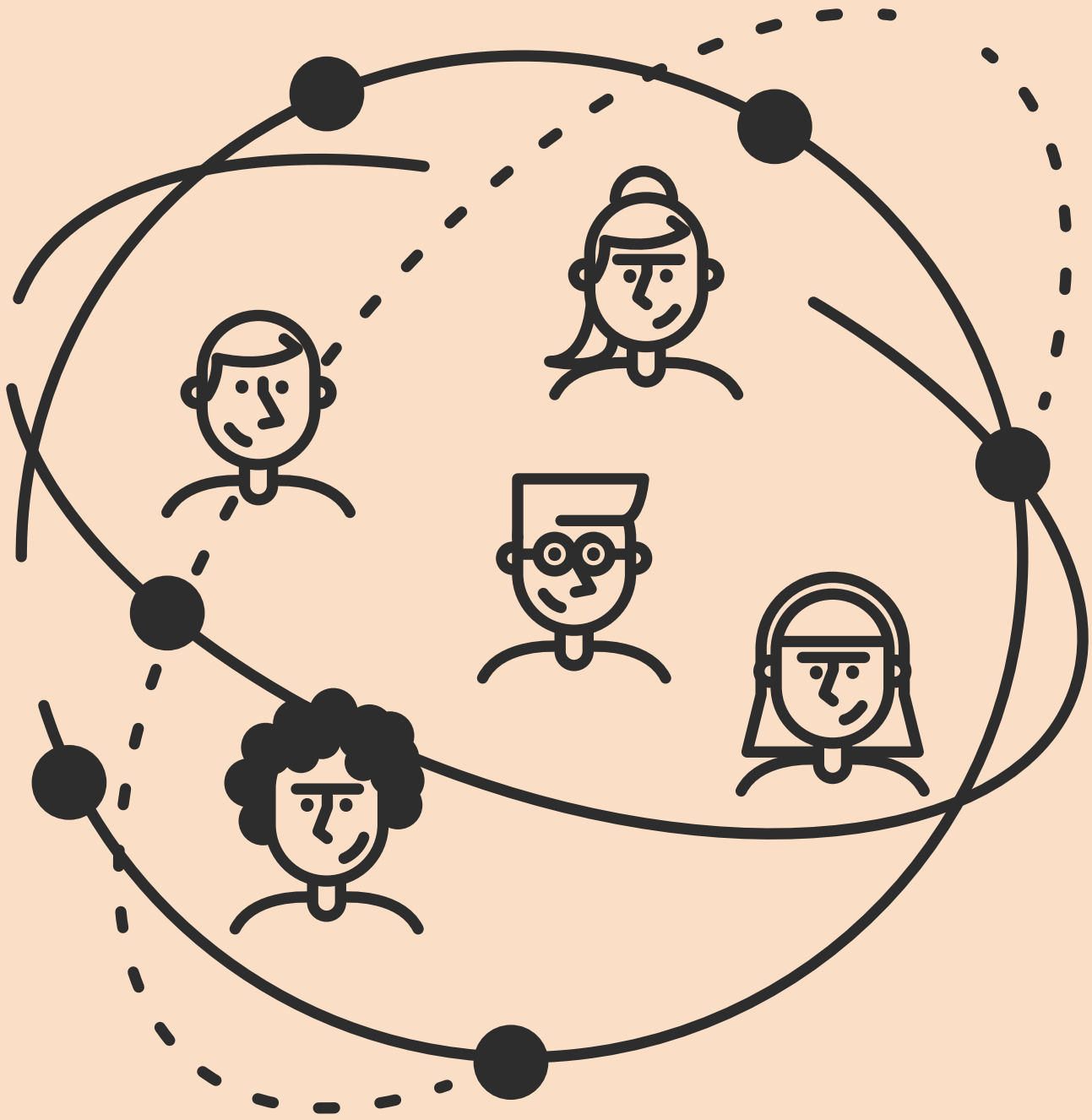
Networked Testing:

Fusing Manual and Automated Testing

A New Paradigm for Quality Assurance



testlio

Table of Contents

Introduction

Successful software leaders know the right mix of app testing tools, and QA services lead to more efficient quality assurance outcomes. Web and mobile application testing have evolved, and today's continuous development environment yields the most productive results with a fine-tuned blend of automation strategy and human-based manual testing best practices.

This guide uses applied use cases to uncover the formula for app testing at its finest. It shares real examples from TV network CW and others who successfully match automated scripts with on-demand testers to meet a need for speed. This guide also explains how the logical principles of Networked Testing improve efficiencies and quality throughout the mobile application testing process. Together, humans and machines run rapid test cycles, conducted in short timeframes, to release flawless apps and end-to-end customer experiences.

