IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 4, Issue 2, December 2024
Impact Factor: 7.57

A Review on AI-Driven Project Management for Transforming Software Engineering Perspectives

Vandana Chaturvedi
Independent Researcher
Vandana.us.tx@gmail.com

Abstract: *The software development life-cycle is getting more and more complicated, which makes it hard for software project managers to keep track of the development of big software systems. This research is based on the premise that AI and ML have a profound impact on software project management and software engineering. In instance, it emphasizes the application of AI methods like intelligent automation in Agile and DevOps models, predictive analytics for allocating resources, NLP for requirements analysis, and ML models for estimating effort and risk. The review also examines the use of AI in driving development in test automation, continuous integration, and deployment processes and how the technologies have improved decision-making, operational efficiency, quality assurance and project success rates. It further addresses the issues of implementation, such as data dependency, model accuracy, ethical issues, and testing complexity, and defines the research agenda in the future, which consists of governance frameworks, AI-driven development platforms, and the changing role of software professionals. In general, the paper provides an extensive summary of the ways AI is transforming the contemporary software project management and engineering processes.*

Keywords: Artificial Intelligence (AI), Machine Learning (ML), Software Project Management, Software Engineering, AI Governance, Intelligent Automation

I. INTRODUCTION

Software engineering practices have been transformative and fast moving due to the change in technology and the increment in the complexity of software requirements. Over the decades of the software development history, the methodologies have been improved and nowadays they are more iterative and agile than the old-fashioned waterfall model [1]. This development has been directed to meet the increased requirement of efficiency, flexibility, and quality of the software products [2]. Software engineering (SE) research has taken a lot of heat over the years for allegedly being unscientific, lacking in maturity, and missing key components like evaluation. There have been claims that SE research is in crisis and that it promotes rather than evaluates. Some have even challenged the fundamentals of SE research.

The software industry has become increasingly dependent on the efficient development of high-quality software as software products affect all areas of daily life [3] [4]. The agile frameworks and especially Scrum are increasingly popular methods of software development, which offer greater flexibility, shorter development cycles, and better quality of the product [5]. Software development lifecycle automation and improvement through collaboration, continuous feedback taking, and automation is the goal of both Agile frameworks and Development and Operations (DevOps) [6] practices. Despite widespread recognition of Scrum and DevOps' benefits [7], very small entities (VSEs) may find the implementation process particularly challenging [8].

The use of AI into project management has grown into a complicated endeavour, impacting businesses' project planning, monitoring, and execution processes. AI-powered solutions streamline decision-making, automate mundane operations, and maximise resource utilisation through the use of machine learning, NLP, and predictive analytics [9]. For the last ten years, AI has been used in project [10]. The last ten years have witnessed the evolution of AI use in project management to include more complex applications of predicting models and real-time risk identification in addition to automation of simple tasks and greatly increases efficiency and minimizes project delays. Decision support provides project teams with

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 4, Issue 2, December 2024
Impact Factor: 7.57

real-time accuracy using AI tools that can display pattern discoveries in extensive datasets. Machine learning models forecast the outcome of sprints, assess how effective the team is, and identify arising issues to increase the number of fast reactions and project success outcomes in both short-term and long-term perspectives. Figure 1 shows how AI has been integrated in project management, its applications, tools, advantages, and best practices.

Figure 1: Integrating Artificial Intelligence into Modern Project Management Practices

A number of persistent problems in software engineering are being addressed to a higher degree with the advent of AI [11]. Software Engineering (SE) phases could be radically altered by the advent of artificial intelligence (AI) [12] applications, which aim to provide better software, boost productivity, and enhance project success rates. Software development teams might benefit from AI in a number of ways, such as with decision-making, project analytics, and the automation of important jobs during the SE phase [13]. The SE phases [14] saw substantial investigation into many AI approaches, including machine learning (ML), natural language processing (NLP), and others. Both the time and the money needed to produce software rise in direct proportion to its size and complexity [15].

A. Structure of the Paper

The paper structure is as follows: Section II presents AI techniques in software management projects. Section III talks about how AI has transformed software engineering practices as seen in Agile, DevOps, and test automation. Section IV summarizes the implementation issues and research future direction. The literature evaluation is found in Section V, and the study is concluded with suggestions for future work in Section VI.

II. AI TECHNIQUES IN SOFTWARE MANAGEMENT PROJECTS

Artificial Intelligence (AI) [16] (with its subdivision as Machine Learning (ML)) represents a new aspect of the software engineering field. The success of software projects is highly dependent on meticulous preparation and implementation. The level of preparation that went into a project might have an effect on the team's performance. Some of these critical components are project needs, cost estimation, and risk reduction. Increased expenses or the project's demise can be the outcome of a poor judgement of this kind. So, a lot of software engineers are trying to figure out how to use AI to make better decisions during project development with less room for error. AI used effectively to software development project decisions has the potential to boost efficiency, improve quality, and decrease development failures and errors. The anticipated application of AI/ML in contemporary software engineering is depicted in Figure 2 below.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 4, Issue 2, December 2024
Impact Factor: 7.57

