# Vendor Landscape: Front-End Mobile Testing Tools

**Automated Testing Improves App Quality**

by John M. Wargo and Diego Lo Giudice

December 4, 2015 | Updated: December 10, 2015

## Why Read This Report

Mobile applications are the primary digital touchpoints for customers and employees, heightening the challenges for application development and delivery (AD&D) teams. Engaging, high-quality mobile apps boost revenue, while poor-quality apps put experience and brand reputation at risk. Testing is at the center of software quality. To cover the variety of devices, operating systems, and networks, AD&D leaders seek greater automation and better tools. This report details the landscape of mobile app testing tools, focusing on functional and user interface (UI) testing.

## Key Takeaways

**Test Automation Is Critical, But Won't Replace All Manual Testing**

UI test automation tools enable AD&D pros to test more aspects of an app, across a wider catalog of devices, faster. They will still need to manually test high-risk business and technical functionality which might be either too expensive or too hard to automate.

**External Device Labs Expand Options For Testing**

Mobile apps can now be tested more easily on a wider range of physical devices, helping developers deliver higher quality apps. Amazon and Google entering the fray will help drive pricing down and make this a more cost-effective solution.

**Real-World Testing Delivers Improved Insight Into App Experience**

Validate user experience (UX) and UI decisions through impartial, third-party eyes. Engineering and quality assurance (QA) know how the app is supposed to work, so they might be too close to see real flaws. Look at crowd-sourced testing options as a way to gain useful insights into the app.

# Vendor Landscape: Front-End Mobile Testing Tools

**Automated Testing Improves App Quality**

by John M. Wargo and Diego Lo Giudice
with Christopher Mines, Joseph Miller, and Tyler Thurston

December 4, 2015 | Updated: December 10, 2015

## Table Of Contents



## Notes & Resources

Forrester interviewed 20 vendor companies, including: CA Technologies, Dynatrace (Keynote), EPAM Systems, Experitest, HP Enterprise, InfoStretch, Micro Focus International, Mobile Labs, Oracle, Perfecto Mobile, Sauce Labs, SmartBear Software, SOASTA, Telerik, Testplant, Torry Harris Business Solutions, Tricentis, Ubertesters, UserTesting, and Xamarin.

## Related Research Documents

The Forrester Wave™: Mobile Infrastructure Services, Q3 2015

The Forrester Wave™: Modern Application Functional Test Automation Tools, Q2 2015

Improving Mobile App Quality Testing

FORRESTER®

Forrester Research, Inc., 60 Acorn Park Drive, Cambridge, MA 02140 USA
+1 617-613-6000 | Fax: +1 617-613-5000 | forrester.com

## Testing Assures The Best Mobile Moments

Software testing matters. In the age of the customer, software developers create and deliver the digital experiences that engage customers. In a mobile moment — the point in time and space when a customer pulls out a mobile device to get what they want in their immediate context — it's a mobile app that typically delivers that experience. In a mobile app, the features the app delivers matter; users won't want to use an app if it doesn't deliver some useful capability. Beyond that initial interest in installing an app, the app's performance and overall quality of experience are what keeps the user engaged with the app. As mobile apps deliver more features, apps become more complex and there are more places where app quality or app performance can suffer. Developers and app testers ensure the quality of mobile apps by thoroughly testing all aspects of the mobile app.

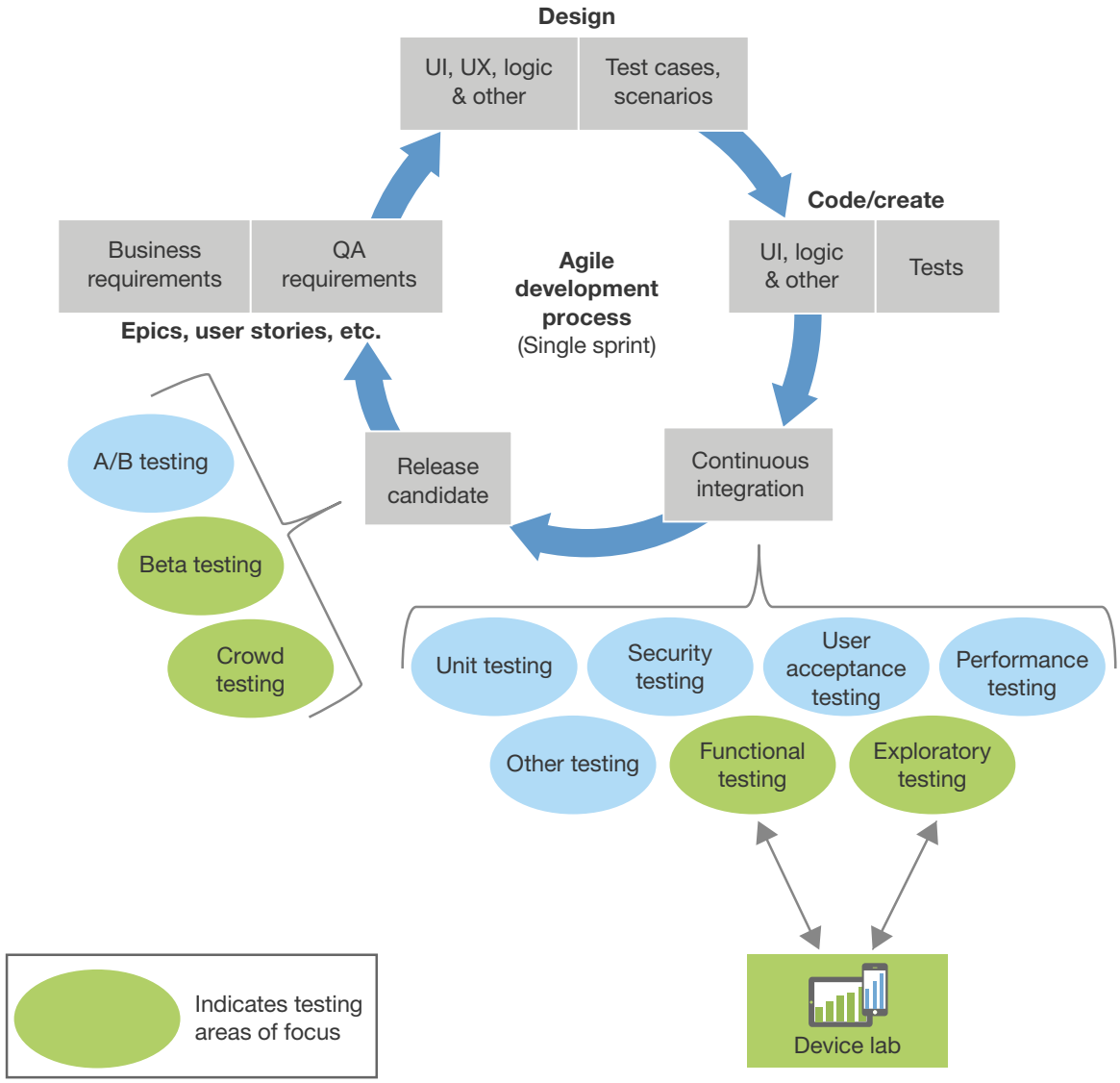
### Testing Methods Of The Past Don't Work For Mobile

