meet with the customers on regular basis in order to validate and verify their requirements [3][7]. Hybrid Agile methods and their effectiveness under different environments are an interesting field of study in software processes [8].

Waterfall

The Waterfall model was introduced by Royce in 1970 [9][10][3][6][11]. The Waterfall model is one of the traditional SDLC model, it follows only the sequential order. It flows steadily downwards. This leads to face these problems - Customer will not get satisfaction, requirements will be in pending, no profit, waste of time [1]. The Waterfall model is the first applied software development strategy, resembling the designs that were used in other industries [12]. The Waterfall methodology was one of the first such defined system development processes [13]. After completing the first phase only we can proceed with next phase [1].

Some important features of the methodology are [1][10]:

- Waterfall follows each step in a linear path.
- Plans all features for simultaneous implementation
- Designs all features
- Implements all features
- Tests all features
- Each phase of development proceeds in order without any overlapping.
- Each phase task to be completed within a specified time period
- The documentation and testing happens at the end of each phase, which helps in maintaining the quality of the project.
- In waterfall model the defect were found very late Figure 1 shows phases of Waterfall model.

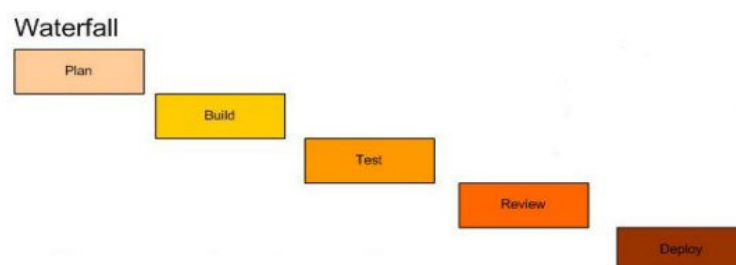


Figure 1: Waterfall method phases [1]

Waterfall model is easy to implement because it is sequential model. The amount of resources required to implement this model are minimal. Proper documentation is followed for the quality of the development. The problems with one phase are never solved completely during that phase and it leads to many problems. If client want the requirement to be changed, it will not implemented [1].

Scrum

Scrum was developed by Ken Schwaben and Jeff Sutherland in the early '90s [1] and is considered to be a framework for developing, delivering and sustaining complex products. Scrum is the most frequently used agile framework [14][15][16][17][18]. In order to define Scrum, all the roles, events, artefacts and rules have to be known and understood [19]. Weekly scrum-of-scrums considered very beneficial, since they distributed information between the teams and revealed possible problems early on [20]. Iterations in Scrum are called sprints. The length of one sprint in Scrum is normally four weeks, but can be also shorter or even longer [20]. Before starting each sprint a sprint planning meeting takes place [20]. Backlogs are lists of items to be developed. In the sprint planning meeting, a product owner with his or her team selects from the product backlog items to be developed during the next sprint to the sprint backlog [20][5]. Sprint planning meeting is conducted by the product owner, team member and the scrum master [21][22]. Product owner is responsible for prioritizing the items which is more important. Sprint planning meeting focusing on what to do and how to do. Changes will not accepted while in Sprint [1][5][23][10]. Daily Scrum implies: Scrum master and development team members report to each other and Daily meeting about 15 minutes [1][5][10]. Three questions are generated in this sprint: What they did yesterday? What they plan to do till next meeting? What difficulties are there in their way? [1][5]

A scrum retrospective meeting involves: talk about their experiences and the problems they face; talk about what to follow and what not further improvement followed and typically 15-30 minutes of the meeting. This will be done after each sprint [1][5]. Figure 2 shows scrum design.

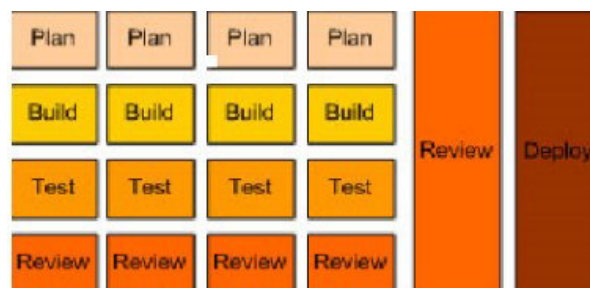


Figure 2. Scrum design [1]

Advantages are [1][13][4][7]:

- Scrum provides customer satisfaction by optimizing turnaround time and responsiveness to requests.
- Increase the quality
- Accept and expect the changes
- Provide better estimates while spending less time creating them
- Be more in control of the project schedule
- Scrum is ideal for rapidly changing, accumulating requirements. – Benefits to customer and project manager
- Scrum is fast, quick and can adapt changes easily
- Freezes schedule
- Short Sprint by short Sprint

- Estimates scope
- Top feature, then next feature – Never changes the schedule, or Sprint
- Adjusts the scope if needed to meet release dates
- Work estimates are much easier.
- Work proceeds and completes more logically.
-

Disadvantage are [1]:

- Documentation is very less
- Team members dedication is very important
- Team work is highly essential
- If team members does not cooperate well, the project will face failure.

