International Journal of Current Engineering and Technology
©2023 INPRESSCO®, All Rights Reserved

E-ISSN 2277 – 4106, P-ISSN 2347 – 5161
Available at http://inpressco.com/category/ijcet

Research Article

Reducing Defect Rates with AI-Powered Test Engineering and JIRA Automation in Agile Workflows

Abhishek Goyal*

Independent Researcher

Received 26 Nov 2023, Accepted 05 Dec 2023, Available online 15 Dec 2023, **Vol.13, No.6 (Nov/Dec 2023)**

Abstract

JIRA automation combined with Artificial Intelligence (AI) has revolutionised defect management and agile processes in software development. This paper aims to review the role that AI methods, including ML and NLP, play in improving the ability to detect defects, minimise the use of test cases, and automate testing. With such tools as customisable as well as agiler and reportable workflows and integrated as well as AI analytic representations, JIRA helps development teams anticipate defects during the working process, arrange priorities and productivity. These benefits are real and cases are showing that defect rates drastically diminished, defect fixes are done so much quicker, and decision-making is improved with this synergy. Possible future developments are the improvement of AI models, integration of new technologies into those models, the implementation of the solutions for widespread teams, and the problems of ethical implementations in AI-driven automation. The paper has presented the current and potential scenarios of AI augmentation to JIRA for improving software testing and defect tracking.

Keywords: *AI-powered testing, JIRA automation, defect management, agile workflows, machine learning (ML), natural language processing (NLP), test case optimisation, software development, predictive defect detection, workflow automation.*

Introduction

The agile method has a long history that began in the late nineteenth century, even before the term agile was coined. Agile refers to more rapid and effective motions. Agile may be applied in any unstable situation since it is defined by its ideals and principles [1]. It encourages teams to act on their own initiative and adapt as much as possible to the quickly changing market conditions [2]. There was a lot of cynicism and a lot of enthusiasm in the early years of agile development, as there is with any new discipline. A multitude of approaches, some more closely aligned with the manifesto's principles than others, emerged on the scene[3].

AI is a set of processes that computers may use to mimic human intellect. You can do more with AI than simply check email addresses or keep tabs on social media [4]. It automates tedious and labour-intensive software testing processes. AI testing essentially comprises automating, analysing, and optimising different testing procedures via the use of sophisticated lgorithms and ML techniques[5].The landscape provides a foundation for understanding and addressing theoretical and practical issues in AI-related software development and design [6]. Software testing is a crucial part of the software engineering life cycle. Without a bug-tracking tool, no software development project would be complete [7].

JIRA Automation

The JIRA software was developed by Atlassian. Additional insights were derived from the selected sample's remarks using a qualitative technique. It takes a lot of work to analyse each comment from different people. Thanks to advancements in technology, natural language processing methods like word2vec and topic modelling can now examine a corpus to extract latent themes and crucial key phrases. One way to find patterns of words in text is by using topic modelling[8]. Automation in Jira helps teams focus on important tasks by simplifying repetitive tasks and complex workflows. Users can easily create automation rules

*Corresponding author's ORCID ID: 0000-0000-0000-0000

DOI: https://doi.org/10.14741/ijcet/v.13.6.7

using a simple, no-code interface[9]. These rules can handle everything from basic tasks to more complex processes shown in Figure 1.

Because JIRA is growing in popularity among developers, it is important to know how this tool affects the organisation of developer communities. There is a defined structure and developers who are organised in the process of developing new open source software, according to several studies[10][11]. For instance, Linux is not the product of a disorganised development process carried out by disorganised developers [12].

Finding, studying, and reducing or eliminating flaws in a product, method, or system is what's known as defect reduction [13]. In common use, a defect is any variation from the intended or expected quality level that results in subpar performance from the product or system. Their causes include design errors, mistakes in production and assembly, problems with used software, or absence of sufficient processes. Defect reduction, therefore, involves the act of using a rational process to ensure that defects are identified and eliminated at all phases within the life cycle of a product or a system[14][15]. Defect identification, root cause analysis, corrective and preventative measures, and continuous improvement initiatives are all included in this [16]. The improvement of the rate of defects is aimed at improving the quality and dependability of a product or a system to increase customer satisfaction, cut expenses, and promote effective production. Defect reduction is a critical aspect of the BOK in a field of Industrial and Systems Engineering (IISE).