Traditional testing process and practices no longer support developers' unprecedented needs for delivery speed and quality; and for mobile app developers and testers, it's even worse.[1] Testing tools for mobile have an important role in enabling a new app-dev life cycle that is fundamentally changed by Agile development and DevOps (see Figure 1). Many of the tests shown apply across multiple types of apps; this report focuses on the tests and enabling technologies that are specific to mobile app testing or have specific requirements for mobile testing.

Dev teams looking to deliver high-quality mobile apps need new approaches to testing because the clock speed of the dev cycle means they have:

- **Limited time to define proper test requirements.** There is no time to write lengthy test strategy, plans, and requirements during fast-paced mobile app development iterations.
- **No way to incorporate user feedback.** Deployed mobile apps get lots of feedback from users. Linear or waterfall development processes don't have the tight feedback loops that help dev teams incorporate ongoing feedback.
- **No time for silo-based handoffs.** When testers and developers work separately, they spend too much time crafting testing artifacts in order to share a common understanding on test needs, goals, and issues. Instead, testers and developers need to work together from the inception of projects.
- **Limited targets for manual testing.** Apps are too complex and dev cycles too short for manual testing of mobile apps. The only exceptions, where manual testing will make sense, will be exploratory testing of functions with high business or technical risk, and UX design and experience validation.

**FIGURE 1** Mobile Testing In An Agile Development Process

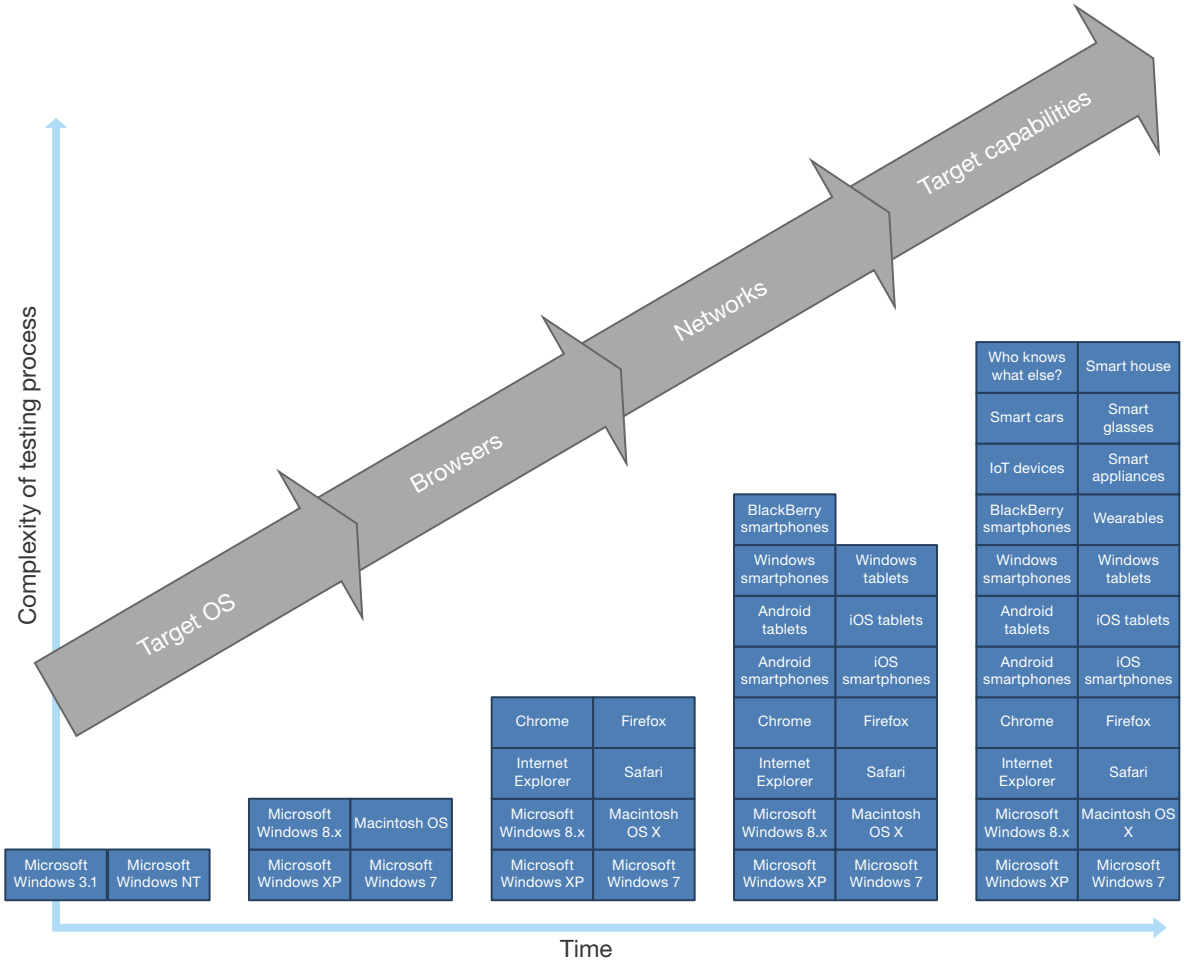



## Mobile App Testing Is Crucial, And Complicated

AD&D pros have to deal with a growing complexity of target environments and requirements for mobile apps (see Figure 2). Testing is right at the center of the hurricane. Business pressure for more and happier customers means constant pressure for new features and better performance. This translates into release schedules that are measured in weeks or days versus the few releases a year that mainframe or desktop apps required. Dev and test teams need to increase the speed of testing as app and environmental complexity increases. This means that mobile tests must:

