AI techniques and tools in Agile Software Development: Preliminary research

Dijana Peras, Zlatko Stapić, Mislav Matijević
University of Zagreb
Faculty of Organization and Informatics
Pavlinska 2, 42000 Varaždin, Croatia
{dijana.peras, zlatko.stapic, mislav.matijevic}@foi.unizg.hr

Abstract*. In recent years, there has been significant growth in research related to Agile software development (ASD), accompanied by a notable rise in the adoption of artificial intelligence (AI) tools and techniques. AI is believed to have the potential to bring about a transformation in agile software development, leading to improved product quality, increased production efficiency, and higher project success rates. Consequently, there is a compelling need to incorporate AI methods and tools into the agile software development process. This study presents the results of a literature review and answers research questions on AI techniques and tools, their purposes, and their benefits when used in the context of ASD. In a multi-phased process, a total of 374 documents were gathered and examined. 28 papers satisfied the inclusion and quality assessment requirements. A total of 7 different AI techniques were identified, of which machine learning (ML) was the one predominantly used. The purposes and benefits of AI techniques were identified and discussed. Recommendations for future research were given to tackle detected research gaps.*

Keywords. Artificial Intelligence, AI, Agile Software Development, Agile Framework, Agile Methodology

1 Introduction

The effective creation of high-quality software has become essential for the software industry as software applications are becoming increasingly popular across every aspect of society. To deliver better software, boost efficiency, and assure project success, agile software development (ASD) could be enhanced with artificial intelligence (AI) solutions. Software development teams may take advantage of AI in a variety of ways, including automation of specific tasks, delivery of project insights, identification of phases of the agile process that are prone to risks, prioritization of effort, etc. (Sofian et al., 2022).

AI techniques like machine learning (ML), heuristic algorithms (HA), deep learning (DL), data mining (DM), data analytics (DA), and natural language processing (NLP) have been widely explored in various software engineering (SE) phases, which involve different activities across all stages of the Software Development Lifecycle (SDLC) (Sofian et al., 2022). Several systematic and mapping literature reviews were published (Biesialska et al., 2021; Perkusich et al., 2020; Sofian et al., 2022), exploring the overall trends in AI techniques and their application to SE. However, no clear distinction was made in AI usage within traditional and agile software development processes, which operate under entirely different assumptions and therefore require special approaches.

This paper aims to provide a preliminary review of AI techniques and tools used in ASD, as well as to investigate the purposes for which AI is used in the context of ASD and the benefits it provides for businesses and users. The distinction is made among agile processes and activities to inspect the scope of AI integration. A broader goal of the paper is to identify research gaps and to build the foundation for the proposal of AI supported framework for agile software development.

The paper consists of five sections. First, a short overview of relevant background on AI and ASD is given. In the second section, the methodology is presented. The third section presents the results of the research. The discussion is made in the fourth chapter. The last chapter presents the main conclusions.

2 Research methodology

A literature review was conducted by three researchers. To maximize the efficiency of the research, two researchers were engaged in each phase of the process. The literature review aimed to identify AI techniques and tools used within ASD. The Literature Review guidelines proposed by Templier and Paré (Templier & Paré, 2015) were followed. The study's protocol is outlined in the sections that follow.

2.1 Formulating the problem

The goal of this paper is to synthesize and analyse the recent research in the area of applying AI to ASD, which is a specific software development paradigm. Namely, ASD is a change-driven methodology for creating software in the context of unpredictable requirements (Perkusich et al., 2020). ASD emphasizes the needs of the customers, collaboration, and fast delivery of software. Due to the presence of the customers, who always provide feedback in accordance with market trends or company values, there is a need for constant decision-making and optimization. Scrum, Extreme Programming, Crystal, and Adaptive Software Development are a few popular agile methodologies.

To clarify the scope of the research, the difference between software process and activities will be briefly explained. A software process (also known as a software methodology) is a set of related activities that leads to the production of the software (Shehab et al., 2020). ASD process consists of three phases, as shown in Figure 1.

