



TESTOPS

A Definitive Guide to Scaling Your Test Automation Program

testim

INTRODUCTION

Two trends drive an increased need for end-to-end testing: growth in web applications and development modernization.

Web applications continue to grow in popularity for B2B and B2C use cases. As browsers have evolved, more of the application code has shifted to the front-end of applications.

At the same time, modern development methodologies demand faster iteration and releases. With shorter, more frequent cycles, the pace of innovation is too fast for manual regression testing.

Modern test automation practices replace slower methods while bringing more diverse testers, including developers, product owners, and automation engineers.

Scaling your tests, teams, and projects efficiently is both a significant challenge and an opportunity to lead.

This ebook will discuss tips and techniques for scaling testing operations (TestOps) more efficiently and effectively.

TABLE OF CONTENTS

01

Why test automation? Why now?

PAGE 04

02

The challenge with growth

PAGE 05

03

Impact of agile methodologies

PAGE 07

04

**Shifting from manual
to automated testing**

PAGE 08

05

Key attributes of scaling

PAGE 09

06

Test planning

PAGE 10

07

Management and organization

PAGE 11

08

Process control

PAGE 12

09

Reporting and insights

PAGE 13

10

Summary

PAGE 14

01 Why test automation? Why now?

The digital footprint quality teams have to manage is increasing each year. The growth is especially prevalent in web applications. Evolving JavaScript frameworks and platform and hosting services enable agile teams to release new functionalities in their web apps quickly. Rather than deprecate old features, developers can adapt them with new UX experiences.

Quality Assurance teams (whether separate or embedded) have a tough choice to make—add resources, take on more risk, or delay releases.

Test Automation has been the saving grace for many software organizations, allowing them to (somewhat) keep pace. However, the continued shift to faster development methodologies and the importance of a digital presence have highlighted vital gaps.

Relying on older test automation frameworks to code functional and UI tests simply takes too long. And the lack of stability in rapidly changing applications reduces the time teams can spend writing new tests, ultimately widening the divide between desired and actual test coverage.

02 The challenge with growth

Automating tests can help reduce the reliance on time-intensive manual regression testing. Many teams have invested heavily in building out test automation suites on open-source frameworks like Selenium.