The differences between Scrum and a waterfall are [1]:

- Different Roles
- Different Meetings
- Different Characteristics
- Different Artifacts
- Different Language

Table 1 shows the differences between a scrum and a waterfall.

TABLE 1. Comparison between Waterfall and SCRUM [1]

Waterfall	SCRUM
Deals with project	Deals with product
Traditional model consists of different phases	SCRUM methodology consists of different sprint
Does not expect changes	Expect changes and accept the changes
More documentation	Less documentation
Project cost determined during planning	Project cost set during project
Probability of success is low	Probability of success is high
Team flexibility and creativity is limited	Team flexibility and creativity is unlimited
Sequential	Overlapping

Spiral methodology

“The Spiral methodology “peels the onion”, progressing through “layers” of the development process. A prototype lets users determine if the project is on track, should be sent back to prior phases, or should be ended [13]”. The phases and phase processes are still linear. Requirements work is still performed in the requirements

phase, design work in the design phase, and so forth, with each of the phases consisting of linear, explicitly defined processes [13]. Figure 3 shows spiral methodology.

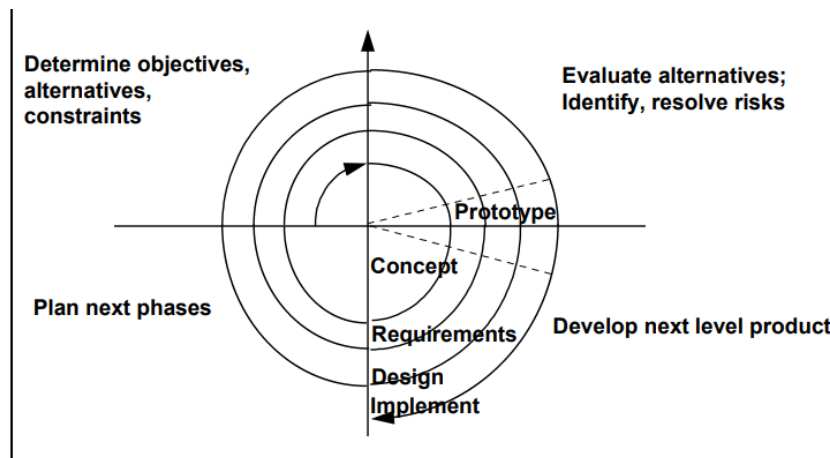


Figure 3: Spiral methodology [13]

Iterative methodology

“The Iterative methodology improves on the Spiral methodology. Each iteration consists of all of the standard Waterfall phases, but each iteration only addresses one set of parsed functionality. Using this approach, one can test the feasibility of a subsystem and technology in the initial iterations.” [13] Further iterations can add resources to the project while ramping up the speed of delivery. This approach improves cost control, ensures delivery of systems and improves overall flexibility. The Iterative approach still expects that the underlying development processes are defined and linear [13]. Table 2 presents a comparison of scrum with methodologies such as waterfall, spiral, iterative.

TABLE 2. Compares the primary SCRUM characteristics to those of other methodologies [13]

	Waterfall	Spiral	Iterative	SCRUM
Defined processes	Required	Required	Required	Planning & Closure only
Final product	Determined during planning	Determined during planning	Set during project	Set during project
Project cost	Determined during planning	Partially variable	Set during project	Set during project
Completion date	Determined during planning	Partially variable	Set during project	Set during project
Responsiveness to environment	Planning only	Planning primarily	At end of each iteration	Throughout
Team flexibility, creativity	Limited - cookbook approach	Limited - cookbook approach	Limited - cookbook approach	Unlimited during iterations
Knowledge transfer	Training prior to project	Training prior to project	Training prior to project	Teamwork during project
Probability of success	Low	Medium low	Medium	High

Kanban

Kanban (visual signal) was first used by workers in Toyota to track processes on their manufacturing system. Teams were able to communicate more effectively as this simple tool provided instance information on what needed to be worked on and when it is needed. Kanban is a scheduling system that provides information about what is needed, when to deliver, and how much to is needed. The entire value chain is controlled ideally by Kanban starting from the supplier to the customer [11].

