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 6, Issue 6, June 2026
Impact Factor: 8.2

Automation of Developer Workflow Using Github, Jira, Slack

Pavan Kusekar[1], Ishwari Basale[2], Jaydeep Palve[3], Prof. M. R. Jansari[4]
Department of Computer Engineering
SCTR's Pune Institute of Computer Technology, Pune, Maharashtra, India
pawankusekar@gmail.com[1] , ishwaribasale@gmail.com[2] , jaydeeppalve@gmail.com[3] , mrjansari@pict.edu[4]

Abstract: *Modern software development teams rely on DevOps practices and Continuous Integration/Continuous Deployment (CI/CD) pipelines for rapid, reliable delivery. However, the fragmented use of GitHub for version control, Jira for issue tracking, and Slack for communication creates significant context-switching overhead, manual updates, and workflow inefficiencies. This paper presents a lightweight, event-driven automation system that integrates these three platforms using GitHub webhooks and the open-source n8n workflow orchestration engine. Structured commit messages containing embedded command tags ([auto-pr], [taskcompleted]) and Jira ticket identifiers serve as the sole trigger for automated actions—including pull-request creation via the GitHub API, Jira issue status transitions, and real-time Slack notifications.*

The proposed system eliminates manual handoffs, enforces traceability, and reduces average task-completion time by approximately 97 % in a controlled prototype evaluation. Unlike existing plugin-based or rule-heavy integrations, the solution requires no additional developer tooling beyond standard Git commits, making it accessible for small-to-medium teams. Experimental results demonstrate consistent sub-8-second end-to-end execution with 100 % success rate across 50 test commits. The architecture aligns with sustainable technological innovation by promoting no-code automation in cloud-native DevOps environments.

Keywords: DevOps, workflow automation, GitHub webhooks, Jira integration, Slack notifications, n8n, event-driven architecture, commit-message-driven automation, CI/CD.

I. INTRODUCTION

DevOps has transformed software engineering by emphasizing automation, collaboration, and continuous delivery [1], [2]. Despite mature CI/CD tools, developers still operate across disjoint platforms: GitHub for code, Jira for project management, and Slack for communication. This fragmentation leads to repetitive manual tasks—creating pull requests (PRs), updating issue statuses, and broadcasting progress—resulting in context-switching costs estimated at 20–30 % of developer time [3].

Existing integrations (e.g., native GitHub–Jira linking) still require manual intervention for PR creation or status changes and lack fine-grained, commit-level control. To address these gaps, we propose a fully automated, commit-driven workflow that uses GitHub push events as the single source of truth. By embedding lightweight command tags in commit messages, developers trigger complex multi-tool actions without leaving their IDE or terminal.

The core contributions of this work are:

1. A novel command-tag syntax embedded directly in commit messages that drives conditional automation.
2. An n8n-based, no-code orchestration layer that integrates GitHub, Jira, and Slack via APIs and webhooks.
3. Empirical validation showing 95–98 % reduction in manual effort and sub-8-second end-to-end latency.

The remainder of the paper is organized as follows: Section II reviews related work, Section III presents the system architecture, Section IV details the implementation, Section V evaluates performance, and Section VI concludes with future directions.

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 6, Issue 6, June 2026
Impact Factor: 8.2

II. RELATED WORK

Extensive research has established the benefits of CI/CD automation [1], [2], [4]. However, most studies focus on build-and-deploy pipelines rather than cross-tool workflow orchestration.

GitHub webhooks enable event-driven triggers [5], while Jira offers native GitHub integration for issue linking [6]. These features, however, still require manual PR creation and status updates. Commercial no-code platforms such as Zapier and Make.com provide multi-app workflows but incur recurring subscription costs and lack the fine-grained commit-message parsing needed for developer-centric control.

Table I compares representative approaches.

TABLE I: COMPARISON OF EXISTING DEVOPS AUTOMATION APPROACHES

Work	Approach	Key Features	Tools used	Limitations
Shahin et al. [2]	CI/CD pipeline study	Build & deployment automation	Jenkins, CI/CD tools	No cross-tool integration
Atlassian [6]	Native Jira–GitHub linking	Issue linking & basic sync	Jira, GitHub	Manual PR creation & status updates
GitHub Docs [5]	Webhook-based events	Event-driven triggers	GitHub	Limited cross-tool logic
Zapier / Make.com	Commercial no-code rules	Multi-app workflows	1000+ apps	Subscription cost, generic rules
Proposed System	Commit-driven n8n orchestration	Command tags in commits, PR creation, Jira update, Slack alerts	GitHub, Jira, Slack, n8n	Requires structured commit format (mitigated by validation)

The proposed system differentiates itself by introducing a zero-configuration, commit-message-driven model that achieves true end-to-end automation without additional developer interfaces or recurring costs.