Organization of the paper

The paper is structured as follows: Section II covers AI-powered test engineering. Section III details JIRA automation in agile workflows. Section IV refers to synergising AI and JIRA automation. A literature review is included in Section V. Conclusions and future research are addressed in Section VI.

Ai-Powered Test Engineering

AI-powered test engineering, or AI testing, is a software testing process that uses artificial intelligence (AI) to improve the testing lifecycle. Methods based on AI for testing produce exhaustive test cases automatically, adapt to changes in real-time and apply machine learning to search for weaknesses in the code base [17].This method enhances total test coverage while at the same time creating efficient test regression. Regardless of their operational setting, a variety of sectors are already adopting the agile strategy[18].

AI Techniques in Defect Detection

AI-powered test engineering uses advanced AI techniques, including ML and NLP, to optimise and automate software testing processes[19].

Machine Learning for Predictive Defect Identification

Another functional usage of ML computations is the studying of historical defect data and the usage of the results in order to predict the risky areas or parts within the software systems. This approach goes a long way in enhancing the detection of defects and minimising some of the glitches to reach the production level [20]. The aggressive method of proactive defect prediction in software development entails estimating potential software problems prior to their occurrence [21].

NLP for Test Case Optimization

It is used for transforming user stories or requirements into more efficient test cases, making it easier for the technical and non-technical staff onsite [22].NLP enable tools such as Functionate to help in generation and administration of a number of test cases reducing the intensive manual work as well as improving coverage. To enhance the testing process, it allows AI to use NLP and translate the requirements into the language it comprehends [23]. The ability to receive feature or user story descriptions from stakeholders who aren't technically savvy is a great asset when it comes to testing. The AI takes these descriptions and turns them into practical test cases, helping both the technical and non-technical team members work together[24].

Benefits of AI in Test Engineering

There are some important benefits in AI-powered test engineering.

Improved Defect Detection Accuracy

Unlike manual testing, AI reduces the number of human errors through automation while also providing extensive testing for latent test cases.

Faster Test Execution and Defect Resolution

With AI automating repetitive tasks, testing cycles are significantly accelerated. Reduced test maintenance efforts and increased overall efficiency are the results of self-healing tests' dynamic adaptation to application changes [25].

Enhanced Test Coverage

AI-driven tools generate comprehensive test scenarios, including complex edge cases, ensuring robust validation of all aspects of the software[26].

Jira Automation in Agile Workflows

An Australian business named "Atlassian" developed JIRA, a program for managing projects. Godzilla is called "Gojira" in Japanese, which is the origin of the name Jira. The program follows the principles of agile development. You may tailor processes in JIRA to fit your specific requirements [27]. Your flow may be shaped and adjusted to your liking, allowing you to trace problems from start to finish according to your procedures. Multiple people have found success using Jira's multiple uses, including agile project management, monitoring service desk tickets, and even writing. To keep track of features that will be added to the product in future releases, developers utilise Agile or Scrum methods like JIRA. Like other Jira approaches, Agile uses roadmaps to keep track of issues [28].

Different Features of JIRA

There are discussed the main features of JIRA Automation in Agile workflows:

Scrum and Kanban Boards

These are visual boards for Agile framework-based task management.While Scrum boards organise work in sprints, Kanban boards deal with the continuous management of workflow to track and prioritise work in real time[29].

Roadmaps

Jira roadmaps provide a big picture of your project timeline and milestones, thus helping teams plan, visualise, and broadcast their projects' long-term strategy or vision and Progress[30].

Bug and issue tracking

Jira are utilised for tracing bugs and issues.Groups can log, track, and manage bugs or issues throughout their life cycle to ensure problems are solved efficiently and do not affect project progress[31].

Agile Reporting

Jira agile reporting provides reports on burndown and velocity charts, which are instrumental in letting the teams know the performance, Progress, and bottlenecks ahead in Agile projects.

Custom Workflows

Custom workflows define several stages and transitions that tasks may undergo[32]. They tailor the workflow to a team's unique processes and ensure that work progresses in an organised and efficient way.

Integrate with over 3000 apps

Jira can integrate with over 3000 applications This feature allows teams to connect with their other tools and platforms, such as Slack, GitHub, and Confluence, to drive productivity and smoothen workflows.

Audit Logs

Jira audit logs track and record changes made within the system. This records all the actions taken for security[33], compliance, and accountability purposes.

Workflows of JIRA

This section explains the main workflows in JIRA.