- **Accommodate multiple target operating systems.** Even if targeting just the two most popular mobile OSes (Android and iOS), each OS release brings new APIs and capabilities. Apps, and the corresponding test suites, must be able to easily adapt.
- **Execute on dozens, if not hundreds, of devices.** Customers use an app around the world and on multiple devices. Each generation of devices brings new capabilities and new 'things' such as sensors that must be tested. Testing environments must be flexible enough to accommodate a large catalog of test targets (devices or emulators/simulators).
- **Deal with multiple form factors.** Responsive apps adjust the position and size of app elements based on available screen real estate and device orientation. Tests must be able to find the components being tested no matter where they are.
- **Use actual physical devices.** The device emulators and simulators provided to developers by device manufacturers and OS vendors are useful during development testing. However, many of these faux devices don't expose all of the hardware capabilities of a device and can't be used for performance testing. Only testing on the real thing(s) will expose an app's capabilities and limitations.
- **Simulate chaos.** Desktop and web applications operate in a more stable environment than mobile apps. Wildly varying network availability and bandwidth coupled with much more limited memory and processor capabilities directly affect app performance and therefore app quality.
- **Run in minutes or hours, not days or weeks.** Pressure on delivery teams for quicker releases coupled with a DevOps approach with continuous integration (CI) means that many tests must run whenever code is checked in or at a minimum once a day. This schedule precludes human involvement in most testing activities and drives dev organizations to implement automated solutions.

**FIGURE 2** App Testing Isn't Getting Any Less Complex




## Automation And Device Diversity Are The Keys To Success

Manual app testing is opportunistic; testers can only test what they decide they have the time and resources to test. This approach leads dev shops to focus testing efforts on important aspects of an application, and lets bugs in less important aspects be discovered by application users.[2] Additionally, many dev organizations test apps only on the most popular devices to save time and money, increasing the opportunities for apps to fail.

A better approach is to use automation tools to expand test and device coverage, testing more app capabilities across a wider range of devices. As Sukesh Soman, head of mobility and open source at Torry Harris Business Solutions, told us: "With automated testing, we're really seeing an improvement in app quality." The improvement occurs simply because an automated shop is testing more parts of the app more often. UI test automation:

- **Fits with today's rapid app delivery schedules.** Automation enables UI tests to happen cleanly within the CI process, executing automatically in sync with other CI tasks instead of as a parallel (and never-ending) task.
- **Frees QA resources to focus more on UX testing.** Validating that an app's UI works as expected is different than validating that the experience is easy and enjoyable.[3] Automated tools can validate that the app's UI components are visible and work as expected; machines can do this better than humans. Validating the mobile app's UX still requires human hands.

Validating that an app's UI works as expected is different than validating that the experience is easy and enjoyable.

- **Protects existing code from issues created through new development.** In today's rapid delivery timelines, QA departments may focus testing on new or updated features and leave testing of old features to last-minute regression testing. By automating all tasks, every feature of an app, old and new, is tested every time, protecting new code from breaking old code.
- **Compresses the development feedback loop.** Existing feedback loops from testing to development, providing testing with an efficient way to notify developers when issues are found, are a key part of the testing process. Automation shortens that loop, getting issues in front of developers in a timelier manner, closer to when a developer's mind is focused on the code.[4]
- **Expands capabilities provided through continuous integration.** Manual app UI testing, by its very nature, cannot be part of the CI processes; instead it runs in parallel and not as frequently. Automating app UI testing adds new capabilities to the CI process and helps AD&D pros deliver better apps; unfortunately, just 25% of mobile developers tell us they currently automate testing processes as part of CI.[5]

## Mobile UI Test Automation Has Arrived But Still Has A Ways To Go

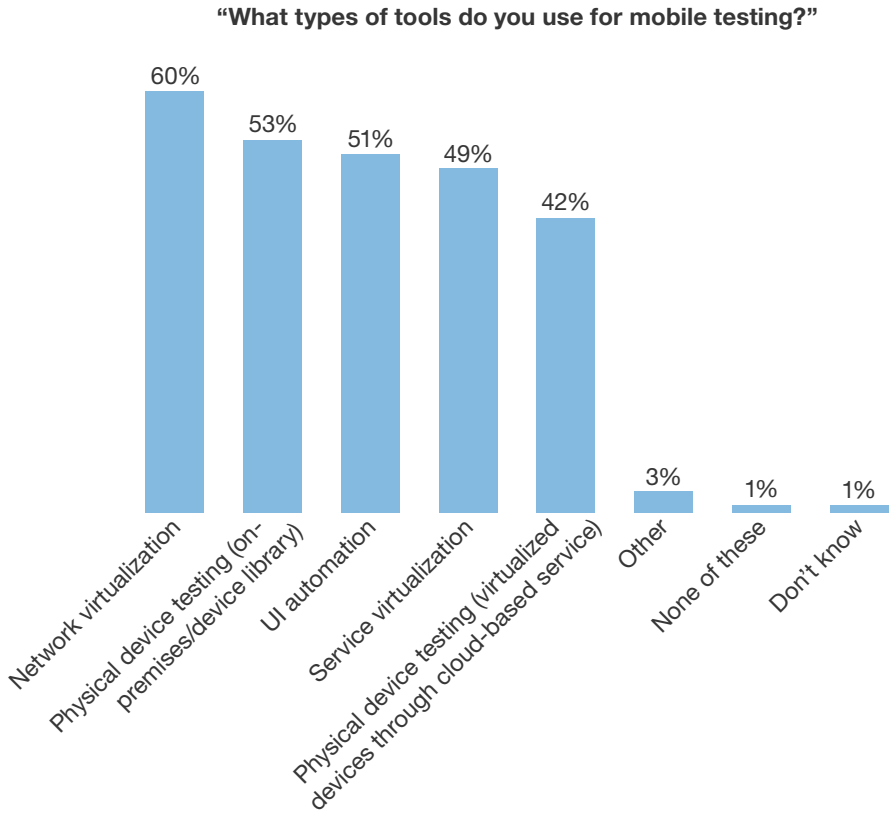
Many mobile testing solutions merely extend existing testing tools into the mobile space; this reflects that many vendors in the space started with enterprise app testing and only recently branched into mobile app testing. UI testing capabilities have been available for mobile web apps for some time, but tools for automated testing of native app UI are more recent arrivals. Forrester's Business Technographics® Global Developer Survey, 2015 tells us that:

- **Half of development organizations have adopted automated UI testing.** Only 51% of developers are currently using automated UI testing (see Figure 3). Automation has made it into many developer tool chests, but there are still a lot of QA engineers poking at mobile devices with their fingers.

