Understanding How GitHub Actions Workflows Fail and Recover

Aref Talebzadeh Bardsiri[1], Alexandre Decan[1,2] and Tom Mens[1]

[1]Software Engineering Lab, University of Mons, Belgium
[2]F.R.S.-FNRS Research Associate

This presentation abstract is based on two papers we published in 2026. The first one titled "On the github actions language: Usage, evolution, and workflow reliability" [1] and the second one titled "GHARuns: A large dataset of github actions workflow execution metadata" [2].

Continuous Integration (CI) and Continuous Delivery (CD) are essential practices in modern software development, enabling teams to deliver high-quality software rapidly and reliably. CI/CD workflows automate the process of building, testing, and deploying code changes, allowing developers to detect issues early and ensure that their software is always in a deployable state [3].

There are several CI/CD platforms, such as Jenkins, GitLab CI, and CircleCI, that provide such automation. Among these, GitHub Actions (hereafter shortened to GHA) has emerged as the most popular choice due to its seamless integration with GitHub repositories, and free service for public repositories [4]. GHA allows developers to design their CI/CD pipelines using YAML configuration files called workflow files, which define a series of jobs and steps that are executed in response to specific events, such as code pushes or pull requests. This flexibility enables developers to create custom workflows tailored to their project's needs, streamlining the development process and improving overall efficiency.

However, beyond adoption, researchers have highlighted that its use comes with multiple challenges. GHA workflows are among the most complex CI/CD configuration files [5], where failure is common [6] and practitioners have difficulties in understanding and writing workflow files [7].

These findings motivated us to study the reliability and maintainability of GHA workflows. In the first step, we conducted a study on the GitHub Actions language and its usage in practice, analysing the effect of workflow size and use of language constructs on workflow reliability and maintainability [1]. In that paper we found that the GHA provides a wide range of language constructs (**197 constructs**) and mapped them based on their purpose to **14 high-level features**. Construct and feature usage analysis revealed that there is a high inequality in the usage of language constructs and features, where a large number of constructs are used by a small number of workflows. We also did a limited study on the workflow executions and found that workflow size and the use of certain features such as shell commands are associated with higher failure rates and lower maintainability.

These observations motivated us to conduct an in-depth study on GHA workflow failures. To this end, we proposed GHARuns [2], a large-scale dataset of GHA workflow execution metadata, whose first version (2026-07-08) contains over 71M+ workflow runs, 122M+ job runs, and 1.3B+ step runs from nearly 46K popular and active GitHub repositories related to software development, covering the period from 2026-02-01 to 2026-06-27 (146 days). The dataset is publicly available for research purposes [8].

For the Benevol'26 workshop, we will primarily focus on presenting results from our previous study [1] and the GHARuns dataset [2]. Not limited to that, we will also present our ongoing research on analyzing the failure patterns and recovery of GHA workflows. We will discuss our preliminary findings. Figure 1 illustrates an example of a workflow run history containing two successful streaks (S2 and S4) and two failing streaks (S1 and S3). A successful streak is a contiguous sequence of successful workflow runs, bracketed by a failed run on each side, while a failing streak is a contiguous sequence

BENEVOL 2026: The 25th Belgium-Netherlands Software Evolution Workshop Madrid, 26-27 November 2026
aref.talebzadehbardsiri@umons.ac.be (A. Talebzadeh Bardsiri); alexandre.decan@umons.ac.be (A. Decan); tom.mens@umons.ac.be (T. Mens)
0009-0005-3719-9716 (A. Talebzadeh Bardsiri); 0000-0002-5824-5823 (A. Decan); 0000-0003-3636-5020 (T. Mens)
© 2026 This work is licensed under a "CC BY 4.0" license.