- Kanban is value optimization theory based agile framework for the transparency of work flow movement and limited work in-progress [24].
- Kanban will allow teams and their members organize more freely [12].
- From this point of view, the following differences will be considered during the research [12]:
- setting up roles in a team is not a requirement
- meetings are not restricted by time-boxed iterations (e.g. sprints)
- the board will be continuously updated and stories can be added anytime if they fit into the current workflow
- any member or team can be the owner of the board
- estimates for tasks are not needed (e.g. time, user points)

Choosing one of these methodologies will be a team decision, based on their style and on the type of project [12]. Table 3 shows the differences between s Scrum and Kanban.

TABLE 3. Scrum vs. Kanban - differences [6][19]

Scrum	Kanban
Team involved in a specific iteration	Optional involvement
Uses speed (velocity) as a measure for improving processes	Uses deadline / lead time as a measure for improving processes
Prescribed estimations	Optional estimations
One sprint backlog belongs to a team	Kanban-board may be shared
Involves using at least three roles (Product Owner / Scrum Master / Scrum Team)	Does not use roles
The Scrum-board is reset between sprints	The Kanban-board does not change, it is persistent
For each sprint, priorities are established on the sprint backlog	Establishing priorities is optional

Scrumban

“In Scrumban, development teams may adapt to production requirements and interests of the stakeholders, without being burdened by the project methodology. Scrumban inherits from Kanban the concept of elimination of elements that might lead to unwanted results, thus avoiding unnecessary processes [6]“. Scrumban may optimize the teams’ effort in order to achieve the quality standards assumed. Scrumban ensures a slow transition from Scrum to Kanban [6]. Table 4 shows the differences between Scrum, Kanban and Scrumban.

TABLE 4. Scrum vs. Kanban vs. Scrumban [6]

	Scrum	Kanban	Scrumban
Instruments	Scrum board, Burn-down charts, prioritized task lists	Kanban board	Working board for visualizing the work-flow and the progress
Roles	Well defined (Product Owner, Scrum Master, Development Team)	No prescribed roles	Team and other needed roles
Planning meeting	Mandatory, Sprint Planning	If team wants and the flow is improved	If team wants and the flow is improved
Daily meeting	Mandatory, every day for 15 minutes	No meeting	Recommended, for continuous work on requirements and reduce idle time
Review/Retrospective meeting	Predefined meetings, activities and durations defined by the framework	Not prescribed	Can be done when needed for process improvement and feedback
Progress	Velocity (sprint content)	WIP (task limit for each step)	Controlled by work-flow state
Product Backlog	List of prioritized and estimated stories, using story points	Task list, each task containing a signal card	Signal card/Time card
Iterations	2-4 weeks (Sprint)	Continuous flow	Continuous flow

Scrumbanfall

Scrumbanfall is an agile integration of Scrum and Kanban with Waterfall model using the mixture of traditional SDLC protocols with the empiricism, agility and workflow management. Scrum is on the base of Scrumbanfall, by keeping Kanban in the center of the Scrum and wrapping Waterfall prior to Scrum Sprints [24]. “A standalone framework Scrum or Kanban or Waterfall cannot provide complete solutions for all the challenges of Software Engineering Management processes. Scrum and Kanban are trending agile software project management frameworks while Waterfall is the first traditional SDLC and still in existence in some of the software development organizations, who have not yet adopted agile frameworks”[24]. Agile frameworks are very suitable for large or medium range of software project, where project requirement and its feasibility is changed frequently over the time, while Waterfall is suitable for the projects having small duration and clear requirement at the initial stage of project, it can be predicated that project requirement may not going to change during its life cycle until the final project delivery [24].

RELATED WORK

One of the earliest surveys on Agile was conducted by the Australian Shine Technologies in 2003. With the majority of the 131 survey participants referring to adoption of Extreme Programming (XP) and around 8% adopting Scrum, 49% stated that Agile reduced development costs, 93% that productivity was better, 83% that business satisfaction was better and 88% that the quality of the software improved. Although a rather early survey, when Agile experience had not been not gained yet, the results from the Agile use are generally in accordance with the outcome of our survey. In 2003 XP was more popular than Scrum [8].

