AVO™
Automation

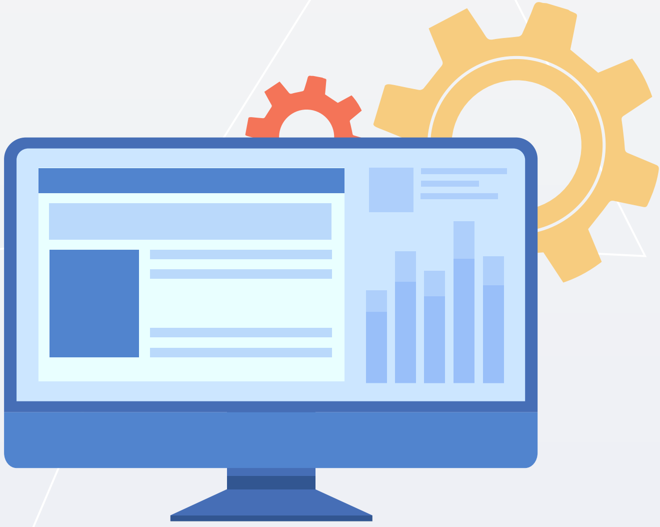
The Ultimate Guide to Automated Regression Testing

Table of Contents

What is regression testing, and why is it important?

Organizations today rely on agility and quality to stay ahead of the curve. Failing to deliver either at the right time could cost heavily. The post-pandemic ecosystem and subsequent economic crisis have made this even more prominent.

To ensure these demands are met, the development and QA teams are constantly pressured to deliver high-quality software faster. Regression testing is one of the primary pillars of the QA team in achieving this.

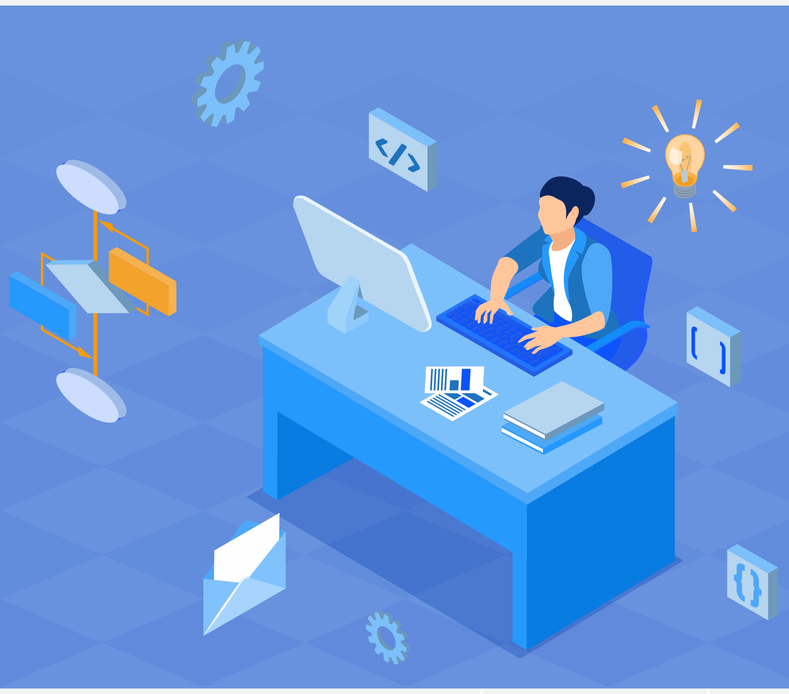


To give you context: have you ever been in a situation when an existing feature in your application stopped working after implementing a new one? Or a bug that you fixed a few months ago makes a grand return after incorporating the new feature? These are the perfect examples of regression – the act of returning to a previous inferior state. Regression testing ensures you don't end up in such situations.

Regression testing is the type of testing where you ensure that the new code changes don't affect the existing features adversely. At the same time, the software continues to perform as intended. It involves running a series of tests in a particular order to ensure the current functionalities are intact.

Ideally, you should perform regression testing every time you add a new feature, enhancement, or fix a bug.

Why should you automate regression testing?



More often than not, regression testing comes across as the testing that makes you go around in circles; because you have to run tests all over again for even a minor code change. Especially when done manually, it can be tedious, time-consuming, and at times boring – the last thing you want to put your team through.

