Agile methodologies in software development: IT students' perspectives

Marija Blagojević, Mladen Janjić, Marina Milošević, Željko Jovanović
Faculty of Technical Sciences Čačak
University of Kragujevac
Čačak, Serbia
marija.blagojevic@ftn.kg.ac.rs; mladen.janjic@ftn.kg.ac.rs, marina.milosevic@ftn.kg.ac.rs, zeljko.jovanovic@ftn.kg.ac.rs

Abstract—**Agile is a popular buzzword these days, especially in software development, although it is recently advocated to be used in other fields, as well. Consequently, there is a growing research interest in principles, methodologies and plethora of other features related to the agile practices. However, research on how to introduce and teach agile principles in education is relatively scarce in comparison to the other fields, although a few research showed interest not long after its emergence. The aim of the paper is to analyze the knowledge of agile methods among students and to conduct a sentiment analysis of students' responses regarding agile methods.**

Keywords-Agile, software development, management.

I. Introduction

The popularity of agile practices (referred as Agile in further text) is commonly associated with the expansion of Internet during the 2000's and later smartphones, as software development needed to be developed rapidly with frequent changes in requirements for new features [1]. This meant that traditional waterfall method was impediment to meeting short deadlines and adapting quickly to the new requirements, not only in software development [2]. Although during the 70's and 80's there were some efforts to improve or even replace traditional waterfall approach to the development of software, sometimes referred to as *heavyweight* methods, it was during the 90's and more formally during the 2000's that so-called *lightweight* methods were properly defined and for which principles and guidance were formulated. One of the most famous proponents of the Agile was Robert C. Marting, better known as "Uncle Bob", also known for coining the SOLID mnemonic.

The new practices in software development also required support in software used for managing software projects, like Microsoft Project and many others. While smaller companies adapted swiftly, Microsoft neglected new trends, which meant that MS Project was relinquished of its use in managing software development, but still held strong in other areas of project management. In expectation of Microsoft introducing support for Agile in MS Project, developers moved to other solutions, currently dominated by Atlassian (founded in 2002), well-known for its applications and services for managing software development and source version control, such as Jira, Trello, Confluence and Bitbucket. Students are still taught MS Project at universities, usually available free of charge through Microsoft Education tools and resources, but principally in areas outside courses related to the software development and programming. Thus, lecturers committed to other solutions or didn't use any at all, just introducing students with some facts about Agile. The covid-19 pandemic put forward the importance of Agile in online teaching [3].

. The objective of the paper is to analyze students' understanding of agile methods and to conduct a sentiment analysis of their responses regarding agile methods. The research purpose is to determine the level of awareness of the significance of using agile methods and to analyze students' sentiments towards agile methodologies.

II. Related Work

With the growing need for junior software developers to experience and gain higher proficiency in Agile, agile principles were applied in programming courses, not just at universities [4]–[21], but also at high school level [22]. This also led to Agile being introduced in courses where it was least expected, e.g., Mathematics [23].

As a consequence, distinctive approaches were developed for the introduction of Agile in education, resulting in different impacts on success and outcomes. It was not only about teaching or learning Agile, but in some cases also revising courses by applying agile principles, which introduced additional requrements, like the neeod of automation to lessen the burden of assessment and feedback [14]. The goal is usually to provide insight into the practice of Agile, upheave students' experience for future work in IT companies, improve collaboration practices and promote Agile values [11].

This usually involves constant monitoring the development of students' teams [12], which further highlights the need of automation, as already mentioned above. The process of applying of Agile has two distinctive perspectives, teaches' and learners' [9], meaning that both parties need certain level of profficiency in using tools and processes in Agile [11]. The first step in acomplishing this may be training and certification of teachers [8], even in some cases following recommendations or guidances provided by professional organizations [22]. In addition to this, there is a concept of integration of pedagogical approaches based upon Agile principles without necesserily teaching about Agile itself [16].

Concerning Agile in higher education in Serbia, research results are scarse, since Agile is not frequently emplyed due to higher requirements for involvement of teaching staff, though some researchers provide experience in teaching Agile in multiple courses over longer periods of time [7]. While a few universities in the region introduced Agile in capstone courses more than a decade ago [7], [22], the lack of wider acceptance of Agile in higher education is still evident.