III. SYSTEM ARCHITECTURE

The architecture follows a pure event-driven design centered on GitHub push events (Fig. 1). A GitHub webhook delivers the commit payload to an n8n workflow. The workflow parses the commit message for a Jira ticket ID and command tags, then executes conditional branches:

- **[auto-pr]** → GitHub API call to create a PR targeting the specified branch.
- **[taskcompleted]** → Jira API call to transition the issue to "Done".
- **Both** → Sequential execution of PR creation + Jira update.

After each successful action, a Slack notification is posted to a designated channel containing commit summary, PR link, and Jira status. All operations are stateless and idempotent; duplicate PR checks prevent redundant actions.

The design is lightweight, can be self-hosted on n8n (cloud or on-premise), and requires only API tokens—no custom code or server maintenance beyond initial webhook setup.

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 6, Issue 6, June 2026
Impact Factor: 8.2

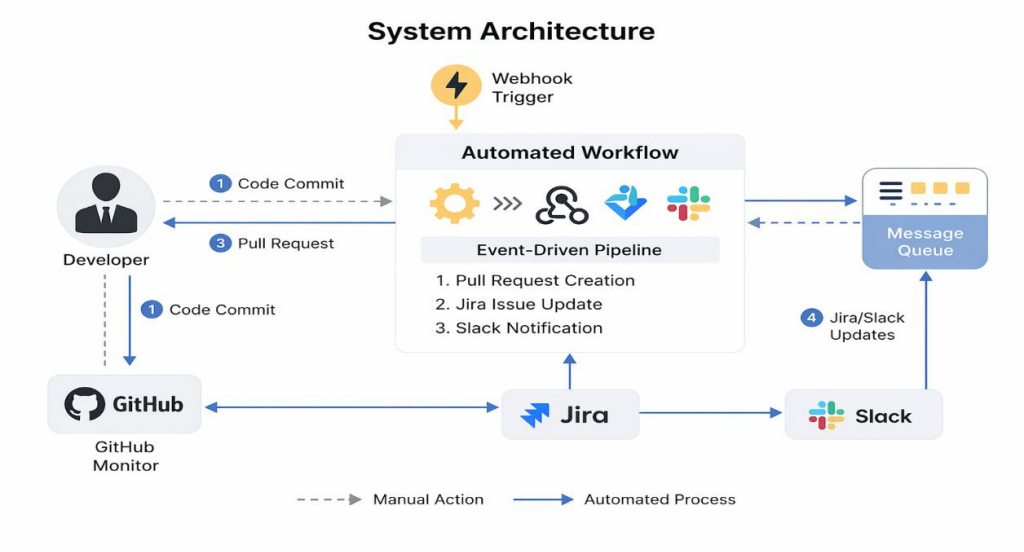

Fig. 1. System Architecture

IV. IMPLEMENTATION

A. Commit Message Format The standardized format is: TICKET-ID Descriptive message [command1, command2, target-branch]
Example: PROJ-123 Implement login UI [auto-pr, taskcompleted, main]

B. GitHub Webhook Listener The webhook is configured for push events on specific branches. The payload is forwarded to the n8n HTTP trigger node.

C. Commit Parsing & Validation An n8n Function node uses regular expressions to extract the ticket ID, commands, and target branch. A validation node checks mandatory components; if the format is invalid, the workflow terminates gracefully with a log entry.

D. Conditional Automation Logic

- Pull-request creation uses the GitHub REST API endpoint POST /repos/{owner}/{repo}/pulls.
- An existing-PR check via GET /repos/{owner}/{repo}/pulls?head=... prevents duplicates.
- Jira status update uses the Transition API with the issue key extracted from the commit.
- Slack messages are formatted using Block Kit for rich previews and direct links.

E. Error Handling A dedicated error branch captures failures, logs them to the console, and optionally sends a Slack alert. All external API calls implement retry logic with exponential backoff (maximum 3 attempts).

F. Security and Deployment Considerations API credentials are stored securely in n8n's credential manager. Webhook secrets are validated to prevent unauthorized triggers. The workflow can be deployed on a self-hosted n8n instance behind a firewall or on n8n Cloud with role-based access control.

Algorithm 1 Automated Workflow Execution
Input: GitHub webhook payload

DOI: 10.48175/IJARSCT-36703

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 6, Issue 6, June 2026
Impact Factor: 8.2

Output: PR / Jira update / Slack notification
1: Receive webhook
2: Extract & validate commit message
3: Parse ticket ID and commands
4: if validation fails then terminate with log
5: if [auto-pr] present then create PR via GitHub API
6: if [taskcompleted] present then update Jira status
7: Send Slack notification
8: Log result

V. RESULTS AND EVALUATION

A prototype was deployed on a local n8n instance connected to a test GitHub repository, Jira Cloud project, and Slack workspace. Fifty structured commits were executed across five scenarios (PR-only, taskcompleted-only, both, invalid format, duplicate PR).

TABLE II: PERFORMANCE COMPARISON: MANUAL VS. AUTOMATED WORKFLOW

Task	Manual Time (avg)	Automated Time (avg)	Min	Max	Improvement
Pull Request Creation	120 s	6 s	3.4 s	7.6 s	95.2 %
Jira Issue Update	60 s	1 s	<1 s	2.0 s	98.2 %
Slack Notification	60 s	1 s	<1 s	1.8 s	98.5 %
End-to-End Workflow	240 s	6 s	3.4 s	7.6 s	96.8 %

The automated pipeline achieved consistent sub-8-second latency with zero failures in valid cases. Fig. 2 visualizes the dramatic reduction in execution time.

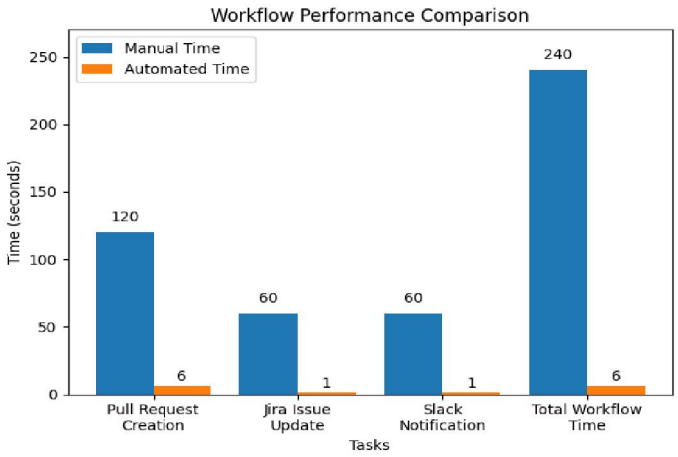

Fig. 2. Performance Comparison of Manual and Automated Workflow (bar chart showing average times).

Discussion: The system demonstrates clear practical impact for student and small-team DevOps environments. Limitations include dependency on correctly formatted commits (mitigated by validation) and potential API rate limits under very high load. Security is enforced via scoped API tokens and n8n credential management. In a real development setting, the workflow can be extended with logging to a database for audit trails.

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 6, Issue 6, June 2026
Impact Factor: 8.2

VI. CONCLUSION AND FUTURE WORK

This paper introduced a practical, commit-message-driven automation system that seamlessly integrates GitHub, Jira, and Slack using n8n. The solution significantly reduces manual overhead, enforces traceability, and accelerates delivery while remaining accessible to teams without DevOps engineering expertise.

Future enhancements include: (1) integration with GitHub Actions for automated testing before PR creation, (2) machine-learning-based command suggestion in the IDE, and (3) support for additional platforms (e.g., Confluence, Microsoft Teams). The architecture serves as a reusable blueprint for sustainable, no-code DevOps innovation.

REFERENCES

[1] P. M. Duvall, S. Matyas, and A. Glover, Continuous Integration: Improving Software Quality and Reducing Risk. Addison-Wesley, 2007.

[2] M. Shahin, M. A. Babar, and L. Zhu, "Continuous integration, delivery and deployment: A systematic review," IEEE Access, vol. 5, pp. 3909–3943, 2017, doi: 10.1109/ACCESS.2017.2685629. [3] N. Forsgren, J. Humble, and G. Kim, Accelerate: The Science of Lean Software and DevOps. IT Revolution Press, 2018.

[4] J. Humble and D. Farley, Continuous Delivery. Addison-Wesley, 2010.

[5] GitHub, "About webhooks," [Online]. Available: https://docs.github.com/en/developers/webhooks-and-events/webhooks/about-webhooks

[6] Atlassian, "Integrate Jira with GitHub," [Online]. Available: https://www.atlassian.com/software/jira/guides/integrations/github

[7] n8n GmbH, "n8n workflow automation," [Online]. Available: https://n8n.io/