System Workflows

These are predefined workflows that come with Jira out of the box [34].They are designed to cover common use cases and are ready to use without much customisation [35].System workflows are often generic. They may be suitable for a variety of projects, making them a quick and easy option for teams that don’t require highly specialised workflows.

Shared Workflows

Shared workflows can be shared across multiple projects within a Jira instance. They are highly customisable.Teams or administrators can create and modify them to meet the specific needs of their projects Shared workflows help maintain consistency across different teams or departments.

Project-specific Workflows

These workflows are specific to a particular project. They are created and customised to address their unique requirements. The specific project flow also makes for quite a lot of flexibility when it comes to project work They allow the teams to set specific four categories, which include stages, transition, and conditions.

Synergizing AI and Jira Automation

The promise of using Artificial Intelligence or AI along with JIRA Automation has made new possibilities for handling defect management in the software development life cycle.It discusses how AI improves the features of JIRA, based on various cases of integration and actual usage [36][37].

Integration Scenarios

Therefore, with the integration of AI with JIRA, a strong foundation is developed for improving the enhance of defect management. These are two main cases showing how AI can be applied in connection with JIRA to optimise the existing procedures and enhance the software development results [38].

Using AI Plugins with JIRA for Defect Prediction

Using AI plugins in JIRA, the development teams can be able to have early signs and forecasts of the planned defects by working with the past project data [39].These tools apply machine learning pattern matching for past issues, for example previous bugs, typical types of errors or congestion in the development process. For instance:

Predictive Analysis: AI can even anticipate the parts of the codebase or some features which are most likely to contain bugs, so that operations, resources, and testing can be appropriately directed [40].

Risk Alerts: AI tools can give early warning for modules that have high chances of having defects through recent commits or complex changes thus reducing the possibility of production failures [41].

Automating Defect Workflows with AI-Driven Analytics

AI-driven analytics revolutionise how defects are tracked, prioritised, and resolved within JIRA. Instead of manually categorising and assigning issues[42],AI-powered automation simplifies the workflow:

Defect Prioritization: AI categorises defects by means of priority depending on issue severity of the application, the reaction of customers, or frequency.

Task Assignment: AI can dynamically assign issues to team members based on workload, expertise, and past resolution times, ensuring an optimised distribution of tasks[29].

Workflow Optimization: Automated insights help identify bottlenecks in defect management, recommending adjustments to improve efficiency.

Case Studies and Real-world Applications

The AI integration with the JIRA automation has again been observed to perform efficiently across scenarios and has gathered quantifiable results in reduction of defects and increase in productivity and overall project delivery [43]. This section provides a discussion of real-world examples of successful implementations and the effects on the software development lifecycles [44].

Examples of Successful Implementations

Several organisations have successfully adopted AI-driven solutions within their JIRA workflows, leveraging the technology to address specific challenges:

E-Commerce Platform: A leading e-commerce company integrated machine learning-based defect prediction tools with JIRA[45].By analysing historical bug data, the AI model predicted high-risk areas in upcoming releases. As a result, the company reduced the occurrence of critical production defects by 40% and improved customer satisfaction through smoother product rollouts.

Technology Firm: A global technology provider implemented natural language processing (NLP) to automatically triage and categorise defects reported by customers[46].The AI system was able to process incoming bug reports, classify them based on priority and severity, and assign them to the appropriate teams. This automation reduced the manual triaging time by 50% and significantly improved response times.

Healthcare Software Company: In a highly regulated industry, a healthcare software company adopted AI-powered analytics to monitor defect resolution timelines[47] By identifying bottlenecks and optimising workflows, the company improved defect closure rates by 25%, ensuring compliance with stringent quality standards.

Measurable Impact on Defect Rates and Productivity

The tangible benefits of integrating AI with JIRA automation are reflected in the following metrics:

Reduced Defect Rates: AI-driven defect prediction and workflow automation allow teams to detect and address potential issues early in the development cycle[48]. Companies have reported up to a 30–40% reduction in post-release defects due to proactive measures.

Improved Productivity: Automation of repetitive tasks, such as categorising and assigning defects, saves valuable time for developers and QA teams[49][50].On average, organisations have observed a 20–30% improvement in team productivity, enabling faster delivery of high-quality software.

Enhanced Decision-Making: AI-powered insights provide teams with actionable data, such as defect trends, recurring issues, and resource utilisation [51]. Project managers may now make decisions based on data, which improves resource allocation and reduces delays.

