AI-Enhanced Scrum: A Literature Review on the Impact of Artificial Intelligence on Scrum Processes

Dijana Peras*[1], Luis de-Marcos[2], and Zlatko Stapić[1]
[1]Department of Information Systems Development, University of Zagreb Faculty of Organization and Informatics Pavlinska 2, 42000 Varaždin, Croatia
[2]Computer Science Department, Polytechnik School, University of Alcalá, Dpto Ciancia Computación, Edificio Politécnico, Ctra Barcelona km 33.1, 28871 Alcalá de Henares, Madrid, Spain
E-mail: dperas@foi.unizg.hr, luis.demarcos@uah.es, zstapic@foi.unizg.hr
*Corresponding author

Keywords: Agile development, AI, artificial intelligence, process enhancement, process improvement, Scrum

Received: July 31, 2025

As artificial intelligence (AI) continues to evolve, its role in software development becomes increasingly relevant. In agile environments, particularly within the Scrum framework, AI shows the potential to improve accuracy, productivity, and overall efficiency. Several Scrum related challenges, such as effort estimation or resource allocation, could be solved by applying AI. Despite the potential, many organizations still struggle with successful AI adoption, mainly due to limited knowledge about AI capabilities and challenges in selecting appropriate AI techniques. This study explores the current state of AI integration in Scrum processes through a literature review. A literature search identified 305 records, of which 18 were included in review. The study examines:(a) how much AI has impacted Scrum, (b) which AI techniques have been used to enhance Scrum processes, and (c) how AI affects automation, assistance, and enhancement within Scrum. Data extraction and coding were conducted using a content analysis, while the synthesized findings were interpreted and discussed using a narrative synthesis approach. The analysis identified seven types of AI techniques (regression, classification, clustering, natural language processing, neural networks, search algorithms, and topic modeling) applied across seven out of nineteen Scrum processes. Most AI support was observed in the Plan and Estimate phase, particularly in effort estimation, user story creation, and release planning. In contrast, no applications were identified for processes in the Implement or Review and Retrospect phases. The impact of AI was categorized into enhancement, assistance, and automation, with a balanced distribution across the categories. The findings suggest that current AI applications in Scrum are concentrated on structured, data-rich processes.

Povzetek: Študija povzema, da se umetna inteligenca v Scrumu trenutno uporablja predvsem za podporo strukturiranim procesom, kot so ocenjevanje napora, uporabniške zgodbe in načrtovanje izdaj, medtem ko je njena uporaba v drugih fazah še omejena.

1 Introduction

Today, artificial intelligence (AI) is used across numerous fields and industries, with its applications continuously expanding. AI is transforming the way people work and has the potential to enhance business processes, making them more efficient. According to Gartner [1], by 2027, 24% of global organizations will be in the planning stage of AI adoption and will begin implementing the most attractive AI use cases. Therefore, organizations seeking a competitive advantage must seriously consider integrating AI into their operations. However, many organizations struggle to implement AI successfully, and a large number of AI initiatives fail to take off [2].

One area where AI has a significant impact is software development. Here, AI could automate tasks, improve project management, enhance code quality, speed up software development, help with decision-making, etc. [3]. Currently, the most popular framework for agile software development is Scrum, adopted by 87% of organizations engaged in agile practices [4]. Scrum encourages a structured way of working that fits with an organization's strategies and culture. It also emphasizes communication, continuous improvement, and transparency [5]. It follows an iterative, incremental approach to increase predictability and reduce risks. Complex tasks are broken down into smaller, easier-to-handle components that can be completed within a month or less. Because of its flexibility and adaptability, Scrum is used across various industries.

AI's abilities in data analysis, pattern recognition, and natural language processing can assist with common tasks within Scrum [6]. AI could assist in the decision-making process, automate repetitive tasks, and provide insights based on historical data. All of these could lead to higher productivity, improved outcomes, and more efficient utilization of resources [7]. By overcoming the drawbacks of manual procedures, the integration of AI within Scrum attempts to create more effective and streamlined software development processes.

The applications of AI in agile software development have been studied extensively, and several articles have been published on this topic [8], [9], [10]. However, while agile methodologies such as Scrum, Kanban, and Extreme Programming share common principles, their structure, roles, ceremonies, and artifacts are very different [11]. How effective AI will be depends on the processes that need to be improved and the problems related to those processes. Therefore, AI applications have to be tailored to the specific needs of each agile methodology. Unlike other agile methodologies, Scrum has a clear process and a high degree of structure. This makes it easier to predict outcomes and identify the right AI solutions. At the same time, Scrum's well-defined roles, artifacts, and time-boxed events constrain when and how AI techniques can be introduced, making it more difficult to experiment with AI solutions. Furthermore, Scrum's time-boxed nature makes it necessary to plan and deliver product on a regular basis. Therefore, AI models might need more time and more historical data to be trained, and they might not deliver insights within a Sprint cycle. Scrum also values team autonomy, so improving tasks like effort estimation or prioritization should be carefully planned so as not to interfere with team ownership. Finally, Scrum relies on periodic inspections and modifications based on outcomes. Since the Product Backlog is constantly changing, AI solutions need to be able to adapt. Given these unique challenges, the paper focuses exclusively on the application of AI within Scrum.

To make the most of the opportunities presented, it is important to understand how to structurally integrate AI into Scrum. However, despite growing interest and individual case studies, there is still no systematic analysis that shows how AI techniques can be used within Scrum processes, identifies the factors that influence the choice of a specific AI technique in a given situation, and explores how Scrum could be improved through the integration of AI. This paper explores the most recent advances in using AI in Scrum and points out areas where more research is needed to fill the research gaps. It aims to determine the specific Scrum processes that AI can enhance, along with the corresponding input and output (data) for each type of identified AI technique, and the extent of AI's impact on Scrum (e.g., automation, enhancement, assistance). Also, the gaps in current research and possible future directions are highlighted. This approach provides a clear understanding of both the possibilities and limitations of using AI in Scrum. Given that the effective application of AI technologies is essential for long-term competitiveness and that many organizations continue to struggle with the successful integration of AI, it is crucial to explore how organizations can integrate AI into their Scrum processes. The main contributions of the study are:

1) an overview of commonly used AI techniques and their alignment with specific processes, input, and output data within Scrum,
2) grouping of AI techniques into broader categories of AI types to allow greater flexibility in technology selection, and
3) an analysis of the trends and future work in the domain of AI-supported Scrum.

The rest of this paper is structured as follows: After a brief introduction in the first section, the second section presents the materials and methods, including the research questions. The third section outlines the research results, while the fourth section is devoted to discussion. The fifth section highlights the research implications and future directions. Finally, the main conclusions are presented.

2 Materials and methods

A literature review was conducted to:

a) assess the current state of efforts in AI-supported Scrum processes,
b) identify the input and output of AI techniques used to improve the Scrum process, and
c) determine the impact of AI on the Scrum process.

Scrum processes were predefined according to the Scrum Body of Knowledge (SBOK™) [11], which is an official manual and set of guidelines for implementing Scrum. The document contains standardized principles, processes, and best practices that Scrum teams and organizations can use to effectively manage projects. Therefore, it is considered a relevant source for the identification of Scrum processes. AI techniques were allowed to emerge through the research. The literature search was performed in January 2025.

2.1 Definitions of key terms

The terms "enhancement," "assistance," and "automation" in the context of AI were defined to ensure clarity and consistency throughout the analysis. These terms refer to different levels of AI involvement in the Scrum process. **Enhancement** means improving or augmenting current processes to make them more effective or efficient without substantially changing how they are carried out. AI acts as an augmenting tool that supports informed execution of tasks. **Assistance** involves AI techniques acting as a support tool for decision-making and planning. Here, AI contributes to the process, while human remains responsible for interpreting the output and making final decisions. **Automation** refers to AI performing tasks with no human involvement. AI executes predefined steps independently based on available inputs. It is mostly found in processes that are repetitive or rule-based.