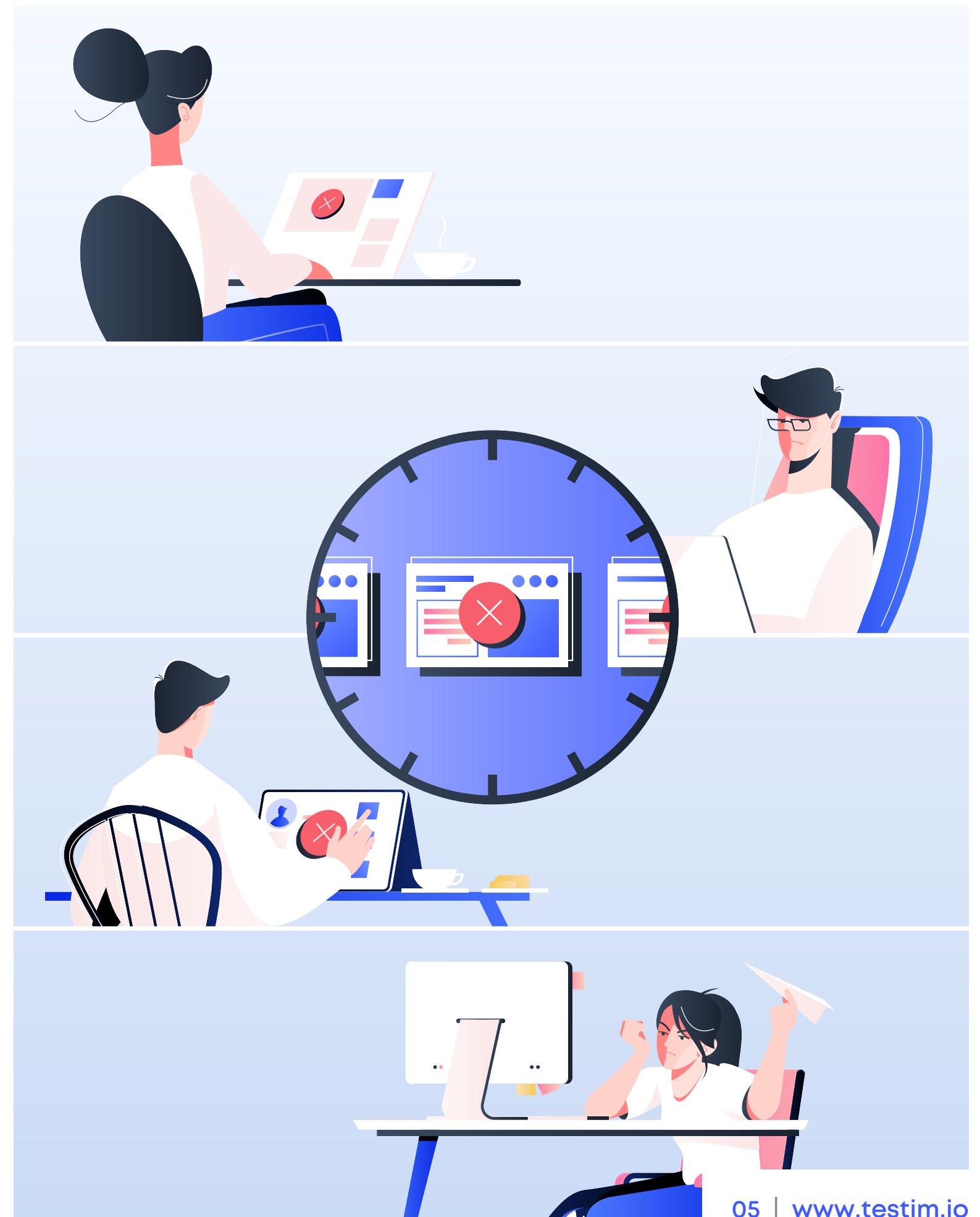
Yet, test automation is still underutilized for many organizations. In fact, according to a 2020 Testim survey, only 25.6% of respondents had automated at least half of their end-to-end tests.

Is too much of a good thing, a bad thing?

Historically in technology, when teams and resources need to scale, efficiency initially suffers, and a new and better approach emerges. Think data storage, network routing, cloud computing, and container services.

Too often, when teams fall behind their quality goals, they have to choose between undesirable options:

- Work harder - spending long hours, including weekends, to catch up. This may help in the short term but typically leads to lower morale and higher turnover.



- Limit coverage - testing some areas of the application while ignoring others. Over time this can translate into significant gaps and higher than acceptable risk.
- Skip testing - pushing releases out the door without adequate testing. Impacts can be mitigated with phased release roll-outs, though over time, quality can suffer, impacting customers and revenue.