III. Methodology

In this study, fourth-year students of the Information Technology study program participated. These students are enrolled in the Software Engineering course, where agile methodology is studied. In addition to theoretical exploration of agile methodology, students have the opportunity to apply and work with various types of exercises in the practical part, following agile principles.

The student "scrum" extended beyond the usual scrum duration, specifically lasting for 3 months (one semester). Within their teams, students were familiarized with the types of meetings and the manner in which they should be conducted, aiming to effectively address challenges in their daily communication.

At the end of the semester, the students were surveyed. The survey was conducted electronically and anonymously, with 46 students participating in the response.

IV. Results and Discussion

Within this chapter, the results of the student survey are presented.

"I have gained enough knowledge through my previous studies that enables me to independently or collaboratively participate in the development of software projects."

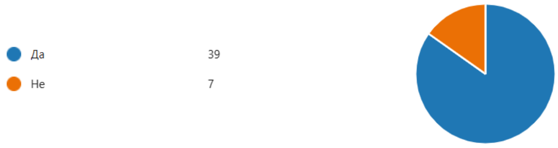


Figure 1. Acquired knowledge

As shown in Figure 1, the majority of students assess that they have acquired enough knowledge to independently or collaboratively develop software projects. Considering that this is the final year of the Information Technology study program, the obtained response is expected.

"I believe that the ability to work in a team is more important than knowledge in the programming field (1 - strongly disagree; 2 - disagree; 3 - undecided; 4 - partially agree; 5 - strongly agree)."

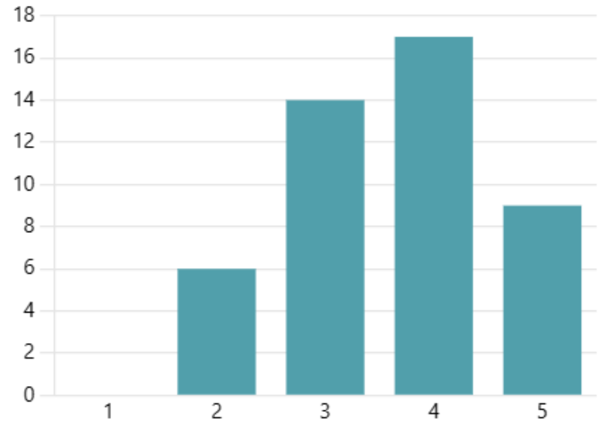


Figure 2. The importance of teamwork

Based on Figure 2, it can be concluded that the majority of students believe that the ability to work in a team is more important than knowledge in the programming field. A portion of students is undecided, implying that, in addition to teamwork, possessing knowledge in programming is equally significant.

"I am familiar with the agile approach to software development (1 - not familiar at all; 2 - slightly familiar; 3 - undecided; 4 - partially familiar; 5 - fully familiar)."

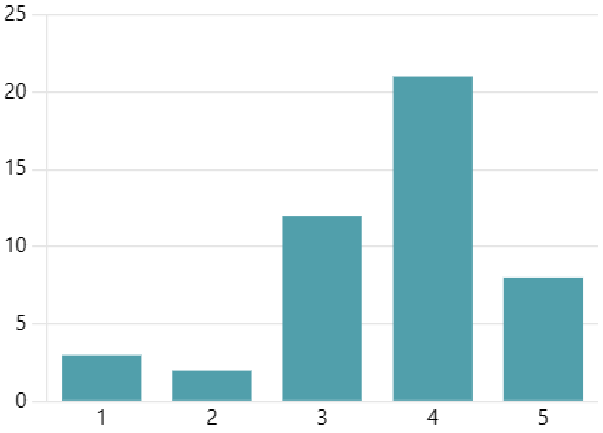


Figure 3. Familiarity with agile approach

Figure 3 indicates that while the majority of students are familiar with the principles of the agile approach, a certain percentage of students are not acquainted with them, despite the fact that agile methodologies are studied both theoretically and practically in the Software Engineering course, including practical exercises.