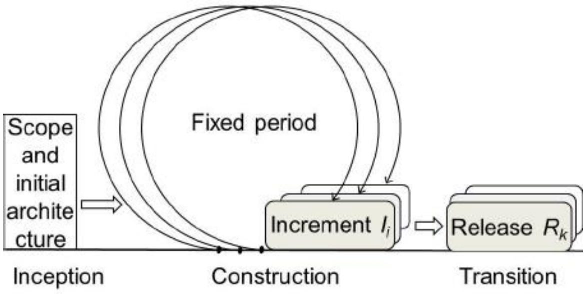

Figure 1. The Agile Software Development Process (ben Othmane et al., 2014)

The project's scope and the initial architecture are determined at the inception phase. The software is developed through a series of iterations during the construction phase. For every iteration, a group of user stories is selected. The requirements are elicited, and the design is updated to address the requirements. Then, a software increment that addresses the requirements is coded. At the end of the iteration, results are presented and the team's efficiency is evaluated. In the transition phase product is tested and prepared as a release for use in production. Multiple activities take place during the ASD process, and each has its own workflow. Some of these activities are task planning, agile estimating (e.g., the time to completion), user stories development, backlog management, agile scheduling, etc.

Today's software is increasingly using artificial intelligence (AI). The ability to quickly, automatically, and effectively make decisions and predictions are considered to be the main benefits of AI techniques and tools, which has led to their adoption in software engineering (SE) (Sofian et al., 2022). In light of the increased interest in AI application in the SE domain, this study seeks to find the answers to the following research questions:

- RQ1: Which AI techniques and tools are used in the context of ASD?
- RQ2: For which purposes are AI techniques and tools used in the context of ASD?
- RQ3: What are the benefits of using AI techniques and tools in the context of ASD?

2.2 Searching the literature

For the selection of primary studies, three databases were used: Web of Science Core Collection (WOSCC), Scopus, and IEEE Xplore (IEEE). The following query was applied to selected databases: *("artificial intelligence" OR "AI")* AND *("agile software development" OR "agile development" OR "agile process" OR "agile processes" OR "agile framework" OR "agile method" OR "agile methodology").* The search period was limited to the last 10 years (2013 – 2023) and to the subject area of Computer Science. Search fields were limited to title, abstract, and keywords. Only articles and conference papers written in English were included in the research.

2.3 Screening for inclusion

Gathering the papers from databases and identifying possibly relevant studies by title and abstract comprised the first two phases of the research selection procedure. Two researchers were included in this phase of the process. One researcher was responsible for screening all papers retrieved during the search phase, while the other author was responsible for screening 15% of the papers from each database. Afterward, the results were compared to see the rate of match between the authors. The following inclusion and exclusion criteria were used:

Inclusion criteria:

- It is clear from the article's title that it is about the use of AI in ASD.
- It seems obvious from the abstract that the article describes and elaborates on the application of AI in ASD.
- It is not clear from the title or abstract to what extent the article covers the application of AI in ASD, but it might be relevant to the research.

Exclusion criteria:

- The title suggests that the article is only marginally related to AI or ASD.
- The abstract suggests that the article is only marginally related to AI or ASD.
- The abstract's focus is not on using AI in ASD, but rather on another topic (i.e., the adoption of agile methodology, development of the ontology, etc.).

After the removal of duplicated papers, the final set of 37 papers was selected for the quality assessment phase, as presented in Table 1.

Table 1. Sources and obtained papers

Source	Initial query	Potentially relevant by title	Potentially relevant by abstract
WOSCC	56	18	10
Scopus	169	27	14
IEEE	149	35	28
Total	374	80	52
Duplicates			15
Final			37

2.4 Assessing quality

Based on the titles and abstracts of 37 papers that appeared to be possibly relevant, the following set of questions for assessing quality was developed:

- Q1: Does the article describe how certain AI technique/tool was used in the context of ASD?
- Q2: Does the article describe how certain activities or processes of ASD can benefit from using AI?
- Q3: Does the article provide an example (i.e., use case) of integrating AI in ASD process?
- Q4: Does the paper only describe the theoretical implications of using AI in ASD?