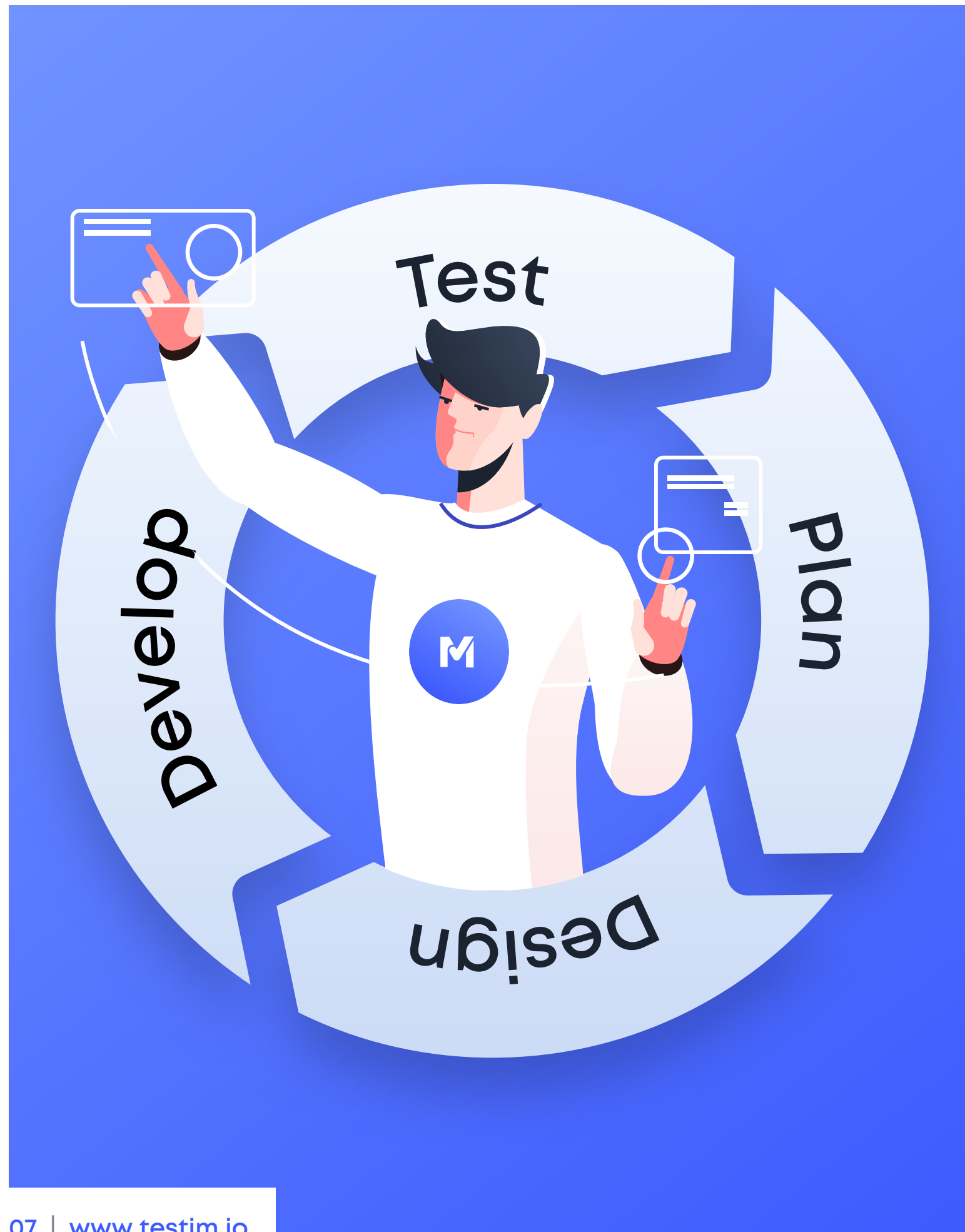
Another common stop-gap approach to handle scale in test automation is to hire additional quality engineers. But, hiring and training scarce, specialized skills takes time and drives up labor costs. Due to the growth in web apps, it's usually only a temporary fix until the next hire is needed.

Often to get out of the catch-up game, you need a new, more modern approach—one that changes the economics of the current model.

According to the World Quality Report for
2020-2021, nearly

90%

of respondents said AI formed the most significant growth area in their test activities. We at Testim agree. AI can help speed test authoring, reduce the cost of test maintenance, and lower an organization's reliance on manual testing.



03 Impact of agile methodologies

The shift to agile development methodologies pressures teams to adopt faster, more flexible testing approaches. Shorter release cycles or sprints translate to frequent application code changes and the need to write and stabilize tests more quickly.

It also means that more team members are involved in the testing process, adding developers, product owners, and business analysts into the mix. These team members have different roles and skillsets, yet with proper tools and procedures, they all can help drive improvements in release quality. Developers understand what is changing in the application. Product owners know the exit criteria for features. QA engineers have expertise in testing from the user's perspective.

One caveat of note: with mixed teams designing, writing, and evaluating test results, you need solid processes and controls to raise the standards of your tests. This can include training, required reviews of test changes by a Subject Matter Expert, or even guidance for test naming conventions.

04 Shifting from manual to automated testing

We showed earlier that test automation is underutilized. It's also growing. Analyst firms like Modor Intelligence and others predict the Software Test Automation market to continue to grow at 14-15% CAGR between 2020 and 2025.