› **Devs perceive UI testing as too hard or too time-consuming.** Just 37% of developers use automated UI testing for all mobile apps (see Figure 4). For many dev organizations, the tools are in place, but the cost of maintaining test scripts is forcing them to reserve test automation for more complex, or longer-lived, apps.

› **Tools need to mature for future requirements.** Almost half, 46%, of developers feel mobile testing tools are sufficient for their current needs, but not for the future, while just 31% feel the tools are sufficient for their current *and future* needs (see Figure 5).

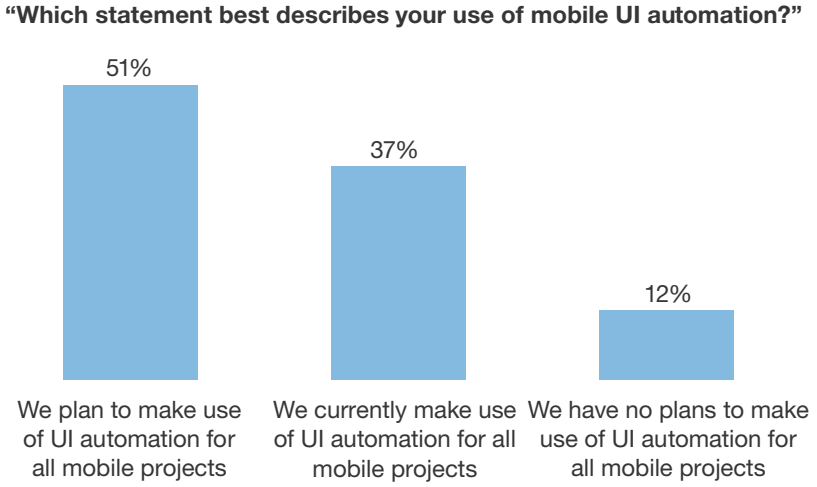
**FIGURE 3** Common Tools Used By Developers




Base: 276 global developers who have worked with both mobile or mobile web apps and quality assurance or testing tools within the past 24 months (multiple responses accepted)

Source: Forrester's Business Technographics® Global Developer Survey, 2015

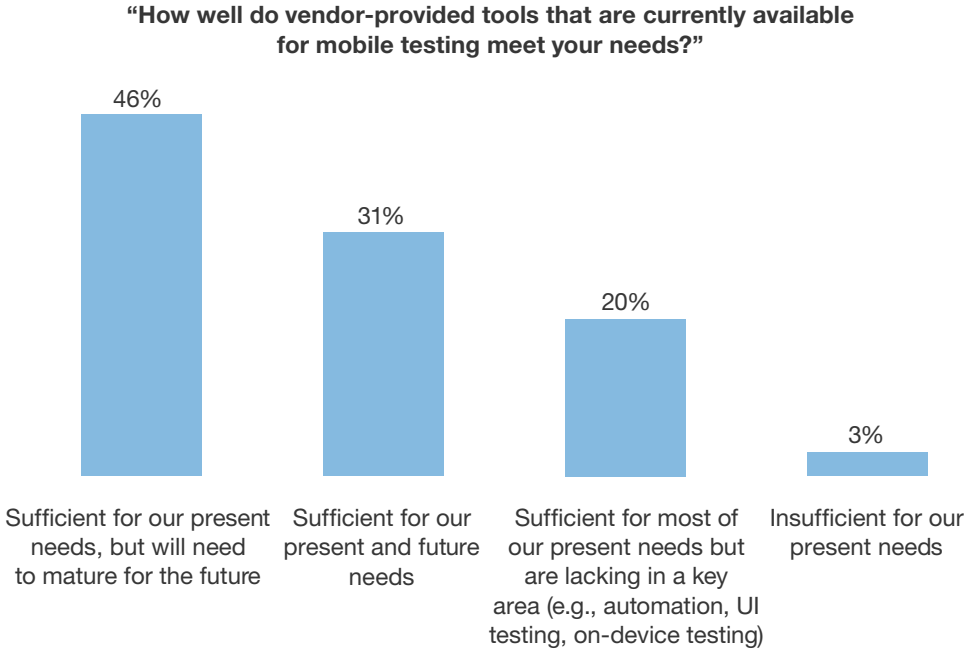
**FIGURE 4** Use Of UI Automation Tools In Development Organizations




Base: 276 global developers who have worked with both mobile or mobile web apps and quality assurance or testing tools within the past 24 months

Source: Forrester's Business Technographics® Global Developer Survey, 2015

**FIGURE 5** Maturity Of Mobile Testing Automation Tools




Base: 276 global developers who have worked with both mobile or mobile web apps and quality assurance or testing tools within the past 24 months

Source: Forrester's Business Technographics® Global Developer Survey, 2015

## Seeking Automation? Survey The Mobile Testing Tools Landscape

A fast-evolving landscape of vendors and products enable development organizations to perform functional tests of a mobile application's user interface. AD&D pros see a widening array of options beyond just testing apps in an internal lab; cloud options are expanding and multiple solutions exist to enable crowdsourced testing as well. In some cases, general-purpose testing tools are expanding to provide mobile capabilities, while other vendors focus solely on mobile. We see the following trends:

› **Traditional enterprise test suite vendors adding mobile-testing capabilities.** Mobile app testing has unique requirements, and mobile-first solutions exist. Existing app testing vendors like Borland Software, CA Technologies, HP Enterprise, and IBM have had to add mobile capabilities in order to compete. In some cases, these mobile capabilities are just a new target for existing test automation tools, while in others the vendors have implemented mobile-specific capabilities that interact seamlessly with existing automation frameworks.[6]

- **Testing vendors standardizing on open source frameworks.** Most web app testing vendors have integrated Selenium to automate functional and cross-browser testing. Mobile app testing tools are now standardizing on Appium for mobile (native, hybrid, and Web) app testing. Supporting a common framework across multiple vendors' tools allows customers to find relevant skills and more easily switch providers while keeping their existing test scripts.
- **Device testing clouds are nascent.** Companies like Keynote (owned by Dynatrace) and Perfecto Mobile have offered cloud-based device testing labs for some time; now other vendors such as Sauce Labs and Xamarin have entered the fray. Before long, many vendors will offer this type of solution, while new entries by Amazon and Google will drive pricing down.
- **Integration of network virtualization tools enhances performance testing.** Mobile app UI testing tools validate that an app is operational; only when coupled with network virtualization tools can developers gather true performance metrics. In this scenario, full testing suites have an edge over bolting multiple products together, which is never optimal.
- **Test vendors embracing test chaos for better results.** True app performance is measured in realistic test scenarios. Testing an app UI when it is the only app running on the device won't find all the problems. Products like InfoStretch Qmetry Mobile and Perfecto Mobile's Continuous Quality Lab allow tests to execute under varying memory and processor load on the device in order to deliver more realistic test results.