Literature Review

In this study, Goyal (2021) discuss about the previous year research on this Reducing Defect Rates with AI-Powered Test Engineering and JIRA Automation in Agile Workflows[52].

In this study, Wilkerson, Nunamaker and Mercer (2012) contrasting the two approaches to software defect reduction—code inspection and test-driven development—and calculating their respective implementation costs For each group, they compared the total number of defects and the cost of implementation. Although code inspection is more costly, it is more successful than test-driven development in reducing faults. Additionally, they discovered that conventional programming techniques were just as good at decreasing defects as test-driven development[53].

In this study, Sarkan Ahmad and Bakar (2011) examine the effects on the requirement process of the agile methodology's application to MIMOS. This article will go over the pros and cons of MIMOS's requirement creation process utilising JIRA and Redmine. Today, Agile development is all the rage because of how quickly and cheaply it can provide results. There will likely be eighty percent of software development projects using the Agile methodology by 2012. The use of agile approach has been implemented in several MIMOS software projects since 2010[54].

In this study, Carneiro, Silva and Alencar, (2018) sought to demonstrate how Scrum may be used for ordinary management by making the required adjustments and providing performance metrics to measure the tool's efficacy. Hence, AM PM differs from conventional PM. The second part of the article shows an example of an agile manufacturing workflow. Consequently, there will be more harmony within the team, enhanced normal workflow, higher quality delivery, better monitoring of operations, and better planning of activities[55].

In this study, Calefato et al (2020) The first step was to establish clear guidelines for the use of the technologies and to emphasise the importance of Slack as the primary platform for internal communication. Our second major update was splitting the Jira Scrum board in half, making one more specific to developers' needs and the other more generalised for managers. They also included Slack connection and automation rules. The results of the second modification reveal that 85 per cent of those who were interviewed felt that their workflow had been enhanced. The significance of businesses considering how to best establish an agile work environment to promote teamwork and open dialogue[56].

In this study, Lunesu et al (2021) intends to introduce and explore a novel method for modelling critical risk variables in agile development through the use of SPSM. They examined three open-source projects, pulling their data from JIRA repositories, to test the simulator and show how the strategy may be utilised. Modelling the agile process and collecting data from the project management tool is part of the methodology. By utilising approaches such as short iterations, feature-driven development, continuous integration, and testing automation, agile methodologies strive to lower the risk associated with software development[57].

In this study, Laanti and Kangas (2015) delves into the discrepancy between the benefits touted by training materials and publications on Agile Portfolio techniques and those claimed by actual Portfolio Kanban Board users. In addition, examine whether the Principles of Large-Scale Agile, compiled from different Agile organisations and shared in the XP2014 keynote speech, can be observed in action by Portfolio Kanban Board users. They conclude by proposing the elimination of the Principle of Pattern Utilisation[58].
Table I provides an overview of studies on Agile methodologies, outlining their objectives, challenges, advantages, and potential areas for future research. These studies highlight benefits such as improved defect reduction, faster delivery, and enhanced collaboration, while also identifying challenges like high costs, tool adaptation complexities, and limitations in large-scale applications. A significant gap remains in understanding the scalability of Agile practices across various organisational contexts and in exploring hybrid approaches to leverage the strengths of different methodologies. Future research could focus on refining tools and frameworks to enhance adaptability and effectiveness in diverse scenarios.

Table 1 Literature review on Agile Methodologies: Objectives, Challenges, Advantages, and Future Research Directions

Study	Objective	Challenges	Advantages	Future Study
Wilkerson, Nunamaker, and Mercer (2012)	Compare software defect rates and costs between code inspection and test-driven development.	Code inspection is more expensive than test-driven development.	Code inspection effectively reduces defects.	Explore hybrid models combining the benefits of code inspection and test-driven development.
Sarkan, Ahmad, and Bakar (2011)	Analyse the influence of Agile methodology on requirement processes at MIMOS.	Managing tools like JIRA and Redmine for requirement development.	Agile supports faster delivery and cost reduction.	Investigate long-term scalability of Agile in large-scale projects at MIMOS.
Carneiro, Silva, and Alencar (2018)	Investigate Scrum's application in routine management and evaluate performance improvements.	Adapting Scrum to routine management requirements.	Improved planning, task prioritisation, team alignment, and delivery quality.	Evaluate the scalability of Scrum adaptations in larger organisations.
Calefato et al. (2020)	Optimize communication and collaboration in Agile environments.	Refactoring tools and ensuring adherence to clear communication rules.	Enhanced workflow efficiency through Slack and automated Jira boards.	Examine the impact of tool integration on distributed Agile teams' productivity.
Lunesu et al. (2021)	Model risk factors in Agile development using Software Process Simulation Modeling (SPSM).	Gathering accurate data for modelling from JIRA repositories.	Risk reduction through Agile practices like iterations, continuous integration, and automation.	Refine simulation models for real-time risk assessment in Agile projects.
Laanti and Kangas (2015)	Evaluate benefits of Agile Portfolio practices using Portfolio Kanban Boards.	Some principles of large-scale Agile practices were not observed in real use cases.	Better visualisation and management of large-scale Agile processes.	Refine and validate Agile Portfolio principles for diverse organisational setups.