As teams shift from manual scripts to automated tests, some benefits are clear, such as cost savings and faster feedback. Rather than spend days manually running through test scripts at the end of a build phase, automated tests run quickly and often, providing more rapid feedback. Automated tests can run across different datasets and browser configurations with almost no time penalty. These time savings can free resources to focus on exploratory testing or complex testing scenarios.

However, to gain the most from this shift, teams also need to consider changes to processes and documents designed around manual testing. For instance:

- Simplify test design documents to focus on key objectives and expected outputs and conditions rather than detailed procedures.
- Consider using your development tracking tools like Jira or GitHub issues to link features to automated tests rather than separate test management tools.



05 Key attributes of scaling

Modern software organizations seek to deliver innovation faster. Each step, from requirements gathering to delivery, needs to be smooth and efficient. We'll focus on different aspects of scaling your test operations more efficiently to keep up with faster software development cycles and more diverse roles and skills.

Testing operations extends from strategy and planning through execution and reporting. Teams need a multi-faceted approach to scaling their test automation projects to avoid shifting bottlenecks from one area to another.

To simplify this discussion, we will focus on four pillars of software testing and discuss how each can be adapted to enable more efficient operations.

Test Planning	It addresses critical aspects of your testing approach. What do you want to test? How do you test it? Who will test it? Consider application risk, the skills and capabilities of your team, and processes needed to establish consistency.
Management and Organization	As you grow, management and organization become more important. Teams need precise work delineation and ownership to avoid duplication. Test lifecycle comes into play as well—tests need to show value and not add workload.
Process Control	When the extended quality team has mixed skills and experiences with testing and automation tools, processes and controls can help maintain high test quality and oversight standards.
Reporting and Insights	Reporting should be more than pass/fail. It should provide a broader view of release readiness, team progress, and areas for growth. The goal is continuous improvement based on data-driven insights.

06 Test planning

Planning should align with the overall testing strategy yet be more specific to the current project or feature in development. Planning determines what and how the capability should be tested and the expected exit criteria.

Planning meetings should occur early in the development cycle and include all relevant stakeholders. Developers understand how new features affect the application and if the changes will have a narrow or broad impact. QA engineers bring testing expertise, including knowledge of current testing assets and limitations. Product owners possess a clear understanding of the expected user experience and motivations. Together, planning helps the team prioritize the tests, often by perceived risk, code complexity, existing coverage, or other variables.



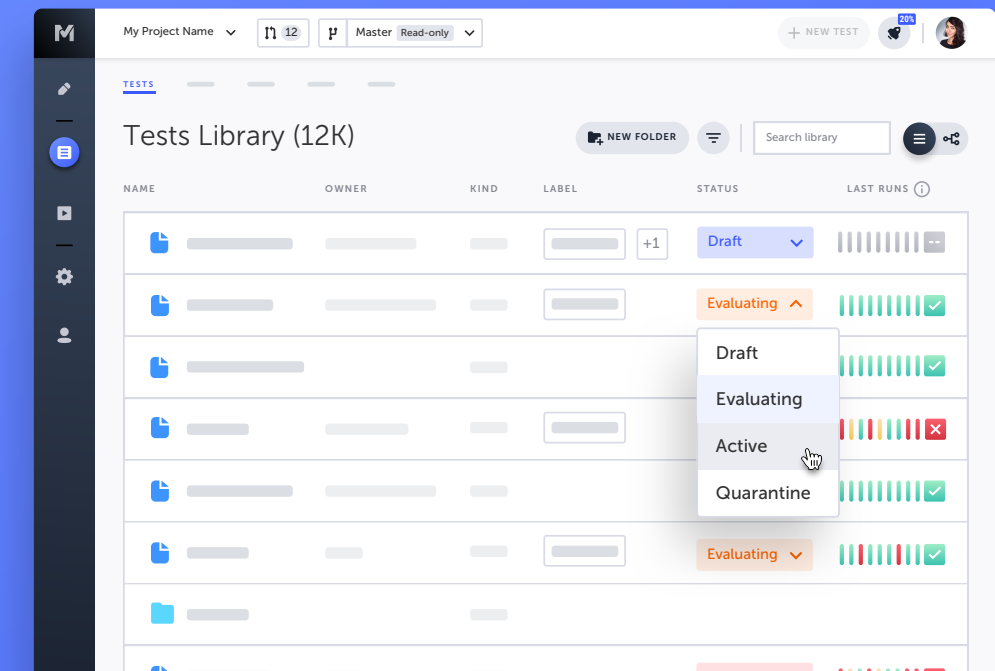
07 Management and organization

Management represents the organization and administration of tests, teams, and tools. It can include coordinating who works on tests or which tests to prioritize.