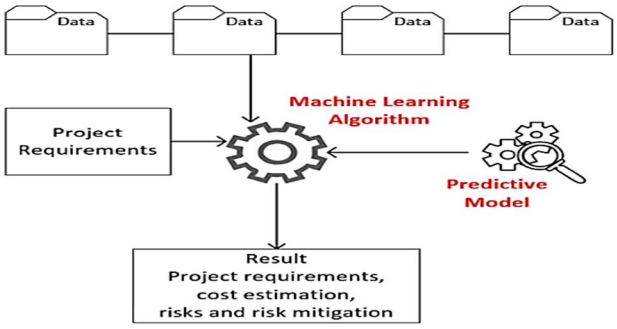

Figure 2: Current AI and ML work with software engineering

A. Machine Learning for Effort Estimation and Risk Prediction

Estimation of effort and risk detection are both established disciplines within the management of software projects, but they are with a high degree of uncertainty, scope changes, and people changes which is difficult [17]. More classical techniques such as experts estimation, estimation by analogy, function point estimation and use case point estimation have given a fair kind of results in controlled environment but the accuracy is not optimal in scattered unstable software environment with turbulent trends where historical information is not consistently foreseeable across other teams or fields. Machine learning algorithms [18], including team velocity, the interactions of developers, and more, can automatically learn patterns from historical data. The number of ML methods are discussed below:

Linear Regression

Two variables, X and Y, are utilised in linear regression to ascertain the association. Here, two variables: one that is independent and one that is dependent. To achieve this, it searches for a line that minimises the set of all squares representing the vertical distances between each set of points and that line.

Support Vector Machine

SVM is a learning method for accurately identifying invisible data. It is uncertain whether the dataset is linearly separable [19]. Another way to apply SVM to non-linear boundaries is by utilising the kernel trick. The data is embedded in a higher-dimensional space via the kernel function to ease separation.

Artificial Neural Network

"Neurones" or "processing units" [20] are the fundamental, interconnected building blocks of an ANN [21]. There are three levels in every ANN: input, hidden, and output [22]. In the first layer, input neurones communicate with each other and the second layer via connections called weights. From there, data is sent to the third layer of output neurones through additional weight connections.

Decision Tree

A DTs is a tool for regression and classification. It streamlines the categorization process and gives a human-friendly modelling approach [23]. This kind of data mining induction technique uses either a depth-first greedy approach or a breadth-first method to split records in a dataset over and over again until every piece of data is categorised according to a set of rules that have already been decided.

Bagging

Bagging, also known as bootstrap aggregating, is a technique that repeatedly uses replacement samples from a dataset in concordance with a smooth probability distribution [24]. The size of each bootstrap sample matches that of the initial

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 4, Issue 2, December 2024
Impact Factor: 7.57

data set exactly. Because replacement sampling is used, some events may show up more than once in the same training set, while others are left out.

Table I shows how different types of ML contribute to better software project management decisions and prediction capability when using AI

Table 1: Machine Learning Techniques for Effort Estimation and Risk Prediction in Software Projects

Technique	Core Concept	Application in Effort Estimation	Application in Risk Prediction	Strengths in Software Projects	Limitations
Linear Regression	Utilised for the purpose of minimising squared error in modelling the linear relationship between dependent and independent variables.	Predicts development effort (e.g., person-hours, cost) based on features like LOC, complexity, team size, story points.	Estimates probability of cost overrun or schedule delay using historical project metrics.	Simple, interpretable, easy to implement; works well with structured historical data.	Assumes linear relationships; sensitive to outliers; limited performance for complex, nonlinear patterns.
Support Vector Machine (SVM)	The kernel trick is used for nonlinear boundaries, and it finds the best hyperplane for regression or classification.	Used in Support Vector Regression (SVR) for nonlinear effort estimation based on multiple project attributes.	Classifies projects into risk categories (high/medium/low risk); detects potential failure patterns.	Effective in high-dimensional datasets; handles nonlinear relationships; robust to overfitting.	Computationally expensive; requires careful kernel and parameter selection.
Artificial Neural Network (ANN)	A network of neurons with multiple layers of connections that may learn complicated patterns by utilizing weighted connections.	Captures complex relationships among project features for accurate effort estimation; useful in agile environments.	Predicts defect risk, failure probability, and schedule slippage using historical project data.	High predictive accuracy; handles nonlinear and complex dependencies; adaptable to large datasets.	Requires large training data; low interpretability; risk of overfitting without proper tuning.
Decision Tree	Tree-based model that recursively partitions data using feature selection criteria.	Identifies key effort drivers (e.g., project type, technology stack) and predicts effort categories.	Classifies project risk levels and identifies critical risk factors (e.g., requirement volatility).	Easy to interpret; visual decision rules; useful for managerial insights.	Prone to overfitting; unstable with small variations in data.
Bagging (Bootstrap Aggregating)	The ensemble approach combines the predictions of several models constructed from bootstrapped samples.	Improves stability and accuracy of effort estimation models (e.g., Bagged Decision Trees).	Enhances reliability of risk prediction by reducing variance and	Reduces overfitting; improves robustness and generalization; effective with	Less interpretable; increased computational cost compared to single models.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 4, Issue 2, December 2024
Impact Factor: 7.57

			model instability.	noisy project data.	

B. Natural Language Processing for Software Requirement Analysis

Software Requirements Engineering (SRE) uses methods like elicitation, design, modelling, and validation to make sure that high-quality software is made. In the past, these procedures have been highly dependent on human analysis and interpretation, which can cause delays, misunderstandings, and erroneous conclusions. Requirements are frequently incomplete or contradictory due to the imprecision of the human language [25]. The requirement handling strategies also need to be more sophisticated to keep up with the increasingly complex software systems. Some have proposed using formal and restricted languages to lessen ambiguity; however, these languages are hard to learn and use, and they lack the expressiveness of natural language. Figure 3 shows the NLP techniques employed in Software Requirements.

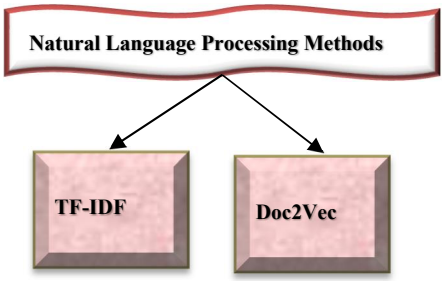

Figure 3: NLP Technologies used in Software Requirement Analysis

Term Frequency-Inverse Document Frequency (TF-IDF)

The term frequency in a text corpus can be determined using TF-IDF. This value is equal to the product of two statistics: one that measures the frequency with which a word appears in a document (the word frequency statistic) and another that measures the frequency with which a phrase appears in documents (the inverse document frequency statistic) [26]. Text processing machine learning pipeline normally starts with the TF-IDF process. Its results are then fed to classifiers and regressors.

Distributed Representations of Documents (doc2vec)

Models like TF-IDF and word2vec [27] and doc2vec acknowledge that words' meanings are lost during the process. By multiplying Doc2vec vectors with TF-IDF vectors, and can utilize both together, but run the risk of retaining information from both approaches.

Enhancing Resource Allocation Efficiency through Predictive Analytics in Software Project Management

Traditional software project management (SPM) methods rely on both predetermined plans and gut feelings from managers when allocating resources. The increasing complexity, dynamics, and uncertainty of modern software projects are too much for these approaches to handle. ML [28], statistical modelling, and historical project data are the innovative components of predictive analytics, which are used to generate data-driven judgements. Predictive analytics [29] assists project managers in being proactive and anticipating resource needs by predicting job durability, developer productivity, fault likelihood, and workload phenomena. This allows them to avoid solving too many problems after they have occurred. According to Table II, one of the most important steps in managing software infrastructure that is both agile and hybrid [30] is moving from reactive to proactive planning.

Table 2: Predictive Analytics in Resource Allocation

Function	Predictive Application	Efficiency Benefit
Effort Estimation	ML models use past task data to forecast task time.	Increases the accuracy of planning and decreases under- and overestimation

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 4, Issue 2, December 2024 Impact Factor: 7.57

Workload Forecasting	Future resource requirements can be predicted using time-series analysis.	Facilitates less chaotic scheduling and safeguards against overloads
Skill Matching	Tasks are assigned by recommender systems according to developer profiles.	Maximises efficiency and effectiveness
Bottleneck Prediction	Estimates when there may be scheduling conflicts or resource overlaps	Reduces downtime and allows for early reallocation
Defect Risk Forecasting	Identifies potentially problematic modules or activities	Determines which quality assurance resources are most important
Dynamic Adjustment	Provision of real-time data to prediction algorithms	Maintains allocation plans that are adaptable to changes as they occur

III. TRANSFORMATIONAL IMPACT ON SOFTWARE ENGINEERING PRACTICES

Information technology project-based organizations deploy more resources to implement processes that contribute to managing and realizing their products and services. The value dimensions of self-organise in IT-project systems are not reimagined to generate new dimensions that align with the IT-project's objective that conform to the organization's overarching mission, despite the complexity of emerging global and multi-national organisations and the difficulty of leading an IT project in the convergence of stakeholders, most of whom come from different backgrounds.

A. AI in Agile Frameworks for Software Development

Agile software development approaches have been effective in recent times due to their feedback-based paradigm, flexibility, and iterative nature. Implementing Agile frameworks such as Kanban and Scrum has greatly enhanced software teams' agility in responding to the demands and whims of the ever-changing market. Another attractive feature of these frameworks is their manageable and quick development cycles. Quicker, higher-quality work can be produced with the help of AI since it can start to fix some of the problems that traditional Agile teams have. With the help of AI-powered technologies, Streamline processes like predictive analytics, optimize resources, and make automated judgements. This empower teams to make more informed decisions when it comes to data. The main roles and usage of AI in Agile frameworks are summarized in Table III, indicating the way AI promotes automation, sprint planning, predictive analysis, resource allocation, and efficiency of the workflow in general

Table 3: Key Roles and Applications of Artificial Intelligence in Agile Frameworks

Area	Role of AI	How AI Contributes	Impact on Agile Teams
Handling Project Complexity	Data Processing & Intelligent Automation	Analyzes large volumes of project data and automates repetitive tasks	Reduces manual workload and supports faster decision-making
Automation of Repetitive Tasks	Task Automation	Automates project tracking, bug detection, and code quality checks	Frees time for creative problem-solving and feature development
Predictive Analysis	Forecasting & Risk Detection	Uses historical data to predict bottlenecks, delays, and risks	Enables proactive planning and reduces project disruptions
Decision-Making Support	Intelligent Recommendations	Suggests resource allocation and sprint priorities	Improves accuracy and speed of managerial decisions
Sprint Planning	Data-Driven Estimation	Evaluates past sprint performance, task duration, and resource use	Produces more accurate sprint forecasts

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 4, Issue 2, December 2024
Impact Factor: 7.57

