

SHOWCASING THOUGHT LEADERSHIP AND ADVANCES IN SOFTWARE TESTING

LogiGear MAGAZINE

The Rapidly Changing Test Landscape



Deploying the Cloud? Make 2013
the Year to Do It Right
Pete Schmitt

Service Virtualization Rises to
cloud app testing challenge
Michael Vizard

What the Cloud Could
Be Like in 2013?
Mandira Srivastava

The Changing Landscape of Software Testing

By Michael Hackett

Letter from the Editor

Editor in Chief

Michael Hackett

Managing Editor

Brian Letwin

Deputy Editor

Joe Luthy

Worldwide Offices

United States Headquarters

2015 Pioneer Ct., Suite B
San Mateo, CA 94403
Tel +01 650 572 1400
Fax +01 650 572 2822

Viet Nam Headquarters

1A Phan Xich Long, Ward 2
Phu Nhuan District
Ho Chi Minh City
Tel +84 8 3995 4072
Fax +84 8 3995 4076

Viet Nam, Da Nang

7th Floor, Dana Book building
76-78 Bach Dang
Hai Chau District
Tel +84 511 3655 33
Fax +84 511 3655 336

www.logigear.com
www.logigear.vn
www.logigearmagazine.com

Copyright 2013
LogiGear Corporation
All rights reserved.
Reproduction without permission is prohibited.

Submission guidelines are located at
[http://www.logigear.com/magazine/
issue/news/editorial-calendar-and-
submission-guidelines/](http://www.logigear.com/magazine/issue/news/editorial-calendar-and-submission-guidelines/)



Change is constant. What's different today is the rate of change. [Moore's law](#) resulted from the observation that the rate of change in computing power is exponential. The products, services and software landscape appears just as dynamic. At the same time, we pretty much take for granted the ubiquitous presence of software running our lives and the conveniences and ease it brings.

The landscape of product development is constantly shifting—mobility is everything and everything thinks. From medical devices to toasters, from radio becoming Pandora, from cash, checks and credit cards becoming online bill pay becoming [mPay](#)—everything is run by software, and more software with logic capability is being deployed. The technologies landscape too is shifting— the Cloud, SaaS, virtualization, cross-platform/non-platform specific mobile apps to HTML5.

Many commentators on technology today state the PC is dead. How will this impact software development and how can test teams be best prepared for this new landscape? For starters, testing cannot remain the same. Software development today is more agile, less documented, faster and even more distributed—but that is last decade's news. Testing has lagged software development but the future demands the lag must be reduced exponentially. This doesn't mean that as testers we are behind the curve, there are teams already running tens of thousands of automated tests on hundreds of virtual environments against daily builds for global financial security systems. There are many very sophisticated test teams. Yet, even their landscape is changing as products and services change.

This issue of LogiGear Magazine examines the seismic shifts changing how we do product development and specifically how we test software. The landscape has changed. What we test has changed—how do we keep up? As much as we are looking at global trends, some people are already there.

This specific issue also reminds me of the mission of LogiGear Magazine. We want to provide a forum for learning new ideas, methods, practices, state-of-the-practice in software testing and a view into the broader landscape of software testing. I tend to focus on the project ahead of me and sometimes lose sight of the broader horizon. In this issue, Mandira Srivastava tells us why 2013 is the year of the cloud; I will discuss the changing software testing landscape; Pete Schmitt shows us how to leverage the cloud across your organization; Michael Vizard discusses how to get around issues of scale using virtualization; and Adrian Woodhead reviews the book, *How Google Tests Software*.

Since we moved in to a Wordpress platform last year, we have built an archive of our past issues and also cross referenced all our content to be searchable on keywords for research on specific topics.

Also, Happy New Year. Happy 2013 and Year of the Snake in the lunar calendar. The Snake is intuitive, introspective, and refined. He can appear cunning and reticent and works very modestly in the business environment. The Snake will plot and scheme to make certain things turn out exactly as they want them to. Hmm. Sounds like many testers I know! Happy New Year. ■

Michael Hackett
Senior Vice President, LogiGear Corporation
Editor in Chief

In this Issue

4 IN THE NEWS

5 WHAT COULD THE CLOUD BE LIKE IN 2013?

Mandira Srivastava

Cloud computing has been helping business enterprises deliver services faster and cheaper compared to all other existing delivery models. 2013 will be the year when the technology is adopted on a larger scale.

7 THE CHANGING LANDSCAPE OF SOFTWARE TESTING

**Michael Hackett,
LogiGear Corporation**

The landscape of software testing is undergoing a fast and dramatic change, driven by the societal, business and technology changes that have placed software everywhere.

16 DEPLOYING THE CLOUD? MAKE 2013 THE YEAR TO DO IT RIGHT

Pete Schmitt, Customer Storage

By thoughtfully and strategically preparing for future cloud opportunities, your organization can address these concerns and fully leverage the benefits of cloud technology across the enterprise.

18 SERVICE VIRTUALIZATION RISES TO CLOUD APP TESTING CHALLENGE

Michael Vizard, IT Business Edge

Service virtualization provides a way to get around that scale issue that is not only practical, but more importantly should lead to better code being deployed the first time every time.

19 SOFTWARE TESTING LANDSCAPE GLOSSARY

Some of the terms used when discussing the cloud, virtualization and SaaS.

20 BOOK REVIEW

Adrian Woodhead, Last.fm

A review of *How Google Tests Software* by James Whittaker, Jason Arbon and Jeff Carollo.

22 VIETNAM VIEW: WHAT THE PHO? A LOOK AT VIETNAM'S SIGNATURE DISH

Brian Letwin, LogiGear Corporation

While Vietnam is known for its regional culinary specialties, there's one dish that transcends geography and can be eaten at any time of day.

In the News

VISTACON 2013 on The Horizon

VISTACON 2013

VISTACON 2013 will continue to bring top experts and professionals to Vietnam to share the latest and greatest developments in Software Testing. With the success and energy from VISTACON 2011, expectations for the conference have risen. So this year, VISTACON 2013 will be covering the latest trends on software testing, including mobile testing, cloud testing, test automation and testing complex and big systems.

The conference, which will take place on April 25 and 26 at the Majestic Hotel in Ho Chi Minh City, will feature industry thought-leaders such as Julian Harty, BJ Rollison and Hans Buwalda. Visit www.VISTACON.vn for more information and special offers on ticket pricing.

Test Engineer of 2013 Announced



On Tuesday, January 29, Peter Spitzer was awarded Test & Measurement World's Test Engineer of the Year Award for 2013. A member of the TRICENTIS team since 2008, Peter is currently the Team-Lead in TRICENTIS Test.

He is responsible for the quality assurance of the TOSCA Testsuite, including creation/planning and implementation of test concepts and reporting the test results to all stakeholders.

He is also responsible for creating, implementing, and optimizing the company's test portfolio, which is what led him to this award. Peter flew all the way from Austria to join the awards ceremony, and was commended on his achievements and his willingness to travel halfway around the world to celebrate the award!

Read [here](#) some of the questions he answered during an interview with EDN Network.

Banks to Experience More Software Glitches This Year



The increasing complexity of banking software is resulting in a number of serious software faults. According to Lev Lesokhin, strategy chief at New York-based software analysis firm Cast, the main problem is that while developers are good at building new functions, they are bad at ensuring nothing goes wrong when new software is added to the old.

In only the last year, these types of faults have resulted in major hiccups for companies such as Paypal, Knight Capital, Bats Global Markets and Ulster Bank.

"Modern computer systems are so complicated you would need to perform more tests than there are stars in the sky to be 100% sure there were no problems in the system," explains Lev Lesokhin. "This means no single person, or even group of people, can ever fully understand the structure under the key business transactions in an enterprise. Testing alone is no longer a viable option to ensure dependable systems."

Blogger of the Month

What Could the Cloud Be Like in 2013?

2013 will be an exciting year for business enterprises that rely on the cloud.

By Mandira Srivastava

Cloud computing has been the buzzword in the world of Information Technology for quite some time and it is likely to retain that status in the coming years.

Cloud computing has been helping business enterprises deliver services faster and cheaper compared to all other existing delivery models.

Small and medium business enterprises have changed their outlook towards technology substantially thanks to the implementation of cloud computing in their core business models. It has brought down infrastructure costs like no other technology has done before, not to mention the easy (remote) access features that have developed in critical business models.

The future looks even more promising than the current scenario. The world as we know it always demands more and we can be sure that we will be seeing many more implementations of cloud computing in as many discrete operational areas as possible. Let's look at what could be the future of cloud computing in 2013.



Collaboration between the Private and Public clouds could be the biggest trend that we witness in 2013 as enterprises are looking for new ways to help their customers benefit from the best of both worlds. A hybrid cloud strategy that provides them with the functionality and reliability of both is thus a very interesting prospect.

Speed could be a decisive factor for next generation cloud computing technologies. Depending upon network bandwidth and speed, cloud based services could be deployed at rates never before imagined. The benefit will be for end customers who rely on cloud-based bottom line services. They would get speedy service rendering resulting in time and cost savings if we take into account charges for service time.

Collaboration between the Private and Public clouds could be the biggest trend that we witness in 2013 as enterprises are looking for new ways to help their customers experience the best of both worlds.

A hybrid cloud strategy that provides them with the functionality and reliability of both is thus a very interesting prospect.

Integration of mobile and cloud computing could become even stronger with mobile applications calling out to back-end services resting on cloud-based platforms. Seamless service delivery will happen from anywhere on the planet.

It is high time people let go of the myths in security regarding cloud computing. 2013 may very well see the end of such myths as cloud delivery platforms have become more secure than ever before with state of the art security firewalls, physical protection and the security of data centers hosting these cloud platforms.

Nothing makes your business more portable than cloud computing. It would be possible to port your data and applications anywhere in the world and all you need is a computer connected to the internet. Backup and recovery measures might be fully loaded onto cloud based platforms thanks to the convenience offered by it; case in point, [GMO Cloud's add-on services](#) on top of their IaaS offering.

Software development companies will be stressing the importance of the cloud like never before. They can get teams of engineers to work together from every corner of the globe and this kind of collaboration would help develop software components quickly and efficiently.

Bring your own device or BYOD is going to be the talk of the town in 2013. We are seeing a paradigm shift in work culture where the freedom and power of IT are actually given to users via a web based interface. No longer is there demand for complex hardware for working on complex software products.

