

Networked Testing:

# New Perspectives on Remote and Distributed QA



# Table of Contents

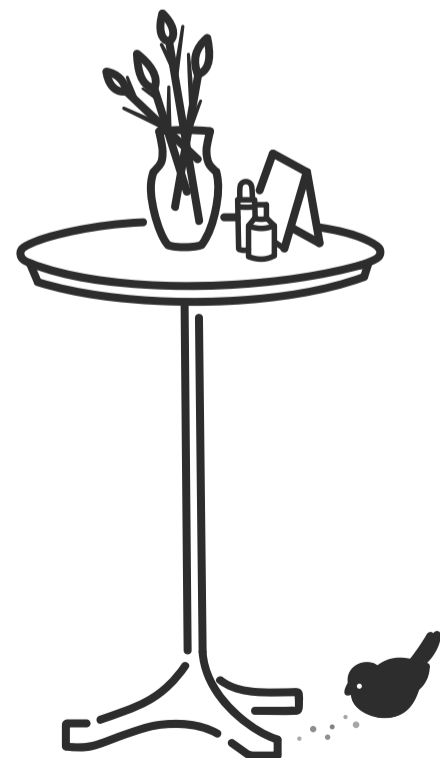
Global trends related to how we live and work	4
Sidebar: Changing consumer behaviors	4
How and where we build and test software has changed, too	5
Sidebar: Another Trend - Distributed Work - Use Case	5
Today's remote-only challenges for agile software and QA teams	6
Top 8 challenges for remote software development and QA teams	7
20 Best Practices for Agile Dev and QA to Go Remote	9
New Ways of Communicating, Connecting and Collaborating	9
Adjust the Dev + QA partnership	10
Leverage Distributed Testing for Speed and Efficiencies	11
Refining Your Remote Test Management Approach	12
Questions to ask when considering a distributed testing approach	14
About Testlio and Distributed Testing	16

# Global trends related to how we live and work

The COVID-19 pandemic has affected consumer behaviors worldwide. As time passes, many of today's practices will become tomorrow's long-term habits. Digital transformation is at the forefront of our changing behaviors.

People are ordering in vs. dining out, skipping the gym for live-stream yoga classes, supervising their children's online learning, dating virtually, video chatting with doctors, and live-streaming entertainment in place of attending in person.

The expectation is for digital experiences that align seamlessly with our current lives. Organizations from nearly all industries are accelerating their digital initiatives to meet the demand for safety, experience, and convenience.



## Changing consumer behaviors

- Fear of catching or spreading COVID-19 means many consumers are proceeding with caution. A Q2, '20 [Deloitte](#) survey shows that 53% of people are not comfortable going to a store. Willingness to go to a bricks-and-mortar store is a new consumer metric with two distinct segments: cautious vs. comfortable. The needs of the "cautious" segment have dramatic implications for building and testing end-to-end, app-driven experiences to enhance pre-COVID shopping behaviors.
- According to [Accenture's summer 2020 report](#) 10 Consumer Trends Impacting CPGs, most consumers are and expect to continue using:
  - In-app ordering 51%
  - Home delivery 45%
  - Live chat 41%
  - Virtual consultation 35%
- Another Accenture study published in Q2, '20 notes that 80% of consumers feel more connected to their communities during COVID-19. And 88% of those people expect that this increased connection is the new normal. More than ever, consumers value the recommendations of family, friends, and neighbors for purchase decisions.

Sound Familiar?

*Which teleconferencing apps do you use? What's been your experience with curbside pickup? Have you found any AirB&B rentals that sanitize?*

# How and where we build and test software has changed, too

Sheltering-in-place mandates to contain COVID-19 forced many organizations in the technology space to quickly embrace remote and distributed work. Employees set up home offices while juggling new work-from-home realities (like virtual stand-ups in between running a daycare for their children). Software teams have had to rethink their approach to coordination, iterative work, meetings, and cross-team coordination.