Conclusion and Future Scope

JIRA automation, together with the application of AI, has brought a significant change in the handling of defects within software development processes. With the help of AI-based approaches, including ML and NLP, it is possible to foresee the defects, choose the proper test cases, and improve the testing sphere. Flexible workflows of JIRA Augmented by AI plugins, offer excellent means for defect prevention, classification, and remediation for development teams. These developments do not only decrease the defect rate but also increase the productivity and decision-making performance. From the case studies elucidated in this paper, it is evident that the integration of JIRA together with AI has tangible value in practice including decrease in production defects, enhancement of defect closure ration and efficient triaging of reported customer issues. This integration saves time by automating simple tasks while providing valuable data that enables teams to prioritise key tasks and make an impact on software quality, customer satisfaction and project timelines.

The future research directions include improving the AI approaches for more precise and specialised defect prediction, exploring the application of generative AI and the integration of blockchain for reinforcing security and performance, and extending the NLP functionality for the more comprehensive feedback analysis and test case generation. Furthermore, there is an opportunity to create AI-based, real-time defect tracking that can identify defects during the coding phase and ideas to design large, dispersed teams and multi-vendor projects openly. Issues that pertain to the bias of the AI predictions and the reasonable disclosure of the decision-making processes will also be essential to initiate as long as the software development process is to be handled by AI.

References

[1] B. Patel, V. K. Yarlagadda, N. Dhameliya, K. Mullangi, and S. C. R. Vennapusa, "Advancements in 5G Technology: Enhancing Connectivity and Performance in Communication Engineering," *Eng. Int.*, vol. 10, no. 2, pp. 117–130, 2022, doi: 10.18034/ei.v10i2.715.

[2] S. Sharma, D. Sarkar, and D. Gupta, "Agile Processes and Methodologies: A Conceptual Study," *Int. J. Comput. Sci. Eng.*, vol. 4, 2012.

[3] T. Dingsøyr, S. Nerur, V. Balijepally, and N. B. Moe, "A decade of agile methodologies: Towards explaining agile software development," 2012. doi: 10.1016/j.jss.2012.02.033.

[4] J. Thomas and V. Vedi, "Enhancing Supply Chain Resilience Through Cloud-Based SCM and Advanced Machine Learning: A Case Study of Logistics," *J. Emerg. Technol. Innov. Res.*, vol. 8, no. 9, 2021.

[5] S. Bauskar, "Business analytics in enterprise system based on application of artificial intelligence," *Int. Res. J. Mod. Eng. Technol. Sci.*, vol. 04, no. 01, pp. 1861–1870, 2022, doi: DOI: https://www.doi.org/10.56726/IRJMETS18127.

[6] V. V Kumar, "An interactive product development model in remanufacturing environment: a chaos-based artificial bee colony approach," *Eng. Int.*, vol. 6, no. 2, pp. 211–222, 2014.

[7] R. Goyal, "The Role Of Business Analysts In Information Management Projects," *Int. J. Core Eng. Manag.*, vol. 6, no. 9, pp. 76–86, 2020.

[8] K. Gorro, K. Gorro, A. Ilano, A. Sebial, E. Ranolo, and E. Vale, "Qualitative Technology Acceptance Evaluation of JIRA in Software Development Using Machine Learning," *Int. J. Adv. Eng.*, 2019.

[9] V. V. Kumar, A. Sahoo, and F. W. Liou, "Cyber-enabled product lifecycle management: A multi-agent framework," in *Procedia Manufacturing*, 2019. doi: 10.1016/j.promfg.2020.01.247.

[10] S. Bauskar, "An Analysis: Early Diagnosis and Classification of Parkinson's Disease Using Machine Learning Techniques," *Int. J. Comput. Eng. Technol.*, vol. 12, no. 1, pp. 54–66, 2021, doi: 10.5281/zenodo.13836264.