Open source cloud computing platforms may see a whole new dimension of demands as they have demonstrated their ability to rival proprietary competitor platforms. They've also shown their flexibility in supporting a plethora of services, not to mention the huge support they offer to anyone using it courtesy of its open source tag.

As far as service delivery is concerned, Software-as-a-Service (SaaS) models will continue to improve and provide faster and less expensive user experiences via simpler interfaces. The number of platforms that can be used to access software as a service would increase with smartphones embracing cloud based applications as we mentioned earlier.

All in all, 2013 will be an exciting year for business enterprises that rely on the cloud. ■

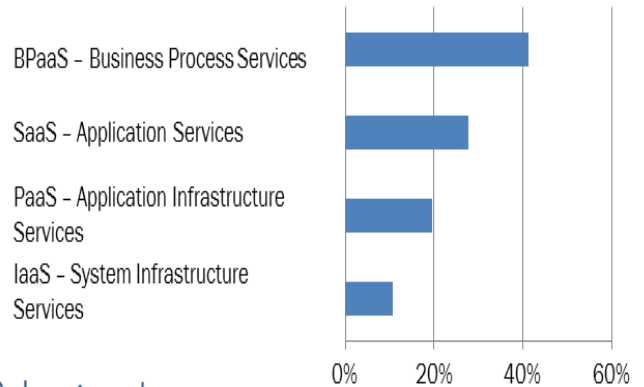
About Mandira



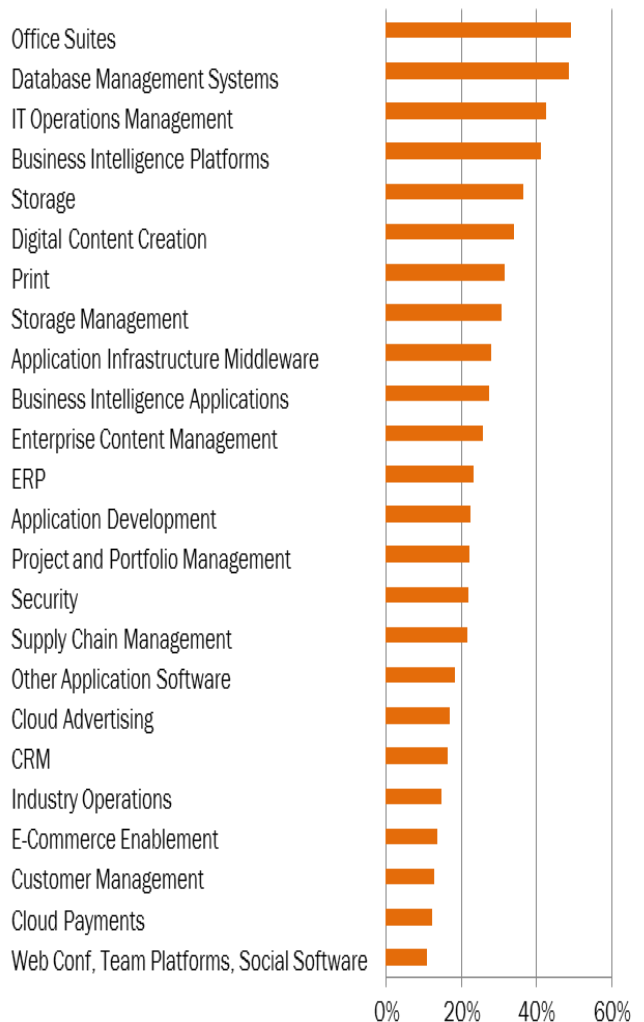
Mandira Srivastava is a fulltime freelance writer who specializes in technology, health and fitness, politics, and financial writing. Equipped with a degree in mass communication, she has worked for both private and corporate clients.

Cloud Computing 5-year Growth Forecast

Segment



Subsegment



Cover Story

The Changing Landscape of Software Testing

*I feel the earth - move - under my feet
I feel the sky tum-b-ling down - tum-b-ling down
- Carol King*

By Michael Hackett, LogiGear Corporation



Introduction

Everything changes. It's the only constant. The landscape of software testing is undergoing a fast and dramatic change driven by societal, business and technology changes that have placed software everywhere.

Computing is ubiquitous. There is hardly anything today that doesn't contain a CPU or information transmission capability, with software to drive it. From smart toasters to passports with microchips that log and share information, and more on the way every day as technologies are crossing and blending. The thinking insulin pump delivers correct doses of life-saving insulin and wirelessly sends reports to you and your doctor's phone and orders a refill when supplies get low. Just a few years ago, who would have thought test strategy for an insulin pump would need to include communication, security and performance testing?

In a relatively short time, financial transactions went from paper checks to online bill pay and now to mPay (mobile payments). The technologies that have enabled such acceleration in product development have changed the testing landscape, dramatically impacting testers and testing.

Software testing skills, training, tools and process have traditionally lagged behind the development of new technologies and products for software programming. Even as everyone seems to have become Agile, there are frameworks for project management (such as Scrum) and many recommendations for development practices like eXtreme Programming (XP), but no common test practice reference to keep up with these more dynamic practices. This article examines some of the larger business trends that impact our products and delivery. We will examine some technology, process and mega-trends to better prepare you for the new normal in software testing. These trends will impact your work. The first step in projecting these impacts is understanding the changes. Let's begin!