What's emerging is a need for flexibility and openness to different approaches to collaborating, partnering, and building great software. As time passes, working from home is another habit that will likely stick post-pandemic for reasons beyond dramatic savings in office space expenses.

- The evolution of fast wifi plus remote tools like Zoom, Microsoft Teams, Slack, and virtual whiteboards turn virtual work into more of a norm.
- Commutes have increased in most major cities each year. A [study](#) from the University of West of England, suggests that adding 20 minutes to your daily commute impacts job satisfaction as much as a 19% pay cut.
- Owl Labs notes that remote work [cuts employee turnover](#) by 25%. After several months of telecommuting during the pandemic, data are emerging to suggest an increase in productivity.

*"We know the future will be increasingly digital, flexible, and remote-friendly, or even remote-first."*

-  
*Microsoft Workplace Insights research, July 2020, HBR*



## Another Trend - Distributed Work - Use Case

With COVID-19, many have realized that work can easily be distributed across the globe and time-zones to increase productivity and improve quality, affordably. For example, distributed software testing comes in handy for agile engineering and QA teams short on testing resources and searching for flexible part-time talent. With platforms like Testlio, an engineering team in New York can upload their builds to a distributed team of on-demand testers before heading home. The global network of remote personnel in various time zones take on the testing while the engineers sleep. An issue report arrives first thing the next morning so the software team can fix the bugs and release with confidence. Distributed and experienced testers can flex in and out of assignments, too, helping with launches, market updates, significant events, and more.

# Today's remote-only challenges for agile software and QA teams

Agile software development requires continuous delivery, parallel development, responsiveness to customer needs, and a willingness to change course during continual iterations. To build as fast as possible through agile QA, developers need to automate testing where possible and frequently hand off small functional sets to testers. By testing small increments in short testing windows, quality rides shotgun with development and is built into an application from the very beginning. The result is faster time to market and applications that are far more responsive to customers' needs.

*"Right now, the virus seems like an accelerator for digital change that was already underway . . . the surprise has been to see the resistance to this digital change suddenly evaporate. What organizations resisted for a decade is now core to survival and innovation. It is exciting, because this digital mindset will persist, and it is highly unlikely companies will try to return to what worked prior to the pandemic.*


-  
**Michael Hendrix,**  
*partner and global design director,*  
*Ideo via [Fast Company](#)*



## Top 8 challenges for remote software development and QA teams

### 1. Frequent releases

Today, brands must deliver high-quality apps with flawless end-to-end customer experiences. To keep up, many web teams deploy constantly (checking in code up to hundreds of times per day). In addition, many mobile apps teams release every 1 or 2 weeks in conjunction with sprints. Testing must be fast and ready to burst into action at critical moments. A recent sweep of release frequency and average reviews on the top 30 iOS shopping apps reveals that app producers that release 3+ times per month have a 7% overall higher app store review (4.8 vs. 4.5).



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

### 2. Automated testing

Agile development depends on automation for speed and scale. However, while fast and obedient, automated tests sometimes miss issues a human can easily detect. Combining automated and manual testing into DevOps processes poses a challenge. How do you network automated and human-based testing into code as it is checked in – ensuring that unit, visual, functional, and other tests are completed?

### 3. Hurried fuzziness

As companies adapt to new consumer behaviors with new products and features, rapid shifts in product focus can degrade user stories' clarity and acceptance criteria. This hurried fuzziness impacts code, the volume of work for developers, and the volume of bugs.

### 4. Added workload

In the same way that development needs to proceed continually in agile, so must QA. Quality assurance teams don't have weeks to analyze requirements and create test cases. Instead, they must think on their feet and rapidly develop new test plans as products evolve. For example, QA may be testing one feature while the development team is working on another so that development can quickly improve an app based on QA feedback. Building an executable testing plan for work from home testers may require more creativity and time.

