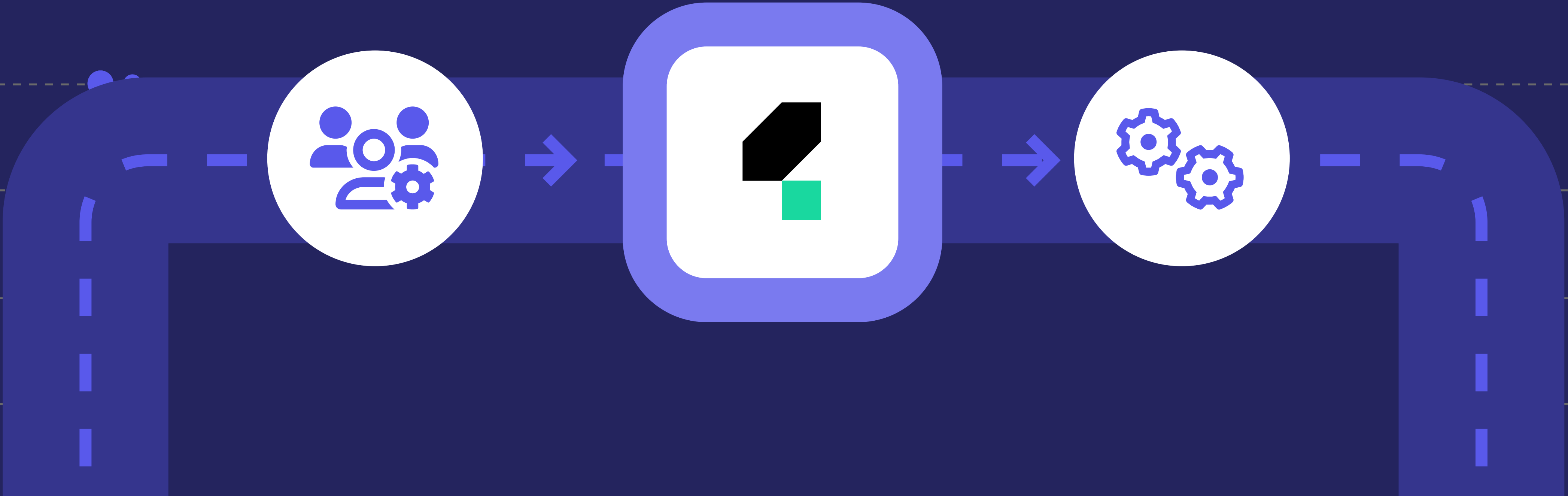
Katalon

From Manual to Automated Testing

5 Essential Steps



Manual Testing and the Need for Speed

Software companies constantly battle between quality and delivering at speed. Manual testers are overwhelmed by thousands of tests under limited time while bearing the risk of undetected bugs in production.

For instance, e-commerce platforms work on multiple operating systems using the same backend. Every time the platform is updated, QEs struggle to complete **thousands of regression test cycles**. With automation implemented, benefits ranging from **test scheduling** to **parallel execution** can shorten the process from weeks to only hours.

A Roadmap From Manual to Automation Testing

- Step 1: Automate what’s repetitve
- Step 2: Choose the “right” tool
- Step 3: Tool/Framework development process
- Step 4: Generate – Execute – Get Reports – Maintain
- Step 5: Monitor the test automation tool’s effectiveness

Step 1 Automate what's repetitve

A general rule of thumb is to automate the **most frequently performed tests**. The list below shows you what to automate and how automation can help.

Test types	How automation helps
Regression tests	Automating regression tests help alleviate time constraints, repetitiveness, and stressfulness while maintaining the functionality of existing features. As found in **The State of Quality Report 2022**, 53% of respondents reported using test automation to do regression testing.
Acceptance tests	Test automation tools and software quality management platforms boost performance consistency by supporting cross-browser/platform testing. However, be selective when filtering out the UAT scenarios to automate, like login flows (with/without CAPTCHA).
API tests	With test automation, automated API regression tests are triggered every time you adjust an API and detect bugs earlier in the process.

Start free trial

Step 2 Choose the “right” tool

The mantra of tool selection is **“There’s no one-size-fits-all tool in automation”** as each team has specific needs, requirements, and resources.

- What application types is the team mainly working with (e.g., web, mobile, API, desktop)?
  - Does it need a custom framework or can a vendor solution get the job done?
- Is the tool suitable with the end-users’ automation expertise (e.g., automation engineer, manual testers, developers)?
- Does the tool provide necessary features to write, run, get reports and debug automated tests?
- Is it easy to debug scripts, update locators and reuse test artifacts?
- Are integrations with tools like CI, Dockers and test management **native or require workarounds?**
- Is there support for technical issues?

> **According to The State of Quality Report 2022, open-source frameworks and libraries are decreasing by 49% in popularity. It can be seen that teams are putting effort into adopting commercial tools for their comprehensiveness and ROI efficiency.**

Start free trial

Step 3 Tool/Framework development

It is a good practice to strategize your tool development when using open-source libraries. Planning keeps you from derailing from your original vision. However, with a pre-built test automation tool like **Katalon**, you can avoid this hassle and move on to step 4 now.

Fundamentally, a test automation tool is a software system that requires project management, programming skills, months or years to build, and continuous maintenance. To develop your test automation tool with an open-source framework, use the same approach and process as you would with other software. Your team should start building basic testing functions first (test generation, test execution) and add functionalities along the way. It is crucial to have the mindset of “start small, improve later”.

A test automation development strategy should include:

- A vision that keeps you committed to the tool objectives.
- The business values of your automation tool.
- The design and features of the framework/tool.
- The timeline and process of test scripting and execution.
- List of in and out-of-scope automation items.

Start free trial

Step 4 Generate-Execute-Analyze-Maintain

TEST GENERATION: Test scripts can be created with full-code or **low-code approaches**. Supporting commonly used programming languages (e.g., Java, Python), open-source testing frameworks and libraries are a great choice for teams with a dedicated development team. A **dual-editor interface for record-and-playback**, drag-and-drop **and scripting** will be needed if there is a mixed number of automation newbies and developers.

TEST EXECUTION: Running tests across **browsers, devices and operating systems** is simpler to do with only one tool. If multiple tools, frameworks or extensions are used, the need to maintain every single one of them will be tremendously time-consuming. Find yourself a tool that supports a wide range of environments, their latest versions and an easy setup process.

TEST REPORTING: Working with open-source frameworks calls for third-party reporting software. In contrast, commercial tools introduce **built-in functionalities** such as sending reports through emails, test failure analysis and many more.

TEST MAINTENANCE: If your tool is built on open-source libraries, the process of updating and maintaining your tests and the framework itself is troublesome. Pre-built vendor tools that follow the **Page Object Model** design enable you to store all objects into one repository so that when updates are needed, changes can be entirely applied.

Increase test automation coverage without a framework to build or pressure to learn coding right away? **Katalon software quality management platform** is just right for learners and quality professionals to shorten functional testing cycles.

Start 30-day trial

Step 5 Monitor the Test Automation Tool’s Effectiveness

Monitoring your test automation adoption helps evaluate the its effectiveness and ensure its business value.
Here is a list of questions to help you find suitable metrics for your team’s testing goal.

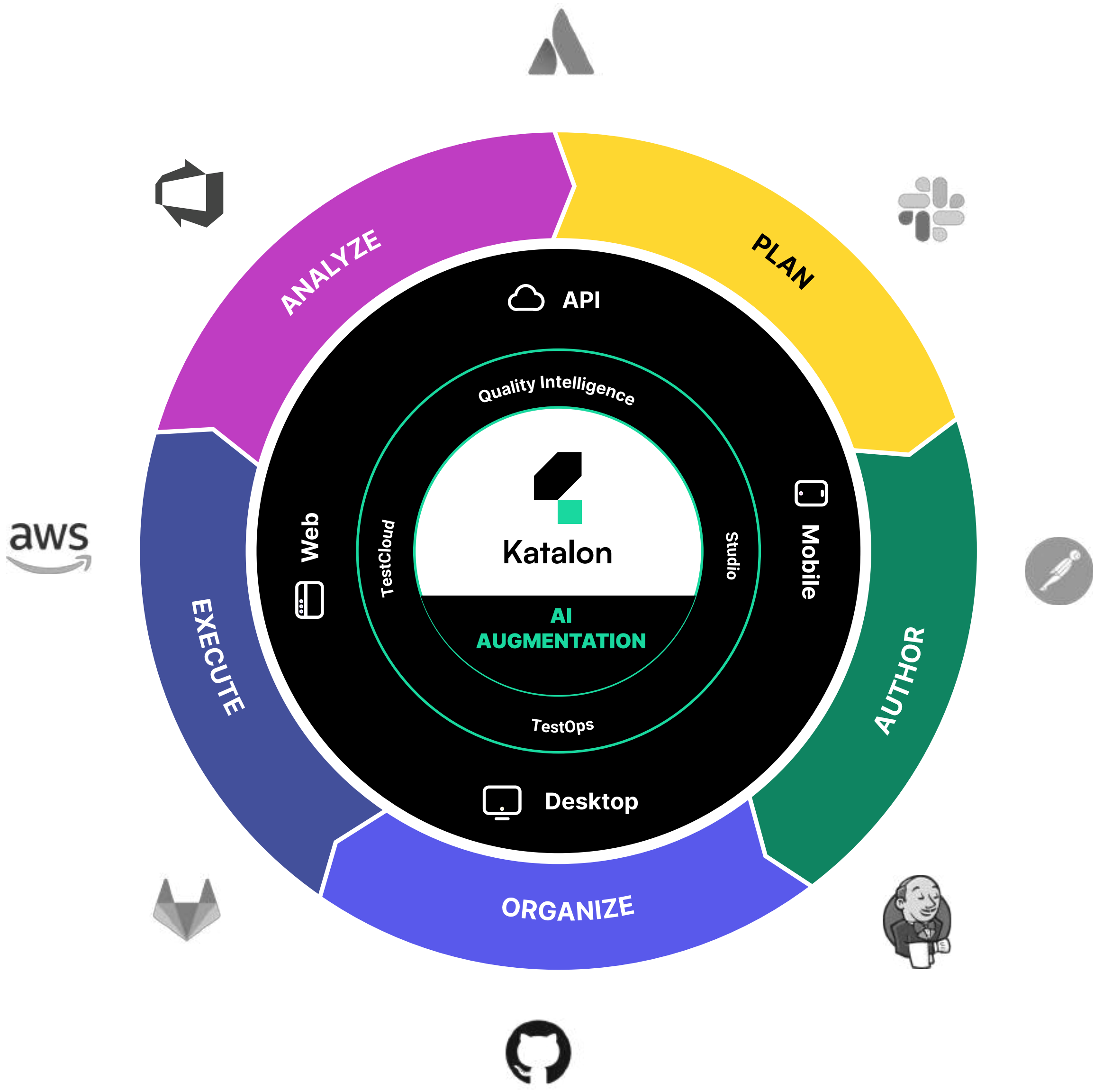
- **How much execution time did you save?**
- **What’s the average time to update or rewrite tests when changes are made to the AUT?**
- **Does it take longer for the same test case to perform overtime?**