"I believe that teamwork on projects is easier to implement using agile methodologies along with version control services (Github, Gitlab, Microsoft DevOps, etc.):"

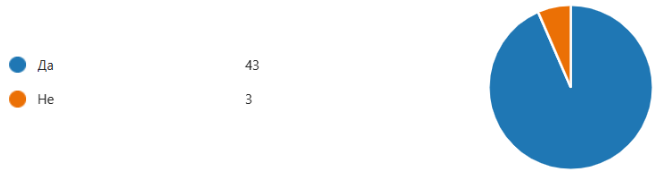


Figure 4. Code versioning services

Considering the results presented in Figure 4, it can be concluded that students are reasonably familiar with the importance of using software for code versioning.

"The time spent on activities related to agile methodologies can be better utilized on specific tasks, such as writing code or software testing (1 - strongly disagree; 2 - disagree; 3 - undecided; 4 - partially agree; 5 - strongly agree)."

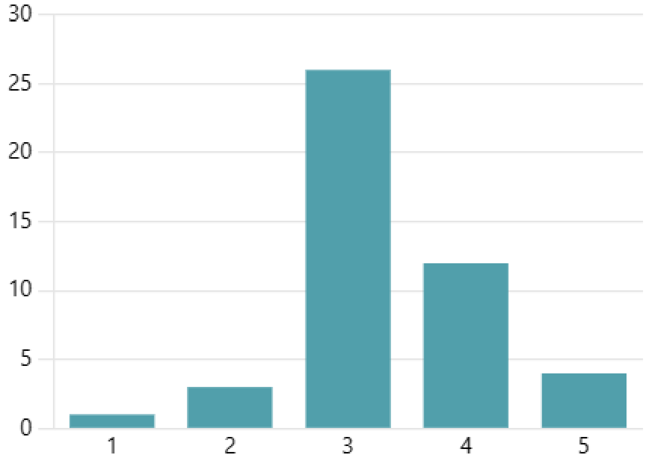


Figure 5. Time utilization

Figure 5 provides an overview of students' opinions on what is more appropriate in terms of time utilization: engaging in activities related to agile methodologies or focusing on specific tasks such as software development and testing.

The majority of students are undecided, followed by those who partially agree with the statement that time spent on activities related to agile methodologies could be better used for specific tasks, such as writing code or software testing.

"Compare the possibility of applying agile methodologies to other activities you have had during your studies (1 - not possible at all; 2 - maybe possible; 3 - I am not sure; 4 - partially possible; 5 - entirely possible)."

