# How Crowdtesting Fits With Software Testing Methodologies

As crowdsourced testing (crowdtesting) becomes an even more popular and robust tool in the software testing community, it's important to recognize the advantages that crowdtesting brings into the fold, across the wide spectrum of various software testing methodologies. While the waterfall and agile development practices become more and more prevalent in the programming world, it's critical to understand both the strengths and weaknesses of typical waterfall and agile testing methodologies, and how the inclusion of crowdtesting can plug any holes that might exist in the typical testing practices during the software development life cycle.

In this article, we'll take a closer look at what crowdtesting is, detail the qualities of a typical agile testing methodology, and then examine the additional and unique benefits that a combination of these methodologies can bring to any development process, so let's get started!

## When Testing and Humanity Collide

At the most fundamental level, crowdsourced testing, or crowdtesting for short, is a testing methodology that takes advantage of the efficiency and benefits of crowdsourcing for use during typical software testing processes. By utilizing a large pool of testers, the development team can better rely on a group of testers which a wide range of attributes, including:

* Numerous Locales & Languages
* Various Levels of Expertise
* Differing Devices & Hardware Configurations
* Assorted Demographics

Given this plethora of varied and unique attributes across the entire base of testers assigned to the project, the QA team can be confident that nearly every angle of attack is being thoroughly tested and retested across the massive spectrum of testers assigned to the project.

## Crowdtesting During Refactoring

In software development, the basic concept of factoring code is a method by which complex systems are broken down into smaller and simpler sub-systems, which are now invariably easier to develop and maintain. By extension, the process of refactoring is simply breaking down *existing* code into smaller chunks, which critically do not alter the final behavior of the code in question.

The process of refactoring is so common and critical within nearly every software development life cycle that any improvement that can be made to the process, whether small or large, is a huge boon to the development team, as well as to the project as a whole. When any system within a project can be dissected and broken down into smaller parts that are easier to understand by everyone involved, the project is always improved and made better.

It should, therefore, come as no surprise that implementation of a crowdtesting methodology into the refactoring process is one simple, yet powerful, step that can greatly improve the quality of the refactoring that is about to take place.

For example, if the codebase for a web project is due for a refactoring pass on the systems that handle user sessions (signing in/out, security, navigating, etc.), it can be incredibly beneficial to perform a crowdtesting pass on those systems *before* any refactoring takes place. More often than not, testers may find bugs or even point to improvements that would dramatically alter the end-result of the system and how it functions, saving a great deal of unnecessary or even wasted coding time during the subsequent refactoring procedures.

## The Human Touch

One of the key tenets of the agile software development methodology is the concept of iteration and incremental evolution of the project throughout the entire software development life cycle. As fresh ideas come to fruition and new systems are developed to bring those ideas to life, often the project will go through varying versions or prototypes, allowing the entire development team, testers, and even clients to experience the project via a hands-on approach.

Not only does this common iterative prototyping practice ensure that projects are of the highest possible quality throughout each stage of the software development life cycle, but it also allows the team to quickly recognize necessary changes and allows the project the flexibility to quickly adapt to these ever-changing needs.

An often overlooked benefit to this frequency of prototyping is the extremely powerful ability to allow thousands of testers, via crowdtesting, to poke and prod at this particular prototype release, examining it in detail for bugs or minor imperfections that in-house developers simply wouldn't notice or even think to look for. At the most fundamental level, utilizing crowdtesting for each prototype release during an agile development life cycle means that each minor version release is given its very own real-world beta testing period. By using the power of crowdsourced testing at every major step along the way, the project is certain to be that much more robust and durable when compared to using only in-house testing methodologies.

## Crowdtesting: A Key Piece in the Puzzle

While crowdtesting is an extremely powerful method of testing throughout the entire software development life cycle, it's critical to understand that crowdtesting is just one more powerful tool in the larger testing toolkit, and recognizing when and where to implement it is key. While unit testing and other developer-driven testing procedures almost always take place, no matter the size of the project, it's important to realize that crowdtesting is another piece of that testing puzzle, alongside unit testing, which should be implemented when and where it provides the most benefit.

While automated testing is a crucial component to the development process, crowdtesting allows for integration of *human testing* into the overall testing process. Unlike automated testing, crowdtesting allows for tests to be performed in the real world, by a wide range of people on a variety of devices and platforms, in environments and locales where people truly live. Even the best automated testing processes or services cannot emulate the power and sheer diversity that crowdtesting provides.

In short, crowdtesting provides feedback and results that are personalized, mirroring the experiences that actual users are likely to have with that product.

**META DESCRIPTION**

A detailed exploration of the benefits of crowdtesting among agile testing, waterfall testing, and other software testing methodologies.

**SOURCES**

* https://en.wikipedia.org/wiki/Crowdsourced\_testing
* https://www.testbirds.com/fileadmin/Whitepaper-Studies/Whitepaper\_Crowdtesting\_Overview\_EN.pdf