Resource Distribution	Smart Allocation	Allocates work to team members according on their past achievements	Optimizes team productivity and efficiency
Real-Time Monitoring	Dynamic Adjustment	Continuously analyzes sprint progress and suggests changes	Maintains alignment with Agile principles and keeps projects on track
CI/CD Integration	Automated Testing & Deployment	Enhances workflow automation in code testing and deployment	Speeds up feedback loops and improves software quality
Distributed Teams Management	Performance-Based Assignment	Analyzes global team performance and suggests task distribution	Reduces coordination issues and improves collaboration
Continuous Improvement	Trend Analysis	Identifies workflow patterns and performance gaps	Supports continuous process optimization

B. Integrating AI into DevOps Frameworks for Software Development

DevOps [31], which is a portmanteau of development and operations, is a change in culture and operations in the SDLC. It focuses on the cooperation, communication, and the integration of the software developers (Dev) and the IT operations (Ops) teams. The main principle of DevOps is to dissolve the old silos between these two essential functions and have a more seamless, efficient, and faster software delivery process. Software deployment and maintenance procedures are being revolutionised by the incorporation of AI [32] and ML into DevOps operations, which opens the door to the possibility of more efficient, dependable, and scalable systems. Intelligent automation comes with AI-based DevOps, which makes it possible to predict data, detect anomalies, and self-heal infrastructure.

AI/ML Use Cases in DevOps

The importance of DevOps procedures is growing as software systems become larger and more complicated, and companies are utilising AI and ML to improve these processes. Recent developments in AI and ML have paved the way for exciting new opportunities to streamline and automate many parts of the software development process. Included in these phases are code integration and testing, deployment, and maintenance. This section covers a variety of applications and uses:

Code Quality Analysis: AI/ML based Automated code quality analysis entails the utilization of smart algorithms to measure and enhance the quality of code [33]. AI/ML models can study codebases to detect the possible problems, impose the coding rules, and propose improvements. The AI/ML models can be used to predict and fix the code problems by examining the historical code updates, defining patterns related to bugs, and training on the past code reviews.

DeepCode: DeepCode is a machine learning application that is used to analyze code and give insights on the possible problems. It uses a sizeable collection of open-source projects to acquire knowledge of coding patterns and unnoticed problems, including a security [34] vulnerability, performance problems, and code smells. Codacy is a tool that is integrated with CI/CD pipelines in order to conduct automated revisions of the code.

Predictive Analytics: Predictive analytics predicting deployment Predictive analytics of deployment implies applying AI to predict when to deploy code changes and at what strategy to apply. Through the analysis of historical deployment data, AI models are able to suggest the optimal deployment time, evaluate the risk, and recommend the best strategy to ensure the risk is minimized. The AI models able to consider variables like the past deployment success rates, the system load, user activity pattern and environmental conditions to determine the most suitable deployment windows.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 4, Issue 2, December 2024
Impact Factor: 7.57

Tools and Technologies

AI and ML are being provided with the DevOps workflow to improve the automation, efficiency, and reliability. These applications vary in terms of machine learning structures to specific application used in the DevOps.

TensorFlow, PyTorch, and Other ML Frameworks: TensorFlow is a machine learning platform that is open-source and developed by Google. It is frequently employed to develop and deploy AI models. It offers a complete ecosystem of training and serving ML models, and thus it is applicable to various applications in DevOps. PyTorch is another well-liked machine learning framework created by Facebook and which has a dynamic computational graph and is easy to use and deploy ML models to DevOps to support various activities, including code quality inspection and real-time monitoring. There are also other machine learning frameworks [35], like Scikit-Learn, Keras and XGBoost used to do special tasks in DevOps. These systems provide a range of models and codes to develop and assess ML models.

Integration with Existing DevOps Tools: Integrating existing AI/ML frameworks with DevOps technologies can greatly enhance their capabilities. To illustrate: Anomaly detection, predictive analytics, and intelligent code analysis are all possible with Jenkins, a popular CI/CD platform, when connect it with AI/ML extensions [36]. The Docker containerization platform has the potential to exploit AI/ML models to optimize resource allocation, control container health, and make scaling decisions automatically. Terraform as an IaC tool can be used with AI-powered resource management tools to optimize provisioning and configuration of infrastructure.

Platforms for AI-Driven DevOps: A number of platforms provide AI-driven functionality in particular to DevOps. The systems on these platforms offer inbuilt monitoring, automation and optimization solutions. Kubernetes, a free container orchestration software, can be enhanced with AI/ML solutions to work better. Datadog is an all-encompassing observability platform that has AI-powered monitoring, logging, and analytics. It has machine learning abilities that allow it to detect anomalies, predictive analytics, and respond to incidents automatically. New Relic is a provider of observability solutions that are powered by AI to monitor and control performance of an app. Its machine learning models have the ability to identify anomalies, forecast performance problems, and give optimization insights on what to do.

C. AI in Test Automation of Software Projects

The ever-changing software development and quality assurance sectors have recently made test automation, which depends on AI, a prominent topic [37]. The combination of AI and test automation has brought new and ground-breaking methods of running software tests and validation. Traditional test automation is based on the development of scripts and test cases [38] that can imitate the interaction with an application [39]. These scripts are run until they detect errors and guarantee quality of software.

Nonetheless, the use of AI brings with itself a completely new aspect of this process, which has a range of benefits:

Augmented Test Script Generation: The AI algorithms can be used to analyze the features of applications and generate test scripts automatically. The time and effort needed to develop and maintain test cases is greatly decreased as a result.