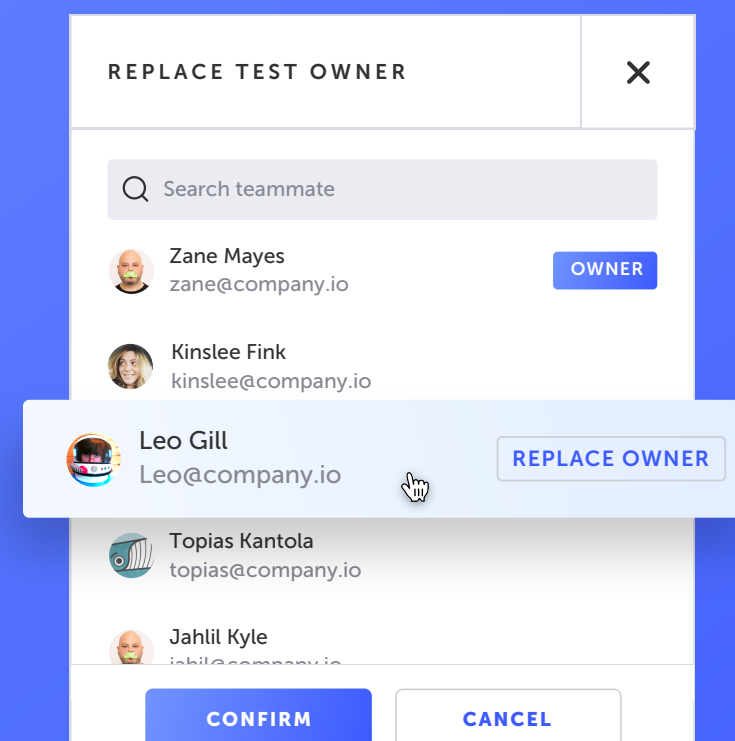
As your project grows, having clear policies can help improve team communication and minimize confusion that causes friction. For instance, set guidance naming conventions of tests or shared components that are descriptive and meaningful. Explain when and how to use labels to improve consistency and clarity. Organize tests into suites or use folders to align with the application, such as by project, user-type, or feature.

Finally, flaky tests can lower confidence, so treat tests like code and test them before they become blockers in your CI.

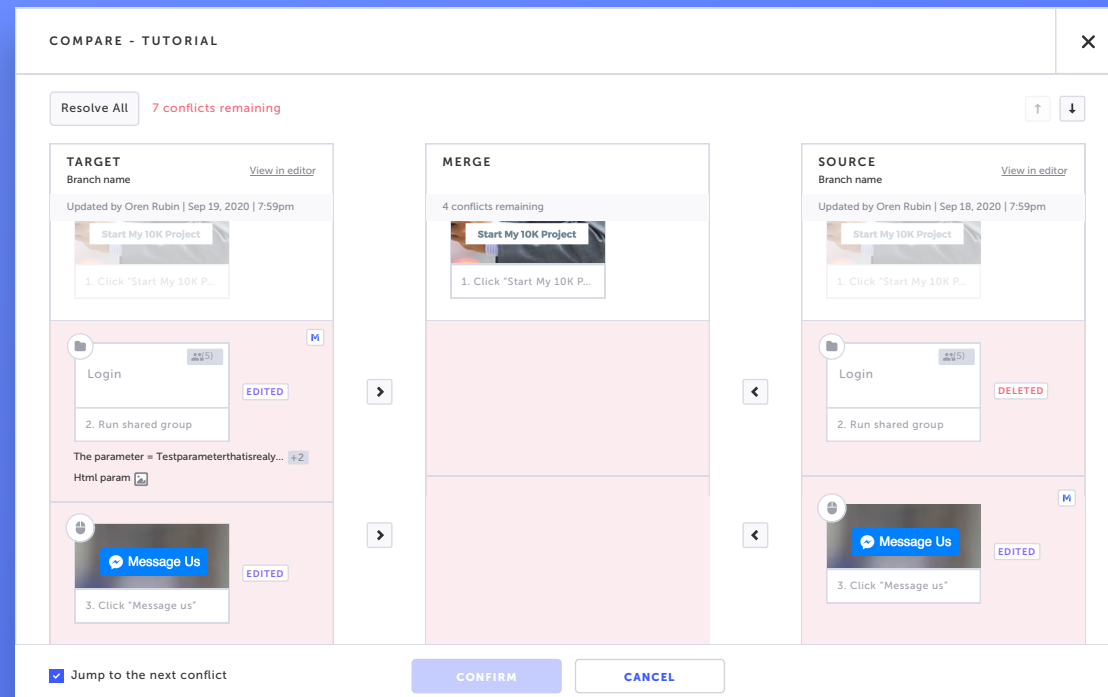
Manage the test lifecycle with test status.



