www.globalapptesting.com
info@globalapptesting.com
UK +44 (0) 330 808 0106
US +1-800-461-2670

linkedin.com/company/global-app-testing
twitter.com/qaops

Global App Testing

Exploratory Testing for Enterprise

Contents

Introduction

Imagine your software suddenly shut down the phones of 60,000 customers. Would they stay loyal to your company or would they light up social media with negative PR?

On Jan. 15, 1990, AT&T waded into a similar scenario when the company's 114 long-distance switches no longer permitted calls to be placed. For nine panic-stricken hours, AT&T and law enforcement fretted over what appeared to be a gigantic cyberattack.

It wasn't.

It was a single bug in AT&T's software. That little bug initially cost the company $60 million in lost revenue.
What happened to AT&T may sound like just a nightmare, but it could easily happen to your business. Project lifecycles have shrunk considerably over the years. Speed is essential to your company's growth and progress, but this breakneck pace leads to a heavy reliance on automated testing.

Many software teams do not have the resources to adequately test their code. Software glitches that make it past your automated testing (and they will) may result in irreversible damage to your company's brand and customer base.

So, businesses need to protect their brand and bottom line from faulty software by incorporating exploratory testing into their software delivery lifecycle, but not by using your father's approach.

Business Drives Innovation

Software quality is critical for modern, product-focused companies and this quality is tightly linked with design methodology.

The software creation process has significantly evolved over the last few decades.

Computer use expanded in the 1960s and 1970s and programming approaches changed to meet growing demand. As PCs proliferated, businesses increasingly sought technology to help address the needs of the marketplace. Software development ramped up to match the needs of businesses, which were experiencing whipsaw change through technology and increasing globalization.

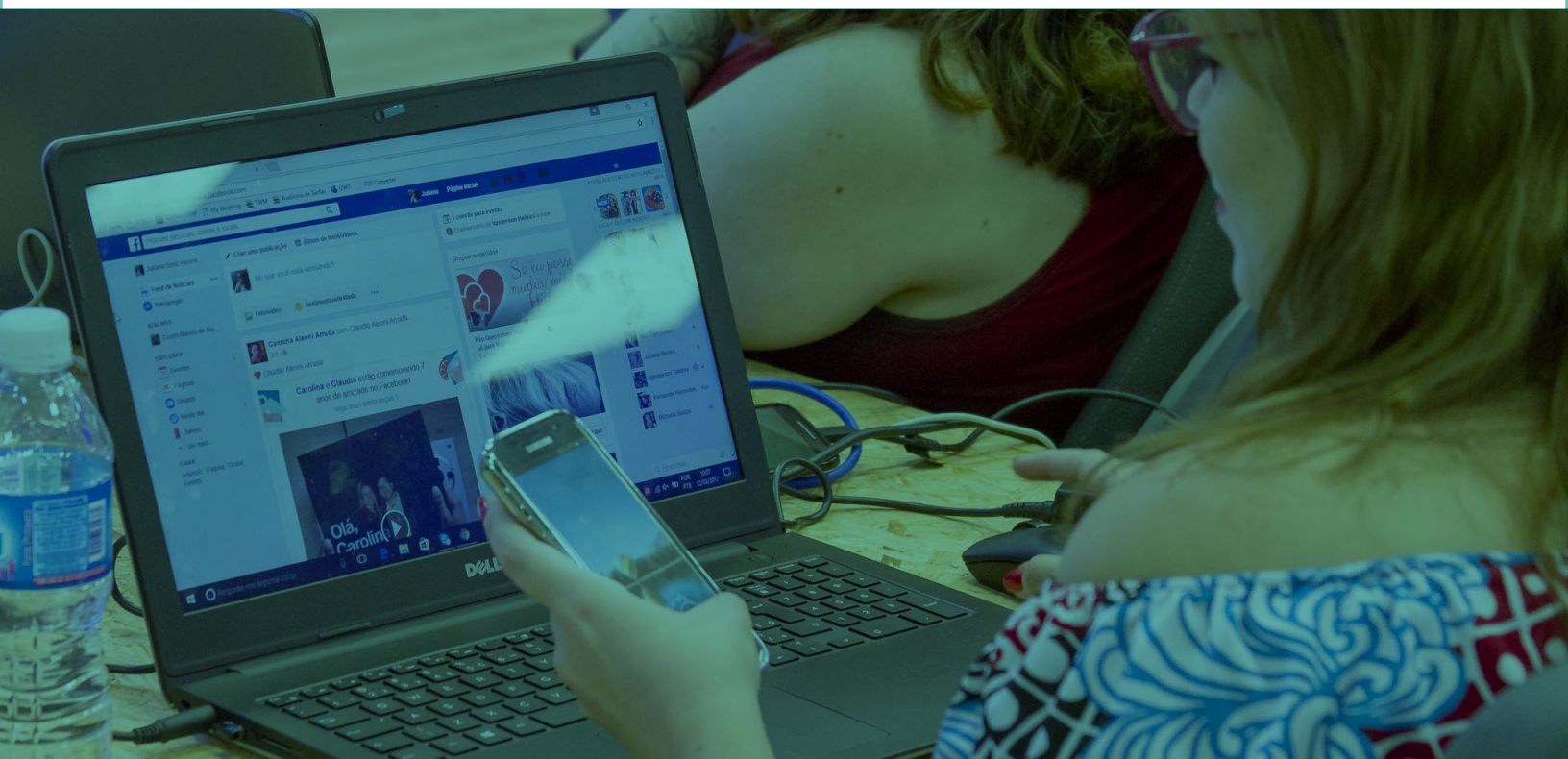
While businesses raced ahead to grasp new opportunities, software teams struggled to effectively support growth with bug-free code. Programming was still in an angst-addled growth phase. It could take years (or more than two decades in some cases) to bring a software product to market or implement it at a company.

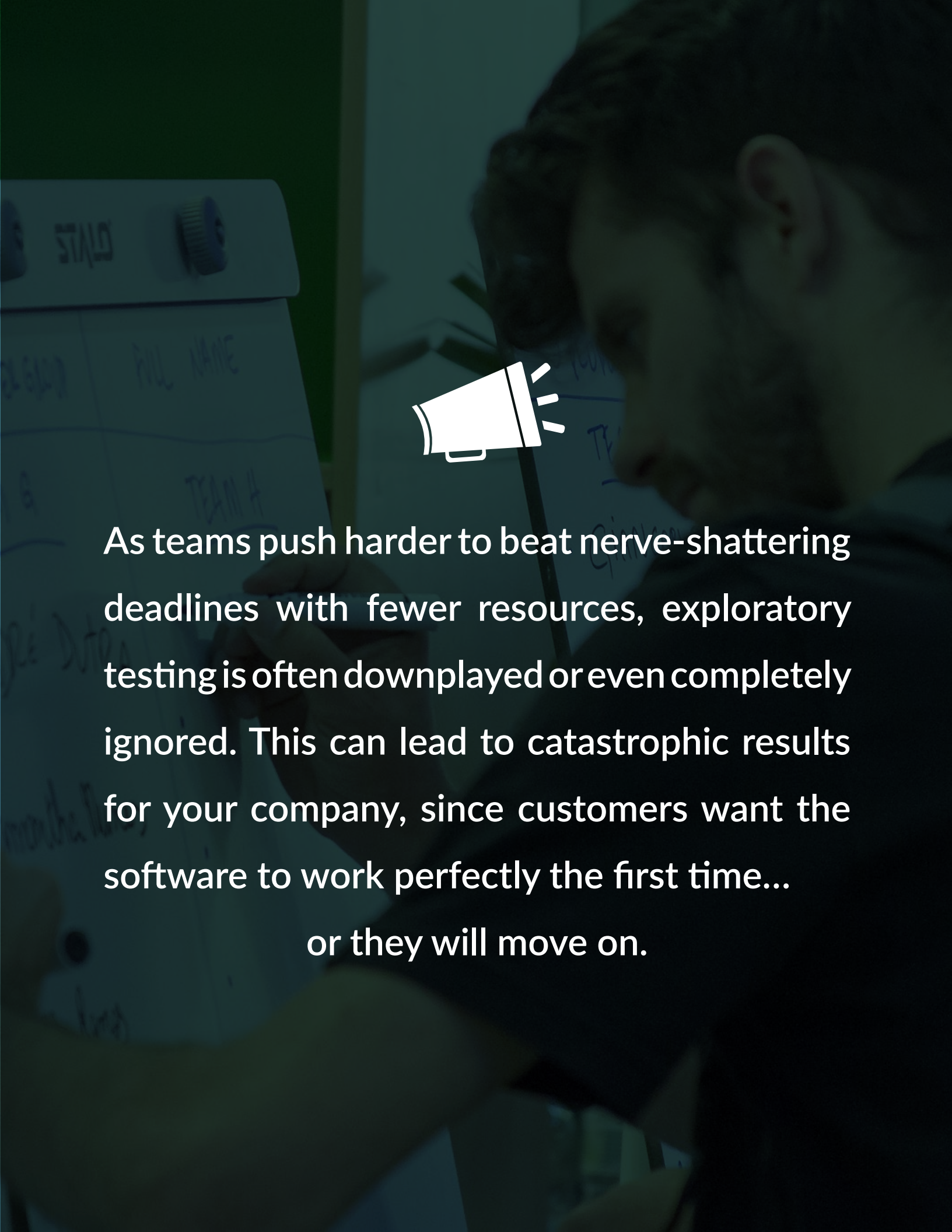
When programs were delivered, they sometimes employed dated languages or technologies. The software may not have functioned on competing systems or failed to fit a company's business model or objectives. The rapidly changing digital world needed a way to speed up the development cycle while still delivering quality products.