Two authors were included in this phase of the research. A ratio of 1:2 was used to distribute the papers among them. Negative responses to questions Q1 to Q3, as well as positive response to question Q4, were eliminatory. The quality assessment analysis resulted in 28 papers which were included in the following phase. Afterward, one paper was excluded from further analysis because the full text could not be retrieved.

2.5 Extracting data

In this stage of the research, 27 papers were examined by two researchers. The number of papers per author was distributed equally. Before the start of this phase, a short discussion was made to assure there are no misunderstandings about the data extraction method. The inductive method was used to answer the defined research questions. Neither the AI techniques and tools nor the ASD processes and activities were predetermined; rather, they emerged through the research, as well as the stated benefits of using AI techniques and tools in the context of ASD. Applicable information from each of the primary studies included in this stage of the review was gathered. Spreadsheets and reference management software were utilized to manipulate data during the analysis. The research results are based on the answers given in these papers to the stated research questions.

3 Research results

The last step of the methodology consists of analysis and synthesis of previously extracted information. Our research findings are displayed in accordance with the stated research questions. The systematization of results and references will be presented in Table 2.

3.1. Which AI techniques and tools are used in ASD?

Answers to the first research question are presented in Figure 2. A total of 7 different AI techniques were identified during the research process.

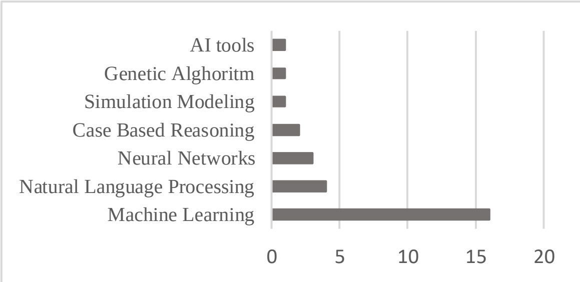

Figure 2. Identified AI techniques and tools used in the context of ASD

The primary AI technique that SE frequently employs is machine learning (ML), which appeared in 16 papers. ML was applied both independently or integrated with other AI techniques to create hybrid techniques. While supervised ML algorithms (e.g., Decision Tree, Linear regression, Logistic Regression, Support Vector Machine, Naïve Bayes, Predictive Analytics, etc.) were mostly used to improve the accuracy of predictions, unsupervised ML algorithms (e.g., K-Means Clustering, KNearest Neighbor, K-medoids, etc.) were used less often, mostly for data segmentation, and usually integrated with supervised ML algorithms. Decision Trees were identified as the most commonly used AI technique (6 papers). Natural Language Processing techniques followed, with 4 papers reporting on this AI technique. This technique was mostly used to automate the derivation of textual information, mainly as a standalone technique. Neural networks were mentioned in 3 papers, Case Based Reasoning in 2 papers, while other AI techniques (Simulation Modeling, Genetic Algorithm) and AI tools (Google Coral devices, TFL mobile SSD object recognition, DLIB correlation tracking) were mentioned in 1 paper.

3.2 For which purposes are AI techniques and tools used in ASD?

As for the second research question, several different purposes of AI techniques and tools were identified, as shown in Figure 3.

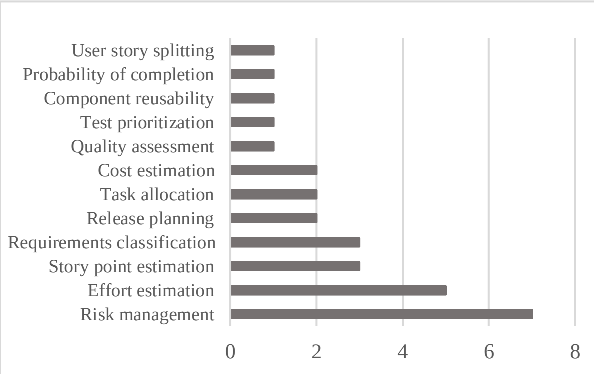

Figure 3. Identified purposes of AI techniques and tools used in the context of ASD

Risk management was identified as the most frequently reported purpose of using AI (7 papers), which is not surprising since the risk of project failure, and time and budget overruns are among the most relevant problems of ASD. Various AI techniques were used to assess the risk, predict failures, prioritize issues, automate the risk management process, and to detect software vulnerabilities. Another almost equally important purpose was effort estimation (5 papers). As the set of requirements for the software to be developed is usually incomplete, effort estimation is uncertain activity, and the use of AI techniques can help improve its accuracy. For similar reasons, AI techniques were used in story point estimation (3 papers). AI techniques were also found to be useful for the purposes of requirements classification (3 papers), release planning (2 papers), task allocation (2 papers), and cost estimation (2 papers). Finally, several different purposes were mentioned only once, including component reusability, quality assessment, test prioritization, probability of completion of features, and user story splitting.

