© JUL 2024 | IRE Journals | Volume 8 Issue 1 | ISSN: 2456-8880

Intelligent Software Agents for Continuous Delivery: Leveraging AI and Machine Learning for Fully Automated DevOps Pipelines

GIREESH KAMBALA
Teach for America

Abstract- The contribution of intelligent software agents, artificial intelligence (AI), and machine learning (ML) in allowing totally automated DevOps pipelines for continuous delivery is investigated in this review paper. Organisations may automate important software development lifecycle activities—including code integration, testing, deployment, and monitoring—by combining artificial intelligence and machine learning technology. The study looks at the advantages of automation—that is, more efficiency, less time-to-market, better software quality, and proactive issue prediction and resolution capacity. Emphasising key technologies like predictive analytics, intelligent monitoring, and autonomous rollback systems, we show how they maximise DevOps practices. Furthermore covered in the paper are the difficulties integrating several technologies, controlling data quality, and guaranteeing scalability in automated pipelines. It offers chances to enhance cooperation among teams for operations, quality assurance, and development. In the end, our work emphasises how intelligent software agents may transform DevOps, boost output, and inspire software delivery innovation.

Indexed Terms- Drug abuse, Relationships, Deterioration, Familial bonds, Mechanisms

I. INTRODUCTION

Intelligent Software Agents fundamentally alter the way modern DevOps systems have been automated, regulated and optimised. Using Artificial Intelligence (AI) & Machine Learning (ML), these agents completely automate the software delivery lifecycle, so insuring that updates, features, and fixes are frequently, consistently, and with least human interaction given. Using such intelligent agents in automated Continuous Integration and Continuous Development (CI/CD) pipelines addresses a broad spectrum of issues including pipeline complexity, scheduling of resources, error detection, and system reliability, so promoting an autonomous ecosystem that fits the goals of modern DevOps approaches. Using artificial intelligence or machine learning, intelligent software agents can process massive volumes of data generated during the creation of software, learn patterns, forecast outcomes, and make real-time decisions to improve general efficiency and reduce the time-to--market for new releases. The main ability of smart software agents to apply advanced ML algorithms to detect prospective pipeline obstacles and inefficiencies defines their functioning. For instance, they can go over historical deployment records, find reoccurring issues including long deployment durations or build failures, and then aggressively suggest or implement repairs for these issues. By enabling teams to address issues before they become more critical, anomaly detection systems checking the pipeline for odd behaviour or breaches from expected norms enhance this predictive capacity even more. These bots let natural language processing (NLP) also scan logs, alarms, or error messages, so transforming challenging technical data into insightful analysis. This helps operations teams and developers to light the cognitive strain on them so free them to focus on more important activities rather than standard troubleshooting. [1], [2].

Figure 1 Workflow DevOps [3]

One of the main advantages of complex software agents is their capacity to enable optimal use of resources and technologies. By means of predictive analytics, these agents will be able to foresee resource needs based on workload patterns, so assuring that enough resources are delivered during peak demand and so save expenses during low demand. They also adaptably scale infrastructure to match varying traffic load, therefore ensuring exceptional performance and availability. Reinforcement learning boosts this capacity even more and always assists agents to maximise their strategies for optimal efficiency by allowing agents learn from past decisions concerning resource allocation. Modern cloud-based systems, in which administration of dispersed systems is built on flexibility and capacity expansion, depend mostly on this adaptation. For smart software agents, support of periodic testing and quality assurance is fairly crucial. Sometimes conventional testing approaches find it difficult to handle issues such flaky tests, insufficient test coverage, and important test case prioritisation. By looking at test data, intelligent agents solve issues using ML models and trend detection pointing up erroneous tests or sites with inadequate coverage. They can dynamically rank test cases to execute first the most essential ones depending on variables including code changes, effect on end-user functionality, and historical defect data. These agents can also generate synthetic test data, replicate numerous user scenarios, and run thorough performance tests to assess software working under many conditions therefore providing high quality assurance all through the delivery process. Security is another field in which intelligent software agents significantly impact. They can quickly find flaws and possible security gaps using modern anomaly detection techniques and threat modelling. By means of continuous monitoring application behaviour, network traffic, and access patterns, these agents enable to identify and minimise security vulnerabilities before they become targets of attack. They may draw attention to unusual login attempts, unauthorised access to private data, or inadvertent configurable changes implying a security concern. Moreover, they could automatically enforce security rules including industry standards and best practices, therefore guaranteeing that apps are safe by themselves. Another area where clever software agents excel is configuration management. A common issue in complicated environments comprising numerous staging, testing, and production configurations that could produce unequal behaviour and deployment mistakes is configuration drift. [4]. Intelligent agents continuously monitor settings across environments, discover deviations, and automatically resolve problems to ensure consistency. In case of issues, they can also monitor changes, use version control for setups, and return to a known good state, therefore providing a continual safety net for controlling challenging systems. Intelligent software agents included in CI/CD pipelines help to improve cooperation among development, operations, and assurance teams. By providing regular platform for debate and decision-making, these agents help teams to operate more effectively, therefore breaking traditional silos and fostering a culture of shared responsibility. Real-time dashboards or analytics provided by intelligent agents, together with great knowledge of pipeline performance measures, help stakeholders to track development, pinpoint areas needing work, and select data-driven decisions to maximise operations. This degree of openness and unity guarantees that every team is focused on shared goals, therefore improving general performance and productivity. Moreover, in DevOps pipelines clever software agents support scalability and agility. Companies in the fast-paced digital marketplace of today have to be able to react swiftly to fit evolving customer expectations, market conditions, and needs. Intelligent agents serve to facilitate this agility by automating repetitive tasks, reducing deployment cycles, and thereby lessening dependency on hand-crafted processes.[5]. Their quick adaptation to new tools, technology, and processes as the business grows helps to maintain the flow current and effective. Companies with advanced IT systems might consider them ideal since they can control systems of distribution of size. Adoption of modern software agents also reduces the hazards related to human mistake, unpredictable conduct, and delays that arise from hand involvement. These agents free workers to focus on innovation or strategic goals via use of robotic repetitive tasks & decision-making methods, hence boosting corporate value and competitive advantage. Since intelligent agents can enforce regulations, provide audit traces, and ensure regulatory compliance, this automation also helps companies to obtain better degrees of compliance and governance. Intelligent agents for software for Continuous Distribution fundamentally change the development, testing, and usage of software. Using the power of machines learning and artificial intelligence to bring to DevOps pipelines formerly unheard-of degrees of digitisation, efficiency, and dependability, these agents solve problems related to complexity, scalability, and security. Their ability to analyse data, project outcomes, and make rapid choices guarantees that businesses might produce high-quality software at speed, so meeting the demands of a market developing in competitiveness and dynamic nature. As the use of these technologies keeps growing, intelligent software agents are predicted to naturally fit the DevOps landscape since they inspire innovation and enable businesses to attain their goals with more confidence and agility [6].

II. LITERATURE REVIEW

Eniola 2025 Fast and consistent releases are absolutely necessary in the dynamic software development scene of today. Key techniques for simplifying development pipelines, improving quality, and increasing time-to----market have become CI (Continuous Integration and Continuous Delivery (CD). Jenkins, an open-source automation technology, is underlined in Eniola 2025's emphasis in enabling flawless CI/CD implementations. Jenkins streamlines workflow automation with tools including its strong connection with revision control systems, pipeline-as-code capability, and large plugin ecosystem. This article also explores best practices, deployment strategies, and rollback mechanisms to maximize CI/CD efficiency. Additionally, real-world examples demonstrate Jenkins' ability to improve delivery processes, reduce manual effort, and foster collaboration among teams. Common CI/CD challenges are addressed, and strategies to overcome them are presented, highlighting the importance of cultivating a culture of continuous improvement for sustainable DevOps success [7].