Some survey showed that the most popular among heavyweight alternatives is, as expected, the waterfall model (36.5%) followed by the Spiral model (14.4%) and the Unified Process (12.2%), whereas enterprises tend to adopt also hybrid approaches or reject traditional methodologies completely heading directly for adaptive techniques (36.9%). In this study agile methodologies mostly used. Among Agile methodologies the big winner is Scrum (76.9% of the participants) followed by Extreme Programming (6.4%) and Feature Driven

Development (3.8%), whereas Agile combinations were also indicated. Authors also tried to detect the percentage of projects that were considered successful. Agile projects are generally successful with 54% of the participants indicating an overall success rate over 81% for their projects [8]

The results some research indicate a significant increase in the adoption of Scrum in comparison to other Agile methodologies with many successful project. Nevertheless, continuous studies are necessary to follow its adoption progress and the emerging variants, especially its combination with other production methods, such as Kanban. [8].

In one study, it can be seen that the most popular methodology is Agile, using Scrum. The most popular methodology used by the candidates was Scrum, with 62.5% of them using Scrum at least once. 28.1% of them used Waterfall and only 25% used Kanban. Scrum reported as well the best overall satisfaction in terms of how much they enjoyed using the methodology. In conclusion, the methodology chosen depends on each team and has to be picked specifically for that project, as no approach can satisfy all needs. The tendency is that Waterfall is used mostly by small teams for a small project that have well-defined requirements, while Agile is more flexible and preferred when continuous feedback is important [12].

Some authors are compared by means of simulation techniques an heavy and prescriptive approach, Waterfall, with two agile and less prescriptive process tools, Scrum and Lean-Kanban. Their study has been carried out on under some limiting assumptions, but it can be considered as a valid starting point for further studies. They described some strengths and weaknesses of three software process methods by modeling their environment with a continuous-time simulation tool. Although Lean-Kanban is well known in software development processes, it has not yet investigated in depth in research works. In their model, the 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 [9].

CONCLUSION

Following the results of the study, it can be concluded that each methodology has its strengths and weaknesses. As such, there is no solution for all types of projects. Various factors like the number of people in the team, how inclined to changes the requirements are or the duration of the project should be considered. SCRUM is best if the requirements frequently change. WATERFALL is best if there is no change in the requirements. Scrum and Kanban are trending agile methodologies for software project development and management. In conclusion, the methodology chosen depends on each team and has to be picked specifically for that project, as no approach can satisfy all needs. The tendency is that Waterfall is used mostly by small teams for a small project that have well-defined requirements, while Agile is more flexible and preferred when continuous feedback is important. Waterfall and Spiral methodologies set the context and deliverable definition at the start of a project. SCRUM and Iterative methodologies initially plan the context and broad deliverable definition, and then evolve the deliverable during the project based on the environment. SCRUM acknowledges that the underlying development processes are incompletely defined and uses control mechanisms to improve flexibility. The primary difference between the defined (waterfall, spiral and iterative) and empirical (SCRUM) approach is that The SCRUM approach assumes that the analysis, design, and development processes in the Sprint phase are unpredictable. A control mechanism is used to manage the unpredictability and control the risk. Flexibility, responsiveness, and reliability are the results. Certainly there is no “absolute best” agile development methodology, each project bringing its own goals and requirements. Still, emerging from the sphere of process control and industrial production, Scrum and Kanban methodologies have generated the

Scrumban hybrid. Scrum, Kanban and Waterfall have generated the Scrumbanfall. Future research could relate to examining the extent to which the Scrumban and Scrumbanfall hybrid approaches are applied and whether it is better to apply multiple methodologies at the same time.