From a software perspective, as the system grows, the number of test cases and the size of the regression suite increases. Eventually, manual execution of regression tests becomes a nightmare as it is practically impossible.

Adoption of agile is also another reason. When release cycles get shorter, regression testing creates a bottleneck, resulting in delays, technical debt, and a higher number of production defects.

This is where automated regression testing comes into play. The repetitive and predictable nature of the tests makes them perfect candidates for automation.

Especially with the pandemic in the picture, test automation is more of a friend than a foe, and adopting it gives you a ticket to high business performance.

Apart from solving the core problems of manual testing, and automated regression testing offers you the following benefits.

Advantages of Automated Regression Testing

Improves feedback cycle:

The feedback loop between development and QA teams determines your product's quality and timely delivery. Given that the release cycles in agile are short, a swift feedback loop is crucial. An automated regression testing framework makes it possible. Any possible defects that you wouldn't otherwise test for come to the fore.

Improves consistency and speed:

Since an automation solution takes care of validating your regression tests, human errors are drastically reduced. This enables you to deliver consistent and accurate results each time. Automation of regression test cases can be cost effective since it saves your effort and cost on bug-fixing.

Frees up your employees:

Regression testing larger applications requires significant effort. When you automate this, you liberate your employees of the mundane tasks and empower them to focus on judgment-intensive jobs. The cherry on top – when the test automation solution is no-code, you can test the applications with a few clicks of buttons.

Allows testing round the clock:

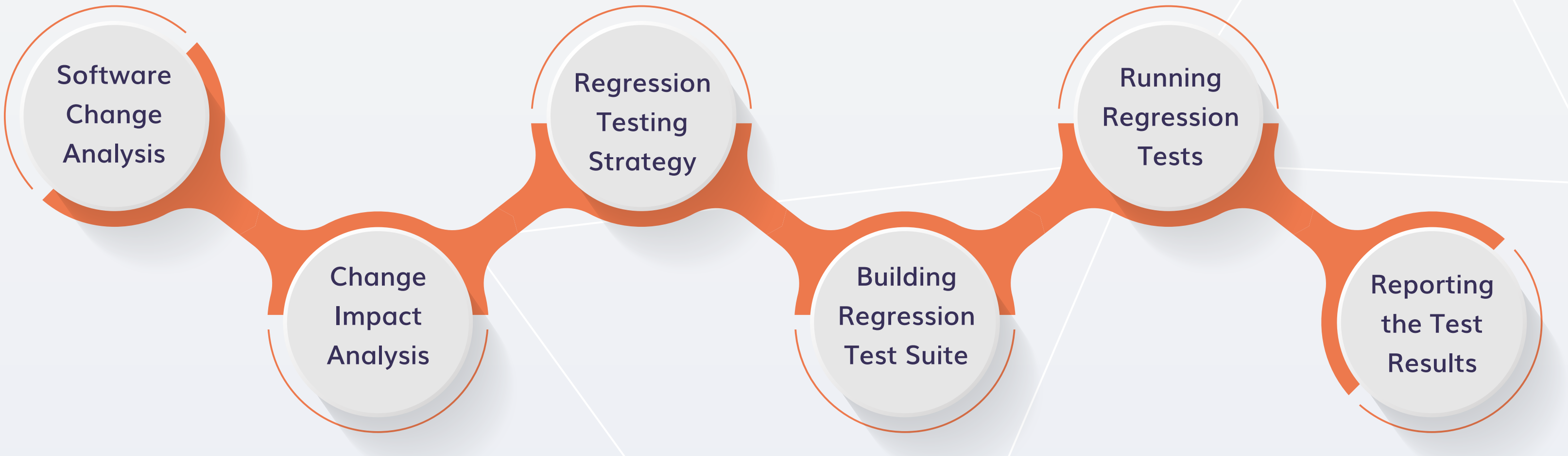
As against manual testing, which is restricted to 8 hours a day per employee, automated regression testing software works round the clock. With features like smart scheduling and intelligent reporting, you can run the tests 24/7. And interpret the logs and identify bugs during working hours.