2.2 Data extraction

To achieve the objectives, three research questions were formulated:

- RQ1: What is the current status of advancements in AI-supported Scrum processes?
- RQ2: What AI techniques have been used to optimize the Scrum process based on the available input and output data?
- RQ3: How has AI improved the Scrum process in terms of automation, assistance, and enhancement?

To ensure the quality and relevance of the selected literature, sources were required to return a substantial number of relevant papers, be accessible to the researchers, and relate closely to the research topic. The search was limited to publications in English and within the field of software engineering. The search string combined AI-related terms with Scrum-related terms and was used to query the Scopus, Web of Science (WoS), and IEEE Xplore databases:

- (“artificial intelligence” OR “AI” OR “AI enhancement” OR “AI process improvement” OR “machine learning” OR “NLP”) AND (“Scrum” OR “agile development”) AND (“project vision” OR “stakeholder* identification” OR “Scrum team*” OR “form team*” OR “epic*” OR “product backlog” OR “release planning” OR “user stories” OR “task creation” OR “task estimation” OR “Sprint backlog” OR “task allocation” OR “issues prioritization” OR “daily standup” OR “Sprint review” OR “Sprint retrospect” OR “retrospect project” OR “reporting”)

In the initial screening phase, studies were included if their titles or abstracts explicitly indicated a focus on improving the Scrum process through AI applications. Additionally, papers whose relevance was not immediately clear from the title or abstract but showed potential were retained for further review. Studies were excluded if their titles or abstracts indicated only marginal relevance to the research objectives. Studies that satisfied those criteria, as well as those with uncertain titles and abstracts, proceeded to the second phase.

The following inclusion criteria were applied on studies that proceeded to the second phase:

- explicit description of the application of AI to improve Scrum processes;
- improvement of at least one Scrum process through the use of AI;
- clear identification of the applied AI techniques;
- description of the impact of AI on the Scrum process (e.g., automation, assistance, or enhancement);
- specification of the input and output data associated with applied AI technique.

Papers were excluded if they:

- discussed AI hypothetically but did not implement any concrete AI techniques;
- lacked methodological detail on AI technique implementation;
- were secondary studies (e.g., surveys, position papers) without original empirical contribution.

The papers that passed this phase were thoroughly read. Information extracted from each included study covered the type of AI used, the input and output data, the specific Scrum processes that were enhanced, and the nature of the AI contribution.

The literature identification, screening, and selection process is summarized using a flow diagram, as shown in Figure 1.

2.3 Data analysis

A comprehensive content analysis of each paper was conducted by one researcher, and reviewed by second researcher. The extracted data were coded and grouped into higher-level categories that aligned with the research questions. Codes represented well known concepts such as the types of AI techniques (e.g., NLP, regression, search algorithms), the type of input data (textual, numerical), and the Scrum phase of the involved process. The categorized information was synthesized narratively to identify relationships between AI techniques, types of data, and Scrum processes. All coded properties of the included studies are reported explicitly in Table 3 and Table 4, which represent the structured outcome of the content analysis. The columns in this dataset included: 1) the Scrum process name, 2) the AI technique name, 3) the type of AI technique, 4) the input data (type and source), 5) the output of the AI technique, 6) the AI contribution (e.g., assistance, enhancement, automation), and 7) the source reference. The rows in this dataset were organized according to the processes and phases of the Scrum framework, as presented in Table 1.

A second researcher independently reviewed the extracted data to verify completeness and accuracy. Where ambiguous or unclear interpretations were identified, the second researcher proposed corrections, which were then jointly reviewed and incorporated when appropriate.

The final results are presented in the *Results* section of the study. They are discussed in detail through textual explanations and summarized in tables to provide a clear and comparative overview of the findings.

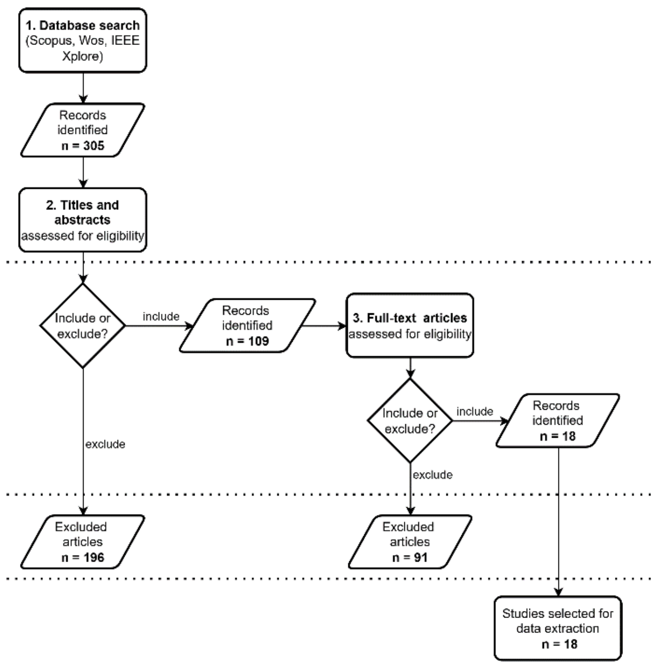

Figure 1: Flow diagram of the research methodology

Table 1: Scrum processes by phases

Phase	Process
Initiate	Create Project Vision Identify Scrum Master and Stakeholder(s) Form Scrum Team Develop Epic(s) Create Prioritized Product Backlog Conduct Release Planning
Plan and estimate	Create User Stories Approve, Estimate, and Commit User Stories Create Tasks Estimate Tasks Create Sprint Backlog
Implement	Create Deliverables Conduct Daily Standup Groom Prioritized Product Backlog
Review and Retrospect	Convene Scrum of Scrums Demonstrate and Validate Sprint Retrospect Sprint Ship Deliverables Retrospect Project

3 Results

The defined search string returned a total of 305 papers (Scopus: 66, WoS: 23, and IEEE: 216), which were initially reviewed based on their titles and abstracts. After the preliminary review, 196 papers were excluded for not meeting the initially defined inclusion and exclusion criteria. The remaining 109 papers that met those criteria advanced to the next stage of selection, where additional inclusion and exclusion criteria were applied. A full-text screening of these papers resulted in a final set of 18 papers included in the literature review. The next phase of the research involved the analysis and synthesis of extracted information from the final set of papers. The research findings are presented according to the stated research questions.

3.1 RQ1. What is the current status of advancements in AI-supported Scrum processes?

Across the reviewed studies, a clear pattern emerged showing that AI applications are highly explored in the *Plan and Estimate* phase, while *Implement* and *Review and Retrospect* phases remain largely unexplored. In total, seven different types of AI were identified: regression, classification, clustering, natural language processing (NLP), neural networks (NN), search algorithms, and topic modeling. Additionally, some studies applied a combination of techniques, such as classification and regression or NLP and classification.

A total of 7 out of 19 Scrum processes were found to be supported by AI. *Task Estimation* and *Conduct Release Planning* were the most studied processes, each appearing in four papers. These were followed by *Create User Stories* (three papers) and *Approve, Estimate, and Commit User Stories* (three papers). Other processes (*Form Scrum Team*, *Create Prioritized Product Backlog*, and *Create Sprint Backlog*) were mentioned once. The following paragraphs will provide an overview of AI-supported Scrum processes categorized by the phases of the Scrum framework: *1) Initiate, 2) Plan and estimate, 3) Implement,* and *4) Review and retrospect*.

AI had the most significant role in Scrum processes within the *Plan and Estimate* phase. Four out of five processes of these phases were identified as affected by AI. The remaining three processes were identified as part of the *Initiate phase*. In Scrum processes related to the *Implement* phase, as well as the *Review and Retrospect phase*, the use of AI was not detected.