Where are we now?

The Business of Software Development

The software development world is driven by business and enabled by technology. Already we take for granted what seemed so new just a few years ago.

From the beginning of software development, testing was thought of as a necessary evil that needed to be *faster*—always—deliver better quality to customers and keep costs down. This idea is not new. Yet the speed and necessity of these are ever-increasing: Faster SDLCs, faster product distribution. Even recently from CD distribution to *downloadable product* to *App download* and SaaS, and even more distributed teams and need for high-volume test automation to meet these demands.

Many people think the move to Agile development practices is a recent shift in the testing landscape. The Agile manifesto, the document that starting it all, was written in 2001—that's 12 years ago. Scrum was presented publicly as a framework for software development in 1995—it is now 17 years old. No longer new. Many companies have moved past Scrum to Kanban—which is a newer SDLC but it is just a digital application of a process introduced in 1953.

While Agile and Scrum are not new, there have been significant changes in product development coupled with Agile. Offshoring and outsourcing Agile development, which we will refer to as distributed Agile, helps maximize efficiency and lead to faster product development. These changing

practices are some of the many pressures on test teams to automate as much testing as possible. What has recently changed the landscape of software testing is the full expectation of teams that test software to be Agile, distribute tasks for best time and cost efficiency and accomplish high-volume test automation simultaneously. Doing this seamlessly and fast is no longer an expectation. It's a given. It's the new normal, and it's not easy. It requires great communication, ALM tools, expert automation frameworks and highly skilled teams. Yet making Agile, distributed testing and test automation effective and efficient while reducing work stress and working at a *sustainable pace* is still out-of-reach for some teams.

Game Changers

There are some big trends in business, technology and products but we need to focus on the disruptive innovations. Not progress or direction adjusters - they are important as well, but wholesale change - the big trends that will impact our work the most and need the most preparation.

These trends are different than adding a new technology; for example, moving from a mobile optimized website to a native app or moving to HTML5. Testers will need new skills and perhaps new tools for testing HTML5. This is progress. There will always be technical progress, new languages, platforms, operating systems, databases and tools. Progress is not a game changer.



Disruptive Innovation

Disruptive innovation is an innovation that creates a new market by applying a different set of values, which ultimately (and unexpectedly) overtakes an existing market. (E.g.: lower priced smart phones). A technology disruptive innovation, or game changer, was moving from procedural to ob-

ject oriented programming. It abruptly and radically restructured software development and it hasn't been the same since.

A disruptive innovation is an innovation that helps create a new market and value network and eventually goes on to disrupt an existing market and value network (over a few years or decades), displacing an earlier technology. The term is used in business and technology literature to describe innovations that improve a product or service in ways that the market does not expect, typically first by designing for a different set of consumers in the new market and later by lowering prices in the existing market.

Just a few years ago, Salesforce.com created disruptive innovation. Salesforce.com's logo showing "No software." tells the story. People initially thought of Salesforce as a software company, but the fact that it's really a service company and produces software as a service is now better understood.

This leads us to the first *mega-trends* or disruptive innovation changing the landscape of software testing:

1. The Cloud and SaaS

Much has been written about SaaS and how the Cloud is taking over. We will focus on how both of these technologies are changing the landscape of software testing. NIST defines cloud computing as a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources—for example, networks, servers, storage, applications and services—that can be rapidly provisioned and released with minimal management effort or service provider interaction.

Often, Cloud Computing and Software as a Service (SaaS) are used interchangeably, but they are distinct. Cloud computing refers to the bigger picture. It is the broad concept of using the internet to allow people to access enabled services—Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). Cloud computing means that infrastructure, applications and business processes can be delivered to you as a service, via the Internet or your own network.

Software as a Service (SaaS) is a service within Cloud Computing. In the SaaS layer, the service provider hosts the software so you don't need to install, manage, or buy hardware for it. All you have to do is connect and use it. SaaS examples include customer relationship management as a service (www.thebulldogcompanies.com/what-is-cloud-computing).

2. Virtualization

Virtualization (or virtualisation) is the simulation of the software and/or hardware upon which other software runs. This simulated environment is called a virtual machine (VM). There are many forms of virtualization, distinguished primarily by the computing architecture layer. Virtualized components may include hardware platforms, operating systems (OS), storage devices, network devices or other resources.

Virtualization can be viewed as part of an overall trend in enterprise IT that includes autonomic computing, a scenario in which the IT environment will be able to manage itself based on perceived activity and utility computing, wherein computer processing power is seen as a utility that clients can pay for on an as-needed basis. A primary goal of virtualization is to centralize administrative tasks while improving scalability and overall hardware-resource utilization. With virtualization, several operating systems can be run in parallel on a single central processing unit (CPU) to reduce overhead costs—not to be confused with multitasking where several programs run on the same OS.

Using virtualization, an enterprise can better manage updates and rapid changes to the operating system and applications without disrupting the user. Ultimately, virtualization dramatically improves the efficiency and availability of resources and applications in an organization. Instead of relying on the old model of “one server, one application” that leads to underutilized resources, virtual resources are dynamically applied to meet business needs without any excess fat. (ConsonusTech).

Simulating software and hardware environments has changed test execution and the entire landscape of software testing.

