# automated web testing

 Summary of different automated web testing options

## Recall

- What is automated web testing?
- What are some popular automated testing technologies?

---

Parasoft Continuous Quality Suite
QA Wolf
ZeuZ

# Overview

In today's highly competitive software development environment, automated **web testing**
tools are critical for creating a solid product while also enabling CI/DI integration, Agile, and DevOps methods to keep up with the constantly changing demand. And web test automation is at the heart of these features. Instead of focusing on shortening testing times, automated testing tools are now responsible for ensuring that all test cases are covered and used to their full potential.

> 📖 Web testing, or web application testing, is a software practice that ensures quality by testing that the functionality of a given web application is working as intended or as per the requirements.

Creating a top-notch web application requires a lot of testing which, if performed manually, can be tedious and time-consuming. For this reason, many QA teams rely on automation to create fast, efficient, and reliable test cases for their web applications.

> 📖 Test automation is the use of software (separate from the software under test) to control the execution of tests. It lets software robots perform repetitive tasks and emulate end-user interaction with the system under test, in order to increase the range, depth, and reliability of one’s quality assurance efforts.

Test automation offloads these routine and repetitive testing tasks from humans to machines. The tests compare actual outcomes with predicted outcomes. This approach can help find bugs in specific operations and simple-use cases (e.g. logging in, creating a new account, and doing password resets).

By automating web application tests, testers are able to save time and effort on monotonous tasks. Automated tests can be run continuously or scheduled at intervals.

This offloads testers from time-consuming tasks, and they can focus on exploratory testing or other tests that require a human perspective.

# What can you Automate?

## Functional Testing

Functional testing is used to ensure that the functionality of the software works as intended for an end-user.

A single end-user can make the whole system crash in minutes, even after unit, integration, and performance tests have passed. This usually happens because the user does something the developers did not expect.

Automating tests ensures that a web application is working as intended. However, functional test automation cannot fully replace the human perspective. That's why you should always supplement your automated test cases with exploratory testing to provide a positive user experience.

If you want to dig deeper into automated functional UI testing, you can get immediate access to this whitepaper on Functional UI testing: An introduction to codeless test automation.

## Regression Testing

Regression testing can be described as "repeated functional testing". It is used to make sure that a software's functionality continues to work after parts of it have been modified with new code or configurations. For instance, when new features are built, regression testing ensures that old features of the software continue to work as intended.

When regression testing is automated, you can quickly and reliably run through simple scenarios and check a variety of changes to get fast feedback. This, in turn, frees up time for testers to focus on product improvements or conduct manual exploration into more unusual cases that require special attention.

To become an expert in automated regression testing, download our whitepaper: How to do regression testing (faster) in agile teams.

## Cross-browser Testing

Cross-browser testing ensures that your web application is performing as expected across different browsers, both on desktop and mobile.

Browsers tend to be updated fairly frequently, meaning that by the time you are ready to deploy your application, it might not work as intended in your target browsers.

Users might have the latest browser version while using an old operating system. By automating cross-browser testing, you make sure that incompatibilities like the one just mentioned are found even before they reach an end-user.

Related reading: How to automate web testing across browsers and devices.

### Performance Testing

Performance testing, such as stress and load testing, ensures that a web application can endure extended periods of activity or peak user loads. Reaching the necessary stress conditions or load level wouldn't be feasible if done manually, therefore automation is key in proving that your application can perform in any situation.

Related reading: How to do application performance monitoring: best practices.

## Before Automating

Before you start automating your web application tests, make sure you draft a test automation strategy to keep you on track. Things to keep in mind before you start automation are:

1. What are the specific requirements of your web application?
2. What do you need to automate?
3. Which test automation tool best suits your requirements and goals?
4. How much maintenance will automation require?

Always start small and, once you're comfortable, start scaling your automation efforts. No one wants to end up with hundreds of automated test cases that are impossible to maintain.

Make sure you don't fall into the 'test automation paradox', where test automation was supposed to enable efficient software development and free up resources, but it actually brings with it an array of new costly tasks and never-ending maintenance.

Successful web application testing requires effective test automation processes, clear communication within the team, an efficient strategy, and an automation tool that doesn't impair testers, but enables them.

If you want to learn more about developing a test automation strategy for effective testing, you can access our guide to web automation.