In Scrum processes within the *Initiate phase*, AI was used to create teams [12] and classify product backlog items (PBIs) [13]. This improved backlog management efficiency and helped in prioritization. More specifically, in order to maximize competence match and redundancy among team members, the Genetic Algorithm (GA) was used to find the best possible team formation. As evidenced by the high accuracy and acceptance rate of the teams created using this technology, AI has the potential to increase the efficacy and efficiency of team formation in Scrum. Release planning, which is frequently a manual process, was also automated with AI. Using historical data, AI techniques have been utilized to forecast project milestones and release dates [14]. Additionally, AI-powered chatbots were employed to assign resources and estimate the amount of time required to finish tasks [15]. Similar user stories have been identified and grouped into releases using NLP [16], [17]. This way, the manual challenge of organizing related stories into a cohesive release was directly addressed. Empirical results demonstrated significant efficiency gains.

AI improved the creation of user stories during the *Plan and Estimate* phase by creating models for assessing text and graphical documents and automatically producing preliminary versions of user stories according to requirement specifications [18], [19], [20]. The goal of this capability is to reduce the amount of time and effort team members usually have to invest in turning statements into tasks. The generated user stories include key attributes, a definition of done, functional and non-functional constraints, and test specifications, and are formatted for easy integration into project management tools.

AI helped make story point estimation more accurate and faster. Different combinations of NLP, NN, and ML were used for this purpose [21], [22], [23], [24], [25], [26], [27], [28], allowing AI to learn from the content of user stories and improve estimation performance. AI techniques have made effort and cost estimation much more accurate and consistent than traditional human-based methods. The findings across sources indicate that no single ML technique is better than the others at predicting effort. This strongly supports the efficacy of ensemble-based models, which use predictions from multiple algorithms to improve results. One of the most important things to understand is the importance of data quality: the level of detail and the number of vague terms in user stories significantly impact the model's effectiveness. Stories that had higher estimation errors were often associated with vague terminology.

Suggestions for improvement include reducing implicit information, using standardized templates for stories, and regularly grooming the backlog [21]. Also, models were trained on labeled historical data. For building reliable models, the quality and completeness of textual requirements are essential. Data preprocessing techniques, like discretization of continuous features and data augmentation, evidently make predictions more accurate and help models learn. This helps managers to explore different project scenarios more effectively.

AI helped create sprint backlogs by assigning tasks to team members based on their skills and availability [29]. This created the most competent Scrum teams for each project. The dataset used for prediction included the employee's service year, training year, KPI score (the number of projects the employee worked on), previous year's rating, education, and awards. For the model to be successful, it is crucial to have this kind of employee data.

The results are presented in the form of a matrix (Table 2), which shows the mapping between Scrum processes and the types of AI techniques applied in the reviewed studies. The rows represent Scrum processes, while the columns represent AI technique types. The numbers in cells indicate the number of reviewed studies in which a specific AI technique was applied to the Scrum process. This overview highlights the concentration of research across Scrum phases and AI technique types.

Figure 2 provides a visual synthesis of AI supported Scrum processes across phases, clearly indicating the concentration of existing research in early planning activities and the lack of AI applications in later phases.

3.2 RQ2. What AI techniques have been used to optimize the Scrum process based on the available input and output data?

Across the studies, the choice of AI technique depended on the type and structure of available data, with NLP being mostly applied to textual data and ML being mostly applied to numerical data. Generally, AI technologies require a large amount of data. The input data consists of prepared information obtained from the environment, which influences the choice of AI technique. Apart from the type and format of input data, the selection of an AI technique also depends on the organization's specific requirements for the process, including the desired output, which may be in textual, numerical, or vector format.

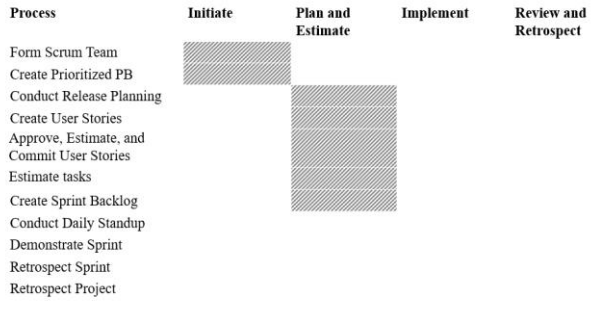

Figure 2: Mapping AI support across scrum phases and processes

Overall, the reviewed papers provided a comprehensive preview of how different AI techniques can be applied to various stages of the Scrum process. They detailed the input data used, the output generated, and the specific AI technique employed in each Scrum process. AI techniques were further categorized into broader types of AI techniques. Understanding these AI types can be more valuable than focusing solely on specific techniques, as it provides a broader perspective on AI's capabilities and limitations, leading to better strategic implementation. Identified AI techniques are a mixture of traditional (e.g., ML, Search algorithms) and advanced (NLP, NN) AI techniques. The choice of a specific AI technique was conditioned by the type of available data input and requirements of the Scrum process (which are reflected in the resulting output). The research identified several AI techniques that were applied within Scrum processes, as presented in Table 3. Detailed dataset properties and performance metrics are reported in the original studies but are not included here due to heterogeneity in experimental designs and evaluation protocols. The following paragraph will summarize the most common application of AI techniques, as identified by previous research.

NLP was a foundational AI technique that made it possible to automate and improve numerous Scrum processes that involve textual data. It was used in processes that generated a lot of unstructured textual data, such as creating user stories [18], [19], [20], and conducting release planning [16], [17]. These processes are mainly part of the second Scrum phase (*Plan and Estimate*). To utilize NLP, it was essential to transform unstructured natural language requirements into machine-readable formats, like detailed user stories, and to create structured developer profiles. Primary input for NLP across the processes was natural language text from requirements documents, user story descriptions, and issue titles/descriptions. Outputs included annotated text, formal specifications, classified sentences, extracted features/keywords, structured tasks, user stories, prioritized issues, etc. Overall, NLP can make AI models used in Scrum processes much more accurate, consistent, and less labor-intensive.

Classification and regression techniques (i.e., Artificial Neural Networks, Naïve Bayes, Decision Trees, and Logistic Regression) were prominently used for prioritizing product backlog items and classifying user stories. For prioritization, inputs included textual features (title, description), and historical priority and story point attributes of existing issues (e.g., from JIRA) [17]. The output was an approximate rank for newly added or modified issues. For story point estimation, classification techniques grouped issues into predefined story point categories based on their characteristics. To improve the accuracy of overall predictions, Decision Trees were used [24]. Inputs were issue descriptions, issue type, and components. The output was a predicted story point class. Overall, classification techniques can help automate the process of making decisions by cutting down the time and reducing the subjectivity that comes with manual estimation methods like Planning Poker.

Table 2: Frequency of AI technique types applied to Scrum processes in the reviewed studies

	Type of AI Technique						
Scrum Phase	Search	Topic Modeling	Regression	Classification	Clustering	NLP	NN
Form Scrum Team	1						
Create Prioritized Product Backlog				1			
Conduct Release Planning	1		1		1	2	
Create User Stories				1		3	
Approve, Estimate, and Commit User Stories		1	1				1
Estimate Tasks			3	1		1	
Create Sprint Backlog			1				

Regression models were used to predict values like the amount of work needed for a user story or task, or the number of story points. These models used historical data, such as story characteristics and developer attributes, to give more accurate estimates than traditional human judgments, which can be biased [25], [26], [27]. For effort estimation, inputs commonly included story priority, number of subtasks, story size (story points), sprint number, and programmer experience (all from historical data). The output was predicted effort in person-hours or estimated story points.