### 5. Drifting teams

Remote teams still connect, but the loss of face-to-face, impromptu conversations, and contact between developers, PMs, and QA engineers may cause drift. A lack of clarity on requirements can lead to more bugs.

### 6. Remote tools

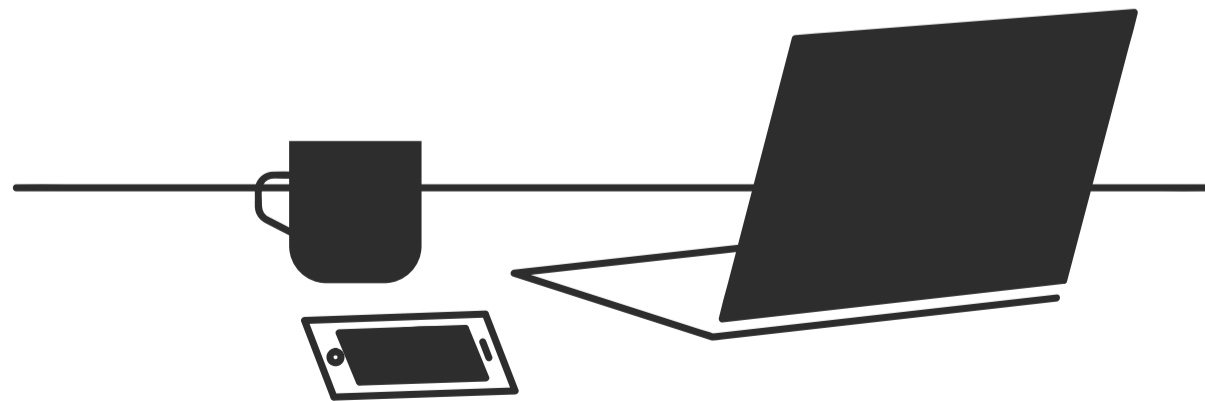
Remote and secure connections for workstations outside the office add a layer of complexity. Moreover, work-from-home internet bandwidth inconsistencies lead to frustrating video conferencing encounters and work productivity issues.

## 7. Lab loss

When in-house labs become less accessible to work-from-home teams, the need to test on a magnitude of platforms, operating systems, devices, and global locations become particularly challenging. Shipping devices to at-home testers is inefficient and offers a host of logistical and sanitation issues. Even before COVID-19 work-in place mandates, office or device labs couldn't provide enough real-world experiences to account for weak WiFi, different network configurations, or the diversity of locations where users consume their apps – from homes, commutes, stores, etc.

## 8. Outsourced frustrations

Even the best agile QA teams experience capacity limitations when it comes to localization, geo coverage, and surges in testing demand. Traditional crowdsourced testing services fall short on tester management, consistent and expert testers, and testing in agile windows with rapid turnarounds. Also, like in-house QA teams, many outsourced device lab partners have lost access to their real devices. They, too, have had to shift to a remote-only environment that impacts agile sprints, real device testing needs, and frequent release cadences.



# 20 Best Practices for Agile Dev and QA to Go Remote

It goes to say that without Agile QA, there is no Agile Development. And without Agile Development, companies face significant lags in application development time, and potentially delivering applications that lack a world-class digital experience.

New challenges impact how software development and quality assurance gets done. We must rethink how we do things and adapt to the remote-only reality for the short-term, and likely long-term. We've curated some of today's best practices for making the transition to remote and distributed QA to not only help you adapt to change, but also enhance your agile development processes.

---

## New Ways of Communicating, Connecting and Collaborating

Maintaining a culture of connection and trust is vital for remote engagement and a sense of psychological safety. Embracing new ways of communicating and connecting means your team is more likely to ideate and propose risky ideas. Also, extra communication effort goes a long way to ensure employees feel a sense of belonging while maintaining a fail-fast-learn-fast mindset.

