

THE ULTIMATE GUIDE TO CONTINUOUS TESTING

Everything you need to know to shift-left testing and reach testing maturity

CONTINUOUS TESTING: TODAY'S GOLDEN STANDARD

At Abstracta, we believe that Agile development, along with the practices that it promotes such as continuous integration, continuous testing and continuous delivery (CI/CT/CD), is the key to remaining competitive in today's technological landscape.

For an Agile environment to flourish in your organization, testing needs to happen earlier on in development than it does in traditional development environments like waterfall. We call this **"shift-left testing"** and it's imperative for Agile teams to truly succeed.

There are several software quality assurance activities to focus on that will help you in your efforts to reach an efficient continuous integration environment, allowing for the quality checks you want to have in each build.

In this guide, we will tackle the various areas in which we can group these activities so you may have a clear picture of what your team must work on in order to make progress in your testing maturity and, ultimately, reach continuous testing!

Let's Begin!

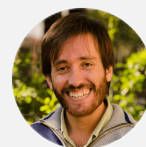


TABLE OF CONTENTS

- [Introduction](#)
- [The Source Code](#)
- [Environments and Infrastructure](#)
- [Bug and Incident Management](#)
- [Test Management](#)
- [Functional Testing](#)
- [Test Automation](#)
- [Performance Testing](#)
- [Security Testing](#)
- [Usability Testing](#)
- [Levels of Testing Maturity](#)
- [Conclusion](#)

ABOUT THE AUTHOR

Federico Toledo is a co-founder and director of Abstracta and holds a PhD in Computer Science from UCLM, Spain. With over 10 years of experience in quality engineering, he's helped many companies to successfully improve their application quality. He is dedicated to testing education, having written one of the first books in Spanish on testing and formed Abstracta Academy. He is also a co-organizer of TestingUY, the biggest testing conference in Latin America.



INTRODUCTION

"Agility basically facilitates competitiveness. To compete in today's environment, you must act and react fast, otherwise your competition will simply beat you to it. Today, the barrier to compete is minimal, and the only way to defend one's stature is by innovating in short iterations and basically meaning adopting Agile."

— Alon Girmonsky, CEO, BlazeMeter

When you shift testing left, among many other benefits, you can achieve **continuous testing**, executing automated checks as part of the software delivery pipeline to obtain immediate feedback on the business risks associated with a software release candidate. Continuous testing also involves early and frequent testing, proper code management, and adjusting the various quality-related tasks over time to maximize the efficiency of the whole process.

Some of the key benefits of continuous testing include:

- Lower the cost of development
- Reduce waste
- Reduce risk upon release
- Increase system reliability
- Release to production faster
- Compete more fiercely in the marketplace



Many teams today are trying to build or refine their continuous testing machine. We believe that in order to improve the results of software production, it's necessary to consider three fundamental pillars that are closely linked together: processes, tools and people. Teams can improve by bettering their processes, but the tools and the team members must also adapt in order for the new and improved processes to stick.



For example, within Agile, teams may find themselves in frameworks like Scrum that create a **process**, for which they will find themselves using various types of **tools** for everything from communication to task management and there will typically be great focus on the motivation and commitment of the **team**.

All aspects of Agile approaches are well designed to be adaptable to change, which is its main focus. The idea is to assume that what the customer needs is not fixed nor established in a contract. Thus, it is essential to work on constant adaptations in a way that does not cause the project costs to skyrocket or become unmanageable.

What is needed to adapt to change and yet maintain a high level of quality?

For any team, the most typical problem that arises when introducing changes in a system is fear: fear of breaking something that was already working before and not realizing it until after it has reached the customer's hands. When a user is given a new version of the application, there is nothing worse than finding that what always used to work no longer does.

THE ULTIMATE GUIDE TO CONTINUOUS TESTING

ENVIRONMENTS AND INFRASTRUCTURE / BUG AND INCIDENT MANAGEMENT

Whereupon, in the third execution cycle full coverage is reached:

	S5	Nexus 4	HTC One
Test 1	%	%	%
Test 2	%	%	%
Test 3	%	%	%

This does not guarantee 100% coverage in each run, but by managing to toggle the test runs, greater coverage is successively achieved.