Self-repairing Test Automation: Automated systems powered by AI can be set up to adapt to application updates and modify test scripts on the fly. This results in less work required for maintenance.

Enhanced Test Data Management: Automated test data generation, management, and delivery of full coverage is possible with AI.

Intelligent Test Execution: AI algorithms can sort critical test cases by importance, based on risk, defect statistics, or usage patterns. It is also optimized to enhance test coverage and minimize execution time.

Dynamic Test Reporting: AI can be used to support intelligent test reporting and analytics, which give meaningful insights into test results, defects, and quality trends.

IV. IMPLEMENTATION CHALLENGES AND FUTURE RESEARCH DIRECTIONS

ML is a subset of AI, which is defined as the intelligence displayed by machines [40]. ML uses datasets to train algorithms, and those algorithms can then be utilised to perform certain tasks autonomously. AI has a lot of potential in

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 4, Issue 2, December 2024 Impact Factor: 7.57

software engineering, especially when it comes to planning software and managing projects. Examines previous research on artificial intelligence and its software engineering uses.

A. Challenges

The majority of these sources concur that software engineering project management would greatly benefit from AI support [39], but they also point out some of the difficulties [41] in putting AI use into practice in teams.

The trustworthiness of AI and ML model-built systems is one such challenge. It has been noted that although computers can mimic human behaviour, real goods cannot replicate human approaches due to the inaccuracy that would result.

The ethical considerations that arise when applying AI models present still another obstacle. Unfortunately, for advanced, continually learning AI systems, the current standard of just doing a single risk assessment before moving on to the next **AI model is often inadequate.**

The testing of ML and AI systems is not without its challenges. Since AI can be wrong even with well-implemented algorithms, testing these systems may require a different strategy than testing typical systems. It can be quite challenging to properly train and deploy an AI system when there is insufficient data because AI systems can only learn from the data that is already available.

The capacity of AI to act in accordance with AI is one of the biggest obstacles facing the AI business. Numerous chat AI attempts have met with disastrous failure, necessitating their removal. As said before, this can be because the AI's current data is influencing its performance.

B. Future Roadmap

There are challenges to using AI in software engineering, there is also possible future work that could mitigate these problems. The future roadmap evaluation of AI in software project management offered in Table IV includes areas of interest, anticipated outcomes, and necessary efforts to facilitate successful, responsible, and scalable AI implementation.

Table 4: Future Roadmap of Software Project & Engineering

Key Area	Future Direction	Impact on Software Projects	Required Actions
Foundational AI Implementation	Train baseline AI/ML systems to address core project management problems	Establish reliable AI support in estimation, scheduling, and risk identification	Develop quality datasets and validate AI tools in real-world projects
Ethical & Logistical Governance	Address ethical, operational, and logistical challenges of AI adoption	Increased trust, transparency, and responsible AI integration	Develop governance frameworks and explainable AI standards
AI-Powered Abstraction Layers	Creation of simple, user-friendly AI-driven development platforms	Coding becomes more accessible; reduced technical barriers	Invest in low-code/no-code and AI-assisted development tools
Transformation of Developer Roles	Shift from manual coding to AI supervision and strategic oversight	Developers take on more administrative and DevOps-oriented responsibilities	Upskill in AI management, system integration, and automation oversight
AI-Driven Project Management & Analytics	AI assists in planning, monitoring, reporting, and predictive data analysis	Automation of technical management duties and better decision-making	Train models on project data and integrate AI into PM workflows

V. LITERATURE REVIEW

Automated decision-making, risk management, collaboration, and operational efficiency are some of the ways in which AI is influencing software project management. This section compares and contrasts articles on the subject. Table V

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 4, Issue 2, December 2024 Impact Factor: 7.57

presents a summary of the focus of each study, the problems tackled, AI structures and solutions, the areas of use and the phases of the project affected.

All the reviewed studies testify to the gradual adoption of Artificial Intelligence in Software Project Management at various stages of the SDLC. The article by K. R. Raghi et al. (2024) takes the concept of AI-based development to full SDLC automation with the help of LLMs and LangChain, showing a comprehensive picture of AI development [42]. In line with this, M. Kanbur, O. P. C, and P. Kulkarni (2023) concentrate on AI-enhanced Agile project management, the predictive analytics and automation enhance the process of planning the sprint, the management of the backlog, and mitigating the risks [43]. Although these studies focus on strategic and strategic decision-making, C. Nitin Rajadhyaksha and J. R. Saini (2022) take it a step further to include Robotic Process Automation (RPA) that involves repetitive administrative duties to enhance efficiency on the process level [44].