- Continue your stands-ups (virtually) and consider additional 1:1 time and midday blocker check-ins. Frequent check-ins help build trust and avoid performance feedback misinterpretations.
- Encourage open communication between your testers rather than isolated, individual work. This nudge gives them permission to cooperate for work distribution, problem-solving, and rapid reproducibility checks.
- For remote and distributed teams, consider remote pair programming tools like Zoom or Tuple (for instant, unscheduled, collaboration via video/screen sharing), Cloud9 or CodeEnvoy (cloud IDEs for hosted development environments), or Visual Studio Pro and Teletype for Atom (live share extensions for popular editors). They improve real-time collaboration and lead to higher quality code and fewer bugs by making continuous code reviews and synchronous collaboration easy.
- View your project collaboration tools as the single source of truth to prevent fractured communications (e.g., JIRA, GitLab, GitHub, and Bitbucket).
- Consider a QA knowledge repository using tools like Google Docs or wikis for others to gain visibility, ask questions, and offer feedback.

## Adjust the Dev + QA partnership

Customers expect new features but won't accept a bad experience. Agile QA works closely with dev to detect risky builds early on in the process. Early detection of functional and usability issues serves as a signal. Not to hold off on developing a feature, but to consider building new features with more intuitive gestures and simpler user flows.

- Very often, the questions asked by the QA team will prevent bugs from happening in the first place. When quality assurance teams participate at the start of the iteration, they begin interpreting requirements, initiating test plans, writing test cases – alongside the development effort. Collaborate early to build a testing plan while development is taking place and invite your testing team to virtual stand-ups if/when helpful to make sure they are aligned with requirements and expectations.

- Integrate the design, build, and reports through a single platform to review and validate all automated and manual test results and defects - side-by-side. Handoff inevitable flakey automated scripts to human testers to build and produce overnight results for an automated-like experience.
- Make sure all specifications are concise and without ambiguity. Looping testing and QA processes into agile sprints can only occur when requirements are not open to more than one interpretation.
- Provide clear documentation (often video-based), clear test and run instructions, and consistent issue reporting guidelines to help remote teams move fast.
- Ensure build testability. Builds with broken functions or crashes are returned by testing teams. To help keep the entire release schedule on track, make sure that developers have taken an initial pass over the build to verify its testability.

## Leverage Distributed Testing for Speed and Efficiencies

Networked testing is an evolution of in-house, outsourcing, and crowdtesting best practices and aligns with the principles of agile development. It builds on the notion that excellent customer experience begins with the best quality and most valuable software. Through efficient and sustainable processes, Networked Testing connects the day-to-day work of engineers with distributed testers to build and release great products at speed, with complete coverage, at compelling economics.

When the pressure's on, and a new release is due, leveraging a distributed testing approach is like having an expert, on-call team with 1000s of real device/OS combinations. Distributed testers from any location perform functional, usability, localization, and other forms of manual software testing.

- Leverage on-demand, large (often 10 to 50 or more), distributed teams of freelance QA experts to rapidly begin and conclude testing.

- Maintaining speed means continuous quality and agile QA is a 24/7 endeavor. Distributing the work to testers in other time zones is the fastest path to early week releases, global coverage, and evening and weekend testing while your engineering team is resting.
- Rethink how you gain complete device and location coverage. While device labs once offered easy access, their cleansing methods and centralized locations sometimes mask real-world experiences. Accordingly, testing thousands of device/OS combinations via local, loaded, real-world hardware — in the hands of humans — can highlight issues not otherwise discoverable.
- Avoid individually-oriented bug bounty testing models because they incentivize the wrong behaviors. Instead, look for services that manage and incentivize testers to cooperate. This approach incentivizes thoughtful testing – leading to higher quality bug detection, collaborative problem solving, and rapid reproducibility checks.
- Pair testers together for complex multi-user features.

