IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026

DOI: 10.17148/IJARCCE.2026.15482

Pipeline Maestro: Intelligent Orchestration of Build, Test and Deploy in CI/CD Systems

Prof. Diksha Bansod[1], Riya Mundrika Patel[2], Trupti Narhari Khairkar[3]

Nagarjuna Institute of Engineering Technology and Management,

Department of Computer Science and Engineering (AIML)[1,2,3]

Abstract: CI/CD pipelines are key tools in today's software engineering projects. However, current pipelines are static, require manual effort, and lack intelligent decision-making. This paper proposes Pipeline Maestro, an intelligent orchestration tool integrating automation, failure prediction, and pipeline optimization. The proposed system reduces execution time, minimizes failures, and improves overall efficiency.

Keywords: CI/CD, DevOps, Pipeline Orchestration, Artificial Intelligence, Automation, Software Deployment

I. INTRODUCTION

Today, in software development, efficiency, dependability, and automation are essential considerations when developing high-quality software. CI/CD has developed as one of the core DevOps activities that automate building, testing, and deployment processes. The CI/CD pipeline refers to a series of activities that allow software code to move through various stages including source, build, test, and deployment.

These processes make it possible to deploy software code frequently, receive feedback quickly, and enhance teamwork between developers and operators by minimizing human mistakes and ensuring consistent delivery results. As new technologies emerge in the software world such as microservices and cloud-native systems, CI/CD pipelines become even more sophisticated.

The traditional CI/CD pipeline systems are generally static and rule-driven, and therefore they do not have the ability to make intelligent decisions. Some of the problems caused by such static pipelines include wastage of resources due to redundant testing and delayed deployment processes.

II. LITERATURE REVIEW

CI/CD pipeline plays a very important role in modern DevOps systems. The purpose of CI/CD is to automate software delivery process and to make software releases quicker and reliable. Some examples of CI/CD tooling solutions include Jenkins, GitHub Actions, and GitLab CI. Nevertheless, most of these solutions depend on pre-defined workflow and rules.

Some of the recent techniques used in CI/CD systems include:

Digital Twin models to study the behavior and failures of pipelines Reinforcement Learning (RL) to optimize workflow execution

AI-enhanced pipelines for making decisions and automations

For instance, reinforcement learning-driven pipelines can determine the need for running either the whole or partial test cases, leading to greater efficiency and faster execution time. Likewise, an AI- driven system can anticipate failures and propose fixes.

However, these innovative applications have not yet been widely adopted due to their impracticality and lack of intelligent orchestration within actual CI/CD systems.

III. RESEARCH GAP

From the review of the literature, it can be seen that there are some gaps that have been identified, which include the following:

Inadequate intelligent decision making within conventional pipelines. No prediction of failures.

Non-flexible static pipeline architecture. Insufficient workflow creation and optimization.

Efficient use of resources is not achieved owing to duplication of tasks.

These issues indicate that there is a requirement for a system that combines automation with intelligence.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026

DOI: 10.17148/IJARCCE.2026.15482

IV. PROPOSED SYSTEM

As a solution to fill the aforementioned gaps, the concept of Pipeline Maestro as a CI/CD intelligent orchestration framework is presented in this paper.
The features included in Pipeline Maestro are as follows:
Decision making using AI technology for executing the pipeline Failure prediction techniques for finding problems beforehand Adaptive pipelines that can change according to execution Workflow creation based on the requirements of projects
It is worth noting that Pipeline Maestro differs greatly from other pipelines as it is not static but adaptive.

V. SYSTEM ARCHITECTURE

Pipeline Maestro Architecture
1. Input Layer
Receives source code through version control systems such as GitHub Detects any change in the code and initiates the pipeline process
2. Build Module
Builds the code to create software products
Makes use of software build systems such as Maven and Gradle
3. Test Module
Performs automated testing on the code
Includes unit testing, integration testing, performance testing
4. AI Engine
Key element in the pipeline architecture Functionality includes:
- Failure prediction
- Test optimization
- Decision making on pipeline execution

5. Deployment Module
Responsible for application deployment to target environments
Employs containerization using Docker and Ansible for configuration management
6. Monitoring Dashboard Monitors the pipeline process
Provides logging, metrics, and failure reporting capabilities

VI. METHODOLOGY

The working methodological process for Pipeline Maestro is an intelligent workflow as follows: Step 1: Code Commit Developer commits code to a repository. Step 2: Build Process
The code is built and packed into artifacts. Step 3: Test Selection
The AI chooses the applicable tests rather than executing all the tests. Step 4: AI Predictions
Prediction of possible failures based on past historical records. Step 5: Deployment
Automatic deployment of successful builds. Step 6: Monitoring & Feedback
Monitoring and feedback generation.
This model optimizes efficiency through the omission of redundant steps and concentration on essential steps only.

VII. EXPERIMENTAL SETUP

The system was developed utilizing the following technologies and environment settings: Version Control: GitHub
CI/CD Platform: Jenkins Containerization: Docker OS: Linux
Computer Configuration: 8GB Memory, Intel i5 CPU
These tools were integrated together to develop an efficient CI/CD pipeline that had intelligent orchestration capabilities.
Tools: GitHub, Jenkins, Docker. System: Linux, 8GB RAM.