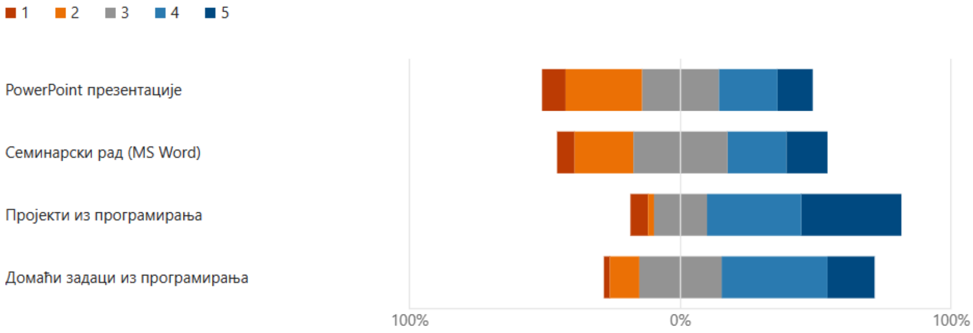


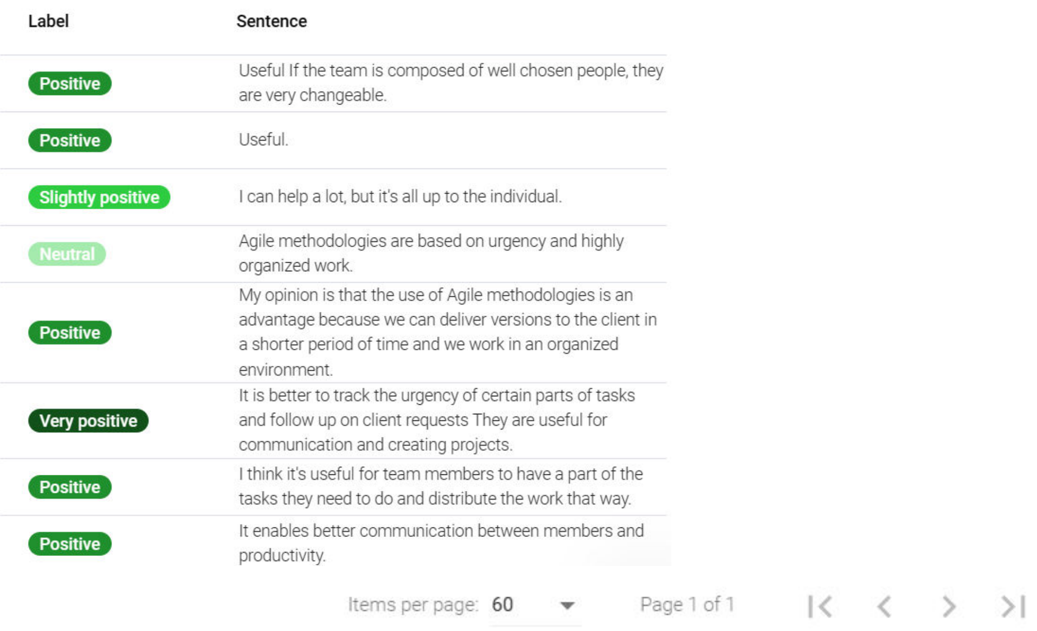
Figure 6. Possibility of applying used activities in agile methodologies

According to Figure 6, the majority of students assess that it is possible to leverage the activities applied within agile methodologies. In addition to closed-ended questions, sentiment analysis was conducted on open-ended responses. Students answered a question related to their perception of the application of agile methods in project development.

Sentiment analysis, often referred to as opinion mining, is a subfield of natural language processing (NLP) that analyzes textual data to discern sentiment or emotional tone conveyed in the text. It categorizes feelings as positive, negative, or neutral based on the content and context of the text. Sentiment analysis is widely used by businesses and organizations to assess customer satisfaction, monitor brand reputation, and understand consumer perception.

Sentiment analysis employs machine learning, statistical analysis, and linguistic rules to examine textual data, such as customer reviews, social media posts, and survey responses. By scrutinizing the language and expressions used in the text, it identifies and extracts subjective information to determine sentiment towards specific topics, products, or services. Sentiment analysis can be conducted at various levels - document level, sentence level, or aspect level, each providing different degrees of insight into conveyed sentiment. Insights gathered from sentiment analysis aid organizations in addressing customer concerns, crafting marketing strategies, and making informed decisions. Despite its advantages, sentiment analysis may encounter challenges such as detecting sarcasm, understanding context, and managing multilingual text analysis.

Through the analysis of individual sentences (responses) in the SpeakAi tool (app.speakai.co), it was determined that the sentences exhibit sentiment ranging from neutral to extremely positive, with no occurrence of negative sentiment. An example of obtaining sentiment for several individual responses is provided in Figure 7. There are a total of 46 responses available in the database.



Label	Sentence
Positive	Useful If the team is composed of well chosen people, they are very changeable.
Positive	Useful.
Slightly positive	I can help a lot, but it's all up to the individual.
Neutral	Agile methodologies are based on urgency and highly organized work.
Positive	My opinion is that the use of Agile methodologies is an advantage because we can deliver versions to the client in a shorter period of time and we work in an organized environment.
Very positive	It is better to track the urgency of certain parts of tasks and follow up on client requests They are useful for communication and creating projects.
Positive	I think it's useful for team members to have a part of the tasks they need to do and distribute the work that way.
Positive	It enables better communication between members and productivity.