3.3 What are the benefits of using AI techniques and tools in ASD?

The answers to the third research question are shown in Figure 4.

There are 10 groups of identified benefits of using AI in ASD. Improvement of the prediction accuracy (of costs, time, effort, failures, task allocation, etc.) was identified as the most reported benefit of AI techniques and tools in the context of ASD (14 papers), followed by decreased risk (5 papers), and decreased development time (4 papers). The decreased cost was stated as a benefit in 3 papers, followed by improved performance and process support which were stated as benefits in 2 papers. Finally, increased value, improved planning, improved quality, and process automation were reported only once as benefits of using AI in ASD.

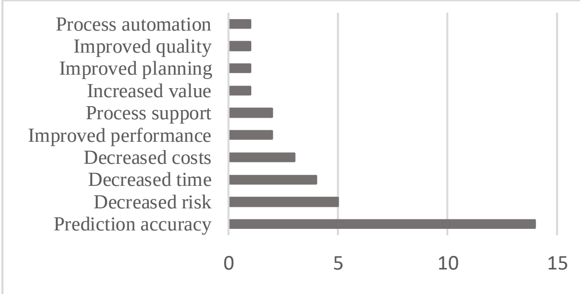

Figure 4. Identified benefits of using AI techniques and tools in the context of ASD

4 Discussion

The value of this paper lies in the identification of the types, purposes, and benefits of AI techniques used in ASD. Contrary to the similar literature reviews which were found during the identification of the need for this research (see Introduction), this paper is the first that tackles agile software development. Since traditional software development (TSD) depends on comprehensive execution of each phase by covering all aspects of the system which is under construction, the choice of AI techniques differs from those that are used to support ASD (Kulkarni & Padmanabham, 2017). As far as we are aware, this literature review is the first that distinguishes agile from traditional software development, as well as the first that deals with AI techniques and tools used within ASD. However, it is important to mention that this paper does not discuss the difference between the AI techniques used in both types of software development, since its primary purpose is narrower and more limited to ASD.

According to the results of the literature review, a variety of AI techniques have been adopted and successfully implemented in ASD. As stated in the previous section, machine learning is without a doubt the most frequently employed AI technique. ML was used for various purposes, such as risk management, effort estimation, story point estimation, requirements classification, task allocation, etc. It was reported that ML has seriously improved prediction accuracy related to cost and effort, as well as decreased risk of failures. On the other hand, other AI techniques were used more rarely, mainly because of their limited scope of impact, but also because of the limited research related to their use within software development. Namely, while ML is well-known and widely recognized, the rise of NLP and neural networks within software development has just recently begun. However, thanks to the promising results of their use in ASD, the expansion of the scope of their use is expected in the next few years.

