FOUR TRENDS RESHAPING THE SOFTWARE QUALITY TESTING MARKET

Telerik
TEST STUDIO

www.telerik.com/test-studio www.twitter.com/teleriktesting

Introduction

The effectiveness of your software quality testing program has the potential to make or break your business. Just consider the experience of eCrowds, a new on-demand, online community created in 2008. Developers married CRM data with blogs, wikis, forums and other "idea exchange" functionality to create a product that would solve the challenges presented by disparate data silos, each with its own user authentication and interface. A single sign-on would deliver immediate and ready access to a host of relevant data.

eCrowds launched with great fanfare, but folded only a year later. In a clear-eyed postmortem of what went wrong, founder David Cummings said performance of the product degraded as it grew in complexity, creating an unacceptable user experience.

"We spent hundreds of hours trying to speed [up] the application with little success," Cummings wrote. "This taught me that we needed to have benchmarking tools incorporated into the development cycle from the beginning..."[1]

The eCrowds experience isn't an anomaly. Marketplace feedback can be quick – and brutal. One-star ratings and negative comments in online forums can kill a product before it has time to get traction, and the failure can damage your brand equity for years to come.

> **"performance of the product degraded as it grew in complexity, creating an unacceptable user experience...**

If you want to succeed in today's "no holds barred" marketplace, it is imperative that you test your application thoroughly to determine how it performs in the field and what the user experience will be like. That means adopting the best testing tools so you can ensure your application works as it was intended.

There are several important factors should you keep in mind, though, as you select a testing solution. In a recent Magic Quadrant report on Integrated Software Quality Suites[2], Gartner analysts point to the many trends impacting the automated software testing marketplace: the drive for productivity, rapidly changing technology and constant updates, the emergence of cloud platforms, the upsurge in the mobility market, distributed development projects, the use of agile development techniques, service oriented architectures, open source testing, application lifecycle management and the emergence of analytics-driven testing.

Each of these trends is making an impact on the testing landscape, but there are four in particular that must be carefully considered for their profound impact. They are fundamentally reshaping the automated testing market and should be taken into account as you make your buying decision. You don't want to invest in tools that will hold you back and limit your options as the future unfolds.

TREND 1:

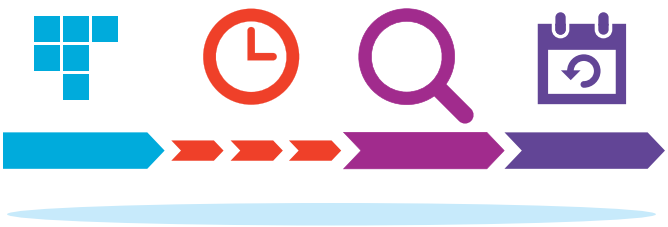
New agile development practices

Industry data shows that agile processes have now overtaken the traditional waterfall approach to development as the methodology of choice.[3] Companies are focusing on rapid, incremental updates to their applications instead of a single, large release that consolidates new features and fixes.

Though agile development is transforming how software is delivered, many testing organizations have yet to catch up to the accelerated cycle times. They are discovering that the manual testing techniques common to waterfall development simply can't be scaled to work well in an agile environment.

Let's say you have 100 tests that are routinely performed before each new software release. Under the previous waterfall approach, your development team would complete the lengthy process of writing new code and hand the product off for testing. Test experts would then write new tests to exercise the new code and conduct a comprehensive regression review to uncover bugs.

With a shortened agile development cycle, though, you don't have time to develop tests for the new feature AND run your



100 baseline tests manually to make certain the changes being made haven't disrupted your existing functionality. If your company is in a two-week sprint before the release of your latest software changes, testers must work in parallel with the development team. As one testing guru said recently, "you can't wait until the entire meal is on the table to determine that the soup needs more salt". Instead, you must find ways to significantly shorten the feedback loop between the testing and development functions.

> **"testers must work in parallel with the development team…**

Some organizations respond to agile time challenges by simply cutting the number of tests performed. Others adopt exploratory testing as a partial, Band-Aid solution and put their software through its paces in the hopes of discovering bugs. Since neither of these approaches is comprehensive, though, you will be taking on new marketplace risk and placing more of the testing burden on end users of your solution.