[11] M. T. Arora, Rajeev and Kumar, Shantanu and Jain, Nitin and Nafis, "Revolutionizing Healthcare with Cloud Computing: Superior Patient Care and Enhanced Service Efficiency," *SSRN*, 2022, doi: http://dx.doi.org/10.2139/ssrn.4957197.

[12] M. Ortu, G. Destefanis, M. Kassab, and M. Marchesi, "Measuring and understanding the effectiveness of JIRA developers communities," in *International Workshop on Emerging Trends in Software Metrics, WETSoM*, 2015. doi: 10.1109/WETSoM.2015.10.

[13] S. G. Jubin Thomas, Kirti Vinod Vedi, "The Effect and Challenges of the Internet of Things (IoT) on the Management of Supply Chains," *Int. J. Res. Anal. Rev.*, vol. 8, no. 3, pp. 874–879, 2021.

[14] V. V. Kumar, F. W. Liou, S. N. Balakrishnan, and V. Kumar, "Economical impact of RFID implementation in remanufacturing: a Chaos-based Interactive Artificial Bee Colony approach," *J. Intell. Manuf.*, 2015, doi: 10.1007/s10845-013-0836-9.

[15] B. Boddu, "DevOps for Database Administration: Best Practices and Case Studies," *https://jsaer.com/download/vol-7-iss-3-2020/JSAER2020-7-3-337-342.pdf*, vol. 7, no. 3, p. 5, 2020.

[16] S. Bauskar, "Predictive Analytics For Sales Forecasting In Enterprise Resource," *Int. Res. J. Mod. Eng. Technol. Sci.*, vol. 04, no. 06, pp. 4607–4618, 2022, doi: https://www.doi.org/10.56726/IRJMETS26271.

[17] D. S. G. Jubin Thomas, Hemanth Volikatla, Veera Venkata Raghunath Indugu, Kushwanth Gondi, "Machine Learning Approaches for Fraud Detection in E-commerce Supply Chains," *Innov. Comput. Sci. J.*, vol. 8, no. 1, 2022.

[18] H. Salehi, S. Biswas, and R. Burgueño, "Data interpretation framework integrating machine learning and pattern recognition for self-powered data-driven damage identification with harvested energy variations," *Eng. Appl. Artif. Intell.*, 2019, doi: 10.1016/j.engappai.2019.08.004.

[19] R. Bibi *et al.*, "Edge AI-Based Automated Detection and Classification of Road Anomalies in VANET Using Deep Learning," *Comput. Intell. Neurosci.*, 2021, doi: 10.1155/2021/6262194.

[20] S. S. Pranav Khare, "The Impact of AI on Product Management : A Systematic Review and Future Trends," *Int. J. Res. Anal. Rev.*, vol. 9, no. 4, pp. 736–741, 2022.

[21] R. Malhotra and J. Jain, "Predicting defects in imbalanced data using resampling methods: An empirical investigation," *PeerJ Comput. Sci.*, 2022, doi: 10.7717/peerj-cs.573.

[22] S. A. and A. Tewari, "AI-Driven Resilience: Enhancing Critical Infrastructure with Edge Computing," *Int. J. Curr. Eng. Technol.*, vol. 12, no. 02, pp. 151–157, 2022, doi: https://doi.org/10.14741/ijcet/v.12.2.9.

[23] R. Arora, S. Gera, and M. Saxena, "Mitigating Security Risks on Privacy of Sensitive Data used in Cloud-based ERP Applications," in *2021 8th International Conference on Computing for Sustainable Global Development (INDIACom)*, 2021, pp. 458–463.

[24] M. S. C. Chan, E. A. del Rio-Chanona, F. Fiorelli, H. Arellano-Garcia, and V. S. Vassiliadis, "Construction of global optimisation constrained NLP test cases from unconstrained problems," *Chem. Eng. Res. Des.*, 2016, doi: 10.1016/j.cherd.2016.03.015.

[25] M. S. Rajeev Arora, "Applications of Cloud Based ERP Application and how to address Security and Data Privacy Issues in Cloud application," *Himal. Univ.*, 2022.

[26] I. K. Nti, A. F. Adekoya, B. A. Weyori, and O. Nyarko-Boateng, "Applications of artificial intelligence in engineering and manufacturing: a systematic review," 2022. doi: 10.1007/s10845-021-01771-6.