Table 2. Systematization of results

AI techniques and tools	Purposes	Benefits	References
Machine Learning (Decision Tree, Random Forest and AdaBoost)	Effort estimation, Cost estimation	Prediction accuracy (effort and cost)	(Rodríguez Sánchez et al., 2023)
Machine Learning (Predictive analytics)	Risk management	Prediction accuracy (failures)	(Batarseh & Gonzalez, 2018)
Machine Learning	Quality assessment	Process support, improved quality	(Poth et al., 2019)
Genetic Algorithm	Release planning	Increased business value	(A. Kumar et al., 2014)
Google Coral devices, TFL mobile SSD object recognition, DLIB correlation tracking	Cost estimation	Decreased costs	(Jüngling et al., 2020)
Natural Language Processing (NLP)	Release planning	Improved planning	(Sharma & Kumar, 2019)
Natural Language Processing (NLP)	Requirements classification	Decreased time and risk	(Almanaseer et al., 2022)
Machine Learning (Decision Tree, Stochastic Gradient, Boosting, and Random Forest)	Effort estimation	Prediction accuracy	(Satapathy & Rath, 2017)
Machine Learning (Logistic Regression)	Probability of completion	Prediction accuracy (time)	(Ameta et al., 2022)
Machine Learning	Risk management	Decreased risk, improved process	(Khanna et al., 2021)
Machine Learning (Random Forest, Decision tree, SGB, and Neural Networks)	Effort estimation	Prediction accuracy (effort estimation)	(S. Kumar et al., 2022)
Machine Learning (Support Vector Machine, Logistic Regression)	Task allocation	Prediction accuracy (task allocation)	(William et al., 2021)
Software Process Simulation Modeling (Monte Carlo simulation)	Risk management	Decreased risk, improved process	(Lunesu et al., 2021)
Machine Learning (K-medoids, and K-means Clustering)	User story splitting	Decreased time, improved performance	(B. Kumar et al., 2022)
Machine Learning	Task allocation	Prediction accuracy (task allocation)	(Singh et al., 2021)
Bi-directional Long Short-Term Memory (BiLSTM), Tree-structured Recurrent Neural Network (RNN), and Convolutional Neural Network (CNN)	Story point estimation	Prediction accuracy (story points)	(Marapelli et al., 2020)
Machine Learning (k-Nearest Neighbors algorithm, Linear Regression)	Effort estimation	Prediction accuracy (effort estimation)	(Sanchez et al., 2022)
Case based reasoning	Risk management	Decreased risk	(Jabeen et al., 2014)
Case Based Reasoning	Risk management	Decreased risk, decreased time	(Mukhtar et al., 2013)
Machine Learning (J48 Decision Tree Classifier)	Test prioritization	Prediction accuracy (defects)	(Butgereit, 2019)
Machine Learning (Deep-SE)	Story point estimation	Prediction accuracy (story points)	(Abadeer & Sabetzadeh, 2021)
Graph Neural Network Text Classification	Story point estimation	Prediction accuracy (story points)	(Phan & Jannesari, 2022)
Natural Language Processing (NLP)	Risk management	Decreased risk	(Shafiq et al., 2021)
Machine Learning	Component reusability	Prediction accuracy (reusability)	(Khan & Lingala, 2022)
Machine Learning (Text Classification Algorithm)	Requirements classification	Improved performance	(Magalhaes et al., 2021)
Natural Language Processing (NLP) and Machine learning	Requirements classification	Process automation	(Lano et al., 2021)
Machine Learning (Naïve Bayes, J48 Decision Tree, Random Forest, Logistic Model Tree)	Effort estimation	Prediction accuracy (effort estimation)	(Moharreri et al., 2016)
Graph Method, Deep Neural Network	Risk management	Process support, decreasing risk	(Srivastava et al., 2022)

During the research process, some major concerns were detected. First of all, only one paper has reported the use of AI tools (Google Coral devices, TensorFlow Lite (TFL) mobile Single Shot Detection (SSD) object recognition, and Dlib correlation tracking) (Jüngling et al., 2020). All other papers were experimenting with the use of different AI techniques on specific activities within the ASD, but without providing concrete tools.

To benefit from this kind of research, businesses should first implement AI techniques according to the stated guidelines, and then monitor and measure the progress of implementation. Since this task is time and resource-consuming, some businesses might be discouraged to get it done. To overcome this obstacle, concrete AI tools should be provided in future research. Secondly, most of the time, the purpose of the AI technique was isolated, i.e., the focus and application of a specific AI technique were very narrow. Out of 28 research papers that were included in the phase of data extraction, 26 of them were using one or two types of AI techniques (see Table 2) for specific activities of the ASD process, e.g., effort estimation, task allocation, prediction accuracy, test prioritization, etc. Only two papers were dealing with the processes of software development and the impact of AI techniques on the ASD process. One of them (Poth et al., 2019) has determined the improvement of the quality of the test process by using ML as a support technique, while the other (Jüngling et al., 2020) has shown how AI tools can be used in the system development lifecycle to save costs. Finally, while the coverage of some activities is satisfactory, other activities (e.g., backlog management, user story development, etc.) are completely neglected. As a consequence, the impact of AI on the whole agile software development process is still unknown. To help overcome detected concerns, our future work should focus on the development of AI supported framework for agile software development. In contrast to the stated benefits of previous research, which were related to specific individual activities of ASD, the purpose of AI supported framework would be to improve the quality of the agile software development process.

