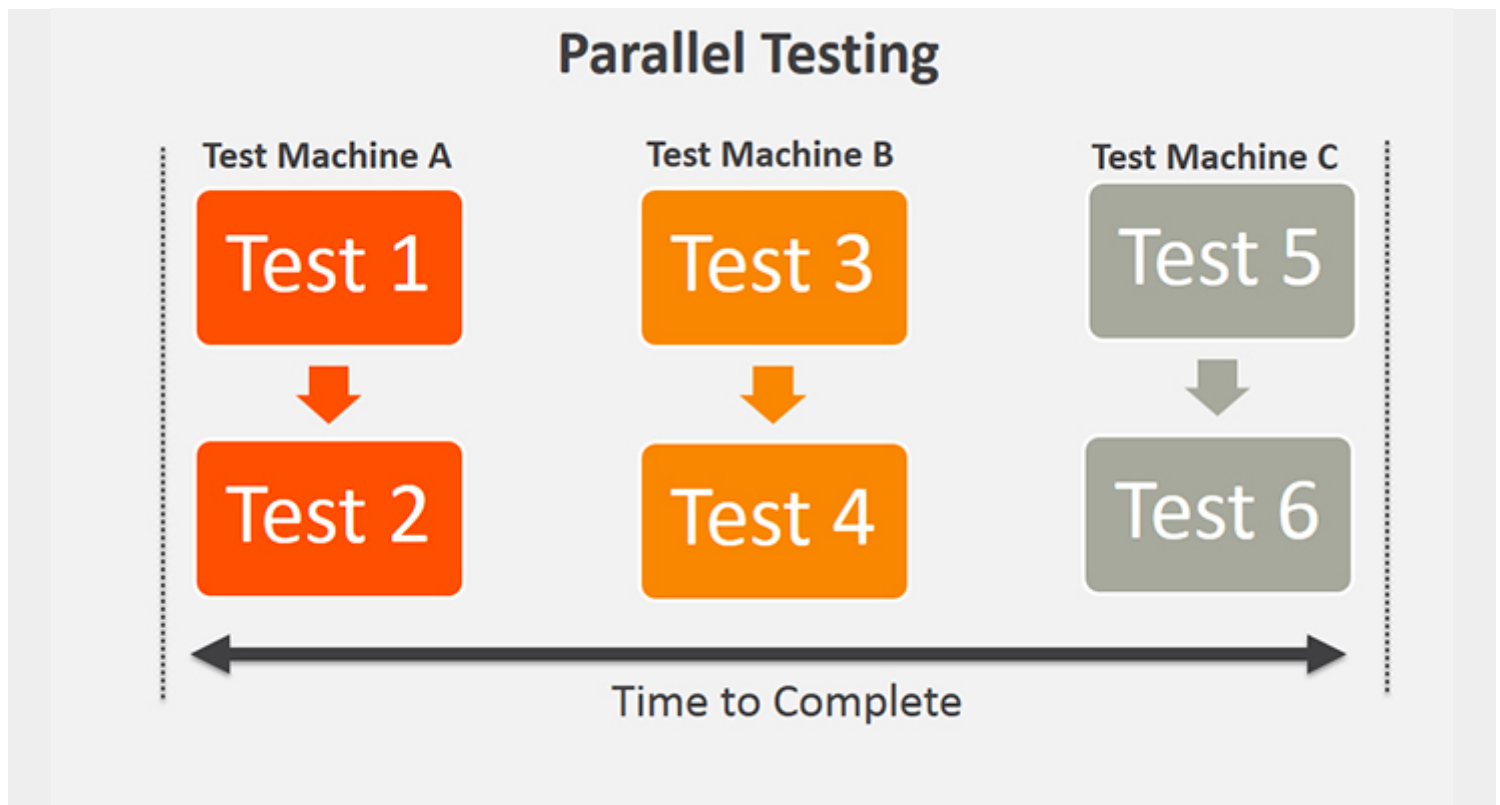


# WHAT IS PARALLEL TESTING? PARALLEL TESTING EXPLAINED



Software development demands that companies stay on top of current trends, innovate to predict future needs, and release software quickly and frequently. Delivered software must be useful, competitive, well-priced – and most importantly, the software must do what it sets out to do.

All of this relies on software testing. Testing is necessary, but the time that strong, sequential testing requires (and that the market demands) means a longer time-to-market, hurting success and profits. Parallel test is a new framework for testing software that radically reduces the cost and time associated with traditional testing methodologies.

## Brief History of Previous Testing Frameworks

Testing is an inherent software development process that has been used for decades as part of the Waterfall approach. In traditional test environments, testing is manual and sequential. Designated testers, anyone from actual developers to potential users, would start using beta software to find pieces that don't work, bugs, inefficiencies – anything the developers could use to review their work and attempt to solve them before software release.

## Sequential Testing



Problems often outnumbered solutions, and software went to market with built-in issues. Companies may have been aware of some, but may not have had the resources (time or money) to resolve them. For decades, manual testing has been the most accessible option, despite the time it takes – and manually-tested software can still contain errors, because it's not an automatic process.

