Strategy for Continuous Testing in iDevOps

Extended Abstract

Peter Zimmerer
Siemens AG
Corporate Technology
Otto-Hahn-Ring 6
D-81739 Munich, Germany
peter.zimmerer@siemens.com

ABSTRACT

This tutorial presentation describes the state-of-the-art in the emerging area of continuous testing in a DevOps context. It specifies the building blocks of a strategy for continuous testing in industrial-grade DevOps projects (iDevOps) and shares our motivations, achievements, and experiences on our journey to transform testing into the iDevOps world.

CCS CONCEPTS

• Software and its engineering → Software creation and management → Software verification and validation → Software defect analysis → Software testing and debugging

• Software and its engineering → Software creation and management → Software development process management → Software development methods → Agile software development

• Software and its engineering → Software organization and properties → Software system structures → Software architectures

KEYWORDS

DevOps; continuous integration (CI); continuous delivery (CD); continuous testing (CT); test strategy; test architecture; mindset

1 INTRODUCTION

Continuous testing is the key for DevOps. But what is the right strategy, especially in industrial domains ("industrial-grade DevOps" – "iDevOps") that face many additional challenges compared to the ideal world of pure IT systems?

This tutorial presentation describes the state-of-the-art in the emerging area of continuous testing in a DevOps context. It specifies the building blocks of a strategy for continuous testing in industrial-grade DevOps projects (iDevOps) and shares our motivations, achievements, and experiences on our journey to transform testing into the iDevOps world.

Especially the mindset and culture is important for any DevOps transformation. But just arguing that "we need motivated people with the right mindset" is not enough; therefore many concrete examples of potential conflicts, misalignment, and new priorities that require a cultural change and different behavior for iDevOps are discussed and explained ("mindset by example").

Attend this presentation and do not only learn what industrial-grade DevOps (iDevOps) is all about and why we at Siemens are driving this forward but be able to use the strategies, tactics, and practices in your projects as a major lever for better testing. We are continuing our journey in this direction – that means a tremendous upgrade and empowerment of testing!

ICSE '18 Companion, May 27-June 3, 2018, Gothenburg, Sweden

ACM ISBN 978-1-4503-5663-3/18/05…$15.00
https://doi.org/10.1145/3183440.3183465

2 AGENDA

- What & Why?
  - Continuous testing defined
  - Specifics of industrial-grade DevOps (iDevOps)
  - Testing areas – What's new and different in continuous testing?
- How?
  - Prerequisites and preconditions for continuous testing
  - Mindset by example
  - Enablers for continuous testing
  - Practices, recommendations, and experiences

3 TECHNICAL BRIEFING SUMMARY

3.1 Definition

Continuous testing has the following two main characteristics and objectives (see figure 1):

- Testing over the whole lifecycle by shift left and shift right, i.e. testing *continuously* from begin to end
- Strategic test automation, i.e. *continuously* reuse and adapt and execute (only the) needed tests in an intelligent manner

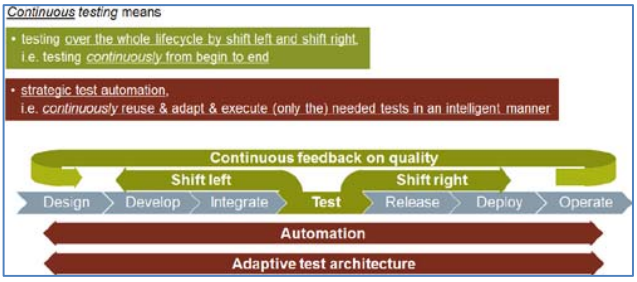

Figure 1: Definition of continuous testing

3.2 What's new and different?

Automated tests for functionality, integrity, consistency, performance, security, any other needed non-functional requirement, regression, etc. are integrated in the whole deployment pipeline: *automated tests gating code*. Quality gates are explicitly defined and efficiently implemented to get faster feedback from tests over the whole pipeline up to production.

A new test strategy results from new test objects (environment, infrastructure as code, deployment pipeline), new test levels and test types (shift left, shift right), new test and quality roles (e.g. DevOps Test Engineer), more virtualization and virtual assets, more API testing and service level testing, new technologies (containers, microservices, contract testing), testing in production (feature toggling, A/B testing, *monitoring is the new testing*), and real-time live dashboards with operational intelligence.

3.3 Prerequisites and preconditions

As a starting point for continuous testing the following practices have to be in place first: source code management, binary repository management, continuous integration, deployment pipeline with continuous deployment, gated check-in, and automated regression testing.

3.4 Mindset by example

Continuous testing requires a cultural change and different behavior. It is very important to understand potential conflicts, misalignment, and new priorities to proactively resolve and overcome these issues.

The discussed examples are related to the area of requirements engineering and features, continuous integration, test approach, test automation and respondence, test environment and test infrastructure, as well as to the area of business objectives and priorities.

3.5 Enablers

There are many different approaches, practices, and techniques that can enhance and foster continuous testing.

Agile, lean, and cross-functional empowered teams are a good basis. A value stream mapping of the deployment pipeline helps to analyze the current status of test activities within the value stream and to create a roadmap on how to manage constraints, eliminate (or even better: prevent) bottlenecks and improve testing within the deployment pipeline. Test-driven development approaches (TDD, ATDD, BDD, specification by example, etc.) are one major driver to make shift left a reality.

Design for testability supports the implementation of features in smaller batches, easy configurability and setup as well as testing in production without (negative) side effects to users. An adaptive and flexible test architecture leveraging the cloud, virtualization, and simulation is the foundation for an adaptive test- und simulation environment. Test automation is hooked up to the build (*automated tests gating code*) to do testing more frequently on smaller batches of changes. The test automation pyramid and the agile testing quadrants have to be enforced, extended, and renewed for (consumer-driven) contract testing, deployment testing, testing in production, and infrastructure testing.

Real-time dashboards for automated tests within the deployment pipeline enforce the collaboration between Dev and Ops and beyond.

3.6 Recommendations

Automated tests are your primary line of defense in your aim to release high quality software more frequently. Integrate testing into the deployment pipeline for all needed non-functional requirements and system qualities. For that distribute different classes of tests along the deployment pipeline and link them to the quality gates. Tests can be carried out against silent features in production to measure quality and impact of changes. Do regular infrastructure smoke tests and define automated deployment test suites to be reused after each deployment. Identify needed number of environments as a balance between flexibility/speed and complexity.

Manual exploratory testing is not made obsolete: it migrates towards the end of the pipeline, leaving computers to do as much work as possible before humans have to get involved. Consider running exploratory testing in parallel with other automated and non-automated tasks. Adopt a measurement-based approach and consider the following metrics and KPIs as a starting point: throughput of code (lead time and frequency) and stability (change failure rate and failure recovery time aka MTTR).

4 KEY TAKEAWAYS

- Understand why and how testing is changing dramatically in the iDevOps world
  - *A strategy for continuous testing is the heartbeat of iDevOps*
  - *Continuous Testing is the linchpin of automated CI and CD*
- Get to know the building blocks of a successful strategy for continuous testing in iDevOps
- Exploit the discussed examples to create and foster an adequate mindset and culture

REFERENCES

[1] P. Zimmerer. 2017. Strategy for Continuous Testing in iDevOps. Presentation at Software-QS-Tag, October 19-20, 2017, Frankfurt (Main), Germany. http://www.qs-tag.de/en/

[2] M. Shahin, M. Ali Babar, and L. Zhu. 2017. Continuous Integration, Delivery and Deployment: A Systematic Review on Approaches, Tools, Challenges and Practices", IEEE Access, 2017. https://arxiv.org/abs/1703.07019