A suite of automated tests can help you address agile testing challenges and reduce your risks by testing more up front. You will be able to begin regression testing early as developers complete working bits of software so you can ensure the changes made aren't having an unexpected impact on other portions of your application. You can concentrate your manual test efforts on code that isn't yet complete. This combination approach allows you to gather lots of information quickly and to shorten the feedback loop during the agile sprint cycle – providing your organization with better, faster and richer sources of data.

SMART BUYING TIP:

Select an automated testing solution that meets the demands of agile testing.

If your company uses an agile development process – or plans to move to agile in the future – look for a flexible, easy-to-use testing solution that is up to the challenge. The most effective tools allow both developers and test teams to collaborate on a common platform and quickly determine how changes will perform in the field.

Your new testing platform should integrate seamlessly with your company's agile project planning tools as well – automatically executing tests, reporting bugs directly to your planning platform, creating an associated task and tracking progress towards a fix – all without manual intervention. With this single, continuous, automated process, developers can walk into the office each morning and see what issues need their attention and will most impact the user experience.

TREND 2:

Emergence of analytics

To get the maximum return from your testing investment, you need to be able to measure the performance of your application under real-world conditions. That means creating various usage scenarios and measuring how your application responds.

Unfortunately you most likely can't test everything – even with automated test tools. That's especially true if you are working within a time-bound agile development cycle. Your application might include tens of thousands of files and hundreds of megabytes of data, all of which need to fall into place in order for your solution to work as intended. Though you can do your best to test against the most common user paths, there will be times when you won't discover an issue until it surfaces in the field.

> **"Embedded software agents will monitor how your application is used and will generate real-time exception reports when things go wrong…**

Today the first red flag typically comes when a user reports the issue to your tech team or writes about the bug on a public forum. Soon, though, new tools will be available to help you get out ahead of the complaint curve. These new, emerging analytics will allow you to be more proactive in understanding the underlying health of your application by tracking, analyzing and benefiting from actual user experiences after launch.

How will the process work? Embedded software agents will monitor how your application is used and will generate real-time exception reports when things go wrong. Those

exceptions get logged into your project management system so a developer can immediately debug the software and deploy a fix – often before an end user notices and reports the issue. You in effect are extending your agile development cycle into the production arena.

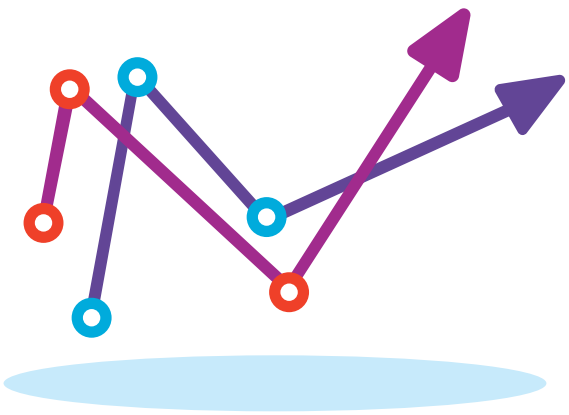
These sorts of analytics become especially important if you have a large, complex application. Think about a typical word processing program, for example. The average user will tap into only a small slice of the software's capabilities – but each individual might use a different slice. It would be virtually impossible to test against every single set of features and the many thousands of options users have for performing tasks in a certain order.

> **"prioritize what tests you need to perform...**

This is a perfect scenario for complementing your traditional automated test program with analytics. By tracking the user experience in real time to see what works well and what crashes, you have the data you need to uncover any bugs you've missed and make quick, continuous improvements.

Facebook and Twitter are examples of companies already using the user experience as an important testing tool. Realistically, neither company can test each new feature against the potential choices made by tens of millions of users. So instead they deploy software builds within a given geographic region, use analytics to collect data and then push the features out more broadly if there are no problems.

Analytics also become incredibly useful when you deploy a feature in two ways to determine which creates the better result. Did color choices, button placement or other differences between the A and B versions impact user behavior? Did one option drive a greater number of purchases, result in more click-throughs or keep individuals on your site longer? By collecting data on actual user behavior, you can see where you are getting the most impact and will be able to make better-informed decisions.



The data uncovered through analytics can even help you prioritize what tests you need to perform. Let's say you work for a global airline and your website offers online ticket booking. To date more than 100,000 travelers have successfully used that feature, and no bugs have been reported. You can rest assured that the function is pretty healthy and can deprioritize it to make room for other tests.