This same strategy applies to web browsers, parameters, or many more variables with which testers have to “play”.

To achieve testing maturity, the necessary environments for test execution are required to ensure that there aren't any additional problems besides those that one hopes to uncover through testing.

BUG AND INCIDENT MANAGEMENT

Incident management is a basic point of efficiency within a development team. From how bugs are managed, it's easy to tell whether or not an organization's testers and developers feel as if they are a part of the same team, with the same goals. This assertion is not only aimed at developers, urging that they collaborate better with testers, but also at testers. Testers should avoid complaining (as they sometimes do) when a bug they reported doesn't get fixed and must understand that not everything needs to be fixed, depending on the team's global vision.

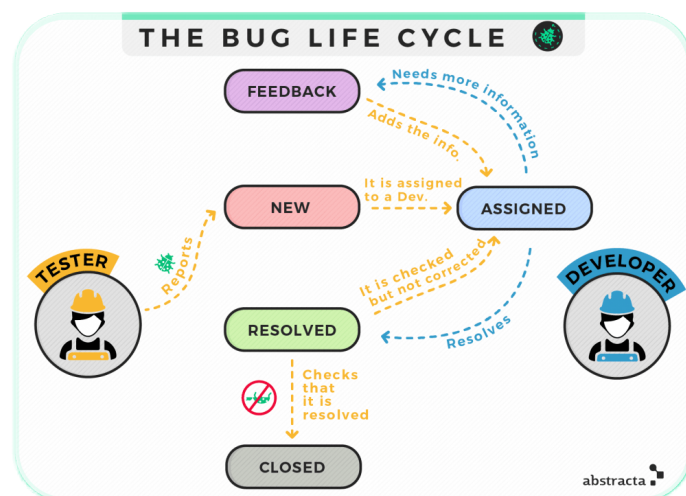
Of course, even when using an incident management tool, there will still be a need to speak face to face, but they do aid in adding clarification. For example, one can comment on and show directly, in the moment, what happened and thus communicate more effectively.

One poor incident management practice that we always find irksome at Abstracta is that with the hundreds of tools available for incident management, some teams still choose email as the designated channel for reporting bugs. Or worse, not using anything at all, and the tester simply taps the developer on the shoulder after finding a bug.

The following scheme summarizes a possible life cycle of a bug (there could be a thousand variations of this according to the team's structure, size, etc).

What's the problem with reporting bugs without recording them with an adequate tool?

Basically, it's impossible to follow up, keep a record, know the status of each incident (if it was already solved, verified, etc.), or even, if there's a team of testers, have it clear what things were already reported by another tester that shouldn't be reported again.



THE ULTIMATE GUIDE TO CONTINUOUS TESTING

TESTING MATURITY LEVELS / CONCLUSION

The model distinguishes three levels for each group of tasks, some being mandatory, recommended or optional. The lines demonstrate the relationship of dependency between them.

Thus, you can see that to have continuous integration, it is necessary to have code quality control, which raises the need to manage versions. Moreover, you should also have a set of automated checks at the unit, API, and GUI levels. But first, In order to have those tests, it is necessary to have separate test environments and to manage them properly.

That is what we, at Abstracta, consider minimal in order to have a good continuous integration strategy, and to reach the highest level of testing maturity.

CONCLUSION

Once you complete the necessary steps to shift left testing and go from “quality assuring” to quality engineering, your organization will reap the highly sought after benefits of being able to efficiently deliver smaller, more frequent releases, keeping up with customer demands, competitors, and market conditions.

The best part is that as more clients and users become satisfied with your software because it consistently delivers on every aspect of quality that matters the most to them, so too will your business feel satisfied, with its increased profits and production capacity.

Additional Resources

- Browse our [blog](#) for even more in-depth information related to each section of this guide
- Watch the [webinar recording](#): Learn How Shutterfly Employs Continuous Performance Tests for Winning Customer Experiences Build After Build
- Download our [ebook](#): A Complete Introduction to Functional Test Automation
- Read the [white paper](#): 10 Mistakes Companies Make When Outsourcing Software Testing
- Follow Abstracta on [Twitter](#), [Linkedin](#), and [Facebook](#)

Have questions or are looking for some assistance in your continuous testing efforts?

Contact us [here](#) or at hello@abstracta.us