

CONTINUOUS TESTING: TESTING EARLY AND TESTING OFTEN

In this fast-changing world, continuous delivery has become a critical concept for many organizations. However, despite embracing continuous integration and continuous deployment, organizations are facing issues in deploying quality software without compromising time and effort. Continuous testing helps address these gaps by automating testing and ensuring immediate identification of issues. This report sheds light on continuous testing, its implementation process and future.



Continuous testing: Testing early and testing often

As per a MarketWatch research, the global DevOps platform market is expected to reach 5360 million USD by the end of 2024, growing at a CAGR of 15.6% between 2019 and 2024. In a world where change is the only constant, industries are adopting DevOps in their attempt to efficiently navigate complex business scenarios. DevOps aims at rapid software delivery through continuous integration and delivery i.e continuous development.

However, in order to enjoy the full benefits of DevOps, companies must introduce continuous testing throughout their software development and delivery processes.

Continuous development process

Continuous development aims at automating and streamlining the process of building, testing and deploying new code into a live environment.

Following are the steps involved in a continuous development process.

1. Develop locally and push your changes to a branch
2. Continuous integration server picks up your changes, builds your code, and runs unit tests
3. Application is deployed to a staging or review environment at the 'delivery step'
4. Once deployed, automated API tests are run against the new version and the user is alerted of any test failures or degradation

in web service. Any failures in this step will prevent new changes from being deployed.

5. After validating the new changes on staging, the application can be deployed to production, after which tests are run again on the live API.

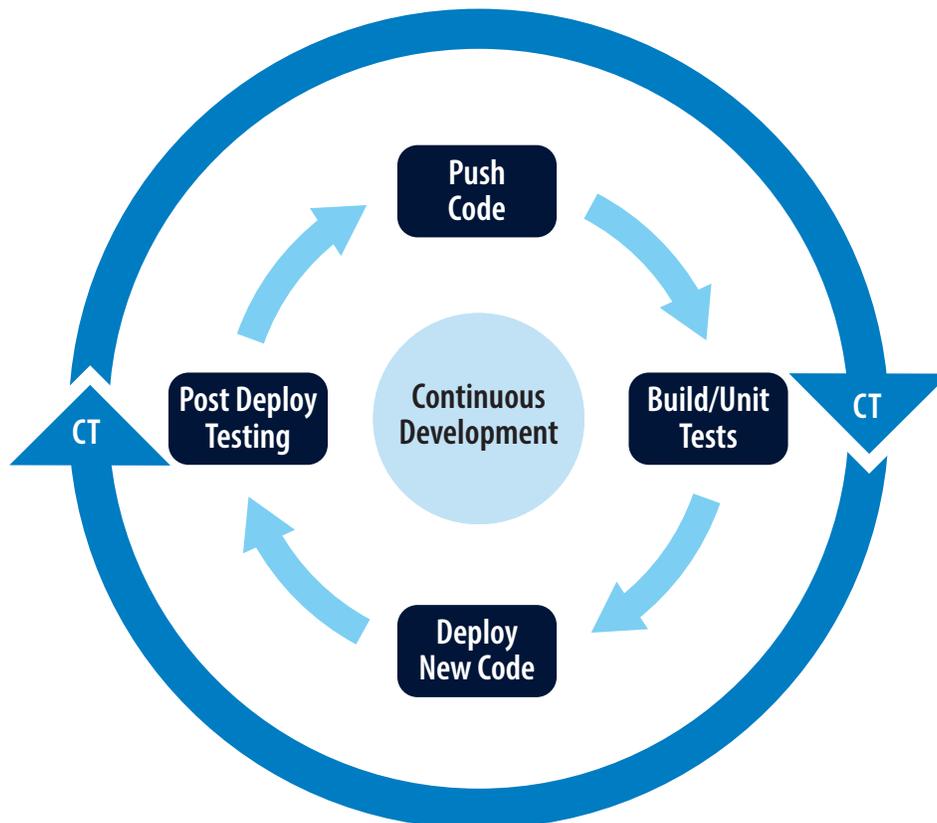
For a continuous development process to be successful, there is a need for early feedback and correction at each step, thereby making continuous testing an integral part of the continuous development process.

Continuous testing

Continuous Testing (CT) is the process of executing automated tests throughout the delivery pipeline giving a developer the earliest

possible feedback about the quality of their code, and a manager timely information about the risks associated with deploying that code. The primary goal of CT is to speed up the process of providing feedback.