Figure 7. Example of labeling

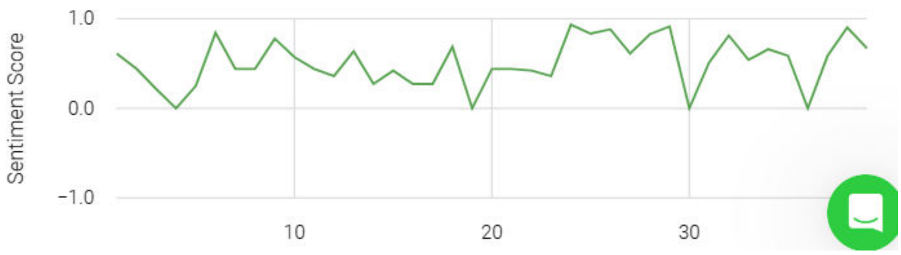
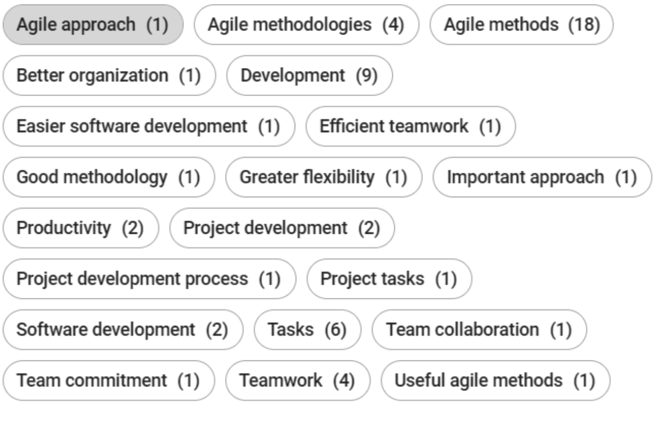


Figure 8. Sentiment Score Display

Figure 8 displays the Sentiment Score. The results indicate that the sentiment is either neutral or positive, with no statements expressing negative sentiment.

Figure 9 illustrates the key words in the dataset. These are keywords related to agile methods (most frequently), followed by tasks (6 repetitions) and teamwork (4 repetitions).

Based on the key words, the most common words associated with agile methods were mapped, and the frequency of these words indicates that students are familiar with the basic concepts of the agile approach.



Agile approach (1) Agile methodologies (4) Agile methods (18)
Better organization (1) Development (9)
Easier software development (1) Efficient teamwork (1)
Good methodology (1) Greater flexibility (1) Important approach (1)
Productivity (2) Project development (2)
Project development process (1) Project tasks (1)
Software development (2) Tasks (6) Team collaboration (1)
Team commitment (1) Teamwork (4) Useful agile methods (1)

Figure 9. Keywords

V. Conclusion

Based on the presented results, several conclusions can be drawn in several ways:

1. Students demonstrate a satisfactory understanding of the importance of applying agile methods in software development. However, a larger percentage of students believe that it is more efficient to dedicate time to concrete activities such as programming, as opposed to activities associated with agile methodology.

2. Through sentiment analysis, it was found that students predominantly express a positive sentiment towards agile methods. Additionally, the analysis of keywords indicates that students adequately recognize key concepts of agile methodologies.

3. Future work is planned to include sentiment analysis of students who attended online classes during the COVID-19 pandemic in comparison to traditional classroom instruction.

Acknowledgment

This study was supported by the Ministry of Science, Technological Development and Innovation of the Republic of Serbia, and these results are parts of the Grant No. 451-03-66/2024-03/200132with University of Kragujevac – Faculty of Technical Sciences Čačak.

References

[1] K. Beck et al., “Manifesto for agile software development,” Agile Alliance, 2010.

[2] S. Malakar, “Agile in practice: practical use-cases on project management methods including agile, kanban and scrum, BPB Publications,” India, 2021.

[3] H.Y. Al-Sholi, O.R. Shadid, K.A. Alshare, M. Lane, “An agile educational framework: A response for the covid-19 pandemic,” Cogent Education, vol. 8, no. 1 (2021), Article ID: 1980939, 17 pages.

[4] P.A. Laplante, “An agile, graduate, software studio course,” IEEE Transactions on Education, vol. 49, no. 4 (2006), pp. 417-419.

[5] B. Bruegge, M. Reiss, J. Schiller, “Agile principles in academic education: a case study,” Sixth International conference on information technology: new generations, 27-29 April 2009, Las Vegas, NV, USA, pp. 1684-1686.

[6] D.F. Rico, H.H. Sayani, “Use of agile methods in software engineering education,” 2009 Agile Conference, 24-28 August 2009, Chicago, IL, USA, pp. 174-179.