## Categorizing Mobile UI Testing Products

Along with the extension of existing app testing tools into mobile, additional categories of UI testing tools offer new testing environments and approaches, each with different value to offer (see Figure 6):

- **Open source UI automation testing frameworks.** Valid open source frameworks like Appium are playing the role Selenium plays in enterprise and web testing space, offering cross-device mobile testing advantages.
- **Commercial UI automation testing tools.** Improving Mobile App Quality Testing Commercial solutions crowd the mobile test automation space, ranging from traditional testing vendors like IBM, HP Enterprise, CA, Tricentis, and Parasoft to native mobile testing vendors like Experitest, Perfecto Mobile, and Xamarin.
- **Mobile device lab testing.** Vendors offer mobile device lab testing environments to deal with the multitude of physical devices.
- **Crowdsourced testing tools.** Crowdsourcing platforms are a new approach aimed at giving development organizations an external perspective on their app.
- **Mobile UI beta-testing.** Beta testing platforms gather in-production user feedback.

**FIGURE 6** The UI Mobile Testing Landscape

| Category | Description/vendors |
|---|---|
| Open source UI automation testing frameworks | Open source community-driven test automation frameworks for web and native mobile applications. These frameworks enable test automation plus portability of tests across automation products.<br><br>Available frameworks include: Appium, Calabash, Robotium, and Selenium. |
| Commercial UI automation testing tools | On-premises or cloud-based environment that manages test execution, results tracking, and reporting. Tests can be executed in an ad hoc manner or driven by CI processes. May offer test recording and editing capabilities, support for manual testing, and integration with device labs.<br><br>Product vendors include: CA, Dynatrace (Keynote), Experitest, HP, IBM, Infostretch, Jamo Solutions, Maveryx, Micro Focus (Borland), Oracle, Parasoft, Perfecto Mobile, Progress Software (Telerik), Ranorex, Sauce Labs, SmartBear, SOASTA, Testmunk, TestObject, Testplant, Tricentis, Xamarin, and Zepyhr. |
| Mobile device lab testing | On-premises or cloud-based systems that enable automated or manual tests to be executed on physical devices. May also offer support for live debugging on devices, data collection (performance and other metrics), and performance adjustment during tests.<br><br>Product vendors include: Amazon, Dynatrace (Keynote), Experitest, Google, Mobile Labs, Perfecto Mobile, Remote Testkit, Sauce Labs, Testdroid, Testin, Testmunk, TestObject, TestPlant, and Xamarin. |
| Crowdsourced UI testing tools | Manage the execution of pre-defined usability tests via crowdsourcing on physical devices. Enable organizations to quickly gain an outsider's perspective on app UX and performance. Use to validate different design approaches or app flow.<br><br>Product vendors include: Applause, Testin, Ubertesters, and UserTesting. |
| Mobile UI beta testing | Manage the delivery of pre-release versions of mobile apps to beta testers. May include collection of crash data and user feedback.<br><br>Product vendors include: Applause, Apple, Microsoft, Test Fairy, and Ubertesters. |

## Open Source Frameworks Enable UI Automation Portability

Automated tests need a mechanism that enables an external process to interact with app running on a mobile device. Automation vendors have built proprietary solutions for this; the open source community has responded with solutions like Appium, Calabash, Robotium, and Selenium. Proprietary solutions only work within a specific vendor's product; adoption of these open source frameworks by many test automation vendors enhances the portability of tests across vendor products. With these frameworks, tests are:

- **Executed manually or driven directly from CI or other automation tools.** It's a simple process to enable these tests through any process. Pass/fail results are returned to the calling process which decides what to do next. Developers can mix tests built with these frameworks directly into their locally executed test scenarios.
- **Scripted using common or proprietary languages.** Scripts for frameworks like Appium and Selenium can be created in a wide range of languages for maximum flexibility such as C#, Java, JavaScript, Objective C, PHP, Python, Ruby. More limited language options are available when using Calabash (Cucumber/Gherkin) and Robotium (Java).
- **Supported by automation vendors.** Many of these frameworks are sponsored by tool vendors. Direct vendor product support means scripts integrate seamlessly with vendors' automation suites. When multiple vendors support a framework, which is the case with Appium and Selenium, existing scripts can be ported between vendor products.

### Commercial UI Testing Vendors Balance Broad Capabilities With Mobile-Specific Features

Most of the enterprise testing vendors in this category support desktop, web, and mobile apps as well as end-to-end tests that include performance testing, functional testing, service virtualization, test data management and more.[7] Dev shops with a "mobile-first" strategy should look at vendors that offer a variety of common features such as management of automated and manual tests, creation, editing, and organization of test scripts, developer dashboard for displaying test results, along with reports and analytics (see Figure 7). Fundamentally, the commercial vendors come from two different backgrounds, with different appeals to developers:

- **Enterprise app testing vendors emphasize consistency.** Testers and developers hate changing their working environments. When testers and developers have existing app-testing tools they're comfortable with, the option of going with what they know will give a jump-start in efficiency that might counterbalance advantages of mobile-specific solutions. Companies like CA, IBM, HP Enterprise, Micro Focus (Borland), Parasoft, Tricentis, and SmartBear are long-term players in this space and include a wide variety of end-to-end testing capabilities in their products.
- **Mobile-focused vendors offer focus and efficiency.** Several vendors started with mobile and are sticking to their roots. Their focus on mobile app testing gives these products efficiencies that can't be found when mobile testing is tacked onto desktop app testing products. Keynote and Perfecto Mobile's Continuous Quality Lab started by delivering cloud-based options for testing mobile apps on physical devices, adding more automation capabilities and additional features over time. Xamarin's focus was enabling cross-platform mobile development before extending into app testing and device cloud capabilities.