[27] Mani Gopalsamy, "An Optimal Artificial Intelligence (AI) technique for cybersecurity threat detection in IoT Networks," *Int. J. Sci. Res. Arch.*, vol. 7, no. 2, pp. 661–671, Dec. 2022, doi: 10.30574/ijsra.2022.7.2.0235.

[28] A. Nogués and J. Valladares, "Agile Methodologies for BI Projects," in *Business Intelligence Tools for Small Companies*, 2017. doi: 10.1007/978-1-4842-2568-4_2.

[29] M. Z. Hasan, R. Fink, M. R. Suyambu, and M. K. Baskaran, "Assessment and improvement of intelligent controllers for elevator energy efficiency," in *IEEE International Conference on Electro Information Technology*, 2012. doi: 10.1109/EIT.2012.6220727.

[30] M. Gopalsamy, "Advanced Cybersecurity in Cloud Via Employing AI Techniques for Effective Intrusion Detection," *Int. J. Res. Anal. Rev.*, vol. 8, no. 1, pp. 187–193, 2021.

[31] R. A. Anoop Kumar, Ramakrishna Garine, Arpita Soni, "Leveraging AI for E-Commerce Personalization : Insights and Challenges from 2020," pp. 1–6, 2020.

[32] R. Bishukarma, "The Role of AI in Automated Testing and Monitoring in SaaS Environments," *Int. J. Res. Anal. Rev.*, vol. 8, no. 2, pp. 846–852, 2021, [Online]. Available: https://www.ijrar.org/papers/IJRAR21B2597.pdf

[33] S. A. and A. Tewari, "Security Vulnerabilities in Edge Computing: A Comprehensive Review," *Int. J. Res. Anal. Rev.*, vol. 9, no. 4, pp. 936–941, 2022.

[34] V. K. Y. Nicholas Richardson, Rajani Pydipalli, Sai Sirisha Maddula, Sunil Kumar Reddy Anumandla, "Role-Based Access Control in SAS Programming: Enhancing Security and Authorization," *Int. J. Reciprocal Symmetry Theor. Phys.*, vol. 6, no. 1, pp. 31–42, 2019.

[35] V. N. Boddapati *et al.*, "Data migration in the cloud database: A review of vendor solutions and challenges," *Int. J. Comput. Artif. Intell.*, vol. 3, no. 2, pp. 96–101, Jul. 2022, doi: 10.33545/27076571.2022.v3.i2a.110.

[36] V. K. Y. Mohamed Ali Shajahan, Nicholas Richardson, Niravkumar Dhameliya, Bhavik Patel, Sunil Kumar Reddy Anumandla, "AUTOSAR Classic vs. AUTOSAR Adaptive: A Comparative Analysis in Stack Development," *Eng. Int.*, vol. 7, no. 2, pp. 161–178, 2019.

[37] A. Moharil, D. Orlov, S. Jameel, T. Trouwen, N. Cassee, and A. Serebrenik, "Between JIRA and GitHub: ASFBot and its Influence on Human Comments in Issue Trackers," in *Proceedings - 2022 Mining Software Repositories Conference, MSR 2022*, 2022. doi: 10.1145/3524842.3528528.

[38] D. Ritter, N. May, and S. Rinderle-Ma, "Patterns for emerging application integration scenarios: A survey," 2017. doi: 10.1016/j.is.2017.03.003.

[39] A. P. A. Singh, "STRATEGIC APPROACHES TO MATERIALS DATA COLLECTION AND INVENTORY MANAGEMENT," *Int. J. Bus. Quant. Econ. Appl. Manag. Res.*, vol. 7, no. 5, 2022.

[40] M. S. Arora, Rajeev, Sheetal Gera, "Impact of Cloud Computing Services and Application in Healthcare Sector and to provide improved quality patient care," no. October 2021, 2021.

[41] B. Boddu, "Serverless Databases Are the Future of Database Management," *https://jsaer.com/download/vol-6-iss-1-2019/JSAER2019-6-1-277-282.pdf*, vol. 6, no. 1, p. 5, 2020.

[42] K. V. V. and S. G. Jubin Thomas , Piyush Patidar, "An analysis of predictive maintenance strategies in supply chain management," *Int. J. Sci. Res. Arch.*, vol. 06, no. 01, pp. 308–317, 2022, doi: DOI: https://doi.org/10.30574/ijsra.2022.6.1.0144.