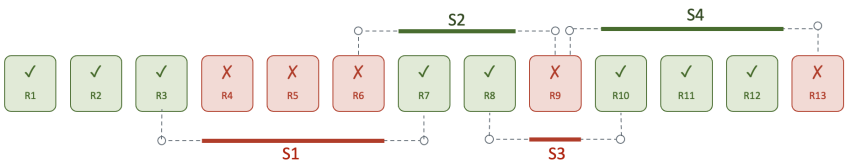


Figure 1: An example of a workflow run history and its streaks. R1 and R2 are left-truncated, since no failed run precedes them, and R13 is right-censored, since no successful run follows it.

of failed workflow runs, bracketed by a successful run on each side. By analyzing the patterns of these streaks, events that triggered or caused the failures, the recovery paths taken to resolve them, and the commits that caused or ended the streaks, we can gain a better understanding of what the life cycle of a GHA workflow looks like and what characterises the more reliable workflows. Among others, we consider the following three research questions: **(RQ1)** *How prevalent are failing streaks in GHA workflows, and how are they distributed across workflows?* We have observed that there is a high inequality in the distribution of the time spent in a failing streak across workflows (Gini: 0.88), while for the time spent in a successful streak, the distribution is more uniform (Gini: 0.37). **(RQ2)** *How long does a streak last, and what ends it?* Our preliminary results show that most failing streaks consist of a single failed run (median: 1 run), while successful streaks tend to be longer (median: 4 runs). **(RQ3)** *Does anything actually change when a workflow breaks or recovers?* Our preliminary results show that a third of the failing streaks end on the very same commit that had just failed, meaning that no change in the repository was needed for the workflow to recover. To further investigate what changed in the remaining cases, we have cloned the repositories to which those streaks belong, and we will compare the commits of the runs in the streaks. It is worth noting that we plan to submit this study to the MSR 2027 conference.

Acknowledgments. This research is supported by F.R.S.-FNRS research projects F.4515.23 and J.0147.24.

References

[1] A. Talebzadeh Bardsiri, A. Decan, T. Mens, On the github actions language: Usage, evolution, and workflow reliability, in: Proceedings of the 42nd International Conference on Software Maintenance and Evolution, F.R.S.-FNRS - Fonds de la Recherche Scientifique, Institute of Electrical and Electronics Engineers (IEEE), 14 September 2026. arXiv:https://orbi.umons.ac.be/handle/20.500.12907/56218.

[2] A. Talebzadeh Bardsiri, T. Mens, A. Decan, GHARuns: A large dataset of github actions workflow execution metadata, in: Int'l Conf. Software Maintenance and Evolution (ICSME), IEEE, 2026. URL: https://orbi.umons.ac.be/handle/20.500.12907/56424.

[3] F. Zampetti, S. Geremia, G. Bavota, M. Di Penta, CI/CD pipelines evolution and restructuring: A qualitative and quantitative study, in: Int'l Conf. Software Maintenance and Evolution (ICSME), 2021. doi:10.1109/ICSME52107.2021.00048.

[4] M. Golzadeh, A. Decan, T. Mens, On the rise and fall of CI services in GitHub, in: Int'l Conf. Software Analysis, Evolution and Reengineering, IEEE, 2022. doi:10.1109/SANER53432.2022.00084.

[5] T. Ghaleb, O. Abduljalil, S. Hassan, CI/CD configuration practices in open-source Android apps: An empirical study, ACM Trans. Softw. Eng. Methodol. (2025). doi:10.1145/3736758.

[6] L. Zheng, S. Li, X. Huang, J. Huang, B. Lin, J. Chen, J. Xuan, Why do GitHub Actions workflows fail? An empirical study, ACM Trans. Softw. Eng. Methodol. (2025). doi:10.1145/3749371.

[7] S. G. Saroar, M. Nayebi, Developers' perception of GitHub Actions: A survey analysis, in: Int'l Conf. Evaluation and Assessment in Software Engineering, 2023. doi:10.1145/3593434.3593475.

[8] A. Talebzadeh Bardsiri, T. Mens, A. Decan, GHARuns: A large dataset of GitHub Actions workflow execution metadata [dataset], Zenodo [Version 2026-07-08], 2026. URL: https://doi.org/10.5281/zenodo.20427454. doi:10.5281/zenodo.20427454.