5 Conclusion

This paper provided a preliminary overview of AI techniques and tools used in agile software development. It also described the purposes for which AI techniques are used in the context of ASD, as well as the benefits they provide for businesses and users. The results of this paper will be mostly valuable for researchers, as a starting point to further investigate AI techniques and tools used in ASD, but also for detecting gaps and promising areas of future research.

The paper's findings demonstrate the diversity of the literature on this topic, describing several AI methods used in ASD. The majority of the papers that were analysed focused on the use of ML techniques within specific activities of the software development process. Also, a great share of papers was dealing with the small set of activities of the software development process (e.g., risk management, agile estimating, task planning, etc.). It is therefore assumed that many more activities of ASD could greatly benefit from the adoption of AI techniques.

During the literature review, a lack of AI tools and a lack of papers dealing with the impact of AI techniques on the ASD process were detected. To help overcome the detected concerns, the authors strive to design and develop AI supported framework for agile software development. Namely, ASD is more than a collection of activities and techniques. It consists of processes, which encompass different events, roles, and artifacts.

Such framework could help the software development teams to maximize the potential of the agile development by maximizing the strengths and opportunities in the team and given context as well as helping to reduce the weaknesses and threats during the whole process. The journey in design and development of such framework is ahead of us, and to determine the overall impact of AI on ASD, the impact of AI on different components of the process should also be determined.

As we have determined that specific AI tools that can be practically applied are lacking, future research could focus on developing concrete AI tools and examining their impact on the entire ASD process. The integration of these tools into the mentioned framework would also be possible and highly desirable.

References

Abadeer, M., & Sabetzadeh, M. (2021). Machine Learning-based Estimation of Story Points in Agile Development: Industrial Experience and Lessons Learned. *2021 IEEE 29th International Requirements Engineering Conference Workshops (REW)*, 106–115. https://doi.org/10.1109/REW53955.2021.00022

Almanaseer, A. M., Alzyadat, W., Muhairat, M., Al-Showarah, S., & Alhroob, A. (2022). A proposed model for eliminating nonfunctional requirements in Agile Methods using natural language processes. *2022 International Conference on Emerging Trends in Computing and Engineering Applications (ETCEA)*, 1–7. https://doi.org/10.1109/ETCEA57049.2022.10009796

Ameta, U., Patel, M., & Sharma, A. K. (2022). Scaled Agile Framework Implementation in Organizations', its Shortcomings and an AI Based Solution to Track Team's Performance. *2022 IEEE 3rd Global Conference for Advancement in*

Technology (GCAT), 1–7. https://doi.org/ 10.1109/GCAT55367.2022.9971968

Batarseh, F. A., & Gonzalez, A. J. (2018). Predicting failures in agile software development through data analytics. *Software Quality Journal*, *26*(1), 49–66. https://doi.org/10.1007/s11219-015-9285-3

ben Othmane, L., Angin, P., Weffers, H., & Bhargava, B. (2014). Extending the agile development process to develop acceptably secure software. *IEEE Transactions on Dependable and Secure Computing*, *11*(6), 497–509.

Biesialska, K., Franch, X., & Muntés-Mulero, V. (2021). Big Data analytics in Agile software development: A systematic mapping study. *Information and Software Technology*, *132*, 106448.

