



# Automate Me if You Can: Latest Trends in Test Automation

By Pavlo Vedilin

# Automate Me if You Can: Latest Trends in Test Automation

Originally posted by [Virtual-Strategy Magazine](#)

Even though the influence of digitization is nothing new, the all-present term of [Digital Transformation](#) has been getting more and more attention in almost every corner of business and everyday life. Where there is the word “digital,” there is an ever-growing demand for software development. Trying to keep your hand on the pulse of this allegedly vague term and push it even further should be your burning ambition.

As the call for software services is increasing over time, the approach to them doesn't remain standing, having moved from desktop to web and now to the cloud, mobile, IoT, VR, etc. To quote a well-known comic book, with great power comes great responsibility: with the advancement of technologies and platforms to be supported, quality requirements are also building up, causing the amount of [QA work](#) to increase flat-out. That is when test automation became a buzzword, firmly holding the fort up until now. It has never been standing still, but rather rapidly developing to fulfill a pressing industry need in tools, instruments and approaches to reduce time-to-market and related expenses without compromising quality of the product. Let's take a sneak peek on [test automation changes and challenges](#), and how to get your company shaped up for them.

## Agile Is the Key

The first thing to mention is that industry keeps relying on agile methodologies when it comes to software development. In short iterations and continuous delivery (CD) environment of agile projects, test automation is almost invaluable for covering regression testing and meeting tight deadlines. According to [Capgemini's research](#), a number of [agile projects](#) increased by 36 percent compared to the previous year, while the percentage of automated tests increased by 45 percent as opposed to 28 percent in 2014. What's more, to meet these deadlines, QAs have to cooperate with Devs hand-in-hand, which blurs the line between

the two roles: QAs are developing test automation items and CI parts, while developers are commonly using a TDD approach and are often involved in performance testing, making the app automatable and other testing-related duties. There is a bit of Dev in every QA, and vice versa.

## DevOps: The Best of Both Worlds

The popularity of agile also encouraged a gradual spread of DevOps, the golden term meaning combining the best of development and operations worlds. DevOps engineers help developers and QAs to focus on coding and testing, rather than setting up and supporting application infrastructure and various test and dev environments. Based on numerous cases from my own experience, DevOps is THE thing and has occupied a well-deserved role of a buzzword too, along with test automation and Agile.

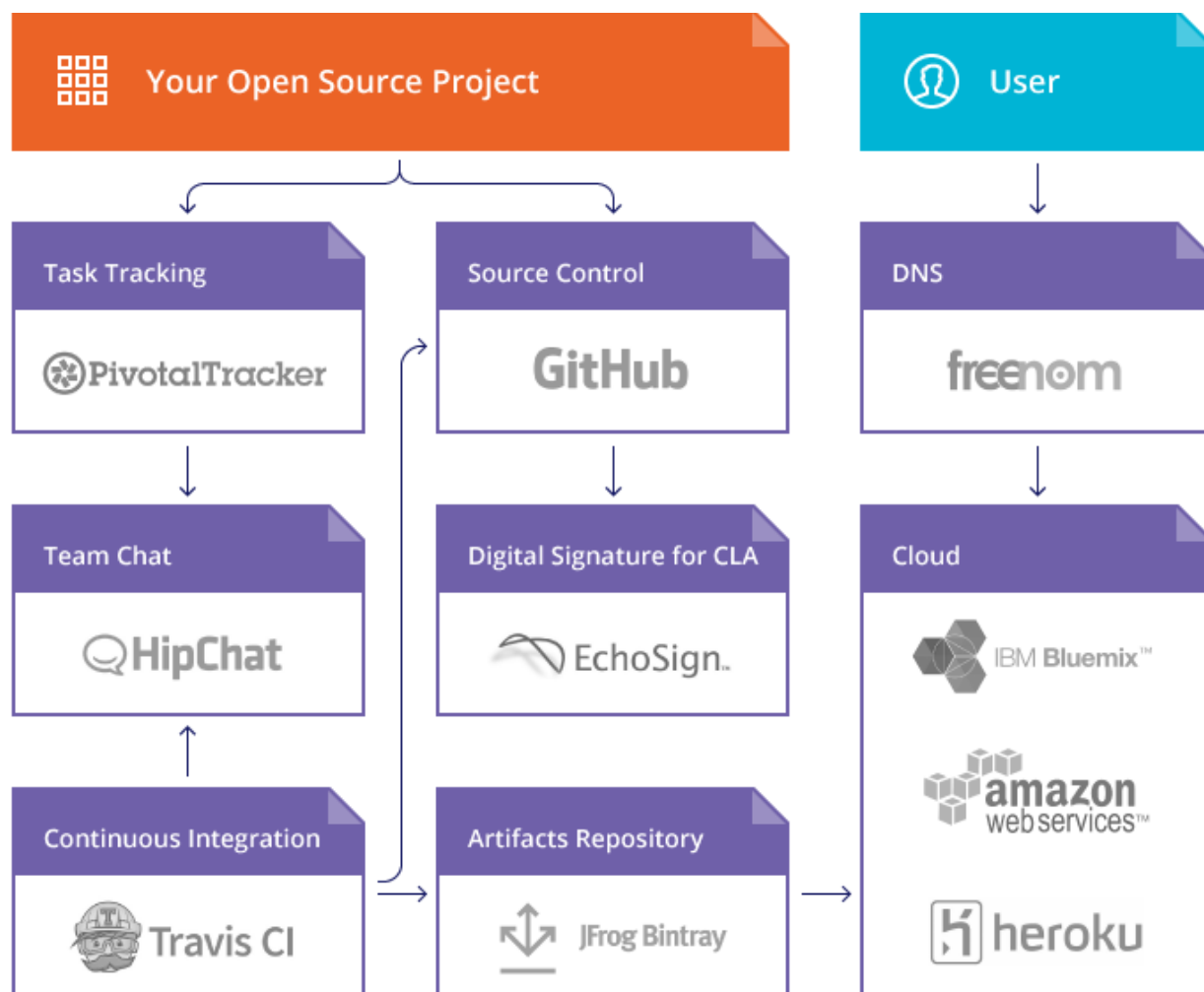
DevOps engineers themselves are knee-deep in automation as well, in a [slightly different way](#) though. What makes automation one of the chief ingredients in cloud services (read more below) is dynamic control panels and markedly adaptable configurations. Bringing up these automation layers via additional services and APIs, cloud providers enable businesses to stage, test and deploy new software automatically from CD tools that put the software pipeline together. This makes QAs' and Devs' lives easier.