Initially, the waterfall methodology took hold, but even this approach was not nimble enough to pivot for constantly changing specs. Agile, developed in 2001, was the natural response of developers frustrated by outmoded software approaches and languages. Agile instituted a new way to manage software development projects - and it implemented testing every step of the way. The problem, of course, is that the testing is largely automated.

Automated testing sounds great in theory, but the practice is far from ideal. This is largely because automation cannot ensure the actual quality of the software tests Developers face three challenges: lack of time to develop scripts, code difficult to test via script, and an inappropriate testing approach.

Automated testing speeds up the cycle, but creates a gap in the finished product's quality. In essence, automated testing leaves out the most important part - discovering the unknown - and gives teams a false sense of code integrity.





As teams push harder to beat nerve-shattering deadlines with fewer resources, exploratory testing is often downplayed or even completely ignored. This can lead to catastrophic results for your company, since customers want the software to work perfectly the first time…

or they will move on.

Satisfying Your Customers

Your target customer may have a unique profile, but one thing is true about all modern shoppers - they want quality products quickly. They will be loyal to your brand if they feel that you consistently deliver quality.

Gallup showed that creating loyal customers through brand engagement supports corporate health, and brand image plays a vital role in customer perception.

For years, Toyota cornered the market on reliability with the Corolla and easily attracted middle-class consumers with the allure of a quality vehicle. Now, compare that with the image of the Ford Pinto. There are, of course, tales of software product quality that hit closer to home. In 1996, the Ariane 5 crashed shortly after launch due to a software issue, a glitch that wasted almost $8 billion.

Customers with a negative experience are three times more likely to tell others about it, and the estimated annual loss to businesses from negative reviews is nearly $540 billion.

Other studies found that 44 percent of users will delete an app right away if it does not work properly the first time, and 79 percent of these consumers indicated that they would not make future purchases from the company's site.

It's no different for software companies, where
customers reward better producers with money and attention. Healthy brand images deliver strong returns. Software companies compete for the trust and loyalty of customers, but issues of quality are a constant threat to brand image and customer attraction and retention.

A study performed by Penn State University revealed some startling findings. The researchers found that negative online reviews had a powerful, detrimental effect on a company - so much that negative ratings actually made consumers avoid the business in the future.

Bugs in software can also create legal problems for your business, like when a government levies a fine for a malfunctioning product or when users file a lawsuit. Since glitches in your software are so expensive and damaging, it makes sense to utilize all the tools at your disposal to extensively test your product before it launches, including incorporating exploratory testing into your quality assurance approach.

Many development companies balk at exploratory testing, citing time constraints and budgets. Performing only automated testing, however, may be even more costly in the long run.

The Nest Incident

The Nest incident in 2016 highlights the challenges facing development companies;

Its self-adjusting thermostat was heavily promoted by utility companies in some parts of the U.S. and touted as a way for consumers to save money.

Millions of customers jumped at the chance to save on their monthly bills but trouble quickly brewed.