Readers will come away from this guide understanding:

- How networked testing helps engineering teams release faster
- How to build a more holistic and efficient QA strategy to avoid coverage gaps
- Best practices, case studies, and tools to leverage people + machines to glide between active and idle testing states, bursting into action only when necessary
- How to synthesize your automated and manual testing results for integrated reporting

Testing apps at scale isn't for the faint of heart

Today's testing environment must be flexible, burstable, and scalable—enabling release candidates to quickly move from engineering to points of distribution (i.e., Google Play, Apple App Store, etc.). Building a reliable and scalable testing and QA strategy for your organization has its challenges.

Challenges for today's QA and Engineering Teams

Frequent releases and CI/CD intensity

Today, we must deliver high-quality apps with flawless end-to-end customer experiences. To keep up, most teams release every 1 or 2 weeks in conjunction with sprints. Testing must be fast and ready to burst into action at key moments.

2 to 4 times

As a best practice, Testlio recommends pushing App Store updates 2 to 4 times per month

*Source: Testlio benchmark research

There's simply too much work

With managing workflow, increasing productivity, and reducing costs, engineering leaders have a lot on their testing plates, including:

- High performance up times and response times
- Consistent OS and hardware changes
- Critical fixes
- Marketing requests
- New functionality
- Seasonal initiatives
- Challenging location, language, and network needs
- Complete device/OS coverage

Networked Testing methodology empowers engineering teams

Networked testing is a modern and flexible software testing methodology whereby automated and manual testing is fast, rhythmic, impactful, and economical. Companies leverage networked testing techniques for automated regression testing, functional testing, usability testing, localization testing, exploratory testing, livestream testing, and more.

Key characteristics of the networked testing include:

- Expert human testers fill gaps missed by automated testing
- Burstable swarm teams provide on-demand coverage only when necessary
- Compressed testing windows lead to efficiencies and fast testing cycles
- Connected real system linked to existing workflows and reporting bring it all together

Networked testing relies on the best of both automated testing and human-based testing. Let's review the pros and cons of automation and human-based testing before discussing the magic formula for integrating the two approaches.

Imagine augmenting your automated scripts with a nimble team of 10-50+ remote, on-demand manual testers bursting into action quickly to swarm the testing surface. This burstable tenet of Networked Testing leverages testing software, automated scripts, and expert human testers working in concert to burst upon the testing surface only when necessary and in compressed testing windows.

- Capacity to address varying release demands at key moments
- Complete coverage for languages, places, and networks that matter
- Testing for all in-the-wild hardware and software configurations

"When well-designed and implemented, networked testing can provide superior value for native app product and engineering teams. It can deliver a tuned approach that helps teams release world-class apps confidently — often at a fraction of the overall cost of other testing approaches."

-
Steve Semelsberger, CEO, Testlio

Forbes

Automated Testing: Pros and Cons

Automation technologies like the Sauce Labs Continuous Testing Cloud, BitBar, Amazon Device Cloud, and Perfecto enable large scale parallelization of automated tests. Instead of running a single test, automators burst onto the testing surface and conduct many tests at the same time. Automated verifications expedite the testing of prescriptive runs quickly and efficiently by way of multiple computers.

Pros	Cons
Fast and efficient	While fast and obedient, automated tests miss obvious issues a human can easily detect. For example, using an automated test to inspect app login flow won't catch an obvious blinking error in the center of the screen without specific instructions within the script.
Scalable	Costs to implement and maintain including the oversight from a dedicated engineer.
Repeatable	Unless you already have automated tests created and debugged, feature code deployed during 2-week sprints is often working before automated tests catch up.

Automated Testing: Best Practices

Embrace technology & partnerships

To keep up with ever accelerating release cycles, QA teams need to embrace process, technology, and partnerships to succeed. Software companies generally rely on engineers, known as Software Developer Engineer in Test (SDET), who can code side-by-side with the developers and write automated tests in parallel to the development.

The challenge is the cost to maintain a staff of expert software testers in addition to the SDETs. To fill this talent gap, partnerships are vital. QA companies like Testlio can expedite both automated and manual testing.

Follow software dev principles

It's important to realize that automated testing IS software development. In this case, the software you are developing is there to test other software, and is used internally. With that in mind, it's important to follow software development principles when writing automated test scripts, such as:

- Source code management
- Using design patterns
- Investing in frameworks

And yes, even tests need to be tested. Thus, testing the tests is also important in automation.

Take an incremental approach

Be mindful of the time and costs involved in creating automated suites.