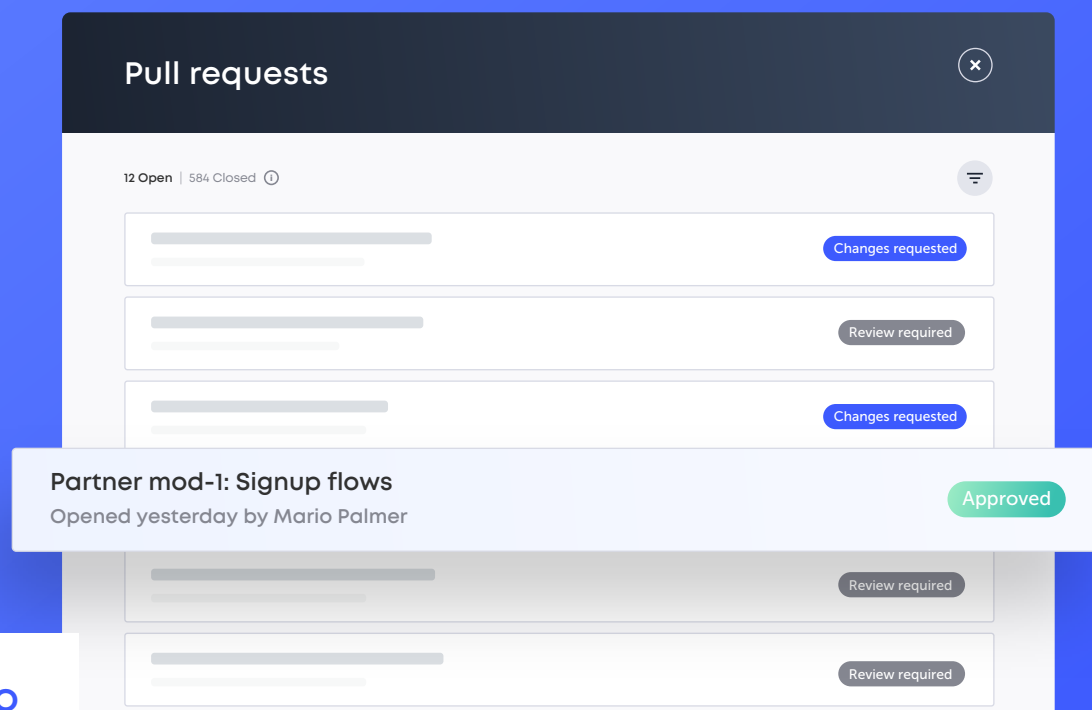
Assign test owners to reduce conflict and improve accountability.



Compare changes at a granular level and cherry-pick what to merge.



Pull requests ensure changes are properly reviewed and approved.



08 Process control

To scale efficiently, you need processes that help ensure consistency and drive the team to higher standards. Processes are even more critical when teams contain diverse skills and varying tool knowledge.