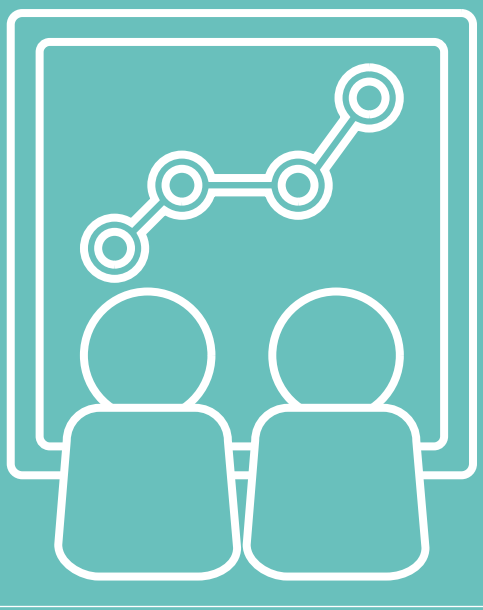
When icy January weather settled over New York, many users thought they were in for a cozy evening. However, on one of the coldest weekends of the year - in the middle of the night - a system bug drained the device's battery, effectively shutting off thermostat control.

Nest was slammed in the media, as New York Times writer Nick Bilton related his personal story of waking up in a rapidly cooling home with an infant. He wasn't the only writer to take on Nest or deliver more bad news about the product's earlier failings in a high-profile media outlet.

Software bugs result in the direct loss of customers, but there are other repercussions, as well. In 2016, software glitches cost businesses an estimated $1 trillion, but that number does not consider how many customers were scared off by negative reviews before even trying a product.

The Limits Of Automation

Automated testing offers speed and it certainly does have a place in a project's lifecycle. It can help catch routine, common problems, but it will not address everything, especially if your testers do not evaluate the code on a broad array of devices. Additionally, if you are not regularly auditing and maintaining your scripts, you may find that they have their own glitches.



For example, your software product must work effectively on mobile and desktop systems and negotiate the quirks of numerous operating systems. While there are similarities across systems, there are many more differences. So many edge cases are possible.

Imagine the variations that can arise in an interaction with a shopping app, such as if the customer does not enter the correct information in the correct order field, inserts code into text boxes, or clicks quickly on multiple buttons. Exploratory testing can focus on the range of interactions between a human and your software to reveal potential flaws or even highlight security risks.

Software interactions between your product and another company's systems are another significant issue. Take Blockchain's digital wallet problem as an example. In 2015, some users running its software on Android devices with the 4.1 or older OS found that the app was generously donating their cryptocurrency to other users without their consent. (Incidentally, this could have resulted in a major loss. In 2015, Bitcoin was trading at about $300, but hit over $16,000 in December 2017.) An exploratory approach that incorporates a broad range of software and devices delivers value across many platforms, staving off more significant glitches and potential multi-user failures.

Automated testing is, essentially, a binary approach to bug hunting; you check to see if a specific condition is true or false. It cannot identify if the test is appropriate for the code in the first place. Determining the type of testing approach is critical to the final product's quality, but it is sometimes difficult to determine possible use cases in a closed corporate testing environment.

Cem Kaner, who first coined the phrase "exploratory testing," is a professor of software engineering at the Florida Institute of Technology.

"The scripted approach means the test stays the same, even though the risk profile is changing," Kaner said.

In order to create realistic tests for your program, you need to create the right tests for many different platforms and people. You need the right blend of automated tests and manual tests. Our gaze naturally turns to the 500-pound gorillas in the room: time and resources.

In a recent survey, more than 50 percent of team leads noted there is not enough time to adequately test, while more than 40 percent said they do not even have the resources to effectively test. (Shockingly, 42 percent confessed that their automated solution does not even test for mobile device usage, which could have far-reaching impacts on their business as more customers go mobile.)

There are not enough devices at most companies to handle the scale needed to create a simulated real-world testing environment. Even the largest of companies lack the budget to operate a testing center that incorporates large volumes of devices with different networks, software, and operating systems, never mind the various complications arising from their geographic locations. Budget or time constraints make handling this testing in-house a liability.

It does not need to be this way. Costly software glitches can be caught before you go live. By choosing the right exploratory testing partner, you can effectively sidestep issues of costs, resources, and time.