3. Ubiquitous Computing

I prefer the term coined at MIT—*Things That Think*.

Ubiquitous computing (ubiqomp) is a post-desktop model of human-computer interaction in which information processing has been thoroughly integrated into everyday objects and activities. In the course of ordinary activities, someone “using” ubiquitous computing engages many computational devices and systems simultaneously, and may not necessarily even be aware that they are doing so. This model is considered an advancement from the older desktop paradigm. The more formal definition of [ubiquitous computing](#) is - “machines that fit the human environment instead of forcing humans to enter theirs.”

There is an explosion of *things that think*, powered by complex algorithms that can process huge amounts of data in near real-time and provide instantaneous results. Smart devices with embedded software systems range from traffic lights and watering systems to medical devices, cars and dishwashers, equipped with scanners and sensors, vision, communication and object-based media (self-aware content meeting context-aware consumer electronics). Increasing sensor density is enabling massive data gathering from everyday objects. At the consumer electronics show, CES, it was projected *350 million IP devices will ship in 2013*. Computing is ubiquitous, everything not only thinks, but collects data, communicates and makes decisions for you. And all of these thinking devices are ready to be tested!

4. Big Data

By now we have all heard the phrase [big data](#). What does it mean? And, more important to us – how will it affect my work?



Big data is a collection of data sets so large and complex that they become difficult to process using on-hand database management tools or traditional data processing applications. The challenges include capture, curation, storage, search, sharing, analysis and visualization. The trend to larger data sets is due to the additional information derivable from analysis of a large, single set of related data, as compared to separate smaller sets with the same total amount of data, allowing correlations to be found to “spot business trends, determine quality of research, prevent diseases, link legal citations, combat crime and determine real-time roadway traffic conditions.”

Some of the often mentioned applications of big data are meteorology, genomics, connectomics, complex physics simulations, biological and environmental research, internet search, finance and business informatics.

[Examples from the private sector:](#)

- Wal-Mart handles more than 1 million customer transactions every hour, which is imported into databases estimated to contain more than 2.5 petabytes of data—the equivalent of 167 times the information contained in all the books in the US Library of Congress.
- Facebook handles 50 billion photos from its user base.
- FICO Falcon Credit Card Fraud Detection System protects 2.1 billion active accounts world-wide.

Imagine if you could overlay your blood pressure history with your calendar and discover your most stressful meetings.

In the future, there will be more pressure sensors, infrared, NFC (Near Field Communication, mobile payments or [mPay](#)). Other sensors will be added, creating new services - automakers adding new features taking advantage of sensor data to help you park, for instance. Progressive insur-

ance has Pay As You Drive feature, lower rates if you share GPS data.

5. Mobility

I know. *Everything is mobile.* You may think it last year's news, but it is generally agreed we are just at the beginning of the mobility tidal wave.

Apple launched the iPhone in 2007 and now 65% of mobile phone use time is in non-communications activities as the eco-system is becoming less focused on voice communication. People are using mobile phones more and more as core pieces of hardware. Shawn G. DuBravac, chief economist and research director for the Consumer Electronics Association states, "The smartphone has become the viewfinder of your digital life." The unexpected consequences are that devices and tablets are becoming hub devices: TV remotes, power notebook computers, blood pressure monitors, etc. *Smartphones are now in 52% of U.S. households; tablets in 40% of U.S. households.* The traditional PC is dead. Long live the tablet. We are testing in the post-PC era.

If we pause here and look at testing implications, ABI Research indicates revenues from mobile application testing tools exceeded \$200 million in 2012. About three-fourths of the revenue was from sales of tools that enhance manual testing, but in the coming years, the market will be driven predominantly by test automation. Growth in the automation segment will push revenues close to \$800 million by the end of 2017.

(\$200 Million Mobile Application Testing Market Boosted by Growing Demand for Automation London, United Kingdom - 22 Oct 2012).

6. Changing Flow of Input

Dozens of companies are working on gesture and voice based controls for devices – gesture, not just touch. *Haptic computing* with touch sensitivity has already been developed for applications from robotics to virtual reality.

Language understanding, speech recognition and computer vision will have application in an ever-increasing number of products. We already have voice activated home health care and security systems with remote healthcare monitoring.

Most of us think voice integration for cars is just for calls. Google's driverless car logged 483,000km/300,000miles last year without an accident, and it's legal in a few US states. It will be important for virtual chauffeurs to reliably understand verbal directions—in many languages and dialects. Would you feel safe being driven in a car 100% operated by software that was tested by someone like you, with your skill set?

The tools and software to test these types of applications will be very different than what we know today. They are likely to be graphical applications much like today's video games to be able to adequately test the input devices.

Testing in the New Landscape

Big changes are happening in the software and hardware product development world. To examine how these things are changing testers, testing and test strategy, we'll use the SP3 framework. This is a convenient framework to examine who tests, the test process and what methods and tools are used.



Components of a Test Strategy – SP3

A test strategy has three components that need to work together to produce an effective test effort. We developed a model called SP3, based on a framework developed by Mitchell Levy of the Value Framework Institute. The strategy (S) consists of:

- People (P1) – everyone on your team.
- Process (P2) – the software development and test process.
- Practice (P3) – the methods and tools your team employs to accomplish the testing task.