Najdawi and A. Shaheen (2021) analyze appropriate governance techniques of AI transformation initiatives, connecting AI implementation and organizational alignment and strategic goals [45]. Growing to operational excellence, Z. Wang, M. Shi, and C. Li (2020) implement machine learning as a part of DevOps to improve monitoring, risk control and system intelligence [46]. D. Quintanilla-Perez et al. (2019) Lastly, discuss collaborative dimensions using Essboard, which strengthens team awareness and vision of the project [47]. Collectively, these studies describe a stratified development, collaboration and process automation to intelligence and end-to-end AI-based software project system

Table 5: Summary of AI-Driven Frameworks in Software Project Management

Reference	Study Focus	Problem Addressed	Application Area	AI / Framework	Tools Proposed	Project Phase Impacted
K. R. Raghi et al. (2024)	Automation of SDLC using LLMs and LangChain	Manual and fragmented SDLC processes; integration challenges in AI adoption	Full Software Development Life Cycle (SDLC)	LLMs, LangChain Framework	AI-driven SDLC automation system	Entire SDLC (Planning, Development, Testing, Deployment)
M. Kanbur et al. (2023)	AI-enhanced Agile Project Management framework	Project risks, inefficient planning, administrative overhead in agile projects	Agile / Scrum Project Management	AI-driven analytics, predictive modeling	AI-supported agile management framework	Planning, Backlog Management, Risk Management
C. N. Rajadhyaksha & J. R. Saini (2022)	Use of RPA in Software Project Management	Time-consuming repetitive administrative tasks reducing efficiency	Software Project Management Processes	Robotic Process Automation (RPA)	RPA tools for task automation	Administrative, Monitoring, Reporting Phases
A. Najdawi & A. Shaheen (2021)	Evaluation of suitable PM methodology for AI transformation projects	Companies' long-term objectives and AI projects aren't complementary.	AI Transformation / Digital Transformation Projects	Comparative analysis of PM methodologies	Methodology evaluation framework	Strategic Planning, Project Governance

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 4, Issue 2, December 2024 Impact Factor: 7.57

Z. Wang et al. (2020)	Intelligent DevOps platform using Machine Learning	Inefficient DevOps alarm handling, business risk and cost control issues	DevOps in Software & Hardware Systems	Machine Learning, DevOps Framework	Intelligent DevOps Platform	Deployment, Monitoring, Operations
D. Quintanilla-Perez et al. (2019)	Collaborative project management using Essboard	Lack of team collaboration and shared project vision in Essence tools	Software Project Collaboration / Essence Framework	Collaborative tool framework (Essence-based)	Essboard Tool	Project Tracking, Review, Collaboration Phase

VI. CONCLUSION AND FUTURE WORK

The AI Techniques in Software Project Management discusses how AI techniques are transforming the contemporary software engineering practice. Overall, this paper finds that AI and Machine Learning can be applied to all project management processes, including effort estimation, risk prediction, requirements analysis, resource allocation, Agile planning, DevOps automation, or test management, to increase the accuracy of decision making, project operational efficiency, and general quality of the software. AI-based predictive analytics allows planning ahead, whereas intelligent automation saves human resources and enhances consistency throughout the software development life cycle. Moreover, requirement engineering and risk assessment processes are empowered through the use of NLP and ML models, and this reduces uncertainties that usually result in the failure of project implementation. In spite of these benefits, data quality and model interpretability, issues of ethical governance, and system validation are paramount in consideration. Sustainable adoption will necessitate dealing with these challenges by providing a strong system of governance and constant improvement of models. The AI, as a whole, offers a revolutionary chance to enhance the success rate of projects and change the future of the software project management.

The future research should be directed at creation of the standardized evaluation frameworks, explainable and transparent AI models, effective governance mechanisms, and practical cases to prove the performance. More research on hybrid AI-human collaborative frameworks, adaptive learning systems, and scalable AI platforms in Agile and DevOps systems will contribute to improved reliability, trust, and pervasive deployments in the software project management

REFERENCES

[1] M. Kuhrmann *et al.*, “What Makes Agile Software Development Agile?,” *IEEE Trans. Softw. Eng.*, vol. 48, no. 9, pp. 3523–3539, Sep. 2022, doi: 10.1109/TSE.2021.3099532.

[2] S. Thangavel, K. C. Sunkara, and S. Srinivasan, “Software-Defined Networking (SDN) in Cloud Data Centers: Optimizing Traffic Management for Hyper-Scale Infrastructure,” *Int. J. Emerg. Trends Comput. Sci. Inf. Technol.*, vol. 3, no. 1, pp. 29–42, 2022, doi: 10.63282/3050-9246.IJETCSIT-V3I3P104.

[3] P. Chandrashekar, “Enhancing Software Application Efficiency Through Design-Centric Methodologies: An Empirical Evaluation,” *ESP J. Eng. Technol. Adv.*, vol. 2, no. 1, pp. 187–196, 2022, doi: 10.56472/25832646/JETA-V2I1P122.

[4] B. V. Swamy, P. P. Barmola, S. Thangavel, S. Kaliappan, H. Patel, and G. Abhyankar, “Cognitive Twins for Predictive Maintenance and Security in IoT Software Systems,” in *2024 4th International Conference on Mobile Networks and Wireless Communications (ICMNWC)*, 2024, pp. 1–8. doi: 10.1109/ICMNWC63764.2024.10872082.