Continuous testing helps validate and get immediate feedback on new changes or new versions of the software made; it is a culmination of many modern development methods such as Test Driven Development (TDD), extreme programming and other collaborative approaches. These tests hooked into continuous integration servers, build pipelines and run through an infinite loop of steps.



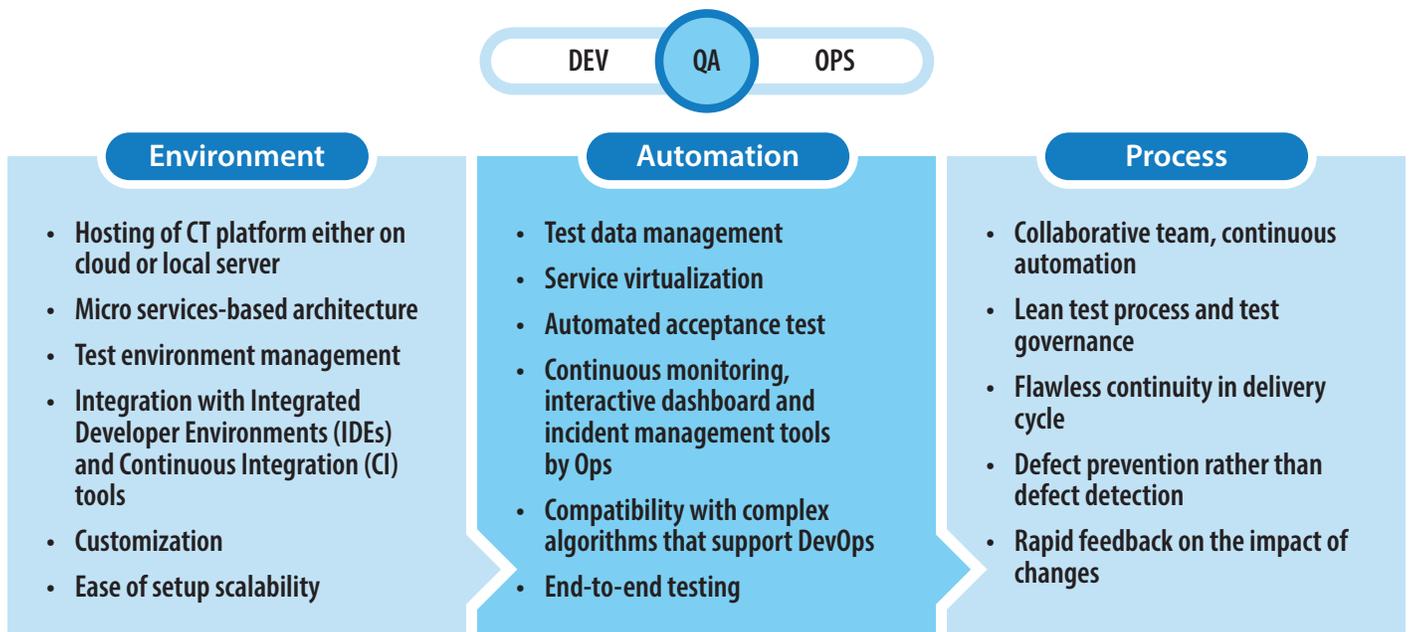
Implementing a continuous testing program

A successful continuous testing program requires good preparation of the environment it operates in. Following is an exhaustive list of requirements for continuous testing.

- Automated generation of test automation script from requirements
- Simulated test environment
- Accessibility to test data on demand and ensuring PII compliance
- Multi-layer/backend 'request and response testing' before UI testing
- Democratisation of performance testing
- Open-source testing tool integration
- Comprehensive cloud-based API testing
- Built-in automated application security testing
- Orchestration and automation of the pipeline to promote artifacts when a test passes
- Harness application insight across SDLC

Readiness of the testing environment, reduction of manual efforts and a change in organizational culture are prerequisites in creating a successful continuous testing program. The below diagram shows elements in respect to environment, automation efforts and processes to ensure a fool proof CT program.