"One time, I had an automation team create a regression test suite for an app that I was leading. They took several months to research the best tools, create a framework, design the tests, and code and test the tests. All pretty good, except for the "several months" part. By the time they were done, the original app had gone through a dozen releases – and the automated tests did not work with the new version of the app.

Now, my approach is to ask for one test to be automated first, then incrementally add to that test, keeping the test suites running on a daily basis, and updated alongside the app as it's updated."

-
John Ruberto
Senior Engagement Manager, Testlio

Automated Testing: Primary Use Cases

Unit tests

Unit tests test a single function (the unit) and replace the rest of the system with other code called "mocks". For example, a unit may be a function that calculates the sales tax on an item.

The purpose of the unit tests is to ensure the code works from a functional perspective. Since the rest of the system is mocked out, unit tests can generally test every permutation possible for the function, including errors and exceptions.

Unit tests run quickly; are usually written in the same language as the underlying code; and are tightly coupled to that code. All of these reasons make them a great candidate for automation.

API Tests

Different parts of an app communicate with each other through APIs (Application Program Interface). API tests will simulate one side of that communication to ensure the other API is functioning as expected.

API tests run pretty fast, but not as fast as unit tests. They are also very reliable, in that they tend to run often without providing false results (also known as flaky tests).

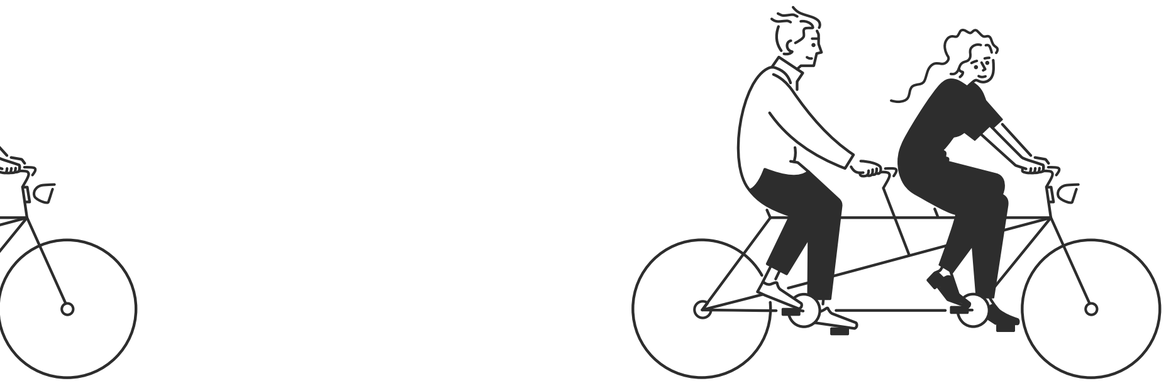
We typically test the business logic of the application using API tests.

UI Tests

UI tests are generally created through frameworks and tools that are accessible to testers that don't have programming skills. This factor helps make UI tests one of the first places for automating tests.

UI tests simulate users interacting with your app, and involve testing the full technology stack for the app, including the back end. When your UI tests are working, they confirm that the many components of your system are working together. A small suite of UI tests can qualify your builds for further testing.

UI tests generally map to existing manual tests, so if your goal is to reduce manual tests, the UI automated tests can be a direct replacement.



Manual Testing: Pros and Cons

Similar to running multiple automated tests in parallel, networked testing enables large scale parallelization of human-based tests. A core tenet of networked testing is the use of on-demand teams of 10-50+ human testers per major run. These testers become familiar with your product over time and help meet variable testing capacity by reliably bursting into action at key moments. Instead of a large staff just in case, you get a large staff of skilled testers just in time.

Pros	Cons
Real Devices and Locations	Not efficient for daily unit, API, or clean (non-flaky) UI testing
Expert feedback for usability, localization, and exploratory testing	Less throughput compared to automation
Finding bugs that scripted or automated tests miss	Pay-per-bug model incentivizes quantity over quality leading to more of your time to filter for important bugs

Manual Testing: Best Practices

Quality of human testers

Testing models have evolved from outsourcing to crowdsourcing to Networked Testing. Outsourcing became too expensive, and the traditional pay-per-bug crowdsourcing approach incentivizes rapid bug detection – producing long lists of low priority issues.

Today's Networked Testing offers greater efficiencies and better overall results at a lower cost:

- Testers augment your in-house team only during periods of peak demand.
- Hourly payment models mean the detection of the highest quality bugs.
- You'll gain thoughtful feedback from experienced testers who understand the state of your product.
- On-demand testers work in short bursts to avoid fatigue and maintain objectivity.

