# Empirical Study of Restarted and Flaky Builds on Travis CI

Thomas Durieux*[1]

[1]CASTOR — CASTOR - Software Research Centre - KTH – KTH Royal Institute of TechnologySE-100 44 StockholmSweden, Suède

**Résumé**

Continuous Integration (CI) is a development practice where developers frequently integrate code into a common codebase. After the code is integrated, the CI server runs a test suite and other tools to produce a set of reports (e.g., the output of linters and tests). If the result of a CI test run is unexpected, developers have the option to manually restart the build, re-running the same test suite on the same code; this can reveal build flakiness, if the restarted build outcome differs from the original build. In this study, we analyze restarted builds, flaky builds, and their impact on the development workflow. We observe that developers restart at least 1.72% of builds, amounting to 56,522 restarted builds in our Travis CI dataset. We observe that more mature and more complex projects are more likely to include restarted builds. The restarted builds are mostly builds that are initially failing due to a test, network problem, or a Travis CI limitations such as execution timeout. Finally, we observe that restarted builds have an impact on development workflow. Indeed, in 54.42% of the restarted builds, the developers analyze and restart a build within an hour of the initial build execution. This suggests that developers wait for CI results, interrupting their workflow to address the issue. Restarted builds also slow down the merging of pull requests by a factor of three, bringing median merging time from 16h to 48h.

*Intervenant