Among the mobile-specific testing products, vendors differentiate with capabilities like:

- **Validating test case coverage.** Several tools help ensure test quality by actively calculating the quality of test coverage for mobile apps. This gives teams an indication of how well the app is being tested and which areas are light. The Tricentis Tosca Testsuite offers automated tools for determining test case coverage, and manages the recording, editing, and execution of tests. CA's Test Case Optimizer generates test cases from requirements documentation and automates recording of test scripts, but relies upon third-party products for test automation.
- **Recording and editing test scripts.** Most solutions provide a UI test recorder that developers and testers use to create test scripts; the difference lies in which options are available for editing tests later. Recording makes tests easier to craft, although enhancing existing scripts is a cumbersome point-and-click affair; some shops will opt instead for rerecording. Solutions like CA's Application Test for Mobile, Experitest, Perfecto Mobile, Sauce Labs, and Telerik Test Studio record scripts to code which can be later edited manually. Others such as HP Enterprise UFT, Tricentis Tosca Testsuite, or SOASTA TouchTest offer test recording capabilities with a proprietary editor.
- **Validating an app's UI.** When UI validation is being performed, validation tools must be able to identify when UI elements have been rendered correctly within the app. Testplant's eggPlant only offers image-based verification, but its robust optical character recognition (OCR) capabilities and its ability to group representative images makes up for any limitations of this approach. Object-based recognition looks for specific UI elements and is more accurate and resilient for mobile app testing; this is the only option available from Experitest, Sauce Labs, and Tricentis. Solutions such as those from HP Enterprise and Keynote offer both options.
- **Remote debugging.** When troubleshooting tricky app issues, especially those that manifest themselves only on specific devices, developers often need to resort to debugging the app directly from their integrated development environment (IDE). This usually means that the device is physically connected to the development system. Being able to debug against any available device, no matter the location, dramatically simplifies the process of stamping out bugs. Of the mobile testing solutions that offer access to mobile device labs, Experitest SeeTest Manual and the Mobile Labs deviceBridge enable live debugging with real devices, no matter where the device is located. While not a formal feature of Testplant's eggCloud, it supports live debugging as well.
- **Varying device load during tests.** Most testing of mobile apps occur when the device, emulator or simulator is running only the app being tested. Other things that are happening on the device can affect performance or, in some cases, cause crashes. Solutions like InfoStretch QMetry Mobile Testing and Perfecto Mobile Continuous Quality Lab enable testers to vary the load on devices during tests, draining memory or processor cycles in order to help expose performance issues. This provides a more accurate measure of app performance and identifying issues caused by device issues.
- **Range of device support.** Most mobile testing solutions support Android and iOS devices. For dev teams looking to support additional mobile device platforms, Dynatrace Keynote, Experitest, Jamo Solutions, Micro Focus, Perfecto Mobile and Testplant eggPlant support BlackBerry and Windows Phone.

**FIGURE 7** Overview Of Sample Test Automation Tools

| Company | Product(s) | Deployment | Automation | | |
|---|---|---|---|---|---|
| | | | Test execution | GUI recording | Verification approach |
| CA Technologies | CA Application Test for Mobile, CA Test Case Optimizer | On-premises | | X | Object |
| Dynatrace | Keynote | Cloud | X | X | Image, Object |
| Experitest | SeeTest Automation, SeeTest Manual, SeeTest Manager, SeeTest Cloud | On-premises | X | X | Image, Object |
| Hewlett-Packard Enterprise | Mobile Center, Sprinter, UFT | On-premises | X | X | Image, Object |
| InfoStretch | Qmetry Test Management, Qmetry Mobile | Cloud, on-premises | X | X | Object |
| Micro Focus (Borland) | Silk Test, Silk Mobile, Silk Central | On-premises | X | X | Object |
| Mobile Labs | deviceBridge, deviceConnect | Cloud, on-premises | | | Object |
| Oracle | MonkeyTalk App Testing Automation Tool, LabManager | On-premises | X | X | Image, Object |
| Perfecto Mobile | Continuous Quality Lab, LocalLink | Cloud, On-premises | X | X | Image, Object |
| Sauce Labs | Sauce Labs | Cloud | X | | Object |
| SmartBear | TestComplete Mobile | On-premises | X | X | Image, Object |
| SOASTA | TouchTest | Cloud, On-premises | X | X | Image, Object |
| Telerik | Test Studio | On-premises | X | X | Object |
| Testplant | eggPlant | On-premises | X | X | Image |
| Tricentis | Tosca Testsuite | On-premises | X | | Object |
| Xamarin | Xamarin Test Cloud | Cloud | X | X | Object |

**FIGURE 7** Overview Of Sample Test Automation Tools (Cont.)

| Company | Manual testing | Functional testing | Device load testing | Devices on cloud/lab | Supported frameworks | OS platforms |
|---|---|---|---|---|---|---|
| CA Technologies | | | | | Appium, Selenium | Android, iOS |
| Dynatrace | X | X | | X | Appium, Selenium | Android, BlackBerry, iOS, Windows |
| Experitest | X | X | | X | Proprietary | Android, BlackBerry, iOS, Windows |
| Hewlett Packard Enterprise | X | X | | X | Proprietary | Android, iOS |
| InfoStretch | X | X | X | | Appium, Selenium | Android, iOS |
| Micro Focus (Borland) | X | X | | | Proprietary | Android, BlackBerry, iOS, Windows |
| Mobile Labs | X | X | | X | Appium, Calabash | Android, iOS |
| Oracle | X | X | | X | Proprietary | Android, iOS |
| Perfecto Mobile | X | X | X | X | Appium, Selenium | Android, BlackBerry, iOS, Windows |
| Sauce Labs | X | X | | X | Appium, Selenium | Android, iOS |
| SmartBear | X | X | | | Proprietary, Selenium | Android, iOS |
| SOASTA | | X | | | Proprietary | Android, iOS |
| Telerik | | X | | | Proprietary | Android, iOS |
| Testplant | X | X | | X | Proprietary | Android, BlackBerry, iOS, Windows |
| Tricentis | X | X | | | Proprietary | Android, iOS |
| Xamarin | X | X | | X | Appium, Selenium, Calabash, Xamarin.UITest | Android, iOS |

## Mobile Device Lab Testing Addresses The Diversity Of Physical Devices

Delivering quality apps is directly affected by the number of devices on which the app is tested. Ad hoc testing against whatever devices are at hand isn't nearly good enough. Without a wide range of target devices to test on, there's no way to know where the app fails in performance, UI quality, and more.