Fully-managed or co-managed approach

Managing human testers takes time. Look for app testing solutions that meet your specific needs – to become your quality team or serve as an extension of your QA team.

- Fully-managed testing means no in-house quality team is needed, which saves management time. You can also leverage a global network of on-demand testers for any device, language, and location.
- Co-managed testing allows you to augment your in-house team with access to testing software coupled with direct access to an on-demand tester network. This flexibility comes in handy for quick fixes and running tests at any time of the day or night. You can perform tests in minutes whenever necessary.

Visibility into the entire process

Whether you depend on a fully-managed or co-managed testing service, you want behind-the-scenes access and flexibility.

- You'll appreciate visibility into the entire testing process – data captured, including the number of bugs, devices, locations, languages, and overall test execution. This data serves as a benchmark on test cycles, issues, and hours to manage.
- You also want to ensure any unused testing hours are applied to more coverage as needed.
- Look for flexibility in how you apply managed app testing service hours so you can apply hours to your needs at the time, whether that's testing or fixed issue verification.
- Look out for limits on app testing software user licenses, hidden fees, and upcharges.

Manual Testing: Primary Use Cases

Exploratory Testing

Software bugs are sneaky – they hide in-between the nooks and crannies of project requirements and user stories.

Structured exploratory testing helps uncover issues missed from automated scripts – before your customers find them.

Usability Testing

Usability revolves around the entire app-driven experience – something that requires human testing on real devices and in real locations.

Usability testing uncovers problem areas where the customer experience is not excellent both inside and outside of the app.

Localization Testing

Users will pass up products whose graphical or UI elements are incompatible with their culture, language, or preferred devices.

Localization testing matches your app with an expert network of global testers to assure your app passes the “locals” test.

Exploratory Testing

What to look for:

- Experienced testers
- Flexible and burstable teams
- Important bug detection vs. duplicates
- Global coverage

Use Case:

Testlio helped a workplace productivity app explore issues uncovered from user feedback left on App Reviews. We developed exploratory test plans, assigned the real-world scenarios to experienced testers who reviewed the end-user complaints, and applied their skills to reproduce the issue and uncover new items. Common problems discovered included long loading times, orientation changes, search bars, speed, performance, and stability issues related to crashes, flashes, and blinking. These are the types of problems that automated testing often misses.

Usability Testing

What to look for:

- Usability recommendations to improve customer experience beyond the app and into the physical world

Use Case:

A retail banking app asked our localization and usability testers to share their regional perspectives on searching, finding, downloading, and using the app for the first time. We created usability interview scripts to capture tester feedback to measure the level of comfort associated with sharing secure information during onboarding. We also conducted usability tests to evaluate the entire onboarding experience and to provide suggestions for making it better. We shared the findings and recommendations with the client design, marketing, and engineering teams to improve the end-to-end customer experience.

Localization Testing

What to look for:

- Design, language, and bidirectional UI issues
- Confirm that localization efforts don’t create breaks

Use Case:

A leading social networking and photo-sharing app is available in 45+ languages throughout the world. The engineering team continuously tests to assure the app is pleasing to global users while checking that localization efforts haven’t resulted in new breaks. They selected Testlio’s European-led client services team to help manage round the clock quality assurance, including functional, linguistic, payments, and localization testing.
Testlio selected 160 testers after screening 6000. The elite group of testers currently perform 1000+ testing hours per month.

Now that we’ve uncovered the best use cases for automation and manual testing, let’s see how to combine the two approaches for maximum speed and coverage. Two real-world examples help you visualize this holistic approach in action.

How to efficiently integrate automated scripts with manual testing

Automated tests take a predictable path. Humans brave the unknown. Together, they provide full coverage to get the job done. But, what's the best approach to merge the strengths of automation with the strengths of human testers?

Where Automation Excels	Where Humans Excel
Unit tests and integration tests when the functionality under test is very stable.	UI and UX testing to test the look and feel of an app
Supporting DevOps with repeatable tests running in parallel to improve results velocity and to provide development teams fast feedback.	Thinking of quality as a solution rather than acting like a robot. For example, taking the time to identify negative reviews in the app store and understanding the user needs and voice enough to build thoughtful test plans.
Repetitive, and data-intensive tests	Judgment: if an automated test fails, it's usually a human to judge whether the test or the product code is at fault.
Happy path testing using known inputs and a clearly defined expected output.	Humans test combinations not anticipated in the automated test cases. They also conduct exploratory testing for higher-level assessment of complex business flow and real-life situations such as interruptions and display image orientation.