Search algorithms (e.g., GA) were used in tasks that required optimization or matching, such as team formation or release planning. They were used with both numerical and textual input data. For example, GAs helped with team formation by matching detailed developer profiles with project needs [12], and they were more accurate than human managers. Input included the technology needs of the projects and detailed profiles of the developers, containing their knowledge (from technology tags) and skills (from structured task descriptions) in the form of feature vectors. The output was the optimal global team formation for a set of target projects. In release planning, GAs were employed to achieve the highest possible business value by determining the optimal sequence for user stories, taking into account dependencies and the team's capacity [14]. User stories, their technical precedence (dependency matrix), assigned business value, estimated effort (story points), and team velocity were all used to plan releases. The output was an ordered list of user stories. This approach provides a strong and adaptable way to automate and improve decision-making in the processes that were previously performed manually.

Topic modeling, like Latent Dirichlet Allocation (LDA), was used to find latent information in user story descriptions that can help estimate the story points [22]. It transformed text descriptions into topic-based vectors, which allowed measurement between issues. This was important for subsequent clustering. The main input was the raw text of the user story, including the title and description. The text was then pre-processed and put together as "issue-context." The output was a numerical vector that shows issue reports, with issues represented by their relevance probabilities to inferred topics. Even though LDA was more accurate than random guessing and mean baselines, its performance for story point estimation was often similar to that of simpler techniques like median estimation. This means that topic modeling may not always give superior outcomes than simpler benchmarks if the quality of the data is not high.

Deep NNs have been used to learn complex semantic features from raw user story text [21]. This made the predictions much more accurate. Common inputs were word embeddings derived from stories, concatenated summary and description texts, and historical data of story points and effort. The output was predicted story points. NN models are powerful, but like other machine learning techniques, their effectiveness depends a lot on the quality and volume of training data they get.

Finally, the combination of different types of AI techniques enabled a more comprehensive and effective analysis of textual content. By combining different AI techniques, companies can deploy the strengths of each one to meet their specific process needs more efficiently. Depending on the process and the desired outcome, the input data varied from numbers to plain text. Numbers were mostly needed for ML models using regression, while the text was used by AI techniques such as NLP, NN, and Topic modeling. Interested readers are offered references for further investigation, where details on input data and the AI techniques used can be found.

Figure 3 shows how different categories of AI techniques have been applied to specific Scrum processes in the reviewed literature, highlighting dominant technique–process pairings.

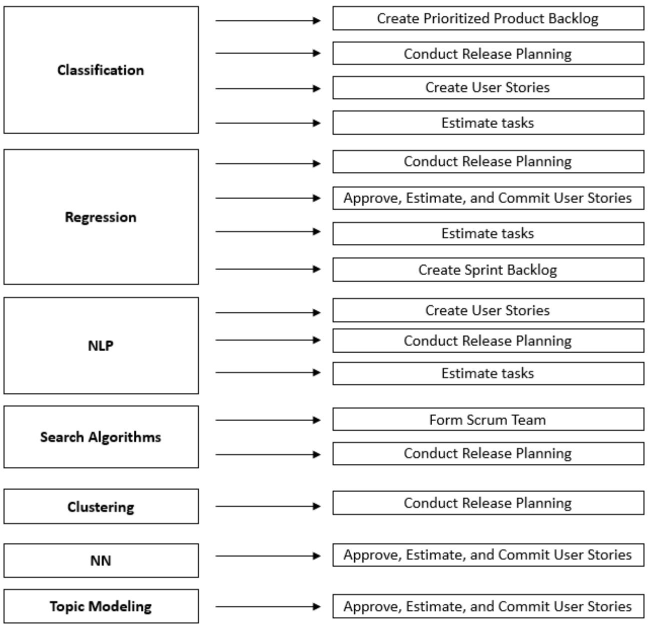

Figure 3: AI Technique types applied to Scrum processes

3.3 RQ3. How has AI improved the Scrum process in terms of automation, assistance, and enhancement?

The papers provide an overview of how AI can improve different stages of the Scrum process. The impact of AI on these processes was categorized as enhancement, assistance, or automation. A balanced distribution of results was observed, with seven papers focusing on AI assisting Scrum processes, five on AI automating them, and six on AI enhancing existing processes. Table 4 summarizes these findings.

Papers [21], [22], [23], [24], [26], [27] reported that AI techniques have enhanced the estimation process through improved data analysis, pattern recognition, and prediction models, enabling more accurate user story and task estimations. More precisely, AI gave value to the Scrum process by providing insights and increasing the accuracy of the estimation process.

Other papers [12], [13], [14], [15], [17], [28], [29] reported the use of AI as a support tool for decision-making and planning. However, AI did not act autonomously, but rather as a supplement to human decision-making in tasks that still required human judgment.

In the context of the AI-supported Scrum process, AI was applied to repetitive [16], [25], rule-based [18], [19], or data-driven jobs [20] with the goal of fully carrying out the process. Such processes can be automated by performing steps independently based on predefined criteria and inputs. AI techniques, especially machine learning (ML) and deep learning (DL), have been widely applied to automate the process of story points estimation. The following is a summary of the trends:

Enhanced estimation: AI techniques significantly reduce the amount of time and labor needed to perform the estimation process, particularly in later project phases when more data is available for training [24]. In addition, AI models produce more consistent and trustworthy estimates because they are less prone to biases and subjective factors [22].

Support for team and resource management: AI addresses multiple team formation issue in Scrum projects [12]. By taking into account specific competencies, such as technical skills deduced from previous project tasks, AI techniques can optimize the distribution of available developers among several software projects. This helps create cross-functional teams without redundant skills. Furthermore, to maximize the number of story points created in a sprint and to distribute available effort among the team, ensemble-based methods and predictive models can be employed [28]. This facilitates the best possible scheduling of developers and stories. NLP-based machine learning techniques can be utilized to dynamically predict the rank of recently added or modified issues [17]. With an estimated rank for incoming issues, project managers can focus on sprint planning and make timely decisions.

Automation of repetitive jobs: LLMs can be used to generate user stories from high-level requirements documents automatically [19]. Senior team members' manual labor can be significantly decreased by this procedure. As for the software specifications, they can be automatically derived from requirements using AI techniques such as NLP and ML [18]. This facilitates the digitization of requirements knowledge, the creation of context-sensitive user stories, and the enhancement of traceability and consistency. Finally, to make release planning for large and complex software projects easier, NLP algorithms can be used to find similar user stories and group them into different releases [16].

Overall, there is a noticeable trend toward automating Scrum's highly manual, time-consuming, and data-rich processes.

4 Discussion

This paper identifies Scrum processes that have been supported by AI. Since the use of AI depends greatly on the type of data available, the required input data for each specific AI technique is listed. Similarly, while the output of a process is strongly related to the problem being addressed, it also influences the choice of AI technique. Therefore, both the input and output data used in previous research were identified.

From a theoretical perspective, the findings of this research align with the Task-Technology Fit (TTF) theory, which argues that technology contributes value only when its capabilities fit the tasks and context in which it is applied [30]. The proposed mapping of AI techniques to Scrum processes, input/output data, and contextual factors operationalizes this perspective by identifying where specific AI capabilities are suitable for particular Scrum tasks.

The motivation for this study partly stems from gaps identified in the author's previous research [3], which revealed that many studies do not differentiate between agile software methodologies. As a result, it is often difficult to extract information about which parts of a process belong to specific agile development methodologies. This lack of clarity makes it challenging to determine how particular AI techniques can be applied to improve Scrum, XP, or Kanban. Conversely, studies that focus specifically on Scrum tend to cover only narrow aspects of the development process (e.g., task allocation, effort estimation). Therefore, the impact of AI on Scrum as a whole remains unknown. Our research tries to cover the aforementioned gaps by providing a clear overview of the phases of Scrum development. It identifies the processes that are part of the Scrum framework and proposes ones that AI can improve. It also identifies the problems that can be solved within each phase of the process (listed in the form of output) and the types of AI techniques that can be used to solve these problems. Furthermore, it gives an overview of the processes for which research related to the application of AI has not yet been done. These results are valuable because they provide state-of-the-art efforts made in the field of AI-supported Scrum, but they also enable us to propose some promising areas for future research.