Butgereit, L. (2019). Using Machine Learning to Prioritize Automated Testing in an Agile Environment. *2019 Conference on Information Communications Technology and Society (ICTAS)*, 1–6. https://doi.org/10.1109/ICTAS.2019.8703639

Jabeen, J., Motla, Y. H., Abbasi, M. A., Batool, D.-B., Butt, R., Nazir, S., & Anwer, S. A. (2014). Incorporating artificial intelligence technique into DSDM. *Asia-Pacific World Congress on Computer Science and Engineering*, 1–8. https://doi.org/10.1109/APWCCSE.2014.7053838

Jüngling, S., Peraic, M., & Martin, A. (2020). *Towards AI-based Solutions in the System Development Lifecycle*.

Khan, F., & Lingala, G. (2022). Machine Learning Techniques For Software Component Reusability. *2022 3rd International Conference for Emerging Technology (INCET)*, 1–6.https://doi.org/10.1109/ INCET54531.2022.9824063

Khanna, E., Popli, R., & Chauhan, N. (2021). Artificial Intelligence based Risk Management Framework for Distributed Agile Software Development. *2021 8th International Conference on Signal Processing and Integrated Networks (SPIN)*, 657–660. https://doi.org/10.1109/SPIN52536.2021.9566000

Kulkarni, R. H., & Padmanabham, P. (2017). Integration of artificial intelligence activities in software development processes and measuring effectiveness of integration. *IET Software*, *11*(1), 18–26. https://doi.org/10.1049/iet-sen.2016.0095

Kumar, A., Nagar, R., & Baghel, A. S. (2014). A genetic algorithm approach to release planning in agile environment. *2014 International Conference on Information Systems and Computer Networks (ISCON)*, 118–122. https://doi.org/10.1109/ICISCON.2014.6965230

Kumar, B., Tiwari, U., & Dobhal, D. C. (2022). User Story Splitting in Agile Software Development using Machine Learning Approach. *2022 Seventh International Conference on Parallel, Distributed and Grid Computing (PDGC)*, 167–171. https://doi.org/10.1109/PDGC56933.2022.10053226

Kumar, S., Arora, M., Sakshi, & Chopra, S. (2022). A Review of Effort Estimation in Agile Software Development using Machine Learning Techniques. *2022 4th International Conference on Inventive Research in Computing Applications (ICIRCA)*, 416–422. https://doi.org/10.1109/ICIRCA54612.2022.9985542

Lano, K., Yassipour-Tehrani, S., & Umar, M. A. (2021). Automated Requirements Formalisation for Agile MDE. *2021 ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C)*, 173–180. https://doi.org/10.1109/MODELS-C53483.2021.00030

Lunesu, M. I., Tonelli, R., Marchesi, L., & Marchesi, M. (2021). Assessing the Risk of Software Development in Agile Methodologies Using Simulation. *IEEE Access*, *9*, 134240–134258. https://doi.org/10.1109/ACCESS.2021.3115941

Magalhaes, C., Araujo, J., & Sardinha, A. (2021). MARE: An Active Learning Approach for Requirements Classification. *2021 IEEE 29th International Requirements Engineering Conference (RE)*, 516–521. https://doi.org/10.1109/RE51729.2021.9714537

Marapelli, B., Carie, A., & Islam, S. M. N. (2020). RNN-CNN MODEL:A Bi-directional Long Short-Term Memory Deep Learning Network For Story Point Estimation. *2020 5th International Conference on Innovative Technologies in Intelligent Systems and Industrial Applications (CITISIA)*, 1–7. https://doi.org/10.1109/ CITISIA50690.2020.9371770

Moharreri, K., Sapre, A. V., Ramanathan, J., & Ramnath, R. (2016). Cost-Effective Supervised Learning Models for Software Effort Estimation in Agile Environments. *2016 IEEE 40th Annual Computer Software and Applications Conference (COMPSAC)*, 135–140. https://doi.org/10.1109/COMPSAC.2016.85

Mukhtar, M., Motla, Y. H., Riaz, M., Khan, M. A., Ahmed, M., Abbas, M. A., Naz, H., & Batool, A. (2013). A hybrid model for agile practices using case based reasoning. *2013 IEEE 4th International Conference on Software Engineering and Service Science*, 820–823. https://doi.org/10.1109/ICSESS.2013.6615431

Perkusich, M., Chaves E Silva, L., Costa, A., Ramos, F., Saraiva, R., Freire, A., Dilorenzo, E., Dantas, E., Santos, D., Gorgônio, K., Almeida, H., &