There are many moving parts within a typical development cycle. But a few best practices will help get you started.

- Seamlessly integrate the design, build, and management of existing automated tests runs with manual tests into your engineering workflows
- Handoff inevitable flakey automated scripts to human testers to build and produce overnight results for an automated-like experience
- Integrate testing reports and insights to review and validate all automated and manual test results and defects

Day in the life of the automation and human-based testing partnership

Sunrise	Sunset	Before Each Release
Automation Unit tests with each build API tests for each build throughout the day	**Automation** Build pushed to tools like SauceLabs or BitBar to automate extensive system & cross-browser tests	**Automation** If necessary, run performance & scalability tests Perform deployment validation tests
Human Developers and QA pair up to conduct manual exploratory testing during the sprint Automation engineer write new automated tests Validate automation test failures and manually test if necessary Recommend which manual tests could be automated	**Human** Build also pushed to human testers in various time zones to run daily regression tests Validate automation test failures and manually test if necessary Recommend which manual tests could be automated	**Human** Functional regression testing Structured exploratory testing Functional and Usability testing for new features Native speakers conduct localization tests for a new language

With Testlio's fully-managed or co-managed options, you gain a team of experts for test strategy, test management, reporting, and partnership. This approach leads to fine-tuned efficiencies and the removal of coverage gaps.

How Testlio Integrates Manual and Automation Testing

01 We recommend and help manage the right blend of manual and automated testing tools

02 We create and maintain your test automation framework, including infrastructure

03 We align and integrate your engineering workflows with Testlio's testing methodology and powerful testing software

04 We recommend which manual tests could be automated

05 We design, build, and maintain accurate test scripts

06 We manage test runs and setups for you

07 We tap into our global network of experienced, on-demand testers to work in parallel with automated scripts, only when necessary

08 We review and validate test results and defects, while also taking care of noise and flaky tests

09 We validate automation test failures and manually test if necessary

10 We create actionable bug reports for true automated tests failures and integrate all reporting into your workflows

Case Studies

National TV Network

One of Testlio's clients is a national TV network that produces apps for all of their major television and mobile device platforms. They require a mix of manual testing to ensure a great user-experience, as well as automated testing to verify the extensive tracking data that they collect. Together, machines and humans worked in parallel to swarm the testing surface in short bursts to quickly identify issues and deliver rapid results. To quickly tackle functional, location, and livestream testing, the TV network combines Sauce Labs automated suites with Testlio's global team of on-demand testers.

For automated testing, the client uses a continuous integration pipeline to fire off a set of tests each time the app is built, and a more comprehensive suite runs nightly. Each evening, the CI system sends the build to a device farm where automated tests run in parallel. The initiation of the tests is completely automatic and controlled by the CI system. When the results are ready, the automated suite sends the results back to the client.

In parallel, Testlio helps them manage manual testing. The CI system sends the build to the Testlio platform, then the testers start their test cycle and run it overnight. Testlio sends the results for review first thing in the morning. With both methods -- automated tests running in the cloud and Testlio manually testing -- they run overnight and provide results in the morning.

Financial Services Application

Another Testlio client is a very successful financial services app, with over a million active users. The app, and its backend, comprise at least a million lines of code, and they release updates every two weeks. To maintain this release velocity, they have a tremendous amount of automated tests that ensure the baseline behavior of the app remains the same. Financial data and workflows are natural allies of automated tests because the results are predictable.

Adding new tests and keeping the others is a very difficult job, and the client's internal QA team is mostly focused on their automation suite. But, they also rely on a QA vendor like Testlio to complete their manual testing.

We bring a team of 20-30 expert testers to do all of the human-based testing in one day, which allows the client team to concentrate on automation. These manual tests include exploratory testing, regression testing, and usability testing.

Testlio is the leader in managed app testing

80
employees

10K+
testers

83
client NPS

1.6B+
collective users

6.5M+
tests executed

350K+
issues reported

Commerce & Retail	amazon	kw KELLERWILLIAMS	SONIC
Finance & Banking	AMERICAN EXPRESS	acorns	BBVA
Health & Wellness	athenahealth	PillPack	STRAVA
Learning & Education	lumosity	Quizlet	UDACITY
Media & Entertainment	CBS	FOX	USA TODAY
Mobility & Travel	AWAY	DAIMLER	Hotels.com
Productivity & Comms	Microsoft	grammarly	wework
Social & Gaming	FORTUNA	Top 10 Social Network	TRILLER
Software & B2B	SAP	DataRobot	HqO
Sports & Fitness	MLS	MANCHESTER UNITED	NBA

See the Testlio Approach to Networked App Testing

Learn more