Test automation is part of the answer; it's also important to collect as many types of physical devices as possible and make them available to developers and testers for manual and automated tests. Options available to AD&D pros include:

- **Build-your-own device lab.** While tightly restricted by budget, many organizations acquire devices and place them in a common area where developers and testers can check them out and use them for their testing. Often this is implemented through creation of a device wall, an efficient way to organize devices and keep them charged.

  Budget is the biggest issue with this approach since devices are expensive, new models are regularly released and, if true network performance is required for testing, data plans need to be acquired. It's basically a full-time job for someone to keep the lab up to date. Additionally, only testers physically local to the lab can make use of the devices.

- **On-premises test labs.** For product teams that can't operate their tests in the cloud (for security or regulatory reasons) or want more control over what devices and capabilities are available, on-premises solutions can be deployed. These solutions include hardware and/or software that make physical devices available to others across the network. In some cases, allowing developers to do live debugging on remote devices as well.

  Several products offer on-premises device lab solutions, for example Experitest SeeTest Cloud, HP Enterprise Mobile Center, Mobile Labs deviceConnect, and Test Plant eggCLoud. These solutions still require dev shops to purchase their own devices, but extend the availability of these devices to testers or test automation systems throughout the organization. Various testing service providers also offer test labs as a service.

- **Cloud-based test labs.** Recognizing the cost and complexity of maintaining the right catalog of devices in multiple languages and regions, vendors have introduced cloud-based solutions. These labs are typically distributed around the world and provisioned with a large catalog of devices. This is the most convenient approach, as it offloads device and OS maintenance to the service provider while providing easy access to devices for all sorts of testing.

  Keynote DeviceAnywhere and the Perfecto Mobile Continuous Quality Lab were pioneers in this space. Sauce Labs and Xamarin Test Cloud now have solutions as well. The recent entries of Amazon with its AWS Device Farm and Google's Cloud Test Lab are going to commoditize these offerings and drive down prices.

## New Tools Enable Crowdsourcing Of App Feedback

As hard as designers, developers, and testers work to implement and test attractive and useful app UIs, it's not until real users put an app through its paces that they truly understand how well it works. App ratings suffer when app owners test UI decisions with real customers, so external, professional

testers are needed instead. For this type of test, crowdsourcing app testing through companies like Applause, Testin, Ubertesters, or UserTesting is an efficient and cost-effective way to expose a mobile app to professional testers. With crowdsourced mobile app testing:

- **Apps are tested on a wide variety of physical devices and network conditions.** Tests are performed on whatever devices the testers use day to day and on whatever network conditions happen to be available. Many testers will have up-to-date hardware, but the app will be exposed to outliers as well.
- **Testers are distributed around the globe for diversity.** Crowdsourced efforts attract participants from all around the world. This is especially helpful when validating internationalization (commonly known as "I18N") efforts or validation of location-based capabilities within an app.
- **Analytics and direct feedback come right to developers.** Applause and Ubertesters provide a software development kit (SDK) that can be baked into an app that allows it to collect details about users' interaction with the app, app crashes, and user feedback. UserTesting records the tester's narrative and a video of app screens as testers perform pre-defined or ad hoc tests. Customers can engage UserTesting analysts to create a detailed summary of test results that highlight specific issues that users faced and issues they perceived with the app's UX.
- **Some testing service providers offer a "protected" crowdsource experience.** Testing service providers like Cognizant, Dell Services, and Tech Mahindra have built service offerings based on crowdsourced testing platforms. Their value proposition is to offer clients a global but closed and controlled crowdsourced testing environment made up of their own employees.

### Beta-Testing Tools Allow Tests While In Production

Another way to get real user feedback is through beta testing. This enables testing an app by less knowledgeable users in unexpected ways, on unexpected devices, and in unexpected conditions. Beta testing is one way to identify the effect third-party apps will have on your app's performance. This approach is more risky from a relationship standpoint if the app is buggy or the UX is off-target, but beta testers should understand this.

Popular solutions for managing delivery of pre-release versions are Applause, Apple TestFlight, Microsoft HockeyApp, Test Fairy, and Ubertesters. They have similar feature sets except for which classes of mobile devices they will support (see Figure 8). Apple's TestFlight solution supports only iOS devices and is designed simply to manage the process of distributing beta versions of mobile apps to testers. Apple has the most arduous restrictions on how mobile apps can be distributed, so something like TestFlight is needed just to simplify the process. Most other solutions support Android and iOS apps; Microsoft's HockeyApp, however, supports a wider range of target platforms adding support for Macintosh OS X and Windows as well. The common features of these products include:

- **Managing the process of sending beta invitations to testers.** Once users are registered, either through self-registration or administrator action, the system sends email notifications to users. Users may receive notification when updated versions of the app are available as well.
- **Distribution of app builds to testers.** This enables over the air (OTA) installation of mobile apps; in some cases, this is through a simple web portal, while in others it is through a separate app on the device. Some solutions track installations of the app for analytics purposes.
- **Automatic crash detection and reporting.** Beta apps will crash, and a critical aspect of beta testing is to collect information on those events. Collecting crash details, the state of the device and app when it happened, plus any insights from beta testers, is required for successful analysis and remediation of the issue.
- **Streamlined user feedback collection.** Effective beta testing processes require feedback from testers. Most provide a shake-for-feedback feature; the user shakes the device and a pop-up window appears where she can type comments.
- **Capture and annotate app screen shots.** We all know that a picture is worth a thousand words; enabling beta testers to grab an image of a screen and mark it up to highlight problem areas streamlines the feedback process and compresses QA's and development's time to comprehend problems. This is especially true when dealing with devices configured for different locales or languages.