Note: We need to make some generalizations here about the state of software testing. While some organizations are now automating 100% of their 10,000 or 100,000 test case regression, some organizations have 100% manual regression of pick and choose coverage. For more automation trends, check out question "Q1" in our 2010-2011 [survey results](#) on automation testing.

There are teams already testing Google cars. Some teams do all the performance and security testing they need. There are teams already on the cutting edge, though many are not. The following descriptions may already be true for you, I am writing in global terms.

People - Skill, distributed/offshore tasks and training

Skills

The shift that is happening with people and skills today is a result of ubiquitous computing and Agile development practices. The variety of skills test teams need today has expanded to the point that hiring traditional testers does not cut it in software development any longer. For this reason, many organizations have had to go offshore to find available resources for projects.

Increased skills are needed for all testing facets. New devices, platforms, automation tools, big data methods, and virtualization all require new levels of expertise. Today, performance issues with virtual machines (VMs) running android emulators is a persistent problem. Automating the test cases to run against the emulators on these VMs will require talented experts to sort out current issues and maintain usable performance.



Many Agile practices espouse cross-functional teams. While this is difficult or impossible for the vast majority of teams, testers need to grow their cross-functional skills to do more than just execute test cases! Testers need to be *multi-functional*, whether that is helping with UI design, code-review, automation engineering or help/documentation. Many organizations today think of testers as software developers who test.

Fully Distributed

We live with global products and teams that are assembled from a globally distributed workforce. Skill, availability, cost, continuous work/speed—these are all driving organizations to distribute tasks for optimal execution.

We know distribution of testing tasks is successful and, at the same time, needs more management oversight, great communication infrastructure, [increased use of ALM tools](#), and cross-cultural training.

The Agile/Scrum move to co-located teams, balanced against *both* the cost and availability of skilled staff, is still playing out on most teams. What has been successful, and what I would advise any team looking for staff optimization, is to first distribute test automation tasks. The level of expertise needed to engineer new test automation frameworks can be successfully separated from test design and jump start automation.

SMEs (Subject Matter Experts)

Not only do testers need to become more technical, all members of the team need to increase subject matter or domain knowledge. Subject matter experts are still essential but they need to work together with other, more technical testing staff on test case design for automation, data selection and additional tasks such as flushing out User Story acceptance criteria and bug advocacy. Ultimately, super specialized test teams will evolve—with domain testers at one end and niche technical testers at the other.

Interpersonal Skills

With the blending of work and home life- working hours to overlap distributed teams, working in bull-pens/cross functional project team areas—people working from home, working at a *sustainable pace* is a key eXtreme Programming practice.

In the new software development paradigm, there is a much closer relationship with product owners, customers, business analysts and testing. Development organizations always need more teamwork, seamless communication, trust, respect and all the other people skills that are essential in a highly technical, distributed environment. Soft skill development is essential, not optional, to thrive in the rapid, distributed development world. Skills include: leading technology workers, cross-cultural communications and using ALM tools.

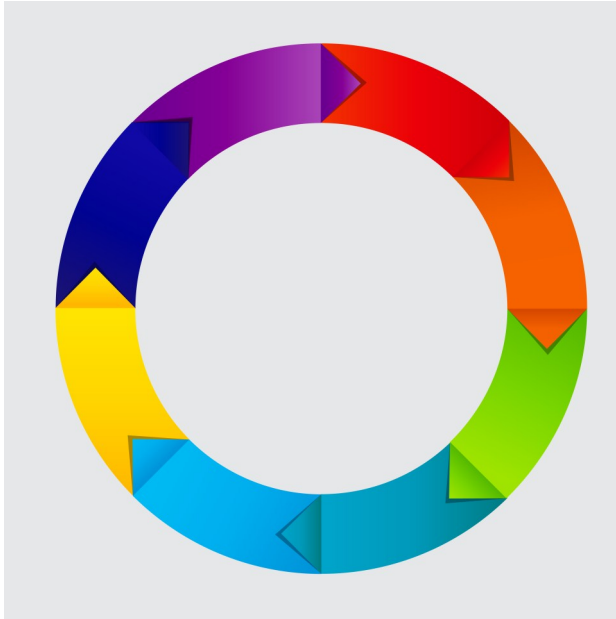
Training

Lean development as a software practice is gaining many followers today. One of the 7 Lean Principles is to *amplify learning*. With the rate of technology, product, and development practice change, it's a no-brainer that there is a constant need for training. For an in-depth view of training in testing, read our [February, 2012 Training issue](#).

Process

Team Process

Software development processes/SDLCs have changed in the past 10 years like never before in the history of software development. The shift toward more metrics, more dashboards and better measurement prevalent in the late 90s



and early 2000's has totally reversed. Now the focus is on leaner processes with the best metric of team success being the delivery of working software. The only metric in Scrum is burndown, velocity is secondary, and bug counts may be moot and turned into user stories. This may be extreme for your organization, but it is definitely the movement.

Software development has been greatly impacted by the use of ALMs (application lifecycle management tools). This is not the place to detail ALMs, (I went into great detail on the subject in a [past article](#)) except to say that their use today is bigger than ever and growing. That these tools are more integrated with testing, automation and test management is new.