Perkusich, A. (2020). Intelligent software engineering in the context of agile software development: A systematic literature review. *Information and Software Technology*, *119*, 106241. https://doi.org/10.1016/j.infsof.2019.106241

Phan, H., & Jannesari, A. (2022). Story point level classification by text level graph neural network. *Proceedings of the 1st International Workshop on Natural Language-Based Software Engineering*, 75–78. https://doi.org/10.1145/3528588.3528654

Poth, A., Beck, Q., & Riel, A. (2019). Artificial Intelligence Helps Making Quality Assurance Processes Leaner. In A. Walker, R. V. O'Connor, & R. Messnarz (Eds.), *Systems, Software and Services Process Improvement* (Vol. 1060, pp. 722–730). Springer International Publishing. https://doi.org/10.1007/978-3-030-28005-5_56

Rodríguez Sánchez, E., Vázquez Santacruz, E. F., & Cervantes Maceda, H. (2023). Effort and Cost Estimation Using Decision Tree Techniques and Story Points in Agile Software Development. *Mathematics*, *11*(6), 1477.

Sanchez, E. R., Maceda, H. C., & Santacruz, E. V. (2022). Software Effort Estimation for Agile Software Development Using a Strategy Based on k-Nearest Neighbors Algorithm. *2022 IEEE Mexican International Conference on Computer Science (ENC)*, 1–6. https://doi.org/10.1109/ENC56672.2022.9882947

Satapathy, S. M., & Rath, S. K. (2017). Empirical assessment of machine learning models for agile software development effort estimation using story points. *Innovations in Systems and Software Engineering*, *13*(2–3), 191–200. https://doi.org/10.1007/s11334-017-0288-z

Shafiq, S., Mashkoor, A., Mayr-Dorn, C., & Egyed, A. (2021). NLP4IP: Natural Language Processing-based Recommendation Approach for Issues Prioritization. *2021 47th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, 99–108. https://doi.org/10.1109/SEAA53835.2021.00022

Sharma, S., & Kumar, D. (2019). Agile Release Planning Using Natural Language Processing Algorithm. *2019 Amity International Conference on Artificial Intelligence (AICAI)*, 934–938. https://doi.org/10.1109/AICAI.2019.8701252

Shehab, M., Abualigah, L., Jarrah, M. I., Alomari, O. A., & Daoud, M. Sh. (2020). (AIAM2019) Artificial Intelligence in Software Engineering and inverse: Review. *International Journal of Computer Integrated Manufacturing*, *33*(10–11), 1129–1144. https://doi.org/10.1080/0951192X.2020.1780320

Singh, M., Chauhan, N., & Popli, R. (2021). Framework for Machine Learning Based Task Allocation in DASD Environment. *2021 Fourth International Conference on Computational Intelligence and Communication Technologies (CCICT)*, 29–34. https://doi.org/10.1109/CCICT53244.2021.00017

Sofian, H., Yunus, N. A. M., & Ahmad, R. (2022). Systematic Mapping: Artificial Intelligence Techniques in Software Engineering. *IEEE Access*, *10*, 51021–51040. https://doi.org/10.1109/ACCESS.2022.3174115

Srivastava, D. K., Makhija, R., & Batta, A. (2022). Software Vulnerabilities Detection in Agile Process using graph method and Deep Neural Network. *2022 International Conference on Advancements in Smart, Secure and Intelligent Computing (ASSIC)*, 1–7. https://doi.org/10.1109/ASSIC55218.2022.10088314

Templier, M., & Paré, G. (2015). A Framework for Guiding and Evaluating Literature Reviews. *Communications of the Association for Information Systems*, *37*. https://doi.org/10.17705/1CAIS.03706

William, P., Kumar, P., Chhabra, G. S., & Vengatesan, K. (2021). Task Allocation in Distributed Agile Software Development using Machine Learning Approach. *2021 International Conference on Disruptive Technologies for Multi-Disciplinary Research and Applications (CENTCON)*, 168–172. https://doi.org/10.1109/CENTCON52345.2021.9688114