Continuous testing is the process of executing series of automated tests, as part of the software delivery pipeline. The QA team acts as a glue between development and operations and as a key enabler for DevOps



A well-set continuous testing process ensures that the application is tested early and often. In order to ensure the readiness of continuous testing, organizations resort to multiple tools and technologies to:

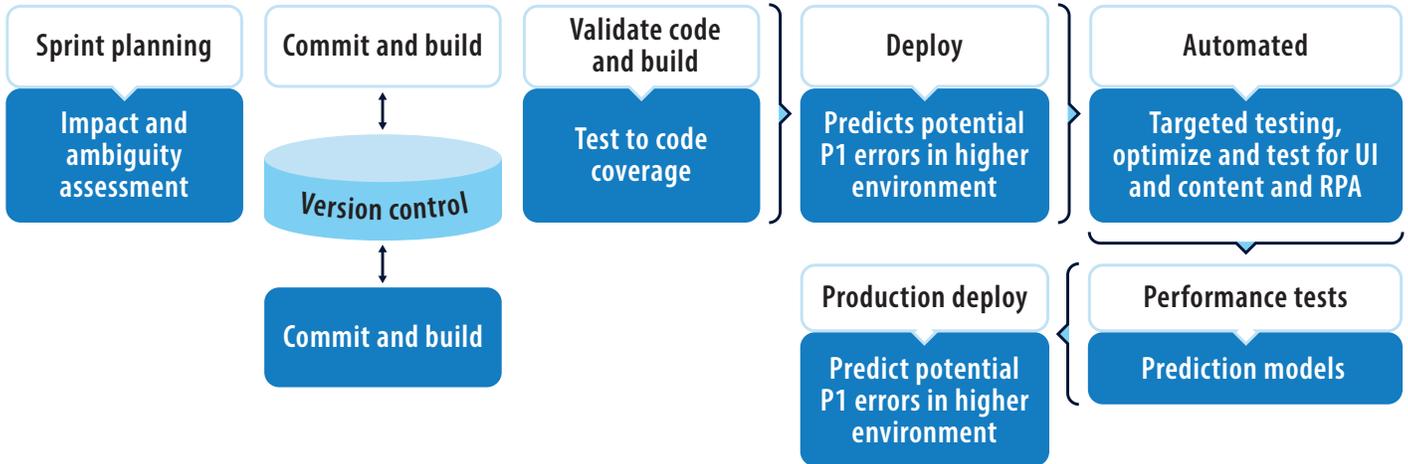
- Generate test data and test cases automatically
- Automate API functional and non-functional testing
- Integrate risk assessment and security testing
- Virtualize testing resources to eliminate bottlenecks
- Test and optimize performance early in development

Future of continuous testing

Today, developers are leveraging multiple technologies to improve the continuous testing process in terms of shorter throughput times,

better test coverage, manual testing elimination and so on. Today, the most frequently used technologies to improve continuous testing in

DevOps are AI and RPA, both of which ensure reliability and efficiency of the continuous testing process and contribute to successful DevOps.



1. Leveraging AI in DevOps continuous testing

AI increases reliability in software applications. Incorporating AI in testing tools enables users to build and execute personalized tests. Here is a thematic representation of how AI will transform continuous testing.

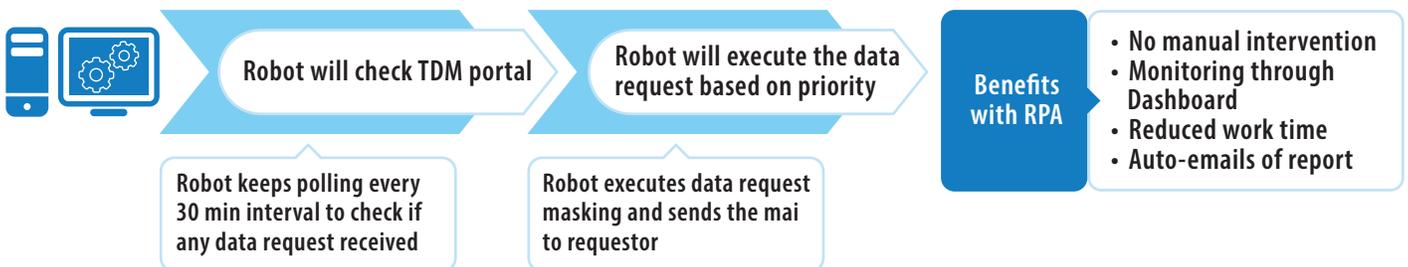
Following are the benefits of leveraging AI in DevOps continuous testing:

- Finding correlation in errors and defect.
- Continuously monitoring test runs to generate actionable insights for QA.
- Anomaly detection in response times and errors.
- Impact analysis to identify the impacted automation scripts for any code change.
- Application health analysis to identify and track application quality, responsiveness and automation script issues across builds.
- UI analysis to highlight user interface changes between test runs.
- Automation script optimization to eliminate redundancies across test scripts.
- Test coverage analysis to identify missing test paths.