## Keep Your Head in the Cloud

Speaking of test and dev, one should mention what these environments rely on: virtualization and cloud platforms such as AWS. By preparing needed virtual images in advance, time spent for setting up these environments may be significantly reduced and if something goes wrong, it is always possible to roll back to the initial image state. In addition, DevOps engineers can automate this process so your environment is ready for test runs with a click of the button (all right, all right, and maybe a few minutes of waiting). Add here some scripts for code compilation and deployment and you will get a lion's share of your CI process ready. Neat.

## Open Your Source

When the IT industry realized that the cost of software is high, an [open source](#) started playing the first fiddle in product development. The most popular modern software tools are open source. And even though test automation tools are far from being exclusively open source, the tendency is clearly changing with its focus shifting from vendor tools to open source tools paired with infrastructure provider services, such as Sauce Labs, BlazeMeter, etc. Here's a guideline for beginners to get armed with free services and set foot in open source:



## Security Testing Is Your Best Friend

Without having to say it, integrating security into an entire software development life cycle (SDLC)—from A to Z (where T obviously stands for Testing)—will prevent you from the undesired post-breach remorse. Arranging one security checkup is good, but having a thought-of security testing plan is even better. And that, kids, is how I met continuous integration and dynamic scan: launching code analysis tools along with automation testing (i.e., implementing them into a CI process) before a release will save you time, money and nerve cells. The goal is to achieve the following pipeline: CI performs code analysis, compiles and tests an application [on a daily basis](#) (the more, the better); then every change that passes the automated tests is deployed to production automatically. This allows dev and test teams to dig up issues bright and early, avoid human errors and detect bugs right away.

## May JavaScript Be with You

A due pinch of recognition should be laid upon JavaScript test automation tools and frameworks. As the demand for test automation is snowballing, more and more QA engineers are looking into writing tests with JavaScript for a number of solid reasons: not only is it easy to be learned, but it also has a huge community, which means there is always someone to offer a helping hand if you are stuck. This enthusiasm is mutual: JavaScript developers are also inquiring into test automation, which results in a much-needed boost in the amount of JavaScript test automation tools and libraries. Most of these tools are not as mature as the similar ones used with Java (as an example), but the scope of attention and a tempo of their growth showcases that JavaScript test automation will soon become a big thing.

## Mobile Is Still the New Black

Since the early days of a single ruling desktop have gone with the wind, the road of enterprise mobility and mobile applications leads to more testing, which in turn triggers QA engineers to look for tools to automate their work. Having realized it, the majority of infrastructure services are now offering cloud mobile devices for testing and automation. Fun fact: [Appium](#), a very popular test automation framework for use with native, hybrid and mobile web apps, is [open source](#) and is built around JavaScript stack.

## Web Services Testing Is In-Thing

Another area of test automation that is capturing attention because of the industry-related trends is web services testing. When a web application is growing on different components within different distributed teams, it becomes difficult to manage monolithic applications and keep high quality. The solution appears to be simple: split a monolithic application into microservices with completely isolated codebases which can be deployed independently. So, many web applications are adopting microservices architecture which means that a web application should be built as a sum of small independent services. This pattern changes the approach of testing such applications toward testing individual services and their integrations with each other. As microservices architecture remains a go-go when it comes to web applications, a call for this sort of testing will spring up.

## Bottom Line

Legit test automation performance is no longer just an increased amount of script code. Pushed by today's software defined world, the nucleus of test automation is moving away from rather conservative regression and smoke-type coverage to nonfunctional requirements. The scope of automation goes up substituting routine testing more and more, which lets engineers focus on context-bound issues.

By [Pavlo Vedilin](#)

# About SoftServe

With over 20 years of experience in digital software development and consulting, SoftServe is a global leader in solving complex business problems, creating industry disrupting technology and accelerating growth and innovation while optimizing operational efficiency. From leading ISV to Fortune 1000 digital enterprises, SoftServe has transformed the way thousands of clients do business with the most innovative technologies and processes in Big Data, Internet of Things (IoT), DevOps, digital commerce, security, and experience design.

Visit [www.softserveinc.com](http://www.softserveinc.com) to learn more.

## **USA HQ**

Toll Free: 866-687-3588

Tel: +1-512-516-8880

## **Ukraine HQ**

Tel: +380-32-240-9090

## **Bulgaria**

Tel: +359-2-902-3760

## **Germany**

Tel: +49-69-2602-5857

## **Poland**

Tel: +48-71-382-2800

## **UK**

Tel: +44-207-544-8414

[info@softserveinc.com](mailto:info@softserveinc.com)

[www.softserveinc.com](http://www.softserveinc.com)