Table 3: AI Techniques and their input and output data listed by Scrum processes

Process	Input	Output	Specific AI Technique	Type of AI Technique	Ref.
Form Scrum Team	Structured task containing task description and technology tags, and a set of target projects' technology requirements	Formed teams for a set of target projects	Genetic algorithm (GA)	Search algorithm	[12]
Create Prioritized Product Backlog	Product backlog items in the form of text	Classification models	Artificial Neural Network (ANN), Naive Bayes (NB), Support Vector Machine (SVM), Logistic Regression (LR), and Decision Tree (DT)	Classification	[13]
Conduct Release Planning	Quantitative data from the backlog, including effort, business value, dependency matrix, and team velocity	Release planning schedule in the form of a vector	GA	Search algorithm	[14]
	User stories in the form of text	Clustered user stories for every release	NLP	NLP/ Clustering	[16]
	Issues and assignees in the form of text	Time estimates and assigned resources in the form of vectors	Extreme Gradient Boosting (EGB)	Regression	[15]
	Issues (description (text), priority, and story points attributes (number))	Final predicted rank (issues prioritization)	NLP	NLP/Classification	[17]
Create User Stories	Requirement statements and epics in the form of text	Expected functionalities expressed as user stories	NLP/Decision Tree Classification	NLP/ Classification	[18]
	Requirements' specifications in the form of text	User stories in text format	NLP	NLP	[19]
	Requirements' specifications	User stories in the JSON format	GPT 4.0	NLP	[20]
Approve, Estimate, and Commit User Stories	User stories in text format	Story point predictions (number)	Deep NN	NN	[21]
	Issues in the form of text	List of topics	Latent Dirichlet Allocation (LDA)	Topic modeling	[22]
	User stories in text format	Estimated story point (number)	Recurrent Neural Network (RNN) and Convolutional Neural Network (CNN)	Regression	[23]
Estimate Tasks	User stories in text format	Hours or days (number)	NLP + NB Classification	NLP/ Classification	[24]
	Story cards (text) and planned effort estimates (number)	Actual effort (number)	NB, J48 DT, Random Forest (RF), Ada Boost (AB)	Regression	[25], [26]
	Total story points required, project velocity, and actual effort (numbers)	Software development effort (number)	DT, Stochastic Gradient Boosting (SGB), RF	Regression	[27]
	Priority of the story (number), number of subtasks for the story, size of the story (story point), sprint of the story (integer), and programmer experience (years)	Total development effort (in person/hours)	Ensemble algorithm (Bayesian Network (BN), Ridge Regression (RR), ANN, SVM, DT, K-Nearest Neighbors (KNN), and Ordinary Least Squares Regression (OLSR)	Regression	[28]
Create Sprint Backlog	Employee data set, effort estimated task list, length of Sprint, Previous Sprint Velocity, Dependencies, Team Calendar (numbers and text)	Job assignment to the best match	SVM, LR, NSL	Regression	[29]

Previous research related to AI in agile development has primarily tackled the application of individual AI techniques to specific tasks, such as effort estimation, task allocation, or backlog prioritization. While these studies demonstrate the technical feasibility of AI in specific contexts, they typically focus on isolated Scrum activities and do not provide a holistic view of the Scrum as a process-oriented system. This study addresses these limitations by mapping specific Scrum processes and phases, and explicitly identifying their required input data, produced outputs, and level of AI involvement. This allows cross-study comparison at the process level and helps to detect underexplored phases. By synthesizing findings and structuring them according to the established framework, this study provides a more comprehensive perspective than previous research. Besides, it highlights methodological gaps and practical challenges that should be addressed in future research.

4.1 RQ1: What is the current status of advancements in AI-supported Scrum processes?

As discussed earlier, our research identified that AI is already being used in 7 out of 19 Scrum processes. Bearing in mind how new this area is, this is still a respectable number, especially since these processes are not marginal elements but core activities. Researchers have mostly studied processes that involve decision-making and contain a lot of historical data.

Table 4: Level of AI involvement in the Scrum process

Process	AI Support	Ref.
Form Scrum Team	Assist in the process of team selection	[12]
Create Prioritized Product Backlog	Assist in classifying product backlog items	[13]
Conduct Release Planning	Assist in release planning	[14], [15]
	Automate release planning	[16]
	Assist in decision-making	[17]
Create User Stories	Build models (automation)	[18]
	Automatic generation of user stories	[19], [20]
Approve, Estimate, and Commit User Stories	Enhance estimation	[21], [22], [23], [26]
Estimate Tasks	Enhance estimation	[24], [27]
	Automate effort estimation	[25]
	Assist in sprint planning	[28]
Create Sprint Backlog	Assist in decision-making	[29]

However, most studies provided a shallow analysis of AI applications, reporting which techniques were used in which processes, without delving into important decision-making aspects such as the factors affecting the use of AI or the integration with existing Scrum practices.

This limits the understanding of where, how, and under what conditions AI can be applied within Scrum. On the other hand, the observed use of AI techniques in early Scrum phases may indicate a practical alignment: these phases are structured, data-rich, and therefore suitable for predictive modeling and classification tasks, while later phases are less structured and require more effort to collect and clean the data. As a result, a large part of Scrum still lacks proper research into AI integration possibilities.

4.2 RQ2: What AI techniques have been used to optimize the Scrum process based on the available input and output data?

The choice of AI techniques seems to depend heavily on the availability of data and its format. According to the results of the literature review, key AI applications within Scrum include NLP, ML, and Search algorithm. Although current research offers concrete AI techniques to solve the specific Scrum problems, it lacks systematic justification for technique selection. Few studies explicitly discussed why a given technique was suitable for the specific Scrum process, and how the input/output data characteristics influenced the choice of AI techniques. This underscores a methodological gap that future work should address by, for instance, developing AI selection frameworks based on context-related factors.

4.3 RQ3: How has AI improved the Scrum process in terms of automation, assistance, and enhancement?

The relatively balanced distribution between enhancement, assistance, and automation suggests that current AI applications in Scrum tend to complement, rather than replace, human activities. This may be caused by the socio-technical nature of Scrum, where many processes rely on collaboration, negotiation, and shared vision, limiting the suitability of full automation. Although several studies demonstrate the technical feasibility of automating specific Scrum processes, these contributions largely represent experimental research rather than evidence of widespread industrial adoption.

By looking into how AI has improved the Scrum process, our research indirectly addressed the question of replacing humans with AI. For instance, the use of AI significantly reduces the amount of effort required for analytical and data-driven tasks. On the other hand, humans are still superior at tasks that need creativity, critical thinking, and collaboration [31]. With current concerns about AI taking over human roles, our study provides valuable insights about how AI affects Scrum and highlights the areas where human skills remain essential.

4.4 Practical implementation challenges

Although the reviewed studies demonstrate promising applications of AI within Scrum, they also highlight several practical challenges that could impact real-world adoption. These challenges are mostly related to data availability and quality, deployment constraints, integration with Scrum practices, and technical limitations of AI models.

Several authors emphasize that AI techniques depend on large and reliable datasets, which are often not available in practice. For instance, insufficient historical data can limit model training for story point estimation or team formation [12], [21]. In addition, user stories frequently contain poor or ambiguous descriptions, which hinders the effectiveness of AI techniques. Historical estimates may also be inaccurate or biased, which reduces the reliability of training data [21]. Since models trained on archival data inherit human errors and biases, AI techniques may provide inaccurate estimations [28]. Finally, domain-specific vocabularies often need to be manually created by experts, which limits the scalability and portability of AI solutions [18], [19].