---

To put together an effective test automation strategy for web applications you need to be really strategic.

Strategic about what parts of the process you should put on auto-pilot and where it would be best to test them, as well:

- don't rush in to test everything in the GUI (like record and reply, for instance); GUI testing tools do come with some significant test maintenance costs, you know
- whenever possible, opt for unit testing instead of GUI testing
- load testing, repetitive tasks, tasks that run on multiple platforms and configurations, tests that need to be "fueled' with multiple data sets… these are just some of the cases that you should automate.

# Popular Automated Web Testing Tools

## Selenium (Top Choice)

Selenium is a free open-source web testing tool. Selenium is the… *"Swiss knife"* type of automated web testing tool. It comes as a heavy package of libraries and tools. It's one of the most versatile automated software testing tools that you can get your hands on. The "de facto" standard in terms of open-source test automation tools, with a large community that you can rely on.

It can run in essentially every:

- Operating system
- Browser
- Automation testing framework
- Programming language

It enables you to write tests without having to understand the Selenium IDE by utilizing the record and playback features. Because of Selenium's excellent support, several of the top browser vendors use it as a native component of their browsers.

- Aids in the development of complex automation scripts.
- The basis for most other software testing tools.
- Parallel testing reduces testing time

And with versatility comes power, since Selenium empowers you to come up with particularly complex browser-centered test scripts that can be used for:

- Regression testing
- Exploratory testing
- Quick reproduction of bugs

### Cons of Selenium

- A certain maintenance overhead (maintaining your tests in Selenium is more expensive compared to unit testing, for instance)
- Building libraries and frameworks to meet specific testing needs does call for above-the-average programming skills (time and effort, as well)

## Watir

A powerful tool for automating tests. A Ruby library in fact, that mimics the behavior of a user interacting with your web app.

- it enables you to create tests that are easy to read and to maintain
- you can run it on your web app, irrespective of the languages that it is written on
- it supports data-driven testing
- from buttons to links, to forms and their responses, it’s built to test all the elements of your web application
- **you can leverage its powerful API handling to extend its capabilities**
- you get to connect it to databases, turn your code into reusable libraries, read data files, export XML

- you get to combine manual browsing with Ruby commands
- it supports cross-browser testing
- it integrates with business-driven development tools: Cucumber, RSpec, Test/Unit

### Cons of Watir

- a relatively small community supporting it (when you compare it to Selenium)
- you need to pair it with other tools to use it to its full potential

## Test Complete

**From cross-browser to regression, to parallel testing, it provides you with all the capabilities that you expect from a robust automation system.**

1500 +real test environments… That's the type of scalability that you get if you opt for this automated software testing.

It's one of those automation testing tools for web applications that you get to use on your mobile and desktop apps, as well. For pretty much any type of automation task that you can think of.

### Cons of Test Complete

Some users have reported errors with object recognition during playback. So, you might want to keep that in mind.

## Test Project

TestProject is a cloud-based open-source automated web testing tool. It allows you to test iOS, Android, and web applications quickly and easily across a variety of platforms. You can even collaborate with teams using Selenium and Appium to ensure efficiency while also maximizing speed. Create coded tests using TestProject's sophisticated SDK or take advantage of the advanced built-in recording capabilities. You can get all of this for free!

- Simple configurations or setups
- Provides a no-code experience
- Addons can be shared and reused within your team.

- Dashboards for detailed reports
- Workflow integrations for CI/CD

## Katalon Studio

An automated web testing tool for web and mobile app development, it's available for both platforms. Appium is built on top of Selenium and Appium to make API and web testing easier. It's compatible with a wide range of other applications, including JIRA, Kobiton, Git, and Slack.

- Conveniently accessible to any type of tester, those with no programming background included
- Ships with a whole set of powerful features
- It is versatile because it is compatible with Windows, macOS, and Linux systems
- With hundreds of pre-programmed keywords, you can quickly and easily create test cases

Some features include:

- customizable execution workflow
- support for image-based testing
- smooth integration with a variety of tools (e.g. TeamCity and Jenkins)
- built-in support for generating test scripts, creating test cases, reporting results, recording actions
- built-in object repository, object re-identification, Xpath
- built-in support for Groovy/Java scripting languages
- visual representation of each step in the test (it’s one of the most beginner tester-friendly tools out there)

## Eggplant

Eggplant, now owned by Keysight Technologies, has been acknowledged for excellence as an automated web testing tool. It is widely popular and for over a decade has been used by all the major analysts, including industry leaders such as Gartner and Forrester.