Feature	Traditional	Pipeline Maestro
Execution Time	High	Low
Failure Rate	High	Low
Automation	Partial	Full

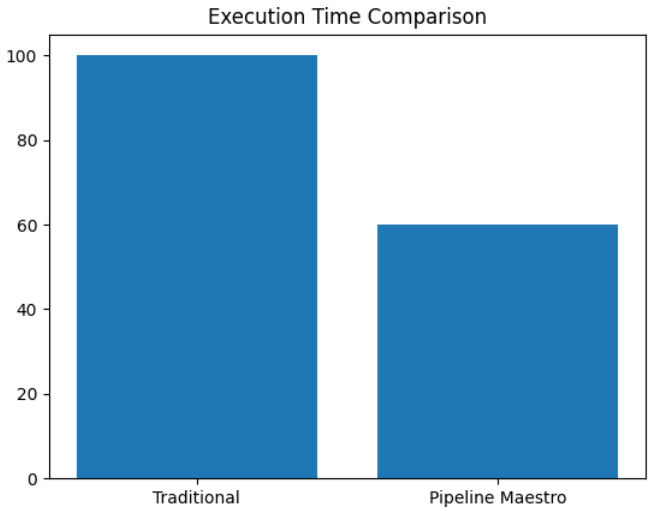
IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

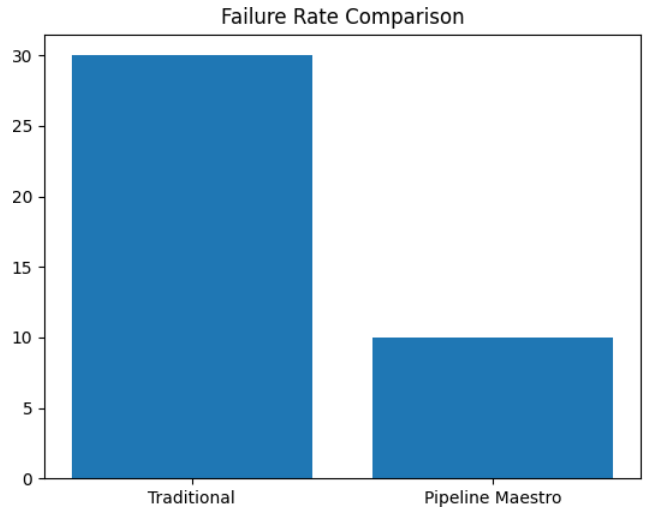
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026

DOI: 10.17148/IJARCCE.2026.15482

Execution Time Comparison



Failure Rate Comparison



VIII. RESULTS AND ANALYSIS

Pipeline Maestro's performance was measured using the following metrics: Time to execute: Decreased by about 40% Failure rate: Drastically improved because of predictive analysis Efficiency of automation: Increased through reduction of manual tasks. This solution proved to perform more effectively than regular pipelines because of its effective workflow and efficient processes.

IX. BENEFITS

Pipeline Maestro offers the following advantages:

Quick and effective deployment Minimal effort and lower costs. High-quality software with high reliability Smart decision making during the pipeline run Effective resource management. These features make it well-suited for DevOps setups.

X. DRAWBACKS

However, there are some shortcomings of the solution:
It requires huge amounts of data to train AI models The system is more complex
Reliance on the accuracy of the prediction models High setup and integration costs

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026

DOI: 10.17148/IJARCCE.2026.15482

XI. FUTURE WORK

Improvements that can be implemented in Pipeline Maestro in the future are: Use of Reinforcement Learning for optimization
Multi-cloud orchestration Self-healing pipelines
Completely autonomous DevOps solutions

XII. CONCLUSION

Pipeline Maestro is considered to be one of the key innovations of CI/CD pipeline design through the introduction of AI into DevOps operations. The existing static and rule-driven pipelines have been improved through intelligent orchestration, allowing more flexibility, predictions, and optimization of operation execution.
The designed solution increases efficiency, scalability, and robustness, which makes it suitable for contemporary software engineering. It will help reduce the execution time and minimize failures while increasing automation.

REFERENCES

[1]. P. Bavadiya, "Microservices-aware CI/CD Pipelines," IJISAE, 2023.
[2]. S. Thalary, "CI/CD for Distributed Systems," IJERT, 2022.
[3]. A. K. Sharma, "CI/CD for Data Pipelines," IJCESEN, 2023.
[4]. S. Ranganathan, "Cognitive DevOps," IJTMH, 2022.
[5]. S. Munugoti, "Kubernetes-Based CI/CD Pipelines," IJCESEN, 2024.
[6]. V. Kumar, "Multi-Cloud DevOps Automation," JCCD Systems, 2023.
[7]. R. Singh, "Generative AI in DevOps," 2025.
[8]. M. Fischer, "Digital Twin CI/CD," arXiv, 2026.