Three challenges driving the future of functional testing
Executive summary

Teams tasked with verifying the functionality of software applications face a mountain of challenges. While the pace of business requires increasing speed and agility from the testing process, applications are growing more complex and often include components that are in the hands of third parties. The rapid adoption of mobile technologies adds yet another layer of complexity, including the challenges brought by diverse devices, operating systems, and network carriers.

Faced with today’s multilevel challenges, many functional testing teams are reaching the breaking point. Yesterday’s tools and processes can’t overcome the challenges of time to market, the onslaught of mobile technologies, and the complexity of today’s hybrid composite applications.

To respond to these challenges, your organization needs to rethink ingrained practices that are out of step with today’s demands. It is time to transform your functional testing approaches with automated capabilities that accelerate the test process and enable the unified testing of end-to-end application functionality.

For application development, it’s a whole new world

The pace of business requires unprecedented speed and agility.

For your software development and quality assurance (QA) teams focused on enterprise applications, the world is changing at lightning-fast speeds.

Your enterprise can no longer wait for lengthy waterfall development projects that might require a year or two of back-end development and testing before an application goes into production. To stay competitive, you need to bring new applications and new software functionality to market in days or weeks, not months or years.

This reality puts your application delivery teams under pressure to greatly accelerate software development cycles without compromising quality or the end–user experience—missteps that compromise your business. What’s more, to continuously deliver new software functionality that drives competitive differentiation, your application teams need to increase the pace of innovation. Some development teams now release new software functionality as often as every day, depending on the problems they are trying to solve.

These modern realities put a great amount of pressure on your functional testing teams. In a world of Agile software development and iterative releases, functional testing must now take place early and often in the development cycle.

Mobility is bringing a whole new set of challenges.

Today’s application development challenges are only going to grow steeper as mobile conquers the world. Already, industry analysts predict that within the next few years, tablet sales will overtake PC sales¹ and more U.S. Internet users will access the Internet through mobile devices than through PCs or other wireline devices.²

This rapid move by consumers and businesses to mobile applications has huge implications for your functional testing teams. Mobility isn’t one thing; it’s a great many things. Mobile users come at an application from a plethora of devices, operating systems, and network carriers—and the test process must account for all of these variables. Mobile applications have all the testing complications of composite applications, but they don’t simply imitate the desktop environment. They have their own user interface requirements, business process flows, and infrastructure dependencies.

Moreover, the mobility challenges go beyond the world of customers. As mobility becomes even more widespread, enterprise applications will increasingly have mobile extensions that enable employees to work more productively. This “mobilization” of enterprise applications brings yet another set of test challenges. Enterprise applications extending to mobile interfaces must be tested just as thoroughly as their PC counterparts.

---

¹ Mobile
² Modern application
³ Integrations into cloud apps
⁴ Hybrid composite app
⁵ Web interface
⁶ API and services
⁷ Database
⁸ 3rd party integrations
Applications are growing more complex.

Applications no longer sit in silos. Nothing works in isolation—everything is connected. Your e-commerce applications, for example, encompass processes for online shopping, collection of user data, bank card verification, shipping choices, and email confirmations.

As your test teams know all too well, hybrid composite applications bring hybrid testing challenges. How do you test a complex system that is not entirely self-contained? What components of a composite application are not available for testing? How do you validate the integration of a mobile front end with your back-end applications and services?

To verify the functionality of a hybrid application, your software developers and QA teams must test both the parts and the whole:

• They need to test the individual pieces of a composite application and their ability to perform their functions properly.
• They need to test the entire application scenario under hundreds of different variables. Did the correct data come up from the database? Did the application work on certain phones? The questions go on and on.

Given their complexity, today’s hybrid applications are challenging to build at the same quality standards associated with conventional enterprise applications.

Functional testing is at a breaking point.

Faced with today’s complex challenges, many functional testing teams are reaching the breaking point. Yesterday’s tools and processes can’t overcome the challenges of time to market, the onslaught of mobile technologies, and the complexity of today’s hybrid composite applications.

These realities drive the need for transformation of your approaches to functional testing. To enable the rapid delivery of new and enhanced applications without sacrificing quality or innovation, your enterprise needs to implement advanced, automated capabilities that streamline and accelerate the functional testing process.

Here’s what needs to happen

How do you respond to today’s functional testing challenges? Begin with these three steps: Rethink ingrained practices, invest in automation, and unify the functional testing environment.

Rethink ingrained practices.

To meet today’s test challenges, your organization will need to rethink a lot of well-established practices. That’s because many of today’s widely used practices have fallen out of step with the evolving demands of functional testing.

In particular, it’s time to rethink:

• Waterfall lifecycles, in which a product is fully built before any serious testing is done
• High maintenance effort for test assets, such as automation scripts, documentation, and data
• Accepted inefficiencies associated with manual testing
• Duplication of effort across different testing teams (and even within a single team)
• The inability to test complete permutations of GUI system configurations and mobile configurations
• The use of disparate tools across the enterprise—which equates to a lack of shared assets, higher training costs, and no leverage of investments

Your QA teams can’t keep doing what they have always done and expect to meet today’s challenges. The status quo no longer works. It’s time to take a step back and reevaluate processes, tools, and overall approaches. Look for opportunities to cut out repetitive steps, introduce centers of excellence, automate manual test processes, and adopt methodologies that accelerate productivity.