Authors also reported difficulties when AI models are applied to real project environments. Inaccuracies may occur when data models are retroactively applied to older projects, as teams cannot always reconstruct past technical details [12]. AI techniques can be prone to errors when projects involve new emerging technologies that were not represented in historical datasets used for model calibration [28]. Moreover, sustaining model quality requires continuous data maintenance and standardization, which may conflict with agile principles [21]. Small or domain-specific datasets further limit the generalizability of AI solutions across different types of projects and domains [12], [21], [28]. Some models rely on simplifying assumptions due to missing data, such as assuming all tasks with the same standard description and tags require similar effort, which does not always reflect actual practice [12].

AI adoption often requires changes in established team behaviors and processes, which can create friction and resistance. For example, data labeling introduces additional effort for developers. The effectiveness of some AI techniques (e.g., Genetic Algorithms for team formation) depends on the availability of multiple similar past projects, which may not always be the case [12].

Overall, these challenges demonstrate that the practical implementation of AI in Scrum is highly dependent on disciplined data practices and sustained maintenance efforts. Although the individual limitations were reported by different authors, they converge into a recurring set of challenges that may affect the feasibility of AI integration in Scrum.

4.5 Limitations and risks of AI in scrum

The analyzed studies highlight several conceptual and ethical risks that may affect the use of AI techniques.

Risks arise from the mismatch between the flexibility of Scrum practices and the rigidity of AI models. For example, AI models often depend on stable, comparable datasets, which may conflict with the variability of Scrum models [21].

AI models often rely on existing human estimates, which may not consistently reflect actual effort and are prone to human biases [21]. This raises concerns about fairness, transparency, and accountability, particularly if model outputs begin to influence decision-making without critical human oversight.

5 Research implications and future work

The results of this paper will be particularly valuable for researchers as a starting point to further investigate AI techniques used in Scrum, as well as for identifying gaps and promising areas for future research. While previous studies have mostly focused on individual agile processes and their improvement through AI, this research takes a holistic approach to Scrum as a whole. This makes it easier for practitioners to identify the specific processes that interest them, recognize problems similar to their own, and discover the types of AI that could be applied to address these challenges.

We argue that the majority of processes that remain intact by AI could greatly benefit from the adoption of AI techniques. In our opinion, processes that have the biggest potential to be improved by AI are the ones from the *Implement* and *Review and Retrospect phases*. For instance, during the Daily Standup, AI can analyze the team's communication patterns and look for signals that might indicate a problem or a potential risk that needs to be addressed [32]. Sentiment analysis can identify early signs of dissatisfaction, burnout, or conflict in team communications. LLMs may also identify recurring problems or dependencies, which could help detect risks. AI could analyze the team's conversations during the *Retrospect Sprint*, identify recurring patterns or themes, and suggest approaches to address the problems [33]. It can compare the teams' performance against the industry benchmark and provide valuable insights that may not be apparent to the team members themselves [34]. Furthermore, AI can help groom a prioritized product backlog by adapting to changes and assuring that the latest priorities and insights are being reflected [35], [36]. Finally, it can analyze teams' progress, track changes, incorporate customers' feedback, and suggest adjustments to the product backlog. In this way, the process would become much more proactive and strategic.

The results indicate that AI can boost Scrum processes by enhancing, assisting, or automating them. A possibility for broader automation exists. For example, the extraction of requirements from various sources could be automated [37], as well as the translation of user stories into technical specifications [34].

During the literature review, a lack of a strategic approach to the implementation of AI within Scrum was noted. Although the papers included in the analysis did list the type and form of data needed to utilize some AI techniques, these are not enough to enable the successful replication of the AI implementation strategy. Therefore, we argue for the need for an AI-supported Scrum framework, whose task would be to guide the Scrum team through the implementation of AI into the Scrum framework. The proposed framework is not intended as a prescriptive, step-by-step procedure directly used by Scrum teams, but rather as a guidance. Key parts of the suggested AI-supported Scrum framework would be:

1) Identifying processes and analyzing the feasibility of AI solutions (made in this paper): Each Scrum process is analyzed to determine its potential for AI integration. The processes are connected with the right AI technique (e.g., using regression algorithms to estimate effort or clustering or classification methods to prioritize a backlog).

2) Creating the alignment matrix (made in this paper): It shows the inputs, outputs, and AI techniques that are required or expected for each targeted process. This alignment could assist practitioners in choosing the right techniques and preparing the required data.

3) Assessing the contextual factors (in progress): Before integration, factors such as the availability of data, the type of data, the technical infrastructure, and the available resources will be examined. This assessment will help guide recommendations and prevent inadequate solutions.

4) Recommending AI techniques (TODO): AI techniques will be recommended based on both objective (e.g., classification, prediction, and information extraction) and contextual factors.

5) Deciding on feasibility (TODO): Before integrating AI into Scrum, it's important to determine if any changes should be made regarding the roles and artifacts, if there are any plugins available for AI tools, what are the criteria for measuring the success of the integration, etc.

Figure 4 illustrates the conceptual framework that synthesizes evidence from the literature into decision support for selecting AI techniques in Scrum. A simulated walkthrough of the framework is presented in Appendix A.

Based on the findings of this research and the identified research gap, several potential directions for future work are suggested.

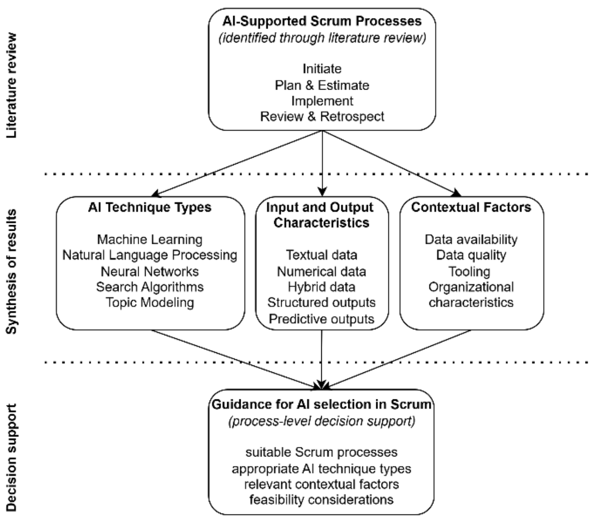

Figure 4: Conceptual framework for AI-supported Scrum

First, a lack of empirical validation of AI techniques in real-world Scrum environments was detected. To evaluate how useful AI-supported solutions are, future research should focus on performing case studies, experiments, and other types of research with Scrum teams. Second, as our research shows, phases such as *Implement* and *Review and Retrospect* remain significantly underexplored. Future work should focus on getting insights from these phases. Third, while this paper proposes key components of a future AI-supported Scrum framework, its complete operationalization remains to be done. Future work should focus on the development of this framework. Fourth, while this paper provides a structured mapping between AI techniques and Scrum processes, it does not aim to compare or benchmark model performance across studies. Due to substantial heterogeneity in datasets, domains, and evaluation protocols, a systematic performance benchmarking or meta-analytic synthesis was considered outside the scope of this review and is identified as an important direction for future empirical research.

Finally, our paper is not without limitations, which we will briefly address. One of the main objectives of this study was to illustrate how AI is transforming Scrum teams. Out of 305 screened papers, only 18 were included in the research. The most common reason for paper exclusion was a lack of case studies, experiments, or industry validations of AI techniques, resulting in hypothetical proposals that were not empirically tested. The second common reason was generic agile processes that couldn't be mapped to Scrum. Other reasons include insufficient detail on data and integration context (e.g., the type of data being used, data source, etc.) and scope mismatch. Only papers that described the concrete improvement of Scrum processes through AI were included in our research. Therefore, there is a possibility that some relevant papers were unintentionally omitted. To minimize this risk, a comprehensive set of carefully chosen keywords was used. Although the final set of papers addressing the area of interest is relatively small (as the field of AI-supported Scrum is still new and specialized), we believe that the widespread use of Scrum in practice justifies our research. Furthermore, we argue that distinguishing Scrum from other agile methodologies was essential for effectively investigating the impact of AI on Scrum processes.