Enhances coverage and scalability:

Manual testing is highly unscalable because no matter how many hours an employee stretches, there is always a possibility of overlooking the obvious. This takes a hit on the quality. Since regression testing is expansive, coverage through manual testing is limited as well. With automation, you can test more in less time, enhancing scalability and coverage.

How to pick the suitable test cases for regression testing?

Coming up with a regression test automation strategy is no easy task, especially for large enterprises. However, a typical flow of regression test strategy looks like this.


Software Change Analysis
Change Impact Analysis
Regression Testing Strategy
Building Regression Test Suite
Running Regression Tests
Reporting the Test Results

The most crucial step is identifying suitable test cases. Selecting and prioritizing the test cases requires you to have a good understanding of the entire application. This is where an intelligent process discovery solution comes into the picture.

With a process discovery solution, you can document the processes and get a fair idea of how the application works end-to-end. You could further put your employees to tasks that require problem solving and judgment – prioritizing the test cases for automation. Though not mandatory, having a process discovery solution in your kitty simplifies your regression testing efforts.

To make regression testing manageable and practical, it is crucial that you select suitable test cases.

Here's a handy checklist that helps you do the heavy lifting without spending tremendous effort and time.

- □ Test cases that have undergone recent functional changes or code changes
- □ The ones that tend to break in production often
- □ Test cases that verify the core business requirements
- □ Integration tests to ensure the flow of processes is seamlessly
- □ Parts of the application that are visible to the end-users
- □ All complex regression tests
- □ Test cases that are critical and take longer time to bug-fix
- □ The ones that are frequently used
- □ A sample of the test cases that passed and failed in the previous cycle

Important note:

Before selecting the regression test cases, make sure your regression test suite is up-to -date and free from all obsolete test cases.

How frequently should you run automated tests?

As mentioned in the earlier sections, you should regression test your applications every time a code change is implemented. In addition, experts recommended performing core regression testing every night or alternate days and complete regression suite execution once a week.

In a few contexts, however, you can't afford to skip regression testing. They are:

- Every new release that's pushed to production
- When you make enhancements to the applications
- Configuration changes
- New integrations with other products
- Most importantly, when you fix bugs

Given that the applications are extensive and software development is primarily agile, it is impossible to run regression tests manually. Thus, it not only impacts the quality but also delays the timelines.

With automation, you can run end-to-end regression tests. Furthermore, if your solution offers a scheduling feature, it makes your life easier since it provides consistent feedback and runs thorough checks without manual intervention.

Should you rule out manual regression testing completely?

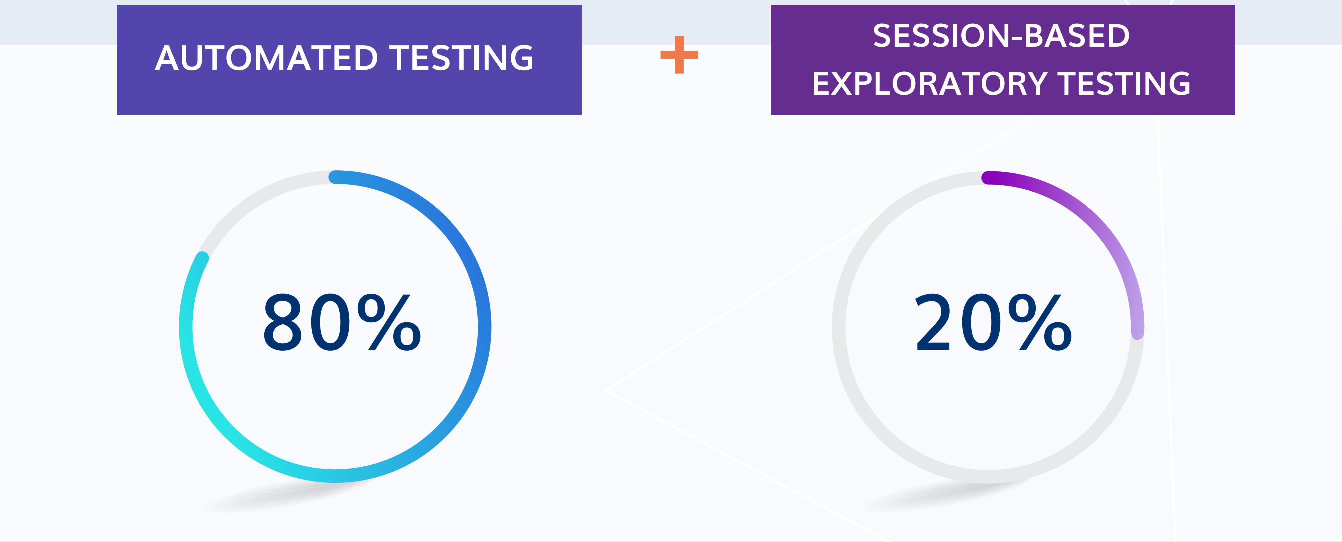
While the typical answer should be 'yes,' it is ideally a 'no.' There is no doubt that automation takes care of the mundane and repetitive tasks, but every business process requires frequent monitoring, evaluating, and mitigating risk; something that only the power of the human mind can do.

Manual testing also works best in overcoming the pesticide paradox of regression testing. According to this principle, if the same set of test cases are executed repeatedly over time, this set of tests may not identify new defects in the system. To troubleshoot this, you need to review and revise your tests regularly, add a new set of tests and delete the existing ones if required. And this is not something that your test automation solution can handle.

Since humans can think out of the box, the chances of identifying a plausible bug are higher. And that's where a hybrid model works best. Embracing automation is a win-win because you save your employees from tedious tasks and channel their skills for higher-value tasks like these.

An ideal agile regression testing strategy includes 80% automated testing and 20% manual exploratory testing.

Ideal Agile Regression Testing = 80% Automated Testing + 20% Manual Exploratory Testing



AUTOMATED TESTING + SESSION-BASED EXPLORATORY TESTING

What capabilities should your test automation solution have?

Automation has the tremendous potential to bring significant business benefits. But, if your automation solution doesn't allow you to be agile and requires you to spend considerable time learning and using it, it compensates for the saved time. So, here are the capabilities your test automation solution should have.

No-code:

A no-code test automation solution enables you to test applications without writing a single line of code. This not only saves your time in writing the scripts but also speedsup your entire regression testing cycle.

Heterogeneous:

When your automated testing solution works seamlessly on the web, mainframe, ERPs, and more, achieving end-to-end testing is not a herculean task. Imagine the amount of time and effort saved in downloading and installing the individual plugins if the solution was platform-dependent?

Intelligent reporting:

The importance of easily understandable and intuitive reports is underestimated. When you can straightforwardly visualize the report, you can pinpoint the bugs quickly and take further actions. Ensure your solution offers a screenshot of each step of execution or a video recording of the same.

Smart Scheduling:

Your solution should allow you to execute multiple scenarios in a single VM independently or in parallel. Especially in the case of regression testing, this saves time and allows multitasking.

Test case reusability:

Reusability of test cases saves you effort and time. Make sure your solution offers this. It should also allow you to get a birds-eye view of the entire test plan.

Why Avo?

With its no-code automation solutions, Avo Automation empowers the largest Fortune 500 companies to transform their business.

With Avo Discover, you can quickly capture, document, modify, and prioritize end-to-end business processes. Avo Assure offers all the capabilities that we just discussed, not just for regression testing but other functional tests as well. The best thing – both Avo Discover and Avo Assure work in unison or individually to help you get the most out of your applications.

- Expand test automation coverage up to 90% by executing end-to-end regression tests repeatedly.
- Achieve over 40% cost savings and increasing employee capacity for innovation.
- Be 2x more productive.
- Leverage about 1400 keywords and reduce testing time.
- Save over 45% of your regression testing effort.
- Deliver high-quality applications at least 3x faster.
- Reduce the production bugs by 80%.

Want to know more about what Avo Automation can do for your business? Reach out to us, and we will schedule a demo for you with our experts.

SAP® Certified
Integration with SAP NetWeaver®

Available on
SAP Store

SAP® Certified
Integration with SAP S/4HANA®

High Performer
Enterprise
SUMMER 2021

High Performer
SUMMER
2021

Users Love Us

AVO™
Automation

Grow with **Avo**

Release quality software faster

Explore Avo Assure

Get a 14-day assisted trial for the test scenario of your choice. **Click here.**