[5] H. A. Ordoñez and C. Cobos, “ISO 29110 en Colombia: de la teoría a la práctica,” *Rev. Guillermo Ockham*, vol. 17, no. 2, pp. 71–80, Dec. 2019, doi: 10.21500/22563202.4299.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 4, Issue 2, December 2024 Impact Factor: 7.57

[6] P. Chandrashekar, "A Survey of Tools, Techniques, and Best Practices: CI/CD Integration in DevOps Workflows," *Int. J. Adv. Res. Sci. Commun. Technol.*, vol. 3, no. 3, pp. 1366–1376, Jul. 2023, doi: 10.48175/IJARSCT-11978V.
[7] Y. Macha and S. K. Pulichikkunnu, "A Survey of DevOps Practices for Machine Learning and Artificial Intelligence Workflows in Modern Software Development," *ESP J. Eng. Technol. Adv.*, vol. 4, no. 3, pp. 200–208, 2024, doi: 10.56472/25832646/JETA-V4I3P121.
[8] H. Annous, L. Livadas, and G. Miles, "OffshoreQA: A Framework for Helping Software Development Outsourcing Companies Comply with ISO 9001:2008," in *2010 5th IEEE International Conference on Global Software Engineering*, IEEE, Aug. 2010, pp. 313–315. doi: 10.1109/ICGSE.2010.43.
[9] J. Gil Ruiz, J. Martínez Torres, and R. González Crespo, "The Application of Artificial Intelligence in Project Management Research: A Review.," *Int. J. Interact. Multimed. Artif. Intell.*, vol. 6, no. 6, pp. 54–66, Jun. 2021, doi: 10.9781/ijimai.2020.12.003.
[10] N. Berente, B. Gu, J. Recker, and R. Santhanam, "Managing Artificial Intelligence," *MIS Q.*, vol. 45, no. 3, pp. 1433–1450, Sep. 2021, doi: 10.25300/MISQ/2021/16274.
[11] S. Singamsetty, "Transforming Data Engineering with Quantum Computing: A New Frontier for AI Models," *Int. J. Comput. Math. Ideas*, vol. 16, no. 03, pp. 3066–3077, 2024, doi: 10.70153/IJCMI/2024.16303.
[12] M. El Samad, G. Nasserddine, and A. Kheir, "Introduction to Artificial Intelligence," in *Artificial Intelligence and Knowledge Processing*, Boca Raton: CRC Press, 2023, pp. 1–14. doi: 10.1201/9781003328414-1.
[13] J. Copeland, *Artificial intelligence: A philosophical introduction*. 2015.
[14] T. C. R Mitchell, J Michalski, *An artificial intelligence approach*. 2013.
[15] H. K. Dam, "Artificial intelligence for software engineering," *XRDS Crossroads, ACM Mag. Students*, vol. 25, no. 3, pp. 34–37, Apr. 2019, doi: 10.1145/3313117.
[16] S. Singamsetty, "AI-Based Data Governance: Empowering Trust and Compliance in Complex Data Ecosystems," *Int. J. Comput. Math. Ideas*, vol. 13, no. 03, pp. 1007–1017, 2021, doi: 10.70153/IJCMI/2021.13301.
[17] K. Patel and S. Gupta, "Prediction of Software Defects for Quality Assurance and Improvement Based on Machine Learning Methods," in *2024 2nd International Conference on Advances in Computation, Communication and Information Technology (ICAICCIT)*, IEEE, Nov. 2024, pp. 400–406. doi: 10.1109/ICAICCIT64383.2024.10912382.
[18] S. Thangavel, S. Srinivasan, S. B. V. Naga, and K. Narukulla, "Distributed Machine Learning for Big Data Analytics: Challenges, Architectures, and Optimizations," *Int. J. Artif. Intell. Data Sci. Mach. Learn.*, vol. 4, no. 3, pp. 18–30, Oct. 2023, doi: 10.63282/3050-9262.IJAIDSML-V4I3P103.
[19] S. K. Shevade, S. S. Keerthi, C. Bhattacharyya, and K. R. K. Murthy, "Improvements to the SMO algorithm for SVM regression," *IEEE Trans. Neural Networks*, vol. 11, no. 5, pp. 1188–1193, 2000, doi: 10.1109/72.870050.
[20] B. Yegnanarayana, *Artificial neural networks*. 2009.
[21] K. Gurney, *An introduction to neural networks*. 2018.
[22] T. M. Khoshgoftaar, E. B. Allen, J. P. Hudepohl, and S. J. Aud, "Application of neural networks to software quality modeling of a very large telecommunications system," *IEEE Trans. Neural Networks*, vol. 8, no. 4, pp. 902–909, Jul. 1997, doi: 10.1109/72.595888.
[23] J. R. Quinlan, "Learning With Continuous Classes 2," *Mach. Learn.*, vol. 92, pp. 343–348, 2006.
[24] L. Breiman, "Bagging predictors," *Mach. Learn.*, vol. 24, no. 2, pp. 123–140, Aug. 1996, doi: 10.1007/BF00058655.
[25] S.-C. Necula, F. Dumitriu, and V. Greavu-Șerban, "A Systematic Literature Review on Using Natural Language Processing in Software Requirements Engineering," *Electronics*, vol. 13, no. 11, p. 2055, May 2024, doi: 10.3390/electronics13112055.
[26] H. Demian, "Natural Language Processing and Machine Learning Methods for Software Development Effort Estimation," *Stud. Informatics Control*, vol. 26, no. 2, Jun. 2017, doi: 10.24846/v26i2y201710.
[27] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," *31st Int. Conf. Mach. Learn. ICML 2014*, vol. 4, pp. 2931–2939, 2014.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 4, Issue 2, December 2024 Impact Factor: 7.57