[7] V. Devedžić, S.R. Milenković, “Teaching agile software development: a case study,” IEEE Transactions on Education, vol. 54, no. 2 (2011), pp. 273-278.

[8] V. Kamat; S. Sardessai, Agile practices in higher education: a case study,” Agile India, 17-19 February, 2012, Bengaluru, India, pp. 48-55.

[9] R. Romeike, T. Göttel, “Agile projects in high school computing education: emphasizing a learners’ perspective,” 7th Workshop in primary and secondary computing education (WiPSCE ’12), 8-9 November 2012, Hamburg, Germany, pp. 48-57.

[10] M. Kropp, A. Meier, M. Mateescu, C. Zahn, “Teaching and learning agile collaboration,” IEEE 27th Conference on Software Engineering Education and Training (CSEE&T), 23-25 April 2014, Klagenfurt, Austria, pp. 139-148.

[11] M. Kropp, A. Meier, R. Biddle, “Teaching agile collaboration skills in the classroom,” IEEE 29th International Conference on Software Engineering Education and Training (CSEET), 5-6 April 2016, Dallas, TX, USA, pp. 118-127.

[12] M. Marques, S.F. Ochoa, M.C. Bastarrica, F.J. Gutierrez, “Enhancing the student learning experience in software engineering project courses,” IEEE Transactions on Education, vol. 61, no. 1 (2018), pp. 63-73.

[13] P. Salza, P. Musmarra, F. Ferrucci, “Agile methodologies in education: a review,” In: D. Parsons, K. MacCallum (eds), Agile and lean concepts for teaching and learning, Springer, Singapore, 2019.

[14] R. Chatley, “Applying Lean Learning to Software Engineering Education,” In: D. Parsons and K. MacCallum (eds.), Agile and Lean Concepts for Teaching and Learning, Springer Nature Singapore Pte Ltd. 2019, pp. 285-302.

[15] T.F. Otero, R. Barwaldt, L.O. Topin, S.V. Menezes, M.J.R. Torres, A.L. de Castro Freitas, “Agile methodologies at an educational context: a systematic review,” IEEE Frontiers in Education Conference (FIE), 21-24 October 2020, Uppsala, Sweden, 2020, pp. 1-5.

[16] J.H. Sharp, A. Mitchell, G. Lang, “Agile teaching and learning in information systems education: an analysis and categorization of literature,” Journal of Information Systems Education, vol. 31, no. 4 (2020), pp. 269-281.

[17] H.Y. Al-Sholi, O.R. Shadid, K.A. Alshare, M. Lane, “An agile educational framework: a response for the covid-19 pandemic,” Cogent Education, vol. 8, no. 1 (2021), Article ID: 1980939, 17 pages.

[18] M. Neumann, L. Baumann, “Agile methods in higher education: adapting and using eduScrum with real world projects,” IEEE Frontiers in Education Conference (FIE), 13-16 October 2021, Lincoln, NE, USA, pp. 1-8.

[19] R.M. Paul, Y. Jazayeri, L. Behjat, M. Potter, “Design of an integrated project-based learning curriculum: analysis through Fink’s taxonomy of significant learning,” IEEE Transactions on Education, vol. 66, no. 5 (2023), pp. 457-467.

[20] C. Marnewick, “Student experiences of project-based learning in agile project management education,” Project Leadership and Society, vol. 4 (2023), Article ID: 100096, 10 pages.

[21] D. Çulha, “Applying competition-based learning to agile software engineering,” Computer applications in engineering education, vol. 24, no. 3 (2016), pp. 382-387.

[22] V. Mahnic, “A capstone course on agile software development using Scrum,” IEEE Transactions on Education, vol. 55, no. 1 (2012), pp. 99-106.

[23] J. Pócsová, D. Bednárová, G. Bogdanovská, A. Mojžišová, “Implementation of agile methodologies in an engineering course,” Education Sciences, vol. 10, no. 11 (2020), Article ID: 333, 19 pages.

[24] M. Tushev, G. Williams, A. Mahmoud, “Using GitHub in large software engineering classes. an exploratory case study,” Computer Science Education, vol. 30, no. 2 (2020), pp. 155-186.