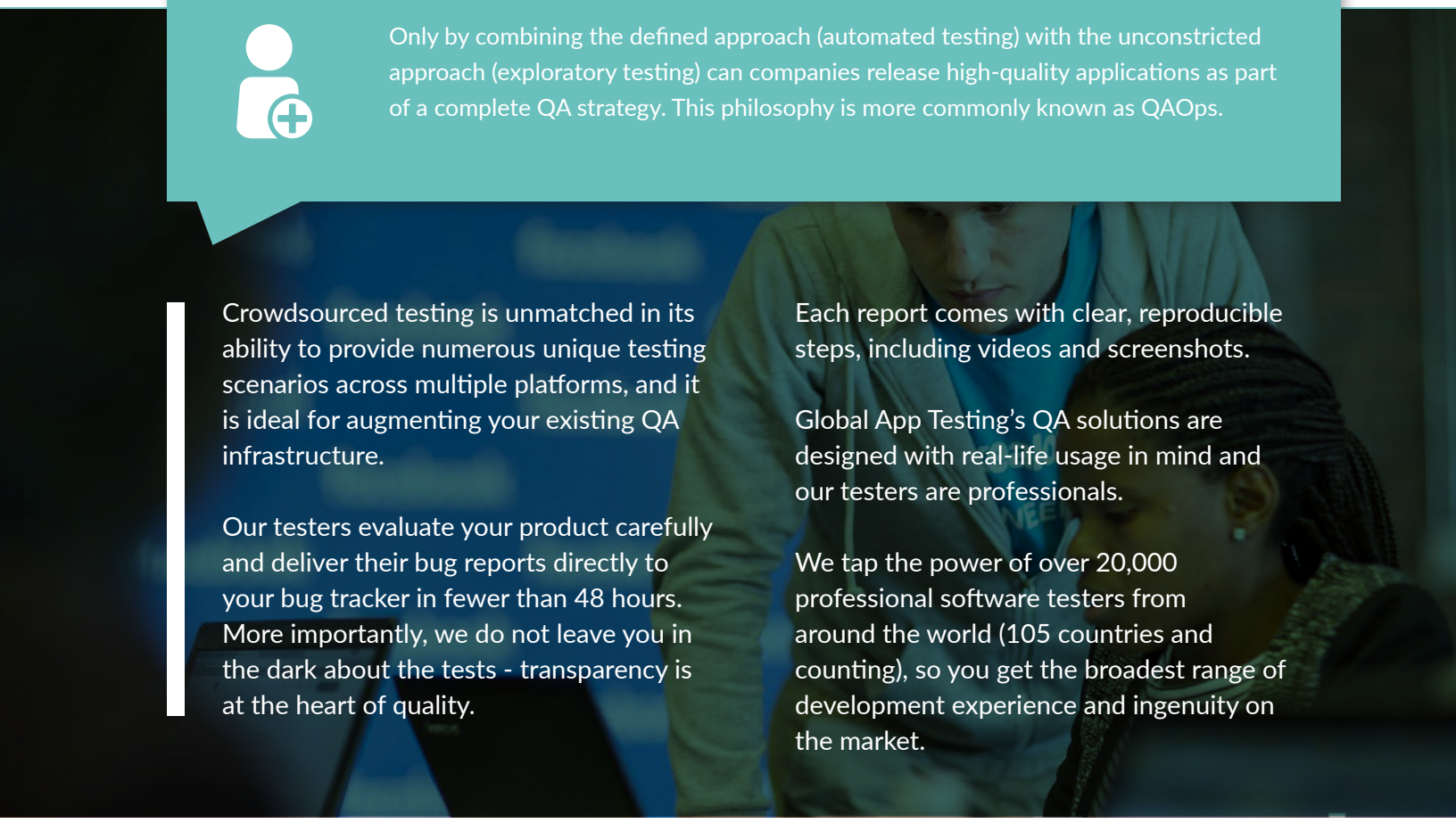
A New Exploratory Testing Approach

Automated testing will never disappear, but it needs support from exploratory testing. At Global App Testing, we understand how vital bug-free software is to your company's reputation and bottom line. Exploratory testing is a differentiator, an absolute necessity in today's fast-paced software marketplace.

We provide exploratory testing that is easy for your business to scale, manage, and afford. We can provide this service due to our unique approach in creating the testing environment with the right resources for your software, websites and applications.

Exploratory testing has traditionally been expensive and difficult to scale because you need a large number of testers in a variety of environments to really understand what could possibly go wrong. We crowdsource our testers to deliver value, so you can scale the number of testers you need depending on the project.

Only by combining the defined approach (automated testing) with the unconstricted approach (exploratory testing) can companies release high-quality applications as part of a complete QA strategy. This philosophy is more commonly known as QAOps.



