Testing and Continuous Integration at Scale: Limits, Costs, and Expectations.

Kim Herzig
Microsoft Corporation
Redmond, USA
kimh@micrsoft.com

ABSTRACT

Build and verification systems and processes have changed significantly in the past decade and so did the corresponding development processes. In fact, build and verification systems are a lower bound of how fast a company can ship new features—the more time spent to compile and test code the worse the company's ability to compete on the market. In other words, verification is key but needs to be affordable—in terms of money but also in terms of time. This keynote will be about some fundamental concepts of modern industry software testing, their cost and limits as well as expectations of software developers and teams against build, test, and development processes.

ACM Reference Format:
Kim Herzig. 2018. Testing and Continuous Integration at Scale: Limits, Costs, and Expectations.. In *SBST'18: SBST'18:IEEE/ACM 11th International Workshop on Search-Based Software Testing , May 28–29, 2018, Gothenburg, Sweden.* ACM, New York, NY, USA, 1 page. https://doi.org/10.1145/3194718.3194731

1 INTRODUCTION

Software testing is a key component of software development processes that companies rely on to deliver high quality products and features. However, time requirements to compile and verify code changes imposes a lower bound of how fast the company can ship code to customers. Thus, there is high pressure to drive build and verification durations down without lowering the quality of the systems under test as it impacts to marketâĂŤdelivering new features and being the company that shipped them first.

On the build engine and services side, there have been major advances and improvements in the past decade. To improve code velocity, e.g. time between first commit and code being shipped to customers, and to enable verification of an increasing number of code changes, build systems have changed significantly in the last decade. The times of running builds on a single (multi-CPU) machine are gone. Modern build and verification systems are cached and highly distributed services that provide built-in compile and test selection and operate across entire data centers using elastic machine pooling. While being a huge success and enabling software developers with more incremental code changes and being able to verify earlier and faster, it also changed the expectations developers have against build performance. Nowadays, every pull request or code review iteration is expected to run its own verification build, which ideally completes within minutes to minimizing context switches for engineers and to keep code velocity high.

This has two immediate consequences. First, build services are high capacity, low latency services operating at a very large scale (10k+ machines across different data centers and regions). And although being highly complex, every disruption in these build and verification services are high severity incidents that may potentially delay important feature releases or even critical bug fixes. Secondly, there is no room to lengthen (end-to-end) build times or to report false alarms as each false alarm could break a release pipeline and may cause severe disruptions to many involved parties. Consequently, any changes to these development pipelines or technologies must ensure to operate within clear boundaries: (a) reduce time without impacting quality, (b) improve quality without regressing build speed, (c) reduce operative cost without negatively impacting speed nor quality. Any approach that does not yield any of these expected returns of investments will not be accepted as operational feasible. This talk will cover some basic insights and principals of modern, cached, distributed build systems and the challenges these systems impose. The goal is to provide an overview of limits, costs, and expectations around build and verification services at Microsoft, which we know to be representative for many other industry companies. We will walk over some examples of common assumptions and provide some basic build and test cost figures and formulas to check if or when these assumptions hold. The talk will also provide an overview of high important issues we are working on, which may serve as motivation to work in these areas and to provide operational feasible solutions.

2 ABOUT THE AUTHOR

Kim Herzig is a Senior Software Engineering manager at Microsoft, Redmond, USA. He is leading the Analytics teams for the Tools for Software Engineers (TSE) team. TSE owns large parts of the Microsoft's engineering system, e.g. build, unit testing, integration testing, security and static analyses, etc. Kim and his team are responsible to gather, pipeline, and analyze TSE tool telemetry to give TSE engineers insights into their tool and services performance. His team also helps Microsoft internal (first party) customers to optimize their development processed and pipelines to gain optimal performance and high code velocity.

ACM ISBN 978-1-4503-5741-8/18/05…$15.00
https://doi.org/10.1145/3194718.3194731