[43] R. Bishukarma, "Adaptive AI-Based Anomaly Detection Framework for SaaS Platform Security," *Int. J. Curr. Eng. Technol.*, vol. 12, no. 07, pp. 541–548, 2022, doi: https://doi.org/10.14741/ijcet/v.12.6.8.

[44] P. Mittal and P. Mehta, "Optimization of Software Development Process by Plugin Integration with Jira – A Project Management Tool in Devops," *SSRN Electron. J.*, 2020, doi: 10.2139/ssrn.3564436.

[45] S. K. R. Anumandla, V. K. Yarlagadda, S. C. R. Vennapusa, and K. R. V Kothapalli, "Unveiling the Influence of Artificial Intelligence on Resource Management and Sustainable Development: A Comprehensive Investigation," *Technol. \& Manag. Rev.*, vol. 5, no. 1, pp. 45–65, 2020.

[46] A. Goyal, "Scaling Agile Practices with Quantum Computing for Multi-Vendor Engineering Solutions in Global Markets," *Int. J. Curr. Eng. Technol.*, vol. 12, no. 6, pp. 557–564, 2022.

[47] V. K. Yarlagadda, "Harnessing Biomedical Signals: A Modern Fusion of Hadoop Infrastructure, AI, and Fuzzy Logic in Healthcare," *Malaysian J. Med. Biol. Res.*, vol. 2, no. 2, pp. 85–92, 2021.

[48] Prity Choudhary and Vikas Jalan, "Enhancing Process Comprehension through Simulation-Based Learning," *Int. J. Adv. Res. Sci. Commun. Technol.*, pp. 919–924, Dec. 2022, doi: 10.48175/IJARSCT-14400R.

[49] M. R. Kishore Mullangi, Vamsi Krishna Yarlagadda, Niravkumar Dhameliya, "Integrating AI and Reciprocal Symmetry in Financial Management: A Pathway to Enhanced Decision-Making," *Int. J. Reciprocal Symmetry Theor. Phys.*, vol. 5, no. 1, pp. 42–52, 2018.

[50] M. Z. Hasan, R. Fink, M. R. Suyambu, M. K. Baskaran, D. James, and J. Gamboa, "Performance evaluation of energy efficient intelligent elevator controllers," in *IEEE International Conference on Electro Information Technology*, 2015. doi: 10.1109/EIT.2015.7293320.

[51] K. Patel, "Quality Assurance In The Age Of Data Analytics: Innovations And Challenges," *Int. J. Creat. Res. Thoughts*, vol. 9, no. 12, pp. f573–f578, 2021.

[52] R. Goyal, "The Role Of Requirement Gathering In Agile Software Development: Strategies For Success And Challenges," *Int. J. Core Eng. Manag.*, vol. 6, no. 12, pp. 142–152, 2021.

[53] J. W. Wilkerson, J. F. Nunamaker, and R. Mercer, "Comparing the defect reduction benefits of code inspection and test-driven development," *IEEE Trans. Softw. Eng.*, 2012, doi: 10.1109/TSE.2011.46.

[54] H. M. Sarkan, T. P. S. Ahmad, and A. A. Bakar, "Using JIRA and redmine in requirement development for Agile methodology," in *2011 5th Malaysian Conference in Software Engineering, MySEC 2011*, 2011. doi: 10.1109/MySEC.2011.6140707.

[55] L. B. Carneiro, A. C. C. L. M. Silva, and L. H. Alencar, "Scrum Agile Project Management Methodology Application for Workflow Management: A Case Study," in *IEEE International Conference on Industrial Engineering and Engineering Management*, 2018. doi: 10.1109/IEEM.2018.8607356.

[56] F. Calefato, A. Giove, F. Lanubile, and M. Losavio, "A case study on tool support for collaboration in agile development," in *Proceedings - 2020 ACM/IEEE 15th International Conference on Global Software Engineering, ICGSE 2020*, 2020. doi: 10.1145/3372787.3390436.

[57] M. I. Lunesu, R. Tonelli, L. Marchesi, and M. Marchesi, "Assessing the risk of software development in agile methodologies using simulation," *IEEE Access*, 2021, doi: 10.1109/ACCESS.2021.3115941.

[58] M. Laanti and M. Kangas, "Is Agile Portfolio Management Following the Principles of Large-Scale Agile? Case Study in Finnish Broadcasting Company Yle," in *Proceedings - 2015 Agile Conference, Agile 2015*, 2015. doi: 10.1109/Agile.2015.9.