SMART BUYING TIP:

Select an automated testing partner who can help you use the power of analytics to make data-driven decisions.

Though the use of analytics in feedback for testing is still in its infancy, there is no doubt that it holds the potential to revolutionize the testing landscape. Look for an automated testing partner known for a visionary approach and exhibiting a proven track record for staying on top of emerging trends. You want solutions that will evolve over time, help you track and make sense of huge amounts of data, and arm you to make smarter decisions.

TREND 3:

Scalability in the cloud

The rise of cloud computing continues to transform how applications are delivered. But it is also impacting the testing landscape. Companies now can test at a scale they previously couldn't have afforded and can see how large levels of simulated traffic will impact their application's performance.

Before the emergence of the cloud, testing teams faced tough tradeoffs. They either invested heavily in infrastructure to support testing at peak load levels, or they simply used what resources they had available and hoped for the best. Neither is a great option.

Investing in infrastructure can be incredibly costly and requires deep pockets. The experiences of one major North

American insurance company provide a vivid example. The firm's website is critical to its operations – serving both its clients and the tens of thousands of agents who represent the company's products. To ensure the site performs as intended and can handle peak demand generated by advertising and promotional campaigns, testing is a top priority. Before the availability of the cloud, the company built a three-acre test lab equipped with 5,000 servers to conduct peak-load testing. The capital investment, infrastructure and operational costs were enormous – but worth every dollar. The company's very livelihood depended on continuous uptime.

> **"Companies now can test at a scale they previously couldn't have afforded…**

Less well-heeled companies typically limit their testing program to their available infrastructure. But without the capacity to test for true anticipated peak loads, they haven't had a clear view of the challenges they face. This "hoping that all is well" approach is akin to playing Russian roulette with your company's brand reputation. If the objective is to grow your business, you want to make certain you can handle the demand as large volumes of potential customers visit your site.

Fortunately the cloud offers a more effective option and helps companies of all sizes easily leapfrog any costly hardware barriers. Let's say you have a new venture and decide to bet the farm on a $3 million Super Bowl ad. You can easily determine what happens when hundreds of thousands of users hit your website within minutes of your ad airing – without having to raise capital to build your own testing infrastructure. You simply develop the test script and run it in the cloud.

You can rent the server time you need for just pennies per hour – scaling up for large testing scenarios and scaling down once your testing is complete. Compare that to the thousands of dollars involved in purchasing an actual server and you can easily see the advantage. You eliminate both the capital investment and operational costs and can construct more accurate test scenarios without breaking the bank.

SMART BUYING TIP:

Select an automated testing solution that helps you lower your costs by leveraging cloud technology.

Not all testing solution vendors support a cloud-based approach. Nor do they have cloud-based solutions on their roadmap. Look for a vendor who "speaks" cloud and understands your need for affordable options that minimize your hardware investment for test execution. Look for flexible, cloud-based testing features, including virtual test users you can use to exercise your application from a variety of locations.

TREND 4:

Mobility

In his report on the mobility marketplace, Gartner analyst David Willis says a stripped-down, mobile version of your website (or online application) simply isn't enough anymore.[4]

Instead you need to create a cohesive user experience that crosses channels, platforms and sessions.

That cohesive user experience may seem like an insurmountable challenge, though, when you look at what's going on in the marketplace for mobile devices. Developers

who once complained about having to code for four types of browsers are now concerned about how their application renders on hundreds of tablets, smart phones and other mobile devices with an array of operating systems and display sizes. They even have to deal with new form factors – like screens that shift from vertical to horizontal depending on how they are held.

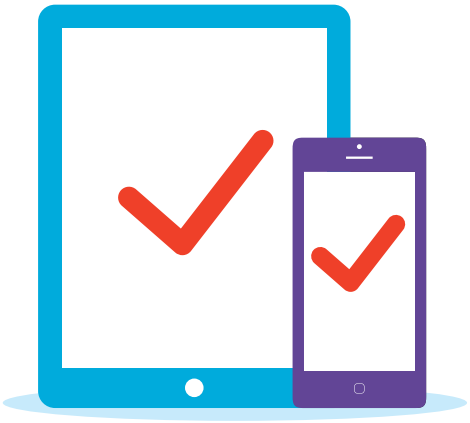
> **“a stripped-down, mobile version of your website (or online application) simply isn’t enough anymore…**

Unfortunately the problem is likely to get worse before it gets better. Rather than a consolidation in the marketplace, we’re seeing an ongoing flurry of new entries – including new device types and new mobile operating systems.

Manufacturers and developers are obviously chasing the burgeoning demand. In the second quarter of 2013, smartphones represented more than half of all mobile phone sales and have now outstripped the sale of traditional mobile feature phones.[5] The Yankee Group predicts that the sale of already popular tablet PCs will increase five-fold between now and 2017, with an average annual compound growth rate of 35 percent.[6]

These mobility market forces are clearly disruptive to the way we build and test software. One example: Mobile users interact with their devices in a more intimate way than they do the typical PC – and in ways that can impact the performance of your application. What happens when a user with broad fingers tries to type on those small buttons you’ve built into your menu? Will double-taps that mimic double mouse-clicks produce the same result as a single tap? Will intense, hard typing give the same or different results?

Device-level testing of website performance becomes a particular challenge. Unlike an application that can be rolled out in stages, you need to know how your site performs on all devices from the day it goes live. If you want to examine the experience of an iOS or Android smartphone user, for example, you need a test that supports the underlying operating system and can collect data on how your site performs in each smartphone environment as you simulate user actions. You need to put the devices through their paces – testing what happens when fingers touch the screen or swipe it in certain ways so you can determine what the response will be.



Mobility has even impacted our very definition of an application. Instead of a one to one mapping of an application across multiple devices, we’re now seeing features decoupled from the underlying operating system. You might need three tests instead of one to determine how the same feature performs across different versions of the same operating system. Test teams need new and innovative tools to help them cope with this new reality.

SMART BUYING TIP:

Select an automated testing solution that helps you deliver the best mobile user experience, regardless of the device.

Flexibility is imperative in this changing marketplace. As you evaluate potential testing partners, look for a provider who is riding the mobility wave and can keep up with new platforms and devices. You should be able to run tests using an actual mobile device instead of an emulator that simulates what your application will look like. Otherwise you won’t really know what will happen when an end user has a phone or tablet device in hand.

Also, make certain you don’t have to “jailbreak,” unlock or modify the mobile device in any way in order to test how your application performs. Instead you should be able to simply run the test application on the device “as is,” right out-of-the-box. You should be able to target both high-level elements and smaller bits, including swipes, zooms, taps and other gestures.

Summing it up

There is no doubt that market forces and new innovations will continue to disrupt the testing landscape. If you decide to delay your selection of an automated testing solution until this "moving train" slows, though, you may be in for a long wait. There simply is no end in sight.

Fortunately there are solutions available today that can meet your current needs AND position you to adapt over time. So ask questions. Evaluate product roadmaps. Notice the language you see on solution websites and in product brochures. Are emerging trends on the vendor's radar and used to actively drive product innovations? If so, you can confidently make a choice that will arm you for whatever the future holds.

ABOUT THE AUTHORS:

Peter Varhol is an Evangelist for Telerik's Test Studio. He's been a software developer and software product manager, technology journalist, and university professor among the many roles in his past, and believes that his best talent is explaining concepts and practices to others. He's on Twitter at @pvarhol.

Steven Vore is an Evangelist for Telerik's Test Studio. He has worked in software support and testing for the better part of two decades, and enjoys exploring ways to make software easier to use. He is a fan of movies and music, and can often be found on Twitter as @StevenJV.

1. Post-Mortem on a Failed Product. http://davidcummings.org/2010/06/04/post-mortem-on-a-failed-product/
2. Gartner Magic Quadrant for Integrated Software Quality Suites. Analysts Thomas E. Murphy and Nathan Wilson. July 11, 2013.
3. Forrester data cited in Agile Adoption Update. Todd Landry. February 18, 2013. http://www.klocwork.com/blog/agile-development/agile-adoption-an-update
4. Mobile and the Nexus of Forces: Creating the New Experience. David A. Willis, Gartner. June 25, 2012.
5. Gartner press release. http://www.gartner.com/newsroom/id/2573415. August 14, 2013.
6. Yankee Group Mobile and Connected Devices Forecast & Monitor. April 2013.