REFERENCES

- [1]. Mahalakshmi, M., Sundararajan, M. Traditional SDLC vs scrum methodology—a comparative study. *International Journal of Emerging Technology and Advanced Engineering*, 3(6), 192-196. (2013).
- [2]. Flora, H. K., Chande, S. V. A systematic study on agile software development methodologies and practices. *International Journal of Computer Science and Information Technologies*, 5(3), 3626-3637. (2014).
- [3]. Umbreen, M., Abbas, J., Shaheed, S. M. A Comparative Approach for SCRUM and FDD in Agile. *International Journal of Computer Science and Innovation*, 2, 79-87. (2015).
- [4]. Amlani, R. D. Advantages and limitations of different SDLC models. *International Journal of Computer Applications & Information Technology*, 1(3), 6-11. (2012).
- [5]. Mahnic, V., Drnovscek, S. Agile software project management with scrum. In *EUNIS 2005 Conference-Session papers and tutorial abstracts* (p. 6). (2005).
- [6]. Stoica, M., Ghilic-Micu, B., Mircea, M., Uscatu, C. Analyzing agile development-from waterfall style to Scrumban. *Informatica Economica*, 20(4). (2016).
- [7]. Azanha, A., Argoud, A. R. T. T., Camargo Junior, J. B. de, & Antonioli, P. D. Agile project management with Scrum. *International Journal of Managing Projects in Business*, 10(1), 121–142. (2017).
- [8]. Kapitsaki, G. M., Christou, M. Where is Scrum in the current Agile world?. In *2014 9th International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE)* (pp. 1-8). IEEE. (2014).
- [9]. Cocco, L., Mannaro, K., Concas, G., Marchesi, M. Simulating kanban and scrum vs. waterfall with system dynamics. In *International conference on agile software development* (pp. 117-131). Springer, Berlin, Heidelberg. (2011).
- [10]. Bannink, S. Challenges in the Transition from Waterfall to Scrum—a Casestudy at Portbase. In *20th Twente Student Conference on Information Technology* (Vol. 182). (2014).
- [11]. Zayat, W., Senvar, O. Framework study for agile software development via scrum and Kanban. *International Journal of Innovation and Technology Management*, 17(04), 2030002. (2020).
- [12]. Andrei, B. A., Casu-Pop, A. C., Gheorghe, S. C., Boianigiu, C. A. A study on using waterfall and agile methods in software project management. *Journal Of Information Systems & Operations Management*, 125-135. (2019).
- [13]. Schwaber, K. Scrum development process. In *Business object design and implementation* (pp. 117-134). Springer, London. (1997).
- [14]. P. Diebold, J-P. Ostberg, S. Wagner and U. Zendler, “What Do Practitioners Vary in Using Scrum?”, In *International Conference on Agile Software Development*, Springer, Cham, pp. 40-51, (2015).
- [15]. L. S. Møller, F. B. Nyboe, T. B. Jørgensen, J. J. Broe, “A Scrum tool for improving Project Management”, In *Proceedings of fifth Student Interaction Design Research Conference*, pp. 30-32, April, (2009).
- [16]. V-P. Eloranta, K. Koskimies, T. Mikkonen, “Exploring ScrumBut—An empirical study of Scrum anti-patterns”, In *Information and Software Technology*, vol. 74, pp. 194-203, (2016).
- [17]. T. Gustavsson, “Benefits of Agile Project Management in a Non-Software Development Context: A Literature Review”, In *Project Management Development – Practice and Perspectives: Fifth International Scientific Conference on Project Management in the Baltic Countries*, pp. 114-124, (2016).

- [18]. Ereiz, and D. Mušić, “Scrum Without a Scrum Master”, In 2019 IEEE International Conference on Computer Science and Educational Informatization (CSEI), pp. 325-328, (2019).
- [19]. Mircea, E. Project management using Agile frameworks. Academy of Economic Studies. Economy Informatics, 19(1), 34-44. (2019).
- [20]. Paasivaara, M., Durasiewicz, S., Lassenius, C. (2009). Using scrum in distributed agile development: A multiple case study. In 2009 Fourth IEEE International Conference on Global Software Engineering (pp. 195-204). IEEE. (2009).
- [21]. Ciurairu, C. Risu, “Usage of Scrum Methodology for Implementing Software Project”, In Managing Innovation and Diversity in Knowledge Society Through Turbulent Time: Proceedings of the MakeLearn and TIIM Joint International Conference 2016, ToKnowPress, pp. 713-721, 2016. (2016).
- [22]. H. Lei, F. Ganjeizadeh, P. K. Jayachandran, P. Ozcan, “A statistical analysis of the effects of Scrum and Kanban on software development projects”, In Robotics and Computer-Integrated Manufacturing, vol. 43, pp. 59-67. (2017).
- [23]. Nishijima, R. T., Dos Santos, J. G. The challenge of implementing scrum agile methodology in a traditional development environment. International Journal of Computers & Technology, 5(2), 98-108. (2013).
- [24]. Bhavsar, K., Shah, V., Gopalan, S. 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. (2020).

Corresponding author:

Dijana KARUOVIĆ
Information Technology
Technical faculty “Mihajlo Pupin”, Zrenjanin
University of Novi Sad
Đure Đakovića bb, Zrenjanin
23000 Zrenjanin, Serbia
dijanakaruoovic@gmail.com