ALMs can greatly facilitate communication with offshore teams. Continuous Integration (CI) is a pervasive XP practice today. The best ALM tool suites have complete CI tooling (source control, unit test execution, smoke test/UI/automated build validation test execution, autobuild, traceability, and reporting) or easy integration for these.

Test Process

Testing often lags behind development in most organizations and test teams are still responding to Agile. Many test teams are struggling in [ScrumBut](#) processes. Most tests are trying to prevent "handoffs" and waste rework as well as being flexible and responsive. As noted above, test teams are still figuring out test practices on Scrum teams and in XP practices. This has been the subject of many articles at [LogiGear Magazine](#).

The biggest test process impacts, or *game changers* to testing in Agile/XP, are CI and the need for high volume test automation. These are detailed in the following "Practices: Methods and Tools" section.

Joint Ownership - Ownership of Quality

It has been a long time since people who test gladly got rid of the QA (quality assurance) job title. However, some unaware people continued to believe in those who test guaranteed quality rather than measure it, report risk, find bugs, or any other achievable tasks. The Agile movement has finally done away with the illusion with more *Teams* today espousing *joint ownership of quality*. There are more upstream quality practices: code review, unit testing, CI, better user story and requirements from need for customer satisfaction and closer communication with test teams. Following the Scrum framework, most teams now have the Product Owner as the ultimate owner of quality.

Speed

The speed of development today is a changing process. The pressure to deploy rapidly and simultaneously on multiple devices or platforms— the ease of distribution, market pressure, cost pressure, *do more with less* pressure— is pushing some teams to abandon or reduce even some universally accepted practices, such as hardening or release sprints, to accommodate the [need for speed](#). Abandoning process for speed is as old as software development itself, but the pressure is greater today. This is risky when testing large and complex systems where test teams need high-volume test automation and excellent risk reporting skills.

SaaS/Cloud

There are evolving process changes in the Cloud, particularly in SaaS. With the immediacy of deployment, rapid change, and ever-smaller or non-existent architecture, some Agile teams cannot keep up. With the speed SaaS products are made available to all customers, patches, hot-fixes or roll-backs are particularly problematic. As always, large-scale test automation is a team's most important defense.

Practice - Methods and Tools

Methods

This area is leading the most dramatic changes in software testing.

Massive Automation– Smoke Tests and Regression Tests

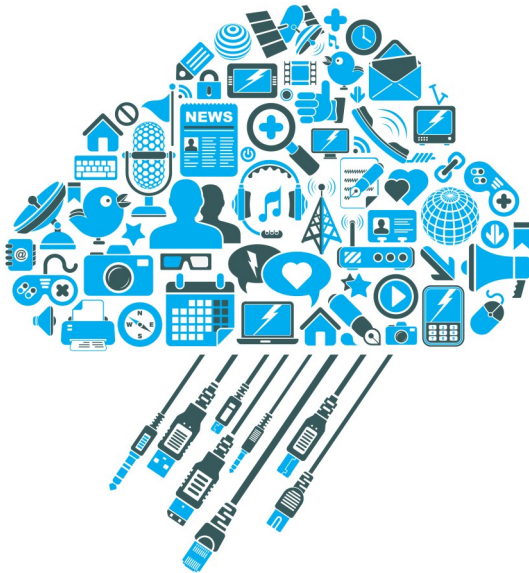
There are two foundation practices that have had a profound impact on the testing landscape. [Continuous Integration](#)—mistaken sometimes to mean only auto build—is the automatic re-running of unit tests, the test team's smoke or UI automated tests, and reporting changed code, fixed bugs and automation results. This process is significant to the new *Agile* speed of development, requiring rapid release, and is dependent on test teams having significant smoke or build validation tests as well as traceability. Using an ALM such as TFS (Team Foundation Server), for example, that has all the pieces for CI built-in, the reporting can detail the code changes and the bug fixes. Gone are the days of getting bad builds, or good builds with no idea what changed. The test team must have a great automated smoke test to have a true CI practice.

Regardless of what other practices you do, or advances your team is making to keep up with the new paradigms in software development, you cannot succeed without massive automation. As we know, old, high-maintenance, random, automation methods will not work today. Automated regression tests are not only for speed, they are necessary for confidence, coverage, focus, information and to free us up to do more exploratory discovery. This last point is often lost in the new Agile development paradigms. With reduced design time and vague user stories, test teams need to discover behaviors (intended or unintended), examine alternative paths and explore new functionality. No one has the time to do this without massive automation.

We also know that scripted, old style automation does not work. I suggest you read Hans Buwalda's [article](#) on Action based Testing for the most advanced methods for test automation.

The Cloud and SaaS

The Cloud



The common use of the cloud in software development today is virtualization of environments to quickly share tools, resources and run simulations of both hardware and software. This becomes more relevant in the context of the current explosion in the mobile applications market for smartphones and tablets. These applications require shorter development cycles and automated virtual regressions making cloud testing cost-effective and efficient.

SaaS

Software-as-a-Service presents unique test method challenges that are different from process changes with SaaS. A positive aspect of SaaS testing is the ability for teams to control the production environment. There is no server side compatibility testing and even client-side access can be restricted and controlled.

The biggest impact of SaaS on test methods is the necessity and growing importance of performance, load, security, and scalability testing, in addition to having our hands full with functionality testing. Where load, performance and scalability testing have often been postponed due to the method's reputation of being expensive, slow, *nice-to-have*, optional, etc., there needs to be crucial discussion before compromising security testing.