5 Conclusion

The study shows how AI is currently being used in Scrum and the level of AI's involvement. It specifies Scrum processes and groups them into phases. It then analyzes the inputs, outputs, and AI techniques (and their types) utilized to improve these processes.

The results indicate that AI has been used the most within the *Plan and Estimate* phase, where it has been utilized to improve effort estimates, assign tasks, and plan releases. ML was the most popular AI method, followed by NLP and other AI methods such as *Search algorithms*. The study shows how these techniques may improve Scrum processes by solving common problems like estimating the right amount of effort or deciding which tasks to do first. It also provides concrete AI-based solutions and identifies processes that AI can improve.

The less-explored phases (*Implement* and *Review and Retrospect*) were identified as promising areas for future research. Finally, the study shows how important it is to have an organized approach for integrating AI into the workplace. Future research should focus on creating a strategic framework for using AI in Scrum so that its full potential can be realized. By doing so, AI could go from being a supplement to an essential component of the Scrum process.

This research is valuable for several reasons. First, to the best of our knowledge, this is the first paper to specifically investigate the use of AI in Scrum. Although several studies have explored the impact of AI on agile software development, this research uniquely isolates and examines only the processes that are part of Scrum. Second, the impact of AI was investigated at the level of the processes. By doing so, we were able to identify the number of processes and the extent to which they were already supported by AI. Third, input and output were identified for each AI technique, enabling practitioners to easily select the potentially suitable AI technique and data sources to achieve the desired outcome. Fourth, specific AI techniques were grouped into broader AI types, focusing on the main problem rather than specific implementations and giving users more options when choosing AI. Fifth, examples of AI use cases in related fields were given to show how they could be used in Scrum and to encourage more research into this area. Finally, this paper provides a strong basis for further research in this promising scientific field.

Appendix A: Simulated Walkthrough of the AI-supported Scrum framework

This appendix presents a simulated walkthrough illustrating how Scrum teams can use it in selecting appropriate AI techniques for their processes. The walkthrough demonstrates how the framework outcomes serve as a decision-support. The scenario is illustrative and aims to clarify the usability of the framework.

A.1 Scenario description

A software development organization applies Scrum and seeks to improve planning accuracy and backlog refinement. The organization has access to user stories, effort estimates, and issue tracking data, but has limited experience with AI.

Step 1 Identifying processes

Based on the results presented in Section 3, the organization identifies Scrum processes that have been empirically supported by AI in prior research. Processes such as *Create User Stories*, *Estimate Tasks*, and *Conduct Release Planning* are recognized as suitable candidates for AI support. Using this information, the team defines its improvement goals (e.g., increasing estimation accuracy and reducing backlog refinement effort) and selects *Estimate Tasks* and *Create User Stories* as target processes.

Step 2 Defining inputs, outputs, and applicable AI techniques

After selecting the target processes, the team examines the required input and expected output data using the alignment matrix. For *Estimate Tasks*, the framework indicates numerical historical data as input and effort values as output, with regression and ensemble-based techniques commonly applied. For *Create User Stories*, textual requirement descriptions are identified as input, with structured user stories as output, supported primarily by NLP techniques. Using this information, the team narrows the set of applicable AI technique types.

Step 3 Assessing the contextual factors

The team evaluates contextual factors identified within the framework, including data availability, data quality, constraints, and organizational readiness. For example, while historical effort data is available and sufficiently structured, user stories vary in quality and level of detail.

Step 4 Selecting AI technique types

Based on selected processes, improvement goals, and contextual factors, the framework guides the team toward appropriate AI technique types (e.g., regression models for *Estimate Tasks*, and NLP for *Create User Stories*). The framework supports informed selection while leaving flexibility in the choice of specific implementations.

Step 5 Deciding on feasibility

For *Estimate Tasks*, the assessment indicates that existing historical effort data and estimation practices are sufficient to support regression-based AI technique types without requiring changes to Scrum roles or artifacts. The team decides to introduce AI as an assistive mechanism for estimation. For *Create User Stories*, the assessment reveals that although textual requirement data is available, variability in story structure may limit the effectiveness of NLP techniques. To address this, the team decides to introduce user story templates and clearer acceptance criteria, while preserving Scrum roles and workflows.

6 Ethics and data availability statement

This study is based exclusively on a review and synthesis of previously published papers. No primary data involving human participants, organizations, or proprietary datasets were collected or analyzed. Therefore, ethical approval was not required. All data used in this study are derived from publicly available records cited in the references.

Acknowledgement

The research was partially supported by the Croatian Science Foundation under Project *SeEduOpsAI: Open*

Platform for AI Enhanced Software Engineering Education (IP-2025-02-4117).

References

[1] "Gartner Forecasts Worldwide Public Cloud Revenue to Grow 17% in 2020," Gartner. Accessed: Nov. 11, 2020. [Online]. Available: https://www.gartner.com/en/newsroom/press-releases/2019-11-13-gartner-forecasts-worldwide-public-cloud-revenue-to-grow-17-percent-in-2020

[2] N. Haefner, V. Parida, O. Gassmann, and J. Wincent, "Implementing and scaling artificial intelligence: A review, framework, and research agenda," *Technol. Forecast. Soc. Change*, vol. 197, p. 122878, Dec. 2023, doi: 10.1016/j.techfore.2023.122878.

[3] D. Peras, Z. Stapić, and M. Matijević, "AI techniques and tools in Agile Software Development: Preliminary research," in *34th Central European Conference on Information and Intelligent Systems: Proceedings*, Zagreb: Sveučilište u Zagrebu, Fakultet organizacije i informatike, 2023, pp. 501–508.

[4] "Scrum Essentials," Gartner. Accessed: Dec. 11, 2024. [Online]. Available: https://www.gartner.com/en/documents/4444699

[5] I. I. Anghel, R. S. Calin, M. L. Nedelea, I. C. Stanica, C. Tudose, and C. A. Boiangiu, "Software development methodologies: A comparative analysis," *UPB Sci Bull*, vol. 83, no. 3, pp. 45–58, 2022.

[6] A. Ibrahim, "AI-Driven Innovation in Software Development: Case Studies and Future Trends," 2024.

[7] S. Arias, A. L. Yacelga, N. B. Arévalo, F. Alvarez, and J. N. Guarderas, "SCRUM Data Analysis and AI Integration for Project Management," Dec. 14, 2024, *Preprints*. doi: 10.22541/au.173415608.84573329/v1.

[8] H. K. Dam, T. Tran, J. Grundy, A. Ghose, and Y. Kamei, "Towards effective AI-powered agile project management," in *2019 IEEE/ACM 41st international conference on software engineering: new ideas and emerging results (ICSE-NIER)*, IEEE, 2019, pp. 41–44.

[9] M. Perkusich *et al.*, "Intelligent software engineering in the context of agile software development: A systematic literature review," *Inf. Softw. Technol.*, vol. 119, p. 106241, Mar. 2020, doi: 10.1016/j.infsof.2019.106241.

[10] J. Haase, P. B. Walker, O. Berardi, and W. Karwowski, "Get Real Get Better: A Framework for Developing Agile Program Management in the U.S. Navy Supported by the Application of Advanced Data Analytics and AI," *Technologies*, vol. 11, no. 6, p. 165, Nov. 2023, doi: 10.3390/technologies11060165.

[11] D. Sunner, "Agile: Adapting to need of the hour: Understanding Agile methodology and Agile techniques," in *2016 2nd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT)*, July 2016, pp. 130–135. doi: 10.1109/ICATCCT.2016.7911978.

[12] A. Costa *et al.*, "A Genetic Algorithm-Based Approach to Support Forming Multiple Scrum Project Teams," *IEEE Access*, vol. 10, pp. 68981–68994, 2022, doi: 10.1109/ACCESS.2022.3186347.

[13] N. Ravikumar, B. Kuhaneswaran, A. Saleem, A. K. Wijeratne, B. Kumara, and G. Herath, "Classification of Product Backlog Items in Agile Software Development Using Machine Learning," in *Handbook of Research on Technological Advances of Library and Information Science in Industry 5.0*, IGI Global, 2023, pp. 306–329.

[14] A. Kumar, R. Nagar, and A. S. Baghel, "A genetic algorithm approach to release planning in agile environment," in *2014 International Conference on Information Systems and Computer Networks (ISCON)*, Mathura, India: IEEE, Mar. 2014, pp. 118–122. doi: 10.1109/ICISCON.2014.6965230.

[15] E. Ebrahim *et al.*, "AI Decision Assistant ChatBot for Software Release Planning and Optimized Resource Allocation," presented at the 2023 IEEE International Conference On Artificial Intelligence Testing (AITest), IEEE, 2023, pp. 55–60.

[16] S. Sharma and D. Kumar, "Agile Release Planning Using Natural Language Processing Algorithm," in *2019 Amity International Conference on Artificial Intelligence (AICAI)*, Dubai, United Arab Emirates: IEEE, Feb. 2019, pp. 934–938. doi: 10.1109/AICAI.2019.8701252.

[17] S. Shafiq, A. Mashkoor, C. Mayr-Dorn, and A. Egyed, "NLP4IP: Natural Language Processing-based Recommendation Approach for Issues Prioritization," in *2021 47th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, Palermo, Italy: IEEE, Sept. 2021, pp. 99–108. doi: 10.1109/SEAA53835.2021.00022.

[18] K. Lano, S. Yassipour-Tehrani, and M. A. Umar, "Automated Requirements Formalisation for Agile MDE," in *2021 ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C)*, Fukuoka, Japan: IEEE, Oct. 2021, pp. 173–180. doi: 10.1109/MODELS-C53483.2021.00030.

[19] P. V. Nistala, A. Rajbhoj, V. Kulkarni, S. Soni, K. V. Nori, and R. Reddy, "Towards digitalization of requirements: generating context-sensitive user stories from diverse specifications," *Autom. Softw. Eng.*, vol. 29, no. 1, p. 26, May 2022, doi: 10.1007/s10515-022-00324-2.

[20] T. Rahman and Y. Zhu, "Automated User Story Generation with Test Case Specification Using Large Language Model," Apr. 01, 2024, *arXiv*: arXiv:2404.01558. Accessed: Aug. 27, 2024. [Online]. Available: http://arxiv.org/abs/2404.01558

[21] M. Abadeer and M. Sabetzadeh, "Machine Learning-based Estimation of Story Points in Agile Development: Industrial Experience and Lessons Learned," in *2021 IEEE 29th International Requirements Engineering Conference Workshops*

(REW), Notre Dame, IN, USA: IEEE, Sept. 2021, pp. 106–115. doi: 10.1109/REW53955.2021.00022.

[22] V. Tawosi, A. Al-Subaihin, and F. Sarro, "Investigating the Effectiveness of Clustering for Story Point Estimation," in *2022 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER)*, Honolulu, HI, USA: IEEE, Mar. 2022, pp. 827–838. doi: 10.1109/SANER53432.2022.00101.

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

[24] C. Ratke, H. H. Hoffmann, T. Gaspar, and P. E. Floriani, "Effort Estimation using Bayesian Networks for Agile Development," in *2019 2nd International Conference on Computer Applications & Information Security (ICCAIS)*, Riyadh, Saudi Arabia: IEEE, May 2019, pp. 1–4. doi: 10.1109/CAIS.2019.8769455.

[25] K. Moharreri, A. V. Sapre, J. Ramanathan, and R. Ramnath, "Cost-Effective Supervised Learning Models for Software Effort Estimation in Agile Environments," in *2016 IEEE 40th Annual Computer Software and Applications Conference (COMPSAC)*, Atlanta, GA, USA: IEEE, June 2016, pp. 135–140. doi: 10.1109/COMPSAC.2016.85.

[26] E. Rodríguez Sánchez, E. F. Vázquez Santacruz, and H. Cervantes Maceda, "Effort and Cost Estimation Using Decision Tree Techniques and Story Points in Agile Software Development," *Mathematics*, vol. 11, no. 6, p. 1477, Mar. 2023.

[27] S. M. Satapathy and S. K. Rath, "Empirical assessment of machine learning models for agile software development effort estimation using story points," *Innov. Syst. Softw. Eng.*, vol. 13, no. 2–3, pp. 191–200, Sept. 2017, doi: 10.1007/s11334-017-0288-z.

[28] O. Malgonde and K. Chari, "An ensemble-based model for predicting agile software development effort," *Empir. Softw. Eng.*, vol. 24, no. 2, pp. 1017–1055, Apr. 2019, doi: 10.1007/s10664-018-9647-0.

[29] P. William, P. Kumar, G. S. Chhabra, and K. Vengatesan, "Task Allocation in Distributed Agile Software Development using Machine Learning Approach," in *2021 International Conference on Disruptive Technologies for Multi-Disciplinary Research and Applications (CENTCON)*, Bengaluru, India: IEEE, Nov. 2021, pp. 168–172. doi: 10.1109/CENTCON52345.2021.9688114.

[30] D. L. Goodhue and R. L. Thompson, "Task-technology fit and individual performance," *MIS Q.*, pp. 213–236, 1995.

[31] M. H. Jarrahi, D. Askay, A. Eshraghi, and P. Smith, "Artificial intelligence and knowledge management: A partnership between human and AI," *Bus. Horiz.*, vol. 66, no. 1, pp. 87–99, Jan. 2023, doi: 10.1016/j.bushor.2022.03.002.

[32] M. D. C. Kop, "Current State of Art of Natural Language Processing Research on User Feedback," Alpen-Adria-Universität Klagenfurt, 2022.

[33] M. Afzaal *et al.*, "Explainable AI for Data-Driven Feedback and Intelligent Action Recommendations to Support Students Self-Regulation," *Front. Artif. Intell.*, vol. 4, p. 723447, Nov. 2021, doi: 10.3389/frai.2021.723447.

[34] K. Sainio and P. Abrahamsson, "Generative Artificial Intelligence Assisting In Agile Project Pain Points Empirical study using ChatGPT," Master's Thesis, Faculty of Management and Business, Tampere University, 2023.

[35] P. Tonella, A. Susi, and F. Palma, "Interactive requirements prioritization using a genetic algorithm," *Inf. Softw. Technol.*, vol. 55, no. 1, pp. 173–187, Jan. 2013, doi: 10.1016/j.infsof.2012.07.003.

[36] C. Zhang, J. Kiseleva, S. K. Jauhar, and R. W. White, "Grounded Task Prioritization with Context-Aware Sequential Ranking," *ACM Trans. Inf. Syst.*, vol. 40, no. 4, pp. 1–28, Oct. 2022, doi: 10.1145/3486861.

[37] A. Henriksson and J. Zdravkovic, "A Data-Driven Framework for Automated Requirements Elicitation from Heterogeneous Digital Sources," in *The Practice of Enterprise Modeling*, vol. 400, J. Grabis and D. Bork, Eds., in Lecture Notes in Business Information Processing, vol. 400. , Cham: Springer International Publishing, 2020, pp. 351–365. doi: 10.1007/978-3-030-63479-7_24.