Tamanampudi 2024 investigates the application of AI and ML in building intelligent feedback loops within DevOps pipelines. The study showcases how big data and real-time monitoring enhance code quality and performance prediction by analyzing logs, monitoring tools, and user interactions. Feedback mechanisms are developed using ML for data acquisition, cleaning, model training, and online prediction. Challenges such as model interpretability, integration, and change management are discussed alongside strategies to resolve them. Case studies illustrate how AI improves code quality and accelerates problem-solving while strengthening development-operations relationships. Future research focuses on collaborative learning, model interpretability, and AI adoption reference models in DevOps pipelines, showcasing the transformative potential of ML-based feedback systems [8].

Tiedekunta 2024 explores the impact of Large Language Model (LLM)-based chatbots in software release processes. The study identifies key features for successful chatbot implementation, including task automation, repetitive action optimization, and improved information sharing. A case study conducted within a specific organization highlights Agile methods, release management, and chatbot functionalities in streamlining releases. Findings reveal that LLM chatbots can optimize workflows without disrupting existing processes if user needs and tool integration challenges are addressed. The research underscores the transformative role of LLMs in enhancing collaboration, efficiency, and reliability in software release management[9].

Tistelgrén 2024 looks at how artificial intelligence fits into software development, with an eye towards NLP tools and coding efficiency. By means of surveys of software professionals, the study reveals adoption hurdles include ethical issues, skill shortages, and integration difficulties. It measures the advantages of artificial intelligence like faster development cycles, better productivity, and more developer experience. Rooted in value creation theories, the thesis bridges the theory and practical divide by placing results inside a theoretical framework. Emphasising AI's ability to improve operations, inspire innovation, and offer competitive benefits while handling the complexity of ethical and technical integration, the study [10].

Goyal 2024 examines techniques for best CI/CD pipeline optimisation using cloud technologies to guarantee dependability and speed of software delivery. Key topics are workflow automation, containerisation for consistent deployment, serverless IT for scalability, and medical monitoring for proactive issue resolving. The study addresses security and scalability issues by stressing ML uses in clever test selection, defect estimate, and resource optimisation. We look at new CI/CD efficiency ideas including microservices, Infra as Code (IaC), and Kubernetes. Practical guidance helps businesses to properly satisfy evolving software development needs.Intelligent program agents for software for Constant Distribution fundamentally changes the direction of software development, testing, and application usage [11].

TABLE.1 LITERATURE SUMMARY

Authors	Methodology	Research gap	Findings
Tran 2024 [12]	The study employs an ML-based FDI framework integrating XAI principles and MLOps guidelines.	Opaque ML models in fault detection lack explainability and efficient development practices	The proposed framework improves FDI performance, operational quality, and model interpretability.
Manchana 2024 [13]	Explores DevOps principles, toolchain, and automation benefits using real-world case studies.	Lack of comprehensive studies on end-to-end DevOps automation across diverse industries.	DevOps automation accelerates delivery, enhances quality, and improves operational efficiency significantly.
Bedoya 2024 [14]	The Research integrates a Large Language Model (LLM) for automated threat discovery and Security Chaos Engineering for flaw identification.	Overreliance on traditional security tools fails to identify vulnerabilities across various stages of the software lifecycle	LLMs enhance threat detection at the design stage, and Security Chaos Engineering improves security flaw identification in agile development.
Sierhieiev 2024 [15]	Behavioral Analysis with Machine Learning (BAML) combines supervised and unsupervised learning to analyze system calls, network traffic, and user interactions.	Traditional methods like SAST, DAST, and IAST struggle with zero-day vulnerabilities, dynamic threats, and complex dependency analysis.	BAML achieved a 94% true positive rate, 11% false positive rate, 93% code coverage, and excels in zero-day vulnerability detection.
Ouaarous 2024 [16]	This study analyzed AI applications in Software Quality Assurance (SQA) using established taxonomies and categorized 5-year literature.	Limited research addresses AI's impact on SQA roles, Human-AI collaboration, and challenges like data quality issues.	AI improves SQA processes with cost reductions, enhanced quality, and promising results from Human-AI collaboration strategies.

