On the Automation and Reuse Practices in GitHub Actions: Results of a Qualitative Survey

Hassan Onsori Delicheh[1], Guillaume Cardoen, Alexandre Decan and Tom Mens

[1]*Software Engineering Lab, University of Mons, Belgium*

Abstract

GitHub Actions is the dominant workflow automation tool for GitHub repositories. Workflow maintenance is often considered a burden for software developers, who frequently face difficulties in writing, testing, and debugging workflows. In the first half of 2025, we carried out an online survey of 419 GitHub workflow maintainers to understand their current automation and reuse practices, challenges, and preferences. We informed about the tasks that tend to be automated using GitHub Actions, the preferred workflow creation mechanisms, and the non-functional characteristics prioritised by respondents. We also examined the practices and challenges associated with GitHub's workflow reuse mechanisms. We observed significant disparities in automation adoption, with core CI/CD tasks being widely automated, but crucial areas like security analysis and performance monitoring receiving less attention. Next to GitHub Actions' built-in reuse mechanisms that are appreciated by many, we observed that copy-pasting remains a prevalent mechanism because it is perceived to be more convenient and allows for more control. These insights highlight opportunities for improved tooling, enhanced support for automation tasks, and better mechanisms for discovering, managing, and trusting reusable components.

Introduction

GitHub Actions is GitHub's integrated CI/CD mechanism. Since its release in 2019 it has become the *de facto* workflow automation tool for GitHub repositories. It enables repository maintainers to automate an unbounded range of automation tasks through workflow configurations. It also provides built-in support for reusable Actions and workflows, which can be shared and reused across workflows and repositories. Despite the widespread use of GitHub Actions, little is known about the automation and reuse practices adopted by workflow developers and maintainers. This hinders the development of best practices and tools to improve workflow automation for collaborative software development.

To fill this gap, we designed and conducted an online survey targeting GitHub workflow maintainers. Using a dataset of workflow commits [1] we identified 6,500 potential respondents having committed at least 10 workflow changes with at least one recent commit. We collected 419 complete responses from practitioners with demonstrated GitHub Actions experience. Data analysis comprised descriptive statistics, non-parametric hypothesis testing, and qualitative coding of free-text responses. The survey addressed two primary goals aimed at understanding GitHub Actions workflow usage and maintenance. Such understanding can ultimately lead to increased effectiveness of workflow usage and reduced maintenance effort.

G1: Understanding adopted workflow automation practices

To reach this goal, the survey included **three** questions related to the tasks being automated by workflows, the mechanisms employed to create these workflows, and the importance of specific non-functional characteristics during workflow maintenance.

1. Task automation. Testing, compiling and building dominate automation efforts, followed by code quality analysis and version management. Some critical tasks receive less attention: security analysis, performance monitoring and compliance checking. This suggests untapped automation potential.

BENEVOL 2025: The 24th Belgium-Netherlands Software Evolution Workshop Enschede, 17-18 November 2025

hassan.onsoridelicheh@umons.ac.be (H. Onsori Delicheh)

0009-0005-7935-4147 (H. Onsori Delicheh)

2. Workflow creation. Workflows are primarily created by adapting existing ones or writing them from scratch. GitHub's starter templates see more limited adoption, indicating a gap between available resources and practitioner needs.
3. Non-functional priorities. Reliability emerges as paramount, followed by security and understandability. Testing and debugging represent major pain points, with practitioners describing a trial-and-error approach due to inadequate tool support.

G2: Understanding workflow reuse practices

This goal comprised **four** survey questions to understand the motivations behind reuse in workflows, the characteristics that influence the selection and usage of reusable workflow components, and the challenges faced when incorporating them into workflows.
1. Reuse mechanisms. Actions developed by others are most frequently used, while reusable workflows see more limited adoption. Copy-pasting remains prevalent: maintainers frequently copy from own workflows, driven by convenience and desire for control.
2. Barriers to adoption. Key barriers include the difficulty to discover suitable Actions, complexity concerns, and limited awareness of the built-in mechanisms of composite Actions and reusable workflows. Trust issues are minimal despite security being a top concern.
3. Action selection criteria. When choosing Actions, respondents prioritise reliability, documentation, maintenance and security. With respect to license compatibility, respondents are uncertain of how this applies in the context of workflows.
4. Dependency Issues. Nearly all respondents encountered issues with Actions, most commonly because of outdated versions, deprecation and breaking changes. These challenges reinforce preferences for copy-pasting over external dependencies.

Conclusion

Our survey results provide solid empirical foundations for understanding GitHub Actions adoption patterns. We can derive several implications and recommendations from the survey results:
Practitioners could expand automation beyond core CI/CD to include security analysis, monitoring, and compliance checking. This could adopt robust dependency management practices and security tools to address Action-related vulnerabilities.
GitHub itself should enhance workflow testing and debugging support, improve template discoverability and customisation, and strengthen documentation for underused reuse mechanisms. Better search and filtering in the Marketplace could address discoverability issues.
Researchers should investigate the barriers to security automation adoption, develop metrics for assessing Action quality, and study long-term impacts of copy-pasting practices on workflow maintainability.

Our findings contribute to the growing body of knowledge on modern CI/CD practices and provide actionable insights for improving workflow automation tools and practices in collaborative software development environments.

Acknowledgments. This research has been submitted to ACM Transactions on Software Engineering and Methodology (TOSEM) and is currently under review. It is supported by the Fonds de la Recherche Scientifique - FNRS under grant numbers T.0149.22, F.4515.23 and J.0147.24.

References

[1] G. Cardoen, T. Mens, A. Decan, A dataset of GitHub Actions workflow histories, in: Int'l Conf. Mining Software Repositories (MSR), ACM, 2024, pp. 677–681. doi:10.1145/3643991.3644867.