**FIGURE 8** Sample Of Beta Test Enablement Tools And Features

| Vendor | Product | Beta invites | Distribute builds | User feedback | Crash reports | Capture screens | SDK | Platforms |
|---|---|---|---|---|---|---|---|---|
| Applause | Applause | X | X | X | X | X | X | Android, iOS |
| Apple | TestFLight | X | X | X | | | | iOS |
| Microsoft | HockeyApp | X | X | X | X | X | X | Android, iOS, OS X, Windows |
| Test Fairy | Test Fairy | X | X | X | X | X | X | Android, iOS |
| Ubertesters | Ubertesters | X | X | X | X | X | X | Android, iOS |

**Recommendations**

## Automate Your Way To Higher Quality Apps

Mobile testing teams must break free of the traditional testing mindset. The rapidly evolving landscape of testing vendors and products can help them:

- **Test mobile apps in chaotic environments.** Varying network availability, bandwidth, and the load on test devices will help deliver a higher-quality app. Enable this through an end-to-end solution like one offered by HP Enterprise or through load performance solutions like SOASTA TouchTest or mobile specific products such as Perfecto Mobile Continuous Quality Lab and InfoStretch Qmetry Mobile.
- **Maintain tests in a world of quick release schedules and feature expansion.** Agile teams pushing for constant customer engagement deliver cool new stuff to test every day. It's a completely different world that can only be mastered through test automation. Look at tools that will analyze an app's UI and method signatures and identify gaps in test coverage, like Tricentis Tosca Testsuite, or that will generate test case maps based on process flow diagrams, CA's Test Case Optimizer.
- **Move testing to the left, making dev and testing more collaborative.** Testers shouldn't be writing test plans, they should instead actively engage with development so automated tests are ready to go when code is checked-in.
- **Gain an outside perspective using beta or crowdsourced testing.** Dev and test teams know an app inside and out — and that's not a healthy perspective for testers. You will find unsuspected issues by engaging users who don't know how the app is supposed to work. Crowdsourced testing is an inexpensive way to gather unique perspectives on your app's usability.
- **Use open source options for portability between vendors.** While not quite a standard option today across vendor products, expect more vendors to adopt Appium and Selenium for mobile app and web testing. Being able to reuse tests will make it less expensive to switch test vendors.

## Engage With An Analyst

Gain greater confidence in your decisions by working with Forrester thought leaders to apply our research to your specific business and technology initiatives.

**Analyst Inquiry**

Ask a question related to our research; a Forrester analyst will help you put it into practice and take the next step. Schedule a 30-minute phone session with the analyst or opt for a response via email.

Learn more about inquiry, including tips for getting the most out of your discussion.

**Analyst Advisory**

Put research into practice with in-depth analysis of your specific business and technology challenges. Engagements include custom advisory calls, strategy days, workshops, speeches, and webinars.

Learn about interactive advisory sessions and how we can support your initiatives.

## Supplemental Material

### Survey Methodology

Forrester's Business Technographics Global Developer Survey, 2015 was fielded to 1,943 developers located in Australia, Brazil, Canada, China, France, Germany, India, New Zealand, the UK, and the US.

Forrester's Business Technographics provides demand-side insight into the priorities, investments, and customer journeys of business and technology decision-makers and the workforce across the globe. Forrester collects data insights from qualified respondents in 10 countries spanning the Americas, Europe, and Asia. Business Technographics uses only superior data sources and advanced data-cleaning techniques to ensure the highest data quality.

### For Technographics Clients: How To Get More Technographics Data Insights

Forrester's Business Technographics Global Developer Survey, 2015 of 1,943 developers located in Australia, Brazil, Canada, China, France, Germany, India, New Zealand, the UK, and the US includes many additional questions and parameters by which you can analyze the data contained in this report.

We can provide additional insights about the consumers highlighted in this report:

› Who they are (e.g., demographics, lifestyle, and interests).

- What they do (e.g., digital, mobile, social behaviors).
- Affiliations they have (e.g., brands used, products owned).
- How they feel (e.g., attitudes, interests).

If you wish to subscribe to Forrester's Business Technographics services, please contact your account manager or data@forrester.com. If you are an existing Technographics client, please contact your data advisor at data@forrester.com.

### Companies Interviewed For This Report

CA Technologies
Dynatrace (Keynote)
EPAM Systems
Experitest
HP Enterprise
InfoStretch
Micro Focus International
Mobile Labs
Oracle
Perfecto Mobile
Sauce Labs
SmartBear Software
SOASTA
Telerik
Testplant
Torry Harris Business Solutions
Tricentis
Ubertesters
UserTesting
Xamarin

## Endnotes

[1] See what quality with speed means and how to achieve it in the following report. See the "Five Must-Do's For Testing Quality At Speed" Forrester report.

[2] The manual testing approach is described in the following Forrester report. See the "Improving Mobile App Quality Testing" Forrester report.

[3] For more information on UX testing and validation, check the following report. See the "How To Modernize User Experience" Forrester report.

[4] For more information on developer context-switching, check the following Forrester report. See the "Brief: Fast Feedback Accelerates Software Time-To-Market" Forrester report.

[5] Source: Forrester's Business Technographics Global Developer Survey, 2015.

[6] For more information on these vendors and their tools, check the following functional test automation Wave. See the "The Forrester Wave™: Modern Application Functional Test Automation Tools, Q2 2015" Forrester report.

[7] For a more complete list of features offered by these enterprise functional test automation tool vendors, check the following report. See the "The Forrester Wave™: Modern Application Functional Test Automation Tools, Q2 2015" Forrester report.

FORRESTER®
CHALLENGE THINKING. LEAD CHANGE.

## We work with business and technology leaders to develop customer-obsessed strategies that drive growth.

PRODUCTS AND SERVICES

- Core research and tools
- Data and analytics
- Peer collaboration
- Analyst engagement
- Consulting
- Events

## Forrester's research and insights are tailored to your role and critical business initiatives.

ROLES WE SERVE

**Marketing & Strategy Professionals**

CMO
B2B Marketing
B2C Marketing
Customer Experience
Customer Insights
eBusiness & Channel Strategy

**Technology Management Professionals**

CIO
- Application Development & Delivery

Enterprise Architecture
Infrastructure & Operations
Security & Risk
Sourcing & Vendor Management

**Technology Industry Professionals**

Analyst Relations

CLIENT SUPPORT

For information on hard-copy or electronic reprints, please contact Client Support at +1 866-367-7378, +1 617-613-5730, or clientsupport@forrester.com. We offer quantity discounts and special pricing for academic and nonprofit institutions.