To speed time to market without sacrificing application quality, your functional testing teams will need to have a greater impact in Agile test methodologies to help bring releasable, tested code into production sooner. Getting test results within the iterative sprints—with enough time for developers to make changes—is imperative.

Another bad practice to recognize is “best-effort testing.” In this informal but widely used practice, testers evaluate functionality to the best of their abilities within the time limits they have been given. When the time is up, the application is released, ready or not, and end users do the rest of the testing. Without time to make any improvements to the process, testers are immediately assigned to the next project. Investment in testing resources, tools, and methodologies must be prioritized or the business risks damage to customer satisfaction.

While your development and test teams have to move faster, your enterprise cannot sacrifice application quality for speed. The goal is to accelerate the delivery of high-quality applications. How do you do that? Look to the following steps: test automation and unified functional testing.

Invest in test automation.

To meet your quality and time-to-market goals, you need to invest in automated functional testing methodologies, tools, and training to allow faster test execution and faster responses to the changing needs of the business. Agile testing cannot be done without automation.

With the right automated test tools in place, your developers can receive instant feedback on changes they made to the code. This allows them to fix problems as they appear and to deliver a higher-quality end product in less time.
Automation is required to help ensure proper test coverage of the growing number of environments and technologies that must be taken into consideration in test processes. In addition, automated testing can help you ensure complete coverage of both legacy code and new software functionality, while greatly accelerating the test process.

For example, you can automate painstaking tasks associated with test case creation and manual testing by integrating test authoring features into your processes, as well as scanners that capture spelling errors, localization, and compliance issues. And you can accelerate time-consuming manual test efforts by integrating tools that enable your developers to convert manual test steps to automated scripts that can be reused by everyone on the team.

To take automation a step further, invest in test frameworks that maximize the reuse of assets and speed up test maintenance and authoring. Frameworks allow you to break processes into functional components that can be reused by other tests. A change in the application can then be reflected once in the reusable component, and then automatically applied to all the tests that use the component within the framework.

To accelerate test automation even more, deploy application-specific automated frameworks for your business applications, such as Oracle and SAP. These packaged solutions can greatly accelerate the test process for the targeted applications.

**Unify the functional testing environment.**

Hybrid composite applications require a unified approach to functional testing. A unified functional testing approach enables developers and QA teams to evaluate both the pieces and parts of an application and the end-to-end functionality of the application.

A unified approach to functional testing automates application functional testing earlier in the software development lifecycle to ferret out defects as soon as they are introduced. Functional testing should be integrated directly into the application development phase and the iterative sprints in the Agile development process.

A unified approach to functional testing puts test automation support for both graphical user interfaces (GUIs) and application programming interfaces (APIs) into one integrated interface.

- Testing the GUI layer means traditional functional validation as well as testing across the plethora of mobile interfaces and intelligent social media technologies.
- Testing the API layer means jumping in with automation right from the start to catch bugs as early as possible. This also means more effective Agile/iterative testing.

- Tying GUI and API testing together allows you to create a true end-to-end validation of the user experience. This is a key way to uncover problems that go undetected with GUI or API testing alone.

**It’s time for a testing transformation**

At HP, we’ve thought a lot about today’s application testing challenges. We’ve come to the conclusion that small changes and incremental advances aren’t enough. Enterprises now need to transform their approaches to functional testing.

That’s the idea behind the tools in the new HP Functional Testing software suite. These tools allow your organization to revolutionize your manual testing, automate functional testing before the GUI is ready, simplify testing through an integrated interface, and create end-to-end automated functional tests for even the most complex hybrid composite applications.

The HP portfolio includes new tools that extend automated functional testing capabilities to mobile devices. With these tools, you can now test your mobile applications on real mobile operating systems in public and private cloud testing environments and tie them to your back-end enterprise systems.

We recognize that speed is as important as quality, and have focused the suite to provide the means to reduce test maintenance overhead, speed execution, and make test authoring easier than ever. To help you further transform your functional testing practice, we offer HP test frameworks and automated framework accelerators tailored for popular applications, such as Oracle and SAP.

And to complete your solution, you can count on HP to provide knowledgeable professional services and an array of complementary partner solutions and services—to create a one-stop shop for test automation.

**To learn more**

To learn more about steps your organization can take to modernize your functional testing practices, visit the HP Functional Testing site at [hp.com/go/functionaltesting](http://hp.com/go/functionaltesting).

---

1 “IDC slashes PC sales forecast as tablets take over,” PC Pro, August 24, 2012.

---

**Get connected**

[hp.com/go/getconnected](http://hp.com/go/getconnected)

Get the insider view on tech trends, support alerts, and HP solutions.

© Copyright 2012 Hewlett-Packard Development Company, L.P. The information contained herein is subject to change without notice. The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates.