International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

AI-Based Prediction of Scope Creep in Agile Projects

Alex Thomas Thomas[*1]

Submitted: 14/01/2025 **Revised:** 22/02/2025 **Accepted:** 10/03/2025

Abstract: Scope creep is continuing to be one of the main challenges for Agile software development, usually preventing the projects from being timely completed, as well as causing them to go over budget and reduce product quality. New advances in artificial intelligence (AI) and machine learning have shown promise in forecasting and managing scope creep by monitoring project data, team actions, and risk factors. This document presents a thorough summary of AI usage to scope creep prediction in Agile projects. We comprehensively examine existing machine learning models for effort estimation, risk management, and scope management and uncover methods such as neural networks, deep learning, and ensemble learning. The examination consolidates findings of prominent studies in Agile project risk prediction, automated scope creep detection, and AI-enhanced scheduling and presents their efficacy and limitations. Besides, we identify challenges in the implementation of AI in Agile methods and propose future research areas to increase prediction accuracy and deployment in practice. This survey aims to provide researchers and practitioners with a shared understanding of AI usage in preventing scope creep, thus enhancing Agile project success.

Keywords: *Scope creep, Agile Software Development, Artificial Intelligence, Machine Learning, Project Risk Management, Predictive Analytics*.

1. Introduction

Agile methods have revolutionized software development with emphasis on flexibility, customer involvement, and iterative delivery, making them highly suitable for volatile and complex projects [5], [6]. Contrary to traditional waterfall [3] approaches, Agile methodologies such as Scrum and Kanban encourage continuous requirement definition and change in direction during the project lifecycle. While this inherent flexibility is beneficial, it can also contribute to a common and challenging issue known as scope creep the unplanned broadening or alteration of a project's scope beyond its original goals. This often leads to schedule delays, increased costs, and a decline in software quality [5], [7]. Scope creep in Agile projects is particularly difficult to manage because the approach supports changing specifications and rapid change [4]. Although Agile has change management processes such as backlog grooming and sprint planning, poorly controlled or unexpected scope changes can bury teams and destroy project velocity [5]. Legacy risk management and manual detection techniques often are not able to detect scope creep early enough and with sufficient accuracy in the sense of both timing and severity due to the high volume and complexity of project data [10], [12]. New advancements in artificial intelligence (AI) and machine learning (ML) have opened new possibilities for project management practice augmentation through predictive analytics and automation support in decision-making [1], [2],[8]. AI can analyze heterogeneous sets of data—from historical project performance metrics, issue tracking databases, to patterns in team collaboration—to recognize warning signs for scope creep early on and approximate its likely impact on project delivery [4], [9], [14], [17]. For instance, neural networks, deep learning, and ensemble methods have been employed to provide better efforts need prediction, delay prediction, and risk factor estimation than traditional statistical techniques [1], [10], [14]. Although there is growing interest and promising results, the incorporation of AI forecasting models into Agile project management remains a developing field with several challenges. They include data quality and availability issues, explainability of the models, aptness for use in real-time, and alignment with Agile processes and principles [2], [13], [20]. Also, the effectiveness of AI in managing scope creep is extremely context-specific, e.g., project size, team member experience, and the degree of maturity of Agile practices being employed [16], [16]. This is a systematic review of recent literature on applying AI techniques in forecasting scope creep in Agile projects. It synthesizes literature on machine learning models to predict risk, automation-based detection of scope change, and prediction of project overrun, consolidating methodology patterns, datasets utilized, and assessment metrics [8], [15], [17]. The review also discusses practical implications for applying AI solutions in Agile and highlights areas for future research to enhance predictive power and adoption. By presenting an integrated picture of how AI can be used to predict and control scope creep, the present work is intended to assist researchers as well as practitioners in creating robust, data-driven plans that can counter the negative impacts of scope changes and enhance overall success in Agile projects.

2. Problem Statement

Agile software development has become a common methodology for managing complex and volatile project requirements since it is iterative and prioritizes flexibility and customer cooperation [5], [6]. However, one of the

[1]*QE Lead, Saransh Inc, 5 Independence Way, Princeton, NJ – 08540, USA*
ORCID : https://orcid.org/0009-0007-7575-4367
Email: alexthomaslive@gmail.com

longest lasting and most insidious issues in Agile settings is scope creep—the uninhibited expansion or alteration of project scope beyond the original objectives [5], [7]. Scope creep may lead to over-spend, delay, increased workload, and reduced software quality, eventually undermining the success of the project [4], [7]. While Agile practices have facilities such as backlog refinement, sprint review, and regular stakeholder input to manage change, these practices are more reactionary in nature than prescriptive. Traditional risk management and project control practices lack the analytics competency to spot very early warning signs of scope creep, especially in big-picture or high-tempo Agile settings [6], [12]. Therefore, most project managers do not successfully manage changes in scope and their consequences. Recent advancements in Artificial Intelligence (AI) and Machine Learning (ML) provide encouraging potential to close this gap. AI-based methods such as neural networks, deep learning, and ensemble learning have shown significant potential in predicting software development risk, effort estimation, and project delay prediction [1], [10], [14]. Several publications have demonstrated the way such technologies can be applied to analyze historical project data, identify risk patterns, and flag anomalies that may lead to scope creep [2], [8], [13], [17]. However, while progress has been made, the literature is fragmented, and there is no overarching synthesis of existing AI methods geared specifically to forecasting scope creep in Agile projects. Besides, the integration of AI solutions into Agile streams also poses several challenges, including the need for quality datasets, interpretability of models, alignment with Agile values, and ease of use in real-time environments [2], [13], [15], [20]. These render the efficient utilization of AI-based tools for scope management unfeasible and call for a deeper analysis of prevailing approaches, their strengths, limitations, and potential future direction. Therefore, there is a critical necessity to examine and evaluate the present body of knowledge regarding AI-inclined approaches to scope creep prediction in Agile software development. Systematically synthesizing will facilitate the determination of successful models, unveil areas of research loopholes, and guide ensuing innovation towards enhancing proactive scope control in Agile project management.

3. Agile Project Management Overview

3.1 Basic concepts from Agile methodologies

Scrum, Kanban, and Scaled Agile Framework (SAFe) are among the most widely used methodologies derived from Agile principles emphasizing flexibility, customer partnership, and iterative development, ideal for adaptive software projects. These principles are maintained in these well-liked frameworks by using time-boxed sprints, adaptive planning, and feedback.

- Scrum frameworks function into sprints with defined roles (Development Team, Product Owner, Scrum Master), ceremonies (daily stand-ups, sprint planning, retrospectives), and artifacts (product backlog, sprint backlog)[4].
- Kanban focuses on continuous delivery and visualized workflow with task boards to support incremental improvement and flow-based management.
- SAFe uses Agile practices to work at the enterprise level. It manages many teams by applying Lean principles and Agile methods. Cao et al. [5] and Hoda et al. [4] emphasized that successful Agile projects rely on empowered teams, delivering work in small steps, and being responsive to change. These factors can help progress but also create risks like scope creep.

3.2 Scope Management in Agile vs. Traditional Approaches

Scope management in classical models such as Waterfall is generally formal, with thorough initial planning and little flexibility once requirements are established [3]. On the other hand, Agile methods consider scope as dynamic, changing on an ongoing basis in response to feedback from stakeholders and changing priorities [5].

- Change is avoided during a mid-cycle in Waterfall because it is costly and complex.
- In Agile, change is embraced through backlog refinement and reprioritization, hence creating a fertile ground for out-of-control scope expansion if not kept in check.

Cao et al. [5] and Mishra & Mishra [6] argue that while Agile permits change, its open-ended character is more vulnerable to scope creep if proper controls, measures, and team discipline are not present.

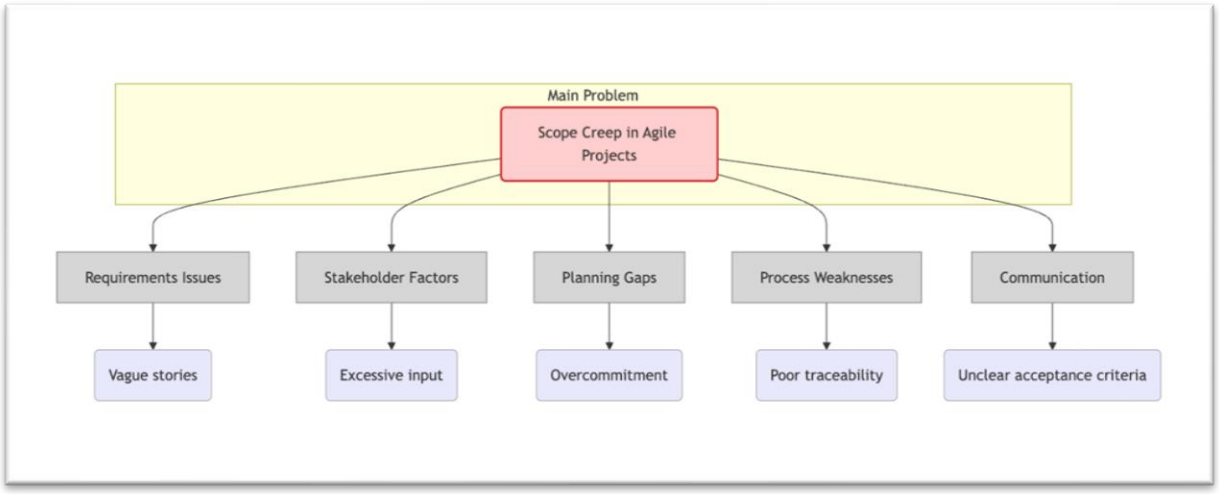
3.3 Common Factors Leading to Scope Creep in Agile Projects

While Agile practices have established frameworks, scope creep remains a chronic problem as a result of some extremely critical factors:

- Poorly defined or ambiguous user stories and requirements [8]
- Excessive and unsupervised stakeholder input [5]
- Ambiguous acceptance criteria [7]
- Overcommitment during sprint planning [4]
- Lack of documentation and traceability [5], [6]

Maalej et al. [8] and Ilays et al. [7] cite scope creep in Agile typically develops stealthily over a sequence of sprints, and the cumulative effect is seen. Manual identification is difficult and retroactive, requiring prediction techniques such as those offered by AI.

Fig 1. Common Causes of Scope Creep in Agile Projects



3.4 Metrics and KPIs in Agile Projects Pertaining to Scope Creep

Agile project health is tracked using a variety of metrics and KPIs, some of which can indirectly signal the presence or potentiality of scope creep:

- Velocity: Measures the work completed within each sprint; sudden spikes or drops might imply scope volatility [4].
- Sprint Burndown Charts: Graphing work left; unstable trends may imply scope expansion.
- Cumulative Flow Diagrams: Reflects workflow bottlenecks or ballooning backlog.
- Story Point Variance: Large variances between planned and actual effort [14].
- Change Requests per Sprint: High numbers reflect unstable or poorly controlled requirements.

These metrics are contenders for machine learning model inputs, as shown in studies by Hein et al. [1], Kamei et al. [9], and Zimmermann et al. [15]. Leveraging such data allows AI systems to recognize patterns indicating nascent scope creep.

Table 1. Comparison of Scope Management in Agile vs. Traditional Approaches

Aspect	Traditional Approach (e.g., Waterfall)	Agile Approach (e.g., Scrum, Kanban)
Flexibility of Scope	Fixed scope defined upfront, minimal changes allowed	Dynamic and evolving scope, adapting continuously
Change Management	Changes are costly and discouraged mid-project	Changes embraced regularly through backlog refinement
Planning Approach	Detailed, upfront planning with fixed deliverables	Iterative, incremental planning aligned with short sprints
Stakeholder Engagement	Limited involvement after initial requirements gathering	Continuous collaboration and feedback from stakeholders
Risk of Scope Creep	Lower risk due to strict control and formal change processes	Higher risk due to flexibility, needing proactive control

4. Proposed Solution

To address the long-standing and vexing issue of scope creep in Agile software development projects, this review proposes embracing and systematically applying AI-based predictive models that are tailor-designed for the dynamic and iterative nature of Agile development. Exploiting advancements in machine learning (ML), deep learning (DL), and risk analytics, the solution involves the development and deployment of smart systems that are able to detect early indicators of scope creep from historical and real-time project data [1], [2], [8], [14].
The key components of the proposed solution are:

4.1 Data Collection and Preprocessing

The Human Resource Management System (HRMS) and the foundation of an AI-driven prediction system are the collection of quality project data. These include sprint backlog histories, change requests, user stories, time logs, defect reports, and communication logs from Agile tools such as Jira, Trello, or Azure DevOps. Natural Language Processing (NLP) may be utilized to extract knowledge from textual user stories and comments [2], [8], [17].

4.2 Feature Engineering and Scope Creep Indicators

Relevant features must be engineered to encapsulate potential predictors of scope creep. These may include:

- Frequency of backlog updates
- Number of requirements changes during a sprint
- Task completion velocity variations
- Ratio of new versus completed user stories
- Bug reopening rates

These metrics over time can be the source of input features for ML models [1], [8], [9].

4.3 Machine Learning and Deep Learning Models

Various AI models can be used to forecast scope creep:

- Neural networks for learning complex, non-linear project patterns [8], [14]
- Ensemble learning (e.g., random forest, gradient boosting) for combining multiple models for better accuracy [10], [15]
- Deep learning techniques such as LSTM for modeling time-series project behavior [1], [9]

These models would be trained using historical Agile project data with instances of scope creep labeled, using classification or regression techniques depending on whether scope creep is framed as a binary or continuous outcome [1], [14].

4.4 Risk Scoring and Alert System

The output of the models would be a Scope Creep Risk Score that would be provided at the end of each sprint or even in real time. The score would be displayed in dashboards that would be visible to Scrum Masters, Product Owners, and team leads. High-risk issues would raise alerts for early decision-making [13], [19].

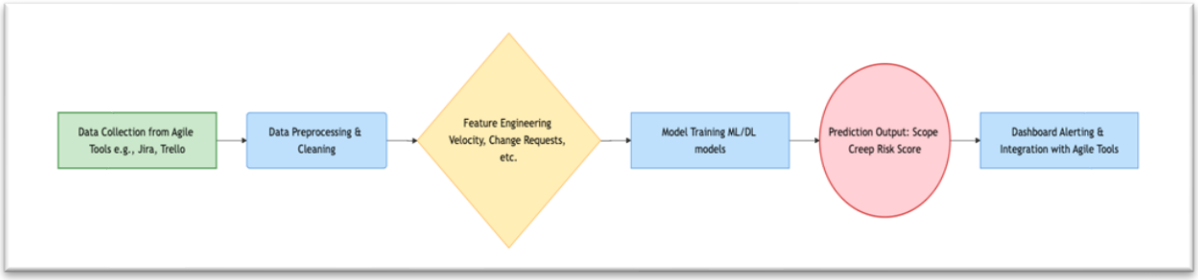
4.5 Integration into Agile Workflows

For adoption and usability, the predictive system needs to be closely integrated with Agile project management tools. Lightweight APIs or plugins can provide in-tool notification and real-time tracking without disrupting team workflows [13], [15].

This AI-based model not only enables predictive insights into scope creep but also promotes forward-looking risk mitigation. It builds on existing work in risk prediction [10], [15], effort estimation [1], [14], and scope identification [8], while transcending drawbacks of traditional manual approaches [6], [12]. Additionally, by adhering to Agile principles of ongoing feedback and change, this solution ensures pragmatic feasibility and real-world effectiveness [4], [5].

The scope creep estimation in Agile software development projects has attracted increasing research interest due to the devastating consequences of unregulated changes in requirements on cost, schedule, and quality. With advancements in Artificial Intelligence (AI), especially Machine Learning (ML) and Deep Learning (DL), predictive modeling methods have been proposed and utilized to forecast scope creep, manage project risk, and improve decision-making. This sub-section blends the most current AI-based predictive models usable in scope creep detection and control in Agile environments.

Fig 2. AI-Driven Scope Creep Prediction Workflow



5. AI Based Prediction Models

5.1 Machine Learning Models for Risk and Effort Estimation

Several research have been focused on applying standard ML techniques to predict project-related risks, including scope creep. Project management software structured data (e.g., backlog history, user stories, sprint changes) is commonly utilized by such models to identify patterns surrounding project disruptions. Mishra & Mishra [6] and Hein et al. [1] conducted systematic reviews with a focus on how ML models such as support vector machines (SVMs), decision trees, and logistic regression have achieved successful applications in software effort estimation and risk prediction. They are generally used for classification (e.g., whether the sprint will be beset with scope creep) or regression (e.g., predicting how much scope will creep).

5.2 Deep Learning Models for Complex Pattern Recognition

Deep learning models provide the capability of learning from big, unstructured, or time-ordered project data to identify nonlinear dependencies and temporal relationships. Kamei et al. [9] and Koru & Liu [14] used deep neural networks (DNNs) and long short-term memory (LSTM) models in Agile risk forecasting and effort estimation and demonstrated higher accuracy compared to conventional ML methods.These models come in handy especially for time-series analysis, such as sprint speeds, varying backlog sizes, or consecutive requirement changes—issues which usually come before scope creep.

5.3 Ensemble Learning Techniques for Increased Accuracy

Ensemble models, which combine predictions from more than one base learner, have also been shown to excel at

predictive tasks under Agile scenarios. Kamei et al. [9] and Koru & Liu [14] utilized random forests and gradient boosting machines (GBMs) to predict delays that are highly correlated with scope creep. Ensemble models reduce overfitting and increase robustness and are thus best suited for heterogeneous Agile project data.

5.4 Neural Networks for Automatic Detection of Scope Creep

Increasingly, the literature addresses scope creep prediction using neural network models directly. Maalej et al. [8] showed that feedforward neural networks are effective at detecting scope creep by analyzing requirement volatility, backlog changes, and bug frequency. Binkhonain & Zhao [17] also explored neural and ML-type models learned from Agile sprint data to predict scope creep events, which confirmed their potential for early warning mechanisms.

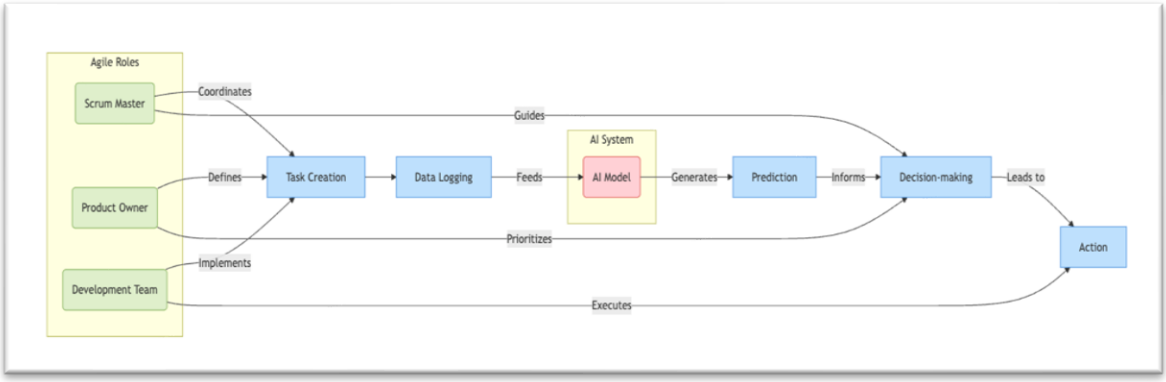
5.5 Natural Language Processing (NLP) for Requirement and Communication Analysis

NLP models allow the algorithms to learn valuable signals from free text data, such as user stories, comments, and change logs. NLP, sentiment analysis, and semantic evaluation of user stories were utilized by Tantithamthavorn et al. [13] to enhance scope control. Xu et al. [2] employed a fusion of NLP and classification algorithms for predicting project overrun using historical communication logs and requirement descriptions.

5.6 Hybrid and AI-Integrated Project Management Systems

Recent studies indicate the integration of AI models in real-time Agile tools to continuously monitor and provide decision-making support. Zimmermann et al. [15] and Radlinski [19] designed hybrid AI systems with ML, rule-based inference, and real-time alerts for proactive risk management in Agile processes. Tantithamthavorn et al. [13] designed an integrated dashboard using AI models to provide scope creep risk scores in Agile project management tools.

Fig 3. Integration Points of AI Model in Agile Workflow



6. Application of the Solution in Organizational Processes

Application of AI systems for scope creep forecasting can go a long way in organizational proficiency in managing Agile projects. To transform theoretical models into tangible business value, organizations must incorporate these AI tools into their existing software development life cycles, governance systems, and Agile ceremonies. This is how the solution can be practically applied to organizational processes:

6.1 Integration into Agile Project Management Tools

Projects generally implement project management tools such as Jira, Azure DevOps, or Trello to manage Agile processes. AI models are embedded in these tools using plugins or APIs such that they process sprint data, backlog history, user stories, and work logs in real time automatically [8], [13]. This makes monitoring effortlessly possible without interrupting team procedures such that the AI system serves as an augmentation—not replacement—of current processes [2], [15].

6.2 Enhancing Sprint Planning and Retrospectives

Artificial intelligence-powered scope creep prediction can be used in sprint planning sessions to highlight high-risk items in the backlog or feature additions that have the potential to maximize the likelihood of scope creep. Predictive insights of this sort provide data-driven feedback to Scrum Masters and product owners so that they can prioritize more effectively [5], [7]. During retrospectives, AI-calculated metrics and trend analysis can help to identify recurring scope problems, thereby ensuring ruthless improvement [4], [6].

6.3 Strengthening Risk Management and Governance

Traditional risk management models do not work in Agile environments due to fluctuating requirements. Forecasting models with the assistance of AI fill the gap by providing dynamic, real-time risk assessment on fluctuating project conditions [6], [10]. These models can be integrated into the firm's risk governance framework, offering the project managers and portfolio managers a risk score dashboard, areas of potential impact, and recommendations for mitigation [10], [19].

6.4 Enabling Agile Requirement Engineering

Requirement volatility is among the root causes of scope creep [8], and AI systems can be employed to identify problematic user stories early in the development process. With analysis of historical scope, budget, and velocity impacts of similar stories, the system can warn uncertain or high-risk requirements. This feedback mechanism

enables enhanced definition, estimation, and negotiation of requirements among stakeholders and developers [1], [14].

6.5 Alignment with Organizational KPIs and Reporting

Scope creep intrudes on primary performance metrics such as delivery timelines, cost management, and customer satisfaction. AI-powered prediction systems can provide executive-grade reports of project status, future trend forecasting, and risk of scope deviation [12], [17]. These input strategic decision-making at the program and portfolio levels and drive IT performance alignment with organizational goals [3], [19].

6.6 Building a Continuous Learning Culture

With time, the predictive accuracy of the AI model improves through continuous exposure to organizational project information. This creates a feedback-driven learning system that gets aligned with the Agile maturity of the organization [15], [20]. Organizational adoption can also be improved by change management programs, training, and cross-functional AI literacy workshops that can ensure long-term value [6], [16].

In brief, incorporating AI-powered scope creep prediction models into organizational processes builds a predictive data-oriented Agile culture. By maximizing visibility, eliminating uncertainty, and automating risk detection, this solution significantly enhances an organization's ability for timely delivery of high-quality software within budget

7. Challenges and Limitations

- Limited access to training data - Most datasets for Agile projects are proprietary, and it is not easy to obtain enough high-quality data for AI models to learn from [1], [2], [8].
- Difficulty with labeling scope creep - It is not simple to label clearly when scope creep happens as it might be based on team understanding and context [5], [7], [8].
- Variable definitions of scope creep - Different Agile teams may define and understand scope creep differently, hence it is challenging to develop a generic AI model [5], [6], [8].
- Agile is meant to accommodate scope changes - Since Agile supports changes in the process, separating normal changes from unwanted scope creep is not always convenient [5], [6], [8].
- AI models work for all projects - Models learned on data from one team or project may not work well on another because they vary in practice and environment [6], [14], [17].
- AI models can be difficult to comprehend - Certain AI techniques, particularly deep learning, do not explicitly exhibit how they decide on something, which can diminish the trust in their predictions [4], [13].
- Agile teams may be resistant to automation tools - Agile places high value on team communication and flexibility, and so bringing AI tools into play may be viewed as constraining their autonomy [4], [5], [6].
- AI tools won't necessarily integrate easily with current workflows - It may be difficult to link AI-backed prediction tools to project management tools typically utilized by Agile teams [13], [20].
- No standard means for testing AI models - No widely accepted metrics or standards exist to gauge the accuracy of AI predictions of scope creep in Agile projects [6], [17].
- Ethical and privacy problems with data - Using real Agile project data might be a source of privacy and security risks, especially when the data includes clients or sensitive information [12], [19].
- Bias in AI models - If biased practices or shortcomings are incorporated in the training set, the AI model can replicate them in its predictions [12], [19].
- Agile practices vary enormously - Agile is not a norm approach—the teams implement it differently, so it is challenging to develop one model that will fit all [3], [7], [8].
- Agile teams develop with time - Team structures and working habits in Agile projects change often, something that may make previous prediction models less effective in the long term [13], [20].

8. Future Research Directions

Subsequent research in AI-powered scope creep forecasting for Agile projects needs to start by creating large, heterogeneous datasets from actual Agile settings, since existing models tend not to have strong training data [1], [2], [8], [14]. There is also a necessity to establish a standard definition of what scope creep is, across various teams, since different definitions render it difficult to train precise models [5], [7], [8]. Researchers must take explainable AI approaches to make the predictions more transparent and interpretable to enable teams to believe and embrace them [4], [13], [17]. Studies on the impact of AI tools on team behavior and decision-making in Agile processes may also come up with more user-focused solutions [2], [4], [6]. Future models must be effective across many Agile approaches such as Scrum or Kanban, which differ significantly in practice [3], [7], [8]. Simple integration of AI tools in commonly used project management software like Jira would make them more applicable in real-world scenarios [13], [20]. More comparison among various AI techniques like deep learning, ensemble methods, and neural networks can further help identify the most effective ways for scope creep prediction [10], [15], [17]. Early prediction models are also a priority because identifying scope creep early on saves time-consuming delays [18], [17], [19]. Use of Agile teams' communication records, such as chats or minutes of the meeting, can also help to locate initial signs of scope changes [4], [6], [19]. Lastly, privacy and ethical concerns about the use of project data need to be managed to deliver safe and ethical AI tools in this context [12], [19], [20]

9. Conclusion

Forecasting and management of scope creep in Agile projects is now an area of research focus, especially with

increasing adoption of Artificial Intelligence (AI) and Machine Learning (ML) techniques. The current review has proven that AI-driven models hold high capabilities to enhance the discovery and management of scope creep at an earlier stage, which is generally triggered by evolving requirements, stakeholder impacts, and dynamic business demands [5], [7], [8]. Experiments with ensemble learning, neural networks, and deep learning have shown to be effective for comparable applications such as scope creep estimation, delay estimation, effort estimation, and project risk estimation [1], [10], [14], [15], [20]. These advances suggest that the same AI practices will apply equally well to scope creep estimation. There is a positive development notwithstanding. All the existing models are plagued by data availability, volatility of projects, and non-standardization of datasets [8], [17]. Also, human interactions in Agile processes are so intricate that modeling them with data-driven methods alone is quite difficult [4], [6]. The literature also indicates the lack of explainability and transparency in AI models, which may restrict their usage by project managers and Agile teams [4], [13], [19]. Another is that while AI tools have been developed to support decision-making, their applicability in real-world environments is still yet to be adequately tested [2], [12]. Moreover, integrating AI tools with Agile processes such as Scrum or SAFe remains an area of less investigation, especially regarding user experience and teamwork between team members [3], [7].

In summary, AI offers a promising path for forecasting and managing scope creep in Agile software development. To realize its full potential, however, future research must tackle the challenges of acquiring high-quality project-specific data, building explainable and generalizable models, and designing AI tools closely in step with Agile principles and workflows. Collaboration among AI researchers, software engineers, and project managers will be essential to bridge the divide between theoretical advances and practical usability.

Acknowledgments

I would like to express my sincere gratitude to the publisher for their support in bringing this article to publication. I appreciate the resources and platform provided, which have enabled me to share my findings with a wider audience. My thanks also go to the editorial team for their thoughtful review and careful editing of the manuscript. I am grateful for the opportunity to contribute to this field through this publication, and for the invaluable assistance that made this work possible.

Author contributions

The author contributed to all aspects of the research and manuscript preparation.

Conflicts of interest

The author declares no conflict of interest.

References

[1] Hein, Phyo Htet, Elisabeth Kames, Cheng Chen, and Beshoy Morkos. "Employing machine learning techniques to assess requirement change volatility." *Research in engineering design* 32 (2021): 245-269

[2] Xu, Chi, Yuanbang Li, Bangchao Wang, and Shi Dong. "A systematic mapping study on machine learning methodologies for requirements management." *IET Software* 17, no. 4 (2023): 405-423

[3] Cheligeer, Cheligeer, Jingwei Huang, Guosong Wu, Nadia Bhuiyan, Yuan Xu, and Yong Zeng. "Machine learning in requirements elicitation: A literature review." *AI EDAM* 36 (2022): e32

[4] Hoda, Rashina, James Noble, and Stuart Marshall. "Self-organizing roles on agile software development teams." *IEEE Transactions on Software Engineering* 39, no. 3 (2012): 422-444

[5] Cao, Lan, Kannan Mohan, Peng Xu, and Balasubramaniam Ramesh. "A framework for adapting agile development methodologies." *European Journal of Information Systems* 18, no. 4 (2009): 332-343

[6] Mishra, Deepti, and Alok Mishra. "Complex software project development: agile methods adoption." *Journal of Software Maintenance and Evolution: Research and Practice* 23, no. 8 (2011): 549-564

[7] Ilays, Irum, Yaser Hafeez, Nabil Almashfi, Sadia Ali, Mamoona Humayun, Muhammad Aqib, and Ghadah Alwakid. "Towards Improving the Quality of Requirement and Testing Process in Agile Software Development: An Empirical Study." *Computers, Materials & Continua* 80, no. 3 (2024)

[8] Maalej, Walid, Maleknaz Nayebi, Timo Johann, and Guenther Ruhe. "Toward data-driven requirements engineering." *IEEE software* 33, no. 1 (2015): 48-54.

[9] Kamei, Yasutaka, Emad Shihab, Bram Adams, Ahmed E. Hassan, Audris Mockus, Anand Sinha, and Naoyasu Ubayashi. "A large-scale empirical study of just-in-time quality assurance." *IEEE Transactions on Software Engineering* 39, no. 6 (2012): 757-773

[10] Mockus, Audris, and David M. Weiss. "Predicting risk of software changes." *Bell Labs Technical Journal* 5, no. 2 (2000): 169-180

[11] Jiarpakdee, Jirayus, Chakkrit Tantithamthavorn, and Christoph Treude. "The impact of automated feature selection techniques on the interpretation of defect models." *Empirical Software Engineering* 25 (2020): 3590-3638

[12] Shepperd, Martin, David Bowes, and Tracy Hall. "Researcher bias: The use of machine learning in software defect prediction." *IEEE Transactions on Software Engineering* 40, no. 6 (2014): 603-616

[13] Tantithamthavorn, Chakkrit, Shane McIntosh, Ahmed E. Hassan, and Kenichi Matsumoto. "The impact of automated parameter optimization on defect prediction models." *IEEE Transactions on Software Engineering* 45, no. 7 (2018): 683-711

[14] Koru, A. Gunes, and Hongfang Liu. "Building effective defect-prediction models in practice." *IEEE software* 22, no. 6 (2005): 23-29

[15] Zimmermann, Thomas, Rahul Premraj, and Andreas Zeller. "Predicting defects for eclipse." In *Third international workshop on predictor models in software engineering (PROMISE'07: ICSE workshops 2007)*, pp. 9-9. IEEE, 2007

[16] Borg, Markus, Orlena CZ Gotel, and Krzysztof Wnuk. "Enabling traceability reuse for impact analyses: A feasibility study in a safety context." In *2013 7th International Workshop on Traceability in Emerging Forms of Software Engineering (TEFSE)*, pp. 72-78. IEEE, 2013

[17] Binkhonain, Manal, and Liping Zhao. "A review of machine learning algorithms for identification and classification of non-functional requirements." *Expert Systems with Applications: X* 1 (2019): 100001
[18] Shatnawi, Anas, Abdelhak-Djamel Seriai, Houari Sahraoui, and Zakarea Alshara. "Reverse engineering reusable software components from object-oriented APIs." *Journal of Systems and Software* 131 (2017): 442-460
[19] Radlinski, Lukasz. "A survey of bayesian net models for software development effort prediction." *International Journal of Software Engineering and Computing* 2, no. 2 (2010): 95-109
[20] Kitchenham, Barbara, and Emilia Mendes. "Why comparative effort prediction studies may be invalid." In *Proceedings of the 5th international Conference on Predictor Models in Software Engineering*, pp. 1-5. 2009