2. Leveraging RPA to automate test data management

With DevOps focusing on continuous integration and delivery, it becomes essential that applications be tested every time they are delivered hence, emphasizing the need for Test Data Management (TDM). TDM helps in creating targeted test data of the right size without cloning the entire production environment. In order to ensure faster throughput and optimal coverage, it is highly advised to automate TDM.

Following diagram gives a glimpse as to how RPA can transform TDM.



Following are the benefits of automating test data management with RPA:

- Helps create stateful test data on demand for end-to-end testing
- Provides stateful test data management seamlessly integrated into test case design and execution
- Extraction and masking of data
- Self-service test data provisioning and management
- Increases testing efficiency
- Reduces skilled resource requirement
- Eliminates manual testing
- Reduces automation effort

Infosys capabilities

Today, most of the organizations deal with tight deadlines and ever-changing demands from informed customer segments. Organizations that are early adopters of agile methodologies and DevOps can help in automating testing needs, accomplishing delivery schedules, and bolstering agile development processes.

Infosys provides different automated solutions to support DevOps continuous testing. Here are a few Infosys offerings that enable DevOps continuous testing:

- **Cloud-based automated test environment** that is fully scalable and can be seamlessly integrated with any of the CI/CD tools that can share data through APIs.
- **Service virtualization** supports codeless UI recording and testing, synchronized user simulation, virtual service environment.
- **Test data management**, sensitive data discovery, data subset, data masking, data request and data extraction, synthetic data generation, data copy, gold copy creation, custom relationships builder, reporting etc.
- **Automated functional testing** includes shift left approach, **automated test script generation** using AI technique on manual test cases, support of TDD/BDD approaches, components, UI testing, and mobile testing.
- **Supports API testing, easy-to-use GUI** for rapid development of automated test cases in a codeless manner and execute functional and regression tests, shift left with functional automation, automation scripting even in the absence of APIs.
- **Load network performance testing** creates user load, distributed real networks, multiple test execution and executes in sequence, captures the results with network performance comparison graph and tests the application performance for distributed users under distributed networks.
- **Process automation** using RPA, QA process automation automates repetitive tasks performed by QA team. Bots can automate testing of each step and its expected action /output and testing of multiple systems is performed by QA team.
- **Leveraging AI capabilities** in DevOps, improved testing efficiency through focused QA of impacted areas, reduced testing cycle time.
- **Customer experience solutions** help evaluate all parameters affecting end customer experience and provides detailed recommendations to improve customer experience numeric scoring. This helps benchmark against past releases or competition, provides better visibility into end-user issues and application quality, and improves customer experience.



Conclusion

Organizations opting for DevOps welcome continuous testing at each stage of product development. This practice helps companies test early and test often, bringing in a cultural shift that fosters communication, collaboration and innovation. Continuous testing, when implemented diligently, allows flawless continuity in the delivery cycle. Integrating a fully automated continuous testing process into the SDLC is the most effective solution for a successful continuous delivery process.

References

Perrow, 2016, November, 10. A real world guide to continuous testing.

<https://techbeacon.com/real-world-guide-continuous-testing>

Infosys internal artifacts

Author

Shashikala J

Senior Associate Consultant – IVS-FS

Shashikala J is a Senior Associate Consultant with 11 years of experience and is currently working with IVS FS team in Infosys. In her ten years of experience in financial services domain, Shashikala has worked on trade finance, equity finance, Forex payments, and United States mortgage projects across six major banking clients.

To know more about the topic please contact icets@infosys.com

For more information, contact askus@infosys.com



© 2019 Infosys Limited, Bengaluru, India. All Rights Reserved. Infosys believes the information in this document is accurate as of its publication date; such information is subject to change without notice. Infosys acknowledges the proprietary rights of other companies to the trademarks, product names and such other intellectual property rights mentioned in this document. Except as expressly permitted, neither this documentation nor any part of it may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, printing, photocopying, recording or otherwise, without the prior permission of Infosys Limited and/ or any named intellectual property rights holders under this document.