

An Introduction to Testing in Production

by Antonella Manna - Thursday, August 18, 2016

<http://blog.belatrixsf.com/an-introduction-to-testing-in-production/>

In today's world, testing in production is not only important, but also critical to ensure complex applications work as we expect them to. Particularly with the move to devops and continuous delivery, testing in production has become a core part of testing software.

However, before determining if it's necessary and useful to conduct testing in production, let's do a brief review of the different environments in the software life cycle:

- **Development Environment:** where the software is created and has the services, development servers and databases.
- **Test Environment:** where the quality assurance (QA) team does all the testing tasks in order to detect and report bugs in the system.
- **Acceptance Environment:** where some final users have the chance to test the system in order to determine if it is fulfilling the requirements.
- **Pre-Production Environment:** the last environment before going live and where the hardware and all the characteristics are very similar to production.
- **Production:** where the application is deployed.

It can be expensive to maintain all these environments. But it's also important to compare the cost of maintaining the environments versus the cost of not detecting bugs on the environments prior to production. Here is a graph:

Meanwhile, this second graph shows data from back in 1996, but the fundamentals still hold true today. We can see the cost of fixing bugs over the different environments (the longer you wait to detect an issue, the more expensive it is to fix), and we see that most of the defects are detected during testing phases:

Limits of the Production Tests

Test environments have some specific characteristics and limitations, such as a smaller capacity, being different from the production environment, and a smaller data volume. These limitations means the results aren't always extendable to the production environment.

Testing in production meanwhile means you can put your software through the same rigors as it will face in the "real world".

When we design testing on production we should have in mind some key aspects:

- The testing data is real, so it should be encrypted
- The modified data is real so the ABM should be reflected on the system and databases
- Load injectors should act like real users

- The environments involve different technologies and it's expensive to have experts in all areas
- It is necessary to find a time with low impact (generally at night)
- It is necessary to have a data set for testing

Sometimes it is useful to create a map with the testing phases and the risks for each phase.

Production Testing Scope

With the limitations we've highlighted, it means it is still necessary to do performance testing over production environments when you are looking for any of the following: memory leaks, locks, resources used in the wrong way, etc. It also has some additional benefits:

- It's possible to analyze the behavior of real users
- It's possible to introduce fails on the real system in order to analyze the capacity of it to recover

There are some aspects which have to be present when production testing is implemented: the user impact, the system security, the responsibility to execute the tests, and avoid affecting other apps which are related to the app being tested.

It's important and useful to maintain the testing on QA and Production environments synchronized in order to compare the results. It's also important to have regular meetings between QA and developer teams, and have a person who's responsible for ensuring quality across the whole process and automatize all the operations where possible.

4 ways to test on production

Canary Servers

Here the name comes from the use of the canaries on the mines. Canaries were used because they are sensitive to dangerous gases and if the canary died or was struggling, then the miners knew that something was wrong and it was time to get out.

It is possible to add a server to act like a canary, in order to implement the test. If a lot of errors or wrong results start appearing, then we know that the new implementation will be problematic.

Third party tool

These are tools that allow you to extract data for the tests in order to replicate information on the real environments based on the tests. They also provide details about the errors detected.

A/B Tests

A/B tests are tests that change a few characteristics of the system and are presented to a few users in order to verify the performance and the interaction with real users.

Interrupters

These provide the chance to gradually build larger features in your code without slowing their release cycles. From a testing perspective, you can activate a full function and display it inside or outside. This allows you to run a series of tests to isolate the effect a particular characteristic is having on the environment. You can also turn it on or off for a subsection of its surroundings or of the user base, allowing more specific analysis.

An example of Production Testing with Azure Web Sites Tool

There are a lot of functionalities that are added to apps like Facebook or Twitter, and only a few users are able to test or see them. This technique is used by companies in order to test visual changes or new functionalities, and verify if users like them or not.

Azure web sites has a portal called Testing in Production that allows this behavior or define traffic percentage to send it to test environments.

To use Azure go to <https://portal.azure.com/> and look for the website: Browse > [Filter by] Websites.

What have been your experiences with testing in production? Let me know by leaving a comment below!