III. INTELLIGENT SOFTWARE AGENTS IN CONTINUOUS DELIVERY

Smart programs have grown in ongoing development (CD) systems into transforming agents able to participate in demanding tasks across the software creation lifetime as independent entities. By means of artificial intelligence (AI) or machine learning (ML) models, these agents reduce the human involvement in CD pipelines, enhance automation, and simplify operations. Their participation guarantees flawless software delivery with least possible downtime from constant monitoring to automating build procedures and even feedback loops. AI-powered agents find possible issues before they manifest by evaluating large amounts of data including code modifications, test results, and deployment metrics, therefore assisting CI/CD systems' decisions. Early detection of code defects, security vulnerabilities, along with other bottlenecks made reachable by these agents's forecasting capabilities improves software functionality and dependability. Intelligent software agents can thus also enable test automation by selecting suitable test cases that depend on code changes, ranking them based on risk & impact, and execution them free from human participation.[17]–[19]. Keeping rigorous standards helps to reduce the required testing time. While all the time they ensure that upgrades are securely carried on through operating without end, intelligent agents can monitor circumstances, handle scalability, and maximise resource consumption during the deployment period.sacrificing user experience In the case of a faulty deployment, these agents can also provide rollback mechanisms, so automatically enabling the restoration of past stable editions and so retaining the operational qualities of the system. By adding feedback loops controlled by ML, these agents can continuously learn from past deployments, hence modifying and enhancing their behaviour depending on historical results and information. This generates a self-improving system that over time gets more efficient, therefore accelerating the supply of software. Adoption of advanced software agents in CD pipelines not only simplifies release cycles but also reduces risk of errors and increases general operational effectiveness. Still, managing difficult procedures, ensuring the quality and interpretability of the data they manage, and ensuring the smooth integration of these agents with present DevOps technologies create challenges. Notwithstanding these challenges, intelligent software agents exhibit great advancement in the pursuit of entirely automated, dependable, scalable CD pipelines [20]–[25].

IV. ROLE OF AI AND ML IN AUTOMATING DEVOPS PIPELINES

By improving several phases of software development, testing, deployment, and monitoring, artificial intelligence and machine learning significantly help to automate DevOps processes. The main objectives of DevOps are to simplify processes, hasten delivery cycles, and enhance cooperation between teams of development and operations. Tools driven by artificial intelligence help to automate repetitive jobs, streamline processes, and offer data-driven insights for improved decision-making. Predictive analytics made possible by machine learning models helps teams to proactively handle possible problems, such system failures or CI/CD pipeline bottlenecks, thereby impacting the production environment. AI and ML are applied to automate test case selection, test prioritising, and defect detectio during the build and testing processes.[26]–[28]n. AI models can find the most pertinent tests to run by examining codebase changes, hence lowering test running times and preserving complete coverage even in this regard. Additionally trained to find trends in past test results, ML techniques can uncover possible flaws or regressions otherwise missed. AI and ML models automate environment design, scalability, and resource optimisation in deployment, therefore guaranteeing robust and efficient installations. AI agents may forecast infrastructure needs and dynamically modify resources depending on workload demands by combining real-time performance measures with past data. Post-deployment is another area where artificial intelligence is important since smart monitoring tools examine system performance and user behaviour to spot anomalies, security flaws, and performance degradation. These systems can react to problems on their own by starting rollbacks or resource scaling, therefore addressing problems. Moreover, by use of feedback loops, artificial intelligence and machine learning may continuously improve the pipeline therefore enabling the system to learn from prior implementations and modify its behaviour for next releases. While lowering human interaction and operational hazards, overall the integration of artificial intelligence and machine learning into DevOps pipelines results in higher automation, efficiency, and dependability.[29]–[31].