Crowdsourced testing is unmatched in its ability to provide numerous unique testing scenarios across multiple platforms, and it is ideal for augmenting your existing QA infrastructure.

Our testers evaluate your product carefully and deliver their bug reports directly to your bug tracker in fewer than 48 hours. More importantly, we do not leave you in the dark about the tests - transparency is at the heart of quality.

Each report comes with clear, reproducible steps, including videos and screenshots.

Global App Testing's QA solutions are designed with real-life usage in mind and our testers are professionals.

We tap the power of over 20,000 professional software testers from around the world (105 countries and counting), so you get the broadest range of development experience and ingenuity on the market.

A Belief in Shared Success

We believe in shared success, so we reward our high-performing testers, ensuring a solid, reliable workforce that delivers quality testing to you in record time.

With access to almost limitless software and hardware combinations and experience with more than 10 platforms, we can definitely help you with your QA.

110,000+ bugs

20,000+ testers

6,800+ apps

Global App Testing

Next Steps

Our fast, accurate results mean that your team can focus on your core product. We've discovered over 110,000 issues in more than 6,800 apps, which has delivered powerful results to our enterprise-level clients. After utilizing our exploratory testing solutions, clients enjoyed a three to five times improvement in their positive reviews and significantly decreased their app store rejection rate.

Exploratory testing can help your company deliver quality software more rapidly and cost effectively than ever before. You get expert guidance during one-on-one consultation with QA experts and ongoing support via dedicated customer success managers.

Enterprise organisations are particularly suited to this model of testing, because it allows extreme scalability of QA operations without the need for additional headcount. Many organisations that we work with cite the ability to obtain vendor budget approval as much easier than additional headcount.

Contact us now and learn more about how you can use exploratory testing to boost your product quality and consumer satisfaction while scaling QA operations.

www.globalapptesting.com
info@globalapptesting.com
UK +44 (0) 330 808 0106
US +1-800-461-2670

linkedin.com/company/global-app-testing
twitter.com/qaops

References

http://www.phworld.org/history/attcrash.htm

https://www.cio.com.au/article/360112/epic_failures_11_infamous_software_bugs/

https://www.scientificamerican.com/article/pogue-5-most-embarrassing-software-bugs-in-history/

https://www.microsoft.com/en-us/research/publication/using-pre-release-test-failures-to-build-early-post-release-defect-prediction-models/

https://techbeacon.com/agility-beyond-history--legacy--agile-development

https://www.cfr.org/backgrounder/naftas-economic-impact

http://agilemanifesto.org/history.html

https://books.google.com/books?id=-izOiCEIABQC&pg=PT313&lpg=PT313&dq=development+teams+do+not+have+enough+time+to+test+code&source=bl&ots=YK6YemZZhx&sig=PXC9u9s8rYyaSZA0Bc6ukX0vq1k&hl=en&sa=X&ved=0ahUKEwi00ffrk_jXAhXRqYMKHQ1TAxkQ6AEIYTAJ#v=onepage&q=development teams do not have enough time to test code&f=false

http://www.gallup.com/services/169331/customer-engagement.aspx

https://blogs.sas.com/content/customeranalytics/2013/11/28/the-impact-of-negative-reviews-on-purchase-decisions/

https://www.globalapptesting.com/blog/how-bugs-impact-your-company-infographic

https://nest.com/press/nest-partners-with-energy-companies/

https://www.nytimes.com/2016/01/14/fashion/nest-thermostat-glitch-battery-dies-software-freeze.html

http://fortune.com/2016/01/21/nest-issues/

http://servicevirtualization.com/report-software-failures-cost-1-1-trillion-2016/

https://www.politico.com/story/2017/05/31/health-records-faulty-software-239004

https://nypost.com/2014/08/12/peeved-law-students-sue-after-exam-software-fails/

https://arstechnica.com/information-technology/2015/05/crypto-flaws-in-blockchain-android-app-sent-bitcoins-to-the-wrong-address/

http://www.kaner.com/pdfs/QAIExploring.pdf

https://www.globalapptesting.com/blog/what-ctos-must-know-about-qa-in-2018-infographic

https://www.statista.com/statistics/274774/forecast-of-mobile-phone-users-worldwide/

https://www.globalapptesting.com/community

https://www.globalapptesting.com/contact-sales?utm_campaign=[eguide]