*"Major testing should occur in the crevices of core product and engineering team work periods. For many companies, this means large-scale testing primarily occurs overnight and/or on weekends — in relation to primary engineering team time zones. To optimize this approach, a multi location team of testers is often optimal."*

-  
**Steve Semelsberger, CEO of Testlio**



## Refining Your Remote Test Management Approach

Combining test management software, integrated systems, services, and a network of testers helps distributed testing integrate into agile development processes. Bringing all of these elements together is not only possible through a networked testing partnership, but often packaged with fully-managed or co-managed options.

- Connect all of your testing systems to a secure cloud-based test management software to gain visibility into edits, version control, and testing progress. The software can also help you manage builds and quickly vet, schedule, and compensate a network of distributed testers from multiple time zones with 1000s of real-world devices. A single platform also allows for test definitions and issue synchronization to tools like JIRA, Azure, Trello, Asana, VSO, Github, and more.
- Ensure your distributed testing coincides with sprints.
- Combine test automation for unit, API, and UI tests with manual testing for exploratory, usability, and localization testing. But don't stop there, integrate your automated and manual testing to enable CI/CD builds, DevOp workflows, project management, monitoring, and more.
- Agile principles offer that working software is the best measure of progress. Real-time test results help you measure the progress of the test cycle and quickly optimize team performance. You'll also want to pull historical reports and analysis on issues regularly to assess which features are most prone to breakage without guessing.
- When partnering with a Networked Testing company, make sure that they deliver quality reports about each test cycle. Otherwise, you'll only have the bug reports, but no analysis into the breadth and depth of testing or the found issues. It can also help determine if any distributed testers don't produce as many reports as their colleagues.



# Questions to ask when considering a distributed testing approach

Shifts in corporate supply and demand due to the global health event require more flexibility to staff up and down as the market environment changes. It's vital to have immediate access to specific skills, expertise, and experience when and where it's required. Quick access to on-demand resources helps agile teams take on surges in demand required to maintain frequent releases.

With thoughtful research, consideration, design, and partnerships, your distributed testing efforts can soar. Both emerging growth companies and the [Fortune 500](#) integrate distributed testing best practices into their sprints. While not complicated, taking the next step may be daunting.

Whether you consider Testlio or another partner to guide you into distributed testing, we've created a checklist of questions to ponder:

- Can our current work-from-home team keep up with a demanding release schedule while maintaining the highest quality customer experience?
- Do we have the expertise or time to fuse both automated and manual testing into our processes?
- Do we have the right mix of tools for remote teams to communicate and collaborate with a distributed team of testers?

- How can we address location and device coverage in a more efficient, expansive, and cost effective way?
- What's the process for managing dozens or more distributed testers from different time zones?
- Do we need a fully-managed testing partner or would a co-managed option better align with our processes?
- How are distributed testers vetted for experience? How many devices do they own, and the number of languages spoken? Will they be consistent from project to project?



# About Testlio and Distributed Testing

Testlio is the leader in managed app testing. With robust client services, a global network of expert testers, and a comprehensive software platform, we provide a suite of burstable, flexible, and on-demand testing solutions: end-to-end, usability, localization, functional, payments, and more. Testlio clients include Amazon, American Express, CBS, Hotels.com, Microsoft, the NBA, and Sonic. Collectively, our clients power over 1.6 billion users. When apps must perform brilliantly, Testlio helps ensure world-class customer experiences. In any location. On any device. In any language.

## Distributed Testing with Testlio

Distributed Testing is like having an expert, on-call team with 1000s of real devices. Distributed testers from any location perform functional, usability, localization, and other forms of manual software testing.

- Scale rapidly with expert, on-demand testing, only when you need it
- Testers vetted for experience and incentivized for collaboration
- Filter testers by skills, location, and devices via local, loaded, real-world hardware
- Choose to co-manage or let Testlio fully-manage your testing





# See the Testlio Approach to Distributed Testing

[Learn more](#)