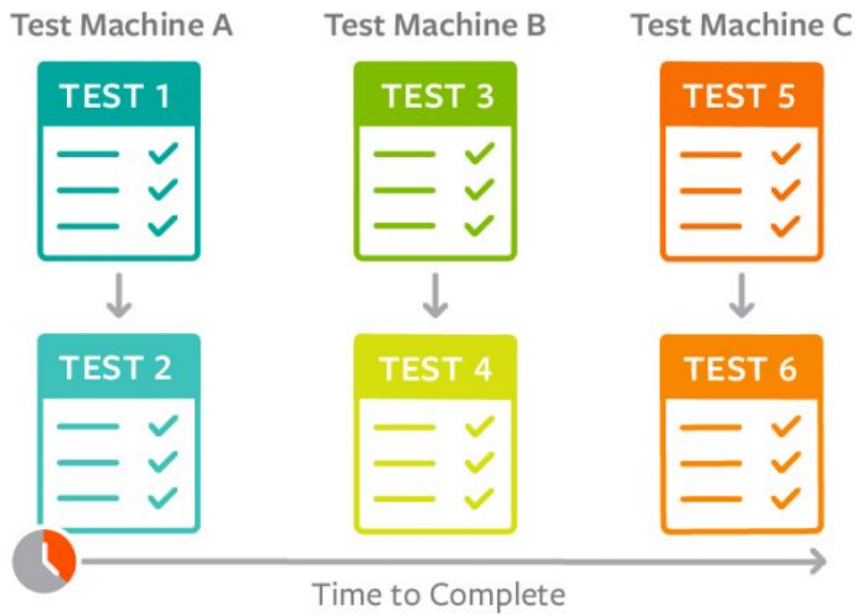
With the advent of agile software development and the separation of [Development Operations \(DevOps\)](#) from IT Operations (ITOps), software development has become more scalable and agile than ever. The extended application of agile and continuous integration principles to testing led to automated testing. Auto testing can reduce time-to-market, essential to successful software launches, without sacrificing quality. Still, automation is expensive and may not keep up with frequent application changes.

Fully embracing automatic testing results in slackness and reticence, particularly because of its high costs.

## Defining parallel testing

**Parallel testing** is a semi-automated testing process that relies on cloud technology and [virtualization](#) to perform tests against several configurations at the same time. The goal of this process is to resolve the limitations of time and budget while still assuring quality.

## Parallel Testing



The parallel testing process can apply to a single application or to components on multiple platforms. Each application or subcomponent, such as operating systems, browsers, and any combination of them, can be tested on a different computer.

While test computers can be individual hardware units, parallel testing more commonly refers to using separate virtual machines within the cloud. (A [virtual machine](#) is an emulation of a computer system, based on computer architecture, and it provides the same functionality of a physical computer.) Whether your testing machines are real or virtual, parallel testing runs more tests than would be practical with sequential testing.

The manual part of parallel testing requires a tester, who interacts with the application via a main “master” computer. The automated part is the test software that connects the master computer with N-controlled machines that run and test the various test environments.

## Parallel testing is not...

Understanding what parallel testing is, requires knowing what it is not. Parallel testing is a process, or framework, for testing, and it should be used as such. It is not a subset of software testing, only a tool within the larger testing toolkit. It also is not an add-on to your pre-existing testing regime. Neither is it a set of final spot-checks. Instead, the parallel testing framework should contain all (or close to all) of the required sequential tests.

## Benefits of parallel testing

Because parallel testing relies on running virtual machines in the cloud, invested testing time can be divided by the number of test machines – so testing time can easily be a fraction of what it is in traditional, sequential testing.

While parallel testing reduces the time and money invested in testing, there are plenty of other

benefits to using parallel testing.

- **Speed.** You only need the appropriate test scripts, you don't need tens or hundreds of extra hours or any more computers than you already use.
- **Affordability.** Leasing test time on a cloud service is much cheaper than building and maintaining an internal testing infrastructure.
- **Wide compatibility.** The limits of sequential testing often mean you can only test the most likely scenarios. Unlimited parallel testing is unlimited means you can test any and all combinations that are useful to users.
- **Continuous and concurrent timing.** The process can test functionality both frequently and rapidly. Developers can submit new code while testing is occurring, which supports and optimizes the widely-used methodologies of continuous integration (CI) and continuous delivery (CD).
- **Current and updated.** Relying on cloud technology means you're always using the latest versions of applications and software.
- **Results-driven.** Testing more scenarios in less time means there is more actionable data towards improvement.
- **Flexibility.** Revert to sequential testing when necessary.
- **Adaptability.** Transition to parallel testing in gradual steps.

## Best uses for parallel testing

Parallel testing has a [variety of testing uses](#). Some scenarios that may be the best fit include:

- Testing across browsers
- Testing compatibility
- Assessing versions
- Testing localization and internationalization
- Review language and validating context
- Ensuring performance
- Testing environment configurations

Product types that particularly benefit from parallel testing can include software with customized UI controls, end user-oriented products, and immature, quickly evolving products that rely on entering the market first to gain users.

Parallel testing is an extension of the logic that applies to all IT sectors. By focusing on practicality, ease, affordability, speed, and scale, parallel testing will likely radically transform testing, leading to continued improvements in software development.

*Original reference image:*