  **If a test takes longer to run than usual, you should further investigate the effectiveness of your automation components.*

- **Has the time to develop the testing tool, assets, or features decreased over time?**

  **This only applies if your test automation tool is self-built from open-source libraries.*

Start free trial



Katalon Quality Management Platform

Low-code web, API, mobile and desktop automated testing

Highlight Capabilities	Description
Low-code and full-code options to create tests	• **Record & Playback:** record actions into runnable test scripts • **Manual Mode:** drag-and-drop keywords to build tests • **Script Mode:** script with Java or Groovy for more advanced scenarios and customized keywords
Remove duplicated efforts with test and configuration reusability across projects	• **Test Artifact Sharing:** import and export test cases, test objects, profiles and custom keywords • **Project Settings:** import and export DesiredCapabilities • **Test Refactoring:** view unused test objects to identify what to prioritize
Shorten and stabilize execution for larger volumes of tests	• **Test Scheduling:** set execution time and date in advance • **Parallel Execution:** run multiple test suites without CPU spikes • **Dynamic Test Suites:** determine test priority via properties and search queries • **Smart Wait:** handle web loading and time lag issues • **Self-Healing:** propose working alternatives to replace broken object locators
Minimize maintenance efforts	• **Time-Capsule:** restore the AUT state when a test fails without reproducing the whole steps to fix broken web objects quicker • **Global Variables:** define variables once, reuse and update when needed
Test multiple AUTs on the most popular OSs	• **All-In-One Platform:** one tool for web, API, mobile and desktop (Windows) testing projects • **OS Compatibility:** Windows, macOS and Linux

Start free trial

Highlight Capabilities	Description
Support various testing methodologies	• **Behavior-Driven Development:** native Gherkin editor and Cucumber-compliant • **Data-Driven Testing:** input values from internal test data and database, Excel/CSV, Oracle SQL, SQL Server, databases with JDBC drivers or a combination of data files • **Keyword-Driven Testing:** drag and drop built-in web UI, mobile, API and desktop keywords to create tests • **Cross-Browser Testing:** web and mobile browsers (Headless, Chrome, Edge, Firefox and Safari) • **Cross-Platform Mobile Testing:** iOS and Android
Natively integrates with ALMs, CI/CD and DevOps tools	• **Application Lifecycle Management:** Jira, TestRail and qTest • **CI/CD:** Azure DevOps, Bamboo, Bitbucket, CircleCI, Jenkins, GitHub, GitHub Action, Gitlab, etc. • **Version Control:** Git, GitHub, GitLab, BitBucket, Azure DevOps Repos, etc.
Migrate legacy scripts and import API objects	• Selenium and Selenium IDE • Postman, SoapUI, WSDL and WADL

Start 30-day trial   **Book a demo**

Katalon

General & Legal Inquiries: business@katalon.com
License & Purchase Inquiries: business@katalon.com
Partnership Inquiries: partner@katalon.com

Follow us