[28] S. Amrale, "Proactive Resource Utilization Prediction for Scalable Cloud Systems with Machine Learning," *Int. J. Res. Anal. Rev. (IJRAR*, vol. 10, no. 4, pp. 758–764, 2023, doi: 10.56472/25832646/JETA-V3I8P119.
[29] N. Kolli and N. K. R. Choppa, "AI Agents for Predictive and Prescriptive Analytics: Enhancing Foresight and Strategy," *Comput. Fraud Secur.*, vol. 2024, no. 7, pp. 275–284, Jul. 2024, doi: 10.52710/cfs.745.
[30] A. Warrier, "Securing and Scaling API Gateways in Hybrid Environments," *United Res. Forum*, vol. 2, no. 2, p. 7, 2024, doi: 10.48550/arXiv.2601.11935 Focus to learn more.
[31] S. K. Chintagunta, "Survey of Containerization , Orchestration , and CI / CD Integration on DevOps in Modern Software Development," *Int. J. Curr. Eng. Technol.*, vol. 13, no. 6, pp. 610–618, 2023, doi: 10.14741/ijcet/v.13.6.14.
[32] S. Katangoori and A. Katangoori, "Data-Centric AI in the Era of Large Volumes: Improving Model Outcomes through Data Quality Engineering," *Am. J. Data Sci. Artif. Intell. Innov.*, vol. 3, pp. 430–457, 2023.
[33] S. K. Chintagunta and S. Amrale, "AI in Code , Testing , and Deployment : A Survey on Productivity Enhancement in Modern Software Engineering," *Int. J. Curr. Eng. Technol.*, vol. 13, no. 6, pp. 627–634, 2023, doi: 10.14741/ijcet/v.13.6.16.
[34] A. Syed, *Supply Chain Software Security*. 2024.
[35] O. C. Oyeniran, A. O. Adewusi, A. G. Adeleke, L. A. Akwawa, and C. F. Azubuko, "AI-driven devops: Leveraging machine learning for automated software deployment and maintenance," *Eng. Sci. Technol. J.*, vol. 4, no. 6, pp. 728–740, 2023, doi: 10.51594/estj.v4i6.1552.
[36] S. Garg, "Graph Neural Networks For Dependency Mapping And Impact Estimation In Ci/Cd Workflows," *Reseachgate.Net*, vol. 6, pp. 439–449, 2021, doi: 10.5281/zenodo.15552164.
[37] S. R. Thota, S. Arora, and S. Gupta, "Al-Driven Automated Software Documentation Generation for Enhanced Development Productivity," in *2024 International Conference on Data Science and Network Security (ICDSNS)*, 2024, pp. 1–7. doi: 10.1109/ICDSNS62112.2024.10691221.
[38] S. K. Chintagunta, "Generative AI Approaches to Automated Unit Test Case Generation in Large-Scale Software Projects," *ESP J. Eng. Technol. Adv.*, vol. 4, no. 1, pp. 150–157, 2024, doi: 10.56472/25832646/JETA-V4I1P121.
[39] R. Khankhoje, "The Power of AI Driven Reporting in Test Automation," *Int. J. Sci. Res.*, vol. 7, no. 11, pp. 1956–1959, Nov. 2018, doi: 10.21275/SR231208194832.
[40] R. Dattangire, R. Vaidya, D. Biradar, and A. Joon, "Exploring the Tangible Impact of Artificial Intelligence and Machine Learning: Bridging the Gap between Hype and Reality," in *2024 1st International Conference on Advanced Computing and Emerging Technologies (ACET)*, IEEE, 2024, pp. 1–6. doi: 10.1109/ACET61898.2024.10730334.
[41] A. Syed, "Navigating Future Challenges," in *Supply Chain Software Security*, Berkeley, CA: Apress, 2024, pp. 491–521. doi: 10.1007/979-8-8688-0799-2_11.
[42] R. Kr, K. Sudha, S. A. M, and Steve Joshua S, "Software Development Automation Using Generative AI," in *2024 International Conference on Emerging Research in Computational Science (ICERCS)*, IEEE, Dec. 2024, pp. 1–6. doi: 10.1109/ICERCS63125.2024.10894980.
[43] M. Kanbur, O. P. C, and P. Kulkarni, "Creative AI in Software Project Management," in *2023 2nd International Conference on Futuristic Technologies (INCOFT)*, IEEE, Nov. 2023, pp. 1–9. doi: 10.1109/INCOFT60753.2023.10425234.
[44] C. Nitin Rajadhyaksha and J. R. Saini, "Robotic Process Automation for Software Project Management," in *2022 IEEE 7th International conference for Convergence in Technology (I2CT)*, IEEE, Apr. 2022, pp. 1–5. doi: 10.1109/I2CT54291.2022.9823972.
[45] A. Najdawi and A. Shaheen, "Which Project Management Methodology is better for AI-Transformation and Innovation Projects?," in *2021 International Conference on Innovative Practices in Technology and Management (ICIPTM)*, IEEE, Feb. 2021, pp. 205–210. doi: 10.1109/ICIPTM52218.2021.9388357.
[46] Z. Wang, M. Shi, and C. Li, "An Intelligent DevOps Platform Research and Design Based on Machine Learning," in *2020 Eighth International Conference on Advanced Cloud and Big Data (CBD)*, IEEE, Dec. 2020, pp. 42–47. doi: 10.1109/CBD51900.2020.00017.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 4, Issue 2, December 2024
Impact Factor: 7.57

[47] D. Quintanilla-Perez, A. Mauricio-Delgadillo, and D. Mauricio-Sanchez, "Essboard: a collaborative tool for using Essence in software development," in *2019 IEEE 10th International Conference on Software Engineering and Service Science (ICSESS)*, IEEE, Oct. 2019, pp. 20–23. doi: 10.1109/ICSESS47205.2019.9040832.