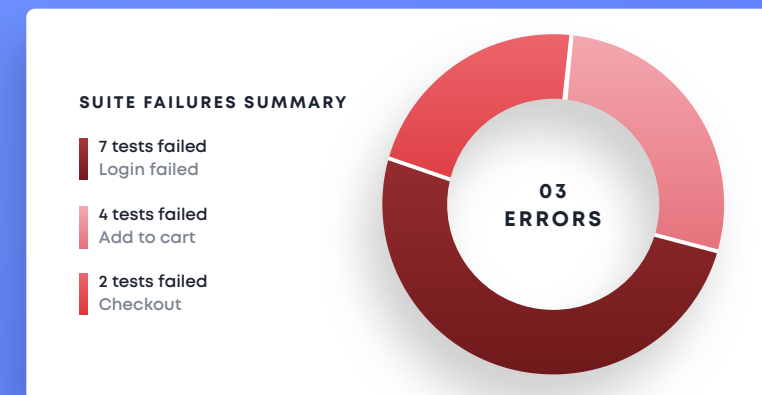
Use process controls to ensure that tests are built and maintained to high standards. Features such as read-only master, three-way merges, required reviews, and pull requests can help ensure test updates get proper visibility and approvals by your subject matter experts.

09 Reporting and insights

To manage at scale, you need to know more than whether a test passed or failed. You need to understand the state of your application's quality, how your team is performing, and what actions you can take to make systemic improvements.

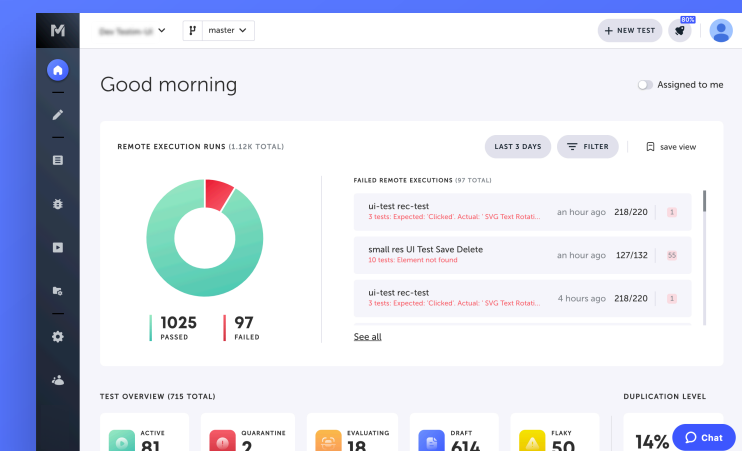
Test results reporting can help you see how your tests are performing. When teams organize tests according to test objectives (such as by feature, project, or user type), they can provide a complete view of release readiness.

Insights refer to the information you derive from your testing and teams to help you work smarter, triage and prioritize work, or make process improvements. Insights at a personnel level can help identify opportunities for training and team productivity gains.



Focus on recurring failures to address systemic issues.

View project or personal dashboards to help prioritize daily work.



Spot trends in test results, errors, or activities and make adjustments.

SUMMARY

TestOps practices...

PLAN AS A TEAM

Leverage collective knowledge of the team to match inherent application risks with appropriate testing plans.

ORGANIZE AND MANAGE

Create a structure enabling team members to find what they need quickly, manage their workflow, and collaborate fluently.

CONTROL CHANGES

Treat your tests like code. Have a subject matter expert review and approve changes.

INSIGHTS

Use data from your tests to inform application quality and identify areas for personnel, systemic, or personnel improvements.

...drive business results

SMARTER USE OF RESOURCES

Improved work prioritization helps minimize quality risk while utilizing the appropriate team members and testing assets.

BOOST TEAM PRODUCTIVITY

Well-organized tests, folders, and suites, along with clear ownership and prioritization, help mixed teams minimize friction and maximize efficiency.

ACHIEVE HIGHER STANDARDS

Well-designed tests are easier to maintain, cause fewer disruptions due to false failures and inspire confidence.

CONTINUOUSLY IMPROVE

Teams can adjust training, fix underlying systemic problems, and habitually enhance and streamline processes with the correct data.



THANK YOU!

**Ready to get started on
your journey to modern
UI testing?**

Get a free account: app.testim.io

Request a demo at: info@testim.io