V. FULLY AUTOMATED DEVOPS PIPELINES

Fully automated DevOps pipelines combining critical concepts including Continuous Integration (CI) and Continuous Creation (CD) simplify the software delivery lifecycle by use of modern automation tools. Reducing hand-made interventions and enhancing communication between the operations and development teams helps these pipelines to improve dependability, speed, and software quality. Key components include smart software agents, artificial intelligence, and machine learning are applied to maximise every process from code development to deployment. Three key traits below allow very efficient completely automated DevOps systems [32].

A. End-to-End Automation of Development and Deployment Processes

From development to production, entirely automated DevOps solutions automate the whole software delivery lifecycle using intelligent software agents, artificial intelligence, or machine learning. Combining Continuous Integration (CI) with Continuous Development (CD) approaches results in an automatically starting pipeline involving code commits, builds, tests, and deployments. Reducing hand inputs and standardising processes ensures uniformity and speed across several steps by means of automation.[32]. From version control infrastructure, the pipeline can automatically produce code, run unit tests, package the software for release. Furthermore, it can start additional steps including security testing, functional, coding standards compliance, and functional, regression, so ensuring that only code meeting criteria finds use in production. This end-to-end automation greatly reduces human error, streamlines delivery, and provides a consistent path for more frequent and more rapid software updates [33].

B. Intelligent Monitoring and Predictive Analytics

Deeply ingrained in entirely automated DevOps systems, artificial intelligence and machine learning models offer real-time monitoring and predictive analytics all through the lifetime of products. These models look at enormous volumes of pipeline produced data like system logs, test results, and build metrics to discover early in the process potential dangers, bottlenecks, or defects. For example, predictive models can forecast build failures based on past trends or pinpoint high-risk code changes most likely to produce regressions.[34]. This proactive approach helps to reduce downtime and minimise costly mistakes by letting developers correct issues before they impact the manufacturing environment. Dynamic resource scaling by AI-powered systems allows them to match varying workloads, therefore ensuring best performance and avoiding system overloads. They can also separately alter infrastructure requirements during implementation [34], [35].

C. Autonomous Rollbacks and Continuous Learning

Mostly depending on their ability to independently start rollbacks and artificial intelligence & machine learning-based continuous learning, completely automated DevOps systems are Should a deployment fail or a detected vulnerability surface, the pipeline can immediately go back to the last stable version, therefore minimising disturbance and ensuring system availability. This rewind system is driven by intelligent agents who track the industrial environment in real-time, assess changes, and act with corrections as necessary. Through continuous education, the pipeline also develops through artificial intelligence models analysing past deployment errors and successes to direct current decision-making. This feedback loop ensures that the pipeline always changes to fit new circumstances, improves accuracy, and best leverages resources for next installs, hence building resilience over time.
[36].

VI. CHALLENGES AND OPPORTUNITIES

Although totally automated DevOps pipelines provide numerous advantages, if deployment is to be successful certain problems must be addressed. The complexity of integration is among the primary challenges. Combining numerous tools and technologies under development—version control systems, testing systems, deployment platforms—requires deliberate planning and collaboration. Incorrect settings or incompatibility between tools could lead to failures, therefore reducing the pipeline's efficiency. Moreover hindering the smooth transition to a totally automated operation is the learning curve connected with using new technologies and tools. Still another challenge is keeping data quality for machines learning and AI models under development. Although in reality data is often incomplete, noisy, or biassed, these models generally rely on high-quality data to create accurate forecasts.[37]. This can jeopardise the effectiveness of the pipeline by erroneous insights or failed forecasts. Moreover important is controlling the scalability of the automated pipeline, especially when the pipeline grows to oversee ever complex software projects. AI-powered systems must be able to dynamically adapt to growing workloads, hence constant resource management and optimisation are absolutely necessary. There are excellent opportunities presented by the evolving landscape of completely automated DevOps systems. Artificial intelligence and machine learning enable predictive maintenance and proactive issue addressing that helps to minimise bottlenecks or failures before they impact output. Intelligent monitoring systems allow the quick improvement in system performance or resource allocation, so improving overall efficiency.[38]. Faster releases, improved software quality, and wiser decision-making are made possible in part by growing maturity of automating tools and technologies. Separating development, operations, and excellence assurance teams enables automation to foster a cooperative culture inside businesses. This drives creativity, continuous improvement, and faster acceptance of new technologies, hence fostering organisational development and competitiveness. By addressing issues and realising the opportunities, companies may create robust, consistent, and efficient DevOps pipelines [39]–[42].

