

# Software Analytics for Continuous Integration

Guilherme Ferreira  
Institute One  
Address One  
gferrei@ncsu.edu

Timothy Menzies  
Institute Two  
Address Two  
menzies@ncsu.edu

## ABSTRACT

By using continuous integration services, one can automate the process of building a system and making it run against existing test suites. However, as it can take a substantial amount of time for a system to be built and ran against test sets, it would be advantageous to know ahead of time whether a build is going to pass or fail. In this paper we show that it's possible to predict, with over 90% precision and accuracy, the status of a build using only commit data, such as commit churn and lines of code added. Moreover, we also show that can be achieved in a relative short amount of time, making just-in-time build prediction a feasible and desirable option. With those, it's possible for developers to alter their changes before triggering a build, thus increasing their chances of a successful build and consequently saving resources and time.

## Categories and Subject Descriptors

H.4 [Information Systems Applications]: Miscellaneous;  
D.2.8 [Software Engineering]: Metrics—*complexity measures, performance measures*

## General Terms

Theory

## Keywords

ACM proceedings, L<sup>A</sup>T<sub>E</sub>X, text tagging

## 1. INTRODUCTION

Continuous integration plays an important part of the entire agile development process[1].

Research Questions: 1: Is it possible to predict a build status using only static commit data?

RQ2? If yes, can we also do so using just the project data, without transfer learning?

## 2. RELATED WORK

## 3. METHODOLOGY

1. Dataset

2. Data preprocessing

3. Algorithms used

## 4. RESULTS

## 5. THREATS TO VALIDITY

Open source data

Citation of Einstein paper [1].

## 6. CONCLUSIONS AND FUTURE WORK

Future: Private data Discover what made the build fail

## 7. REFERENCES

- [1] A. Einstein. Zur Elektrodynamik bewegter Körper. (German) [On the electrodynamics of moving bodies]. *Annalen der Physik*, 322(10):891–921, 1905.