- AI-driven automation that sees technology through consumers' eyes
- User engagement increases by making apps responsive and versatile.
- Streamline test operations and increase testing efficiency.
- Automate mission-critical systems.
- Seamless integration with existing tools

# Cucumber

Cucumber is a free online Behavior Driven Development (BDD) tool. It supports a wide range of languages and is used by major platforms such as PayPal and Canon. Although, Cucumber mainly supports the testing of web environments. It was created to provide users with a better end-user experience.

- Run code on Selenium or other frameworks.
- The test code is written in Gherkin.
- it enables you to reuse code in your tests
- it supports lots of different languages Groovy, Python, Perl, PHP, .NET, Scala
- it grants you great support, since it’s a highly popular automated app testing tool, with a large community
- it enables you to use code along with Watir, Selenium, Capybara
- it’s a cross-platform tool
- quick set up
- it enables you to generate detailed reports
- it integrates with GIT and Jenkins

You can use it to:

- write acceptance tests for your web apps
- perform those tests by running the most representative examples for your app

## Cons of Cucumber

- you might find the default organization a bit… overwhelming

- you depend on external plugins for generating reports, so do expect some integration challenges

## HPE Unified Functional Testing

HPE Unified Functional Testing is a robust cross-platform automated testing tool. It provides extensive automated web testing capabilities as well as the ability for developers to collaborate on the same projects.

It can automate SAP and Java programs, mobile and Visual Basic apps, and other programs in addition to web and desktop applications. For functional and regression testing, it uses a local GUI or a web-based interface.

- The scripting language is VBScript.
- Data-driven testing
- Compatibility across browsers and platforms.

## IBM Rational Functional Tester

Rational Functional Tester (RFT) is IBM's paid automated web testing tool. The primary goals of this program are automated functional testing, regression testing, GUI testing, and data-driven testing. This testing tool supports a wide range of development environments, including Siebel, SAP, Java, Flex, and Dojo, to name a few.

Using RFT's Storyboard Testing, the test script is documented as a sequence of screenshots of the activities taken against the App Under Test. The script can be improved by adjusting the screenshots, which the tester can do during the testing process.

- Tests can be visually edited using natural language and application screenshots.
- Includes Java and VB.Net scripting languages.
- Integrated with Rational Quality Manager.

## Parasoft Continuous Quality Suite

The automated testing tool from Parasoft incorporates quality into the software delivery process to find, diagnose, and fix issues as soon as possible. In addition to static code analysis and unit testing, it includes API testing and service virtualization as well as UI

testing, allowing for rapid deployment while still meeting commercial and security requirements.

- Creating and running AI tests
- Testing low-code apps
- Intuitive dashboard for enhanced reporting
- 120+ messaging formats and protocols
- Pipeline and Agile DevOps integration

## QA Wolf

QA Wolf generates Javascript code while you browse the internet. The same Javascript is then converted into an automated software test. It is ideal for newcomers who do not want to waste time learning difficult programming languages. To debug for more complex testing, developers can easily edit any portion of the test code.

- No need to download or set up anything
- Tests are carried out in complete parallel to obtain quick findings.
- Supports most used browsers: Google Chrome, Mozilla Firefox, and Chromium.
- Execute the plan based on the premise
- Use OpenVPN to connect to a private network.
- Videos and failure records assist you in swiftly identifying problems.
- Vercel and Netlify integrations

## ZeuZ

ZeuZ is an AI-driven automated web testing tool for click-and-test automation. With no need for scripts, manual testers will find it easy to use, while professionals will appreciate its robustness and all-in-one functionality. ZeuZ fulfills the promise of real end-to-end web test automation by integrating CI/CD, intelligent troubleshooting, extensive reporting, and collaborative capabilities.

- Web, mobile, desktop, API, IoT, cloud services
- With or without code, a single UI spanning all technologies in one test case

- All-in-one: plan, execute and manage manual and automated tests
- Recording test scripts, AI-powered object detection, and built-in waiting
- Rich reporting, notifications, collaboration, and batch updates
- A single computer or hundreds of VMs, on-prem or cloud