CONCLUSION

Ultimately, including intelligent software agents, artificial intelligence, and machine learning into completely automated DevOps pipelines will cause a profound transformation in software development and delivery. By automating important processes such code integration, testing, deployment, and monitoring, companies may significantly increase the speed, dependability, and quality of software releases. Stronger and more resilient pipelines are created via predictive maintenance, proactive issue resolution, and ongoing learning made possible by artificial intelligence-driven insights. Still important challenges to be addressed, though, include scalability, data quality control, and tool integration complexity. Notwithstanding these challenges, businesses have significant opportunity to streamline procedures, reduce human error, and foster a culture of constant growth and teamwork. Artificial intelligence and machine learning technologies will enable DevOps pipelines to constantly adapt to enable faster, smarter, more reliable software delivery. As these technologies advance, their relevance will increase and so create new prospects for predictive analytics, intelligent decision-making, and automation. All things considered, smart software agents in DevOps pipelines are prepared for more agility, efficiency, and success in the competitive digital landscape since they will enable firms to approach software development and deployment differently.

REFERENCES

[1] R. McCreadie *et al.*, “Leveraging data-driven infrastructure management to facilitate AIOps for big data applications and operations,” *Technol. Appl. Big Data Value*, pp. 135–158, 2022, doi: 10.1007/9783030783075_7.

[2] S. Johnson, “DEVOPS IN A BOX : AN OPEN-SOURCE STARTER KIT FOR NEW STARTUPS,” vol. 3, no. 2, pp. 143–160, 2021.

[3] A. M. Thakur *et al.*, “Towards a Software Development Framework for Interconnected Science Ecosystems,” *Commun. Comput. Inf. Sci.*, vol. 1690 CCIS, pp. 206–224, 2022, doi: 10.1007/978-3-031-23606-8_13.

[4] A. Alnafessah, A. U. Gias, R. Wang, L. Zhu, G. Casale, and A. Filieri, “Quality-Aware DevOps Research: Where Do We Stand?,” *IEEE Access*, vol. 9, pp. 44476–44489, 2021, doi: 10.1109/ACCESS.2021.3064867.

[5] Y. Ramaswamy, “AI - Optimized Bioinformatics Pipelines in DevOps,” vol. 1, no. 2, pp. 147–161, 2021, doi: 10.56472/25832646/JETA-

V1I2P117.

[6] S. Tatineni and V. R. Boppana, "AI-Powered DevOps and MLOps Frameworks: Enhancing Collaboration, Automation, and Scalability in Machine Learning Pipelines," *J. Artif. Intell. ...*, vol. 1, no. 2, pp. 58–88, 2021, [Online]. Available: https://aimlstudies.co.uk/index.php/jaira/article/view/103%0Ahttps://aimlstudies.co.uk/index.php/jaira/article/download/103/97

[7] E. Ok and J. Eniola, "Accelerating Software Releases : Implementing Continuous Integration and Continuous Delivery with Jenkins," no. January 2024, 2025.

[8] V. M. Tamanampudi, "End-to-End ML-Driven Feedback Loops in DevOps Pipelines End-to-End ML-Driven Feedback Loops in DevOps Pipelines," no. September, 2024, doi: 10.30574/wjaets.2024.13.1.0424.

[9] I. Tiedekunta, "CHATBOTS IN SOFTWARE RELEASE OPTIMIZA- TION : CASE STUDY," 2024.

[10] S. Tistelgrén, "Artificial Intelligence in Software Development: Exploring Utilisation, Tools, and Value Creation," 2024.

[11] A. Goyal, "Optimising Cloud-Based CI / CD Pipelines : Techniques for Rapid Software Deployment," no. December, 2024.

[12] T. A. Tran, T. Ruppert, and J. Abonyi, "The Use of eXplainable Artificial Intelligence and Machine Learning Operation Principles to Support the Continuous Development of Machine Learning-Based Solutions in Fault Detection and Identification," *Computers*, vol. 13, no. 10, 2024, doi: 10.3390/computers13100252.

[13] R. Manchana and C. O. Company, "The DevOps Automation Imperative : Enhancing Software Lifecycle Efficiency and Collaboration The DevOps Automation Imperative : Enhancing Software Lifecycle Efficiency and Collaboration Ramakrishna Manchana," no. July 2021, 2024, doi: 10.5281/zenodo.13789734.

[14] M. Bedoya, S. Palacios, D. Díaz-López, E. Laverde, and P. Nespoli, "Enhancing DevSecOps practice with Large Language Models and Security Chaos Engineering," *Int. J. Inf. Secur.*, vol. 23, no. 6, pp. 3765–3788, 2024, doi: 10.1007/s10207-024-00909-w.

[15] Y. Sierhieiev, V. Paiuk, A. Nicheporuk, A. Kwiecien, and O. Huralnyk, "Detection and prediction of the vulnerabilities in software systems based on behavioral analysis with machine learning," *CEUR Workshop Proc.*, vol. 3736, pp. 239–254, 2024.

[16] R. Ouaarous, I. Hilal, and A. Mezrioui, "On Using Artificial Intelligence in Software Quality Assurance: A State of the Art," *CEUR Workshop Proc.*, vol. 3845, 2024.

[17] A. Mankotia, S. Manager, D. Fueling, and S. Llc, "Impact of AI and Language Models on DevOps and DevSecOps," vol. 14, no. 07, pp. 61–81, 2024.

[18] S. Chittala, "ORCHESTRATING THE CLOUD : AI- ENHANCED RELEASE AUTOMATION IN," vol. 7, no. 2, pp. 864–878, 2024.

[19] A. Shukla, J. K. Vijay, U. Sen, and J. Jain, "From Prototyping to Production: LLM Chains carrying the Software Development Pipeline," vol. 11, no. 3, pp. 193–200, 2024.

[20] S. Shah, "THE RISE OF AI AGENTS IN ENTERPRISE," vol. 15, no. 5, pp. 803–813, 2024.

[21] N. Nivedhaa, "EVALUATING DEVOPS TOOLS AND TECHNOLOGIES FOR EFFECTIVE CLOUD," vol. 1, no. 1, pp. 20–32, 2024.

[22] N. Gadani, "ARTIFICIAL INTELLIGENCE : LEVERAGING AI-BASED TECHNIQUES FOR ARTIFICIAL INTELLIGENCE : LEVERAGING AI-BASED TECHNIQUES FOR SOFTWARE QUALITY Artificial Intelligence in Software Development," no. July, 2024, doi: 10.56726/IRJMETS60018.

[23] S. Kumar, "AI/ML Enabled Automation System for Software Defined Disaggregated Open Radio Access Networks: Transforming Telecommunication Business," *Big Data Min. Anal.*, vol. 7, no. 2, pp. 271–293, 2024, doi: 10.26599/BDMA.2023.9020033.

[24] H. W. Marar, "Advancements in software engineering using AI," *Comput. Softw. Media*

Appl., vol. 6, no. 1, p. 3906, 2024, doi: 10.24294/csma.v6i1.3906.

[25] F. Bayram, "Towards Trustworthy Machine Learning in Production : An Overview of the Robustness in MLOps Approach," no. Ml, 2024, doi: 10.1145/3708497.

[26] J. Kohl *et al.*, "Generative AI Toolkit -- a framework for increasing the quality of LLM-based applications over their whole life cycle," 2024, [Online]. Available: http://arxiv.org/abs/2412.14215

[27] P. Sowinski, I. Lacalle, R. Vano, C. E. Palau, M. Ganzha, and M. Paprzycki, "Overview of Current Challenges in Multi-Architecture Software Engineering and a Vision for the Future," pp. 1–21, 2024, [Online]. Available: http://arxiv.org/abs/2410.20984

[28] B. Eken, S. Pallewatta, N. K. Tran, A. Tosun, and M. A. Babar, "A Multivocal Review of MLOps Practices, Challenges and Open Issues," 2024, [Online]. Available: http://arxiv.org/abs/2406.09737

[29] M. Fu, J. Pasuksmit, and C. Tantithamthavorn, "AI for DevSecOps: A Landscape and Future Opportunities," vol. 1, no. 1, 2024, [Online]. Available: http://arxiv.org/abs/2404.04839

[30] P. Nama, A. Technology, and S. Chinta, "Autonomous Test Oracles : Integrating AI for Intelligent Decision-Making in Automated Software Testing," no. October, 2024.

[31] P. Ganesan and G. Sanodia, "Journal of Artificial Intelligence & Cloud Computing Smart Infrastructure Management : Integrating AI with DevOps for Cloud-Native Applications," vol. 2023, no. October, 2024, doi: 10.47363/JAICC/2023(2)E163.

[32] J. Yang, "Next-Gen Enterprise Architecture : Unlocking AI and Cloud Potential Through DevOps Integration Date : November , 2024," no. November, 2024, doi: 10.13140/RG.2.2.12984.97286.

[33] F. Dine, "Transforming IT Operations : The Power of AI-Enhanced Cloud , DevOps , and DataOps in Enterprise Architecture," no. November, 2024, doi: 10.13140/RG.2.2.31416.97285.

[34] A. Schmelczer and J. Visser, "Trustworthy and Robust AI Deployment by Design: A framework to inject best practice support into AI deployment pipelines," *Proc. - 2023 IEEE/ACM 2nd Int. Conf. AI Eng. - Softw. Eng. AI, CAIN 2023*, pp. 127–138, 2023, doi: 10.1109/CAIN58948.2023.00030.

[35] S. Dinesh, "A Framework of DevSecOps for Software Development Teams," no. June, 2023, [Online]. Available: https://urn.fi/URN:NBN:fi-fe2023073192373

[36] M. Underwood, "Continuous Metadata in Continuous Integration, Stream Processing and Enterprise DataOps," *Data Intell.*, vol. 5, no. 1, pp. 275–288, 2023, doi: 10.1162/dint_a_00193.

[37] A. Sunny, "Artificial Intelligence's Effect on Contemporary Computer Software," no. December, 2023, [Online]. Available: https://www.researchgate.net/publication/376645343

[38] M. Cankar *et al.*, "Security in DevSecOps: Applying Tools and Machine Learning to Verification and Monitoring Steps," *ICPE 2023 - Companion 2023 ACM/SPEC Int. Conf. Perform. Eng.*, pp. 201–205, 2023, doi: 10.1145/3578245.3584943.

[39] C. Birchler, S. Khatiri, B. Bosshard, A. Gambi, and S. Panichella, "Machine learning-based test selection for simulation-based testing of self-driving cars software," *Empir. Softw. Eng.*, vol. 28, no. 3, 2023, doi: 10.1007/s10664-023-10286-y.

[40] A. M. Thakur *et al.*, "Towards a Software Development Framework for Interconnected Science Ecosystems," *Commun. Comput. Inf. Sci.*, vol. 1690 CCIS, no. July, pp. 206–224, 2022, doi: 10.1007/978-3-031-23606-8_13.

[41] P. Pham, V. Nguyen, and T. Nguyen, "A Review of AI-augmented End-to-End Test Automation Tools," *ACM Int. Conf. Proceeding Ser.*, 2022, doi: 10.1145/3551349.3563240.

[42] E. Mosqueira-Rey, E. H. Pereira, D. Alonso-Ríos, and J. Bobes-Bascarán, "A classification and review of tools for developing and interacting with machine learning systems," *Proc. ACM Symp. Appl. Comput.*, pp. 1092–1101, 2022, doi: 10.1145/3477314.3507310.