The value proposition behind SaaS is: let "us" manage the servers, databases, updates, patches, installs and configuring. Everything your IT team used to do when you bought software, you can now buy as a service. Implicit in this proposal is - "We" will manage and secure your data. The biggest hurdle most companies face in letting someone outside the company manage and maintain corporate intellectual property is data security. Security testing is not optional. In some cases, security testing is performed by specialized teams with their own tools. In most cases I see, the test teams are integral to security testing in:

- Test case design.
- Data design.
- Stress points.
- Knowledge of common and extreme user scenarios.
- Reusing existing automation for security testing.

Virtualization

The primary impact of virtualization is the ability to do things faster.

Virtualization makes managing test environments significantly easier. Setting up the environments requires the same amount of work, but regular re-use becomes significantly easier. Management and control of environments through virtualization is now a basic tester skill for:

- Executing tests on multiple virtual servers, clients, and configurations using multiple defined data sets increases test coverage significantly.
- Reproducing user environments and bugs is easier.
- Easy tracking, snapshots and reproducibility to aid team communication.
- Whatever resources are constrained, such as database, service or tools, your test automation and manual execution can be virtualized for independent, faster, shared or not, independent execution.
- Facilitate faster regression testing economically using many virtual machines.
- Virtualization frees-up your desktop.

Performance bottlenecks and problems are common with virtualization test projects. Automation may be more complex to set-up. Daily deployment of new builds to multiple (think 100s) of VMs needs skill to run smoothly. To be effective, your team will probably need a virtualization or tool expert.

Devices That Think

Embedded system testing is very different than application testing. With the explosion in *things that think* and ubiquitous computing/sensors and devices that interact, and often provide some components of *big data*, new test methods are needed.

As an overview of some of the complications:

- How to generate a function call and parameters:
 - Need for new tools, APIs, various input systems.
 - Need for virtual environments.
- Separation between the application development and execution platforms (the software will not be developed on the systems it will eventually run on. This is very different from traditional application development).
- How to simulate real-time conditions.

Difficulty with test validation points or observed behaviors:

- Parameters returned by functions or received messages.
- Value of global variables.
- Ordering and timing of information.
- Potentially no access to the hardware until final phases of development.
- All these will potentially complicate test automation.

With the added worry that many embedded systems are *safety-critical systems* we will need technical training, more tools, new thinking, new methods and more sophisticated test automation for devices that think with no UI.

Big Data

Big Data methods are not jump running a few SQL queries. Saying big data is difficult to work with using relational databases, desktop statistics and visualization packages is an understatement.

Instead, "massively parallel software running on tens, hundreds, or even thousands of servers" is required. What is considered "big data" varies depending on the capabilities of the organization managing the dataset, and on the capabilities of the applications that are traditionally used to process and analyze the data set in its domain. For some organizations, facing hundreds of gigabytes of data for the first time may trigger a need to reconsider data management options. For others, it may take tens or hundreds of terabytes before data size becomes a significant consideration (www.en.wikipedia.org/wiki/Big_data).

Correlating global purchases at Wal-Mart with direct advertising, marketing, sales discounts, new product placement, inventory and thousands of other analyses that can be pulled from the 2.5 petabytes of data creates serious testing challenges. Testing the data, the algorithms, analysis, the presentation, the interface, all at the petabyte levels requires software testing by people like you and I.

This scale of testing will require huge virtualization efforts, a new test environment set-up, management skills, data management, selection and expected result prediction, as well as the obvious data analysis skills. There will be heavy tool use in big data testing. Quick facility in aggressive automation is a pre-requisite.

Mobility

Much has already been written about the increased needs for mobile testing. I will focus on two less discussed ideas.



There is less and less focus on the *phone* aspect of mobile devices, and more focus on intersecting technologies. From common intersections such as Wi-Fi, Bluetooth and GPS to newer and high growth intersections of RFID, the currently hot NFC (near-field communication), ANT+ (used mainly for medical devices), QR Code (quick response code) and a growing variety of scanners. As with things that think, learning new technology, new input tools and methods, capturing output for validation and observation and complexities for test automation are demanded with fast ramp-up.

Security and performance testing are two commonly neglected aspects of mobile testing. Much [lip-service](#) is given to the need for these test methods which need different planning, skill, expensive tools and *time* to execute. In today's fast-at-all-cost delivery environments these test methods very often get postponed until a crisis occurs (check out "[Mobile Testing By the Numbers](#)" for information on how much mobile apps are performance tested). Testers need to be excellent at risk assessment and planning for the times when these are added back into the test strategy.

New Input Technologies

In the past, input variation meant testing mainly keyboards for Japanese and Chinese character input. It's no longer that simple. Even the recent phrase, "Touch, type or tell" must now include gesture. Add in the new and more common [virtual thumbwheels](#) and the range of innovation from companies such as Swype and Nuance and even the most traditional business productivity app must be touch-screen and voice enabled. As with many of the innovations changing the landscape of software testing today, technology specific training, new skills and new tools are a pre-requisite to testing effectively.



Tools

A theme in the new development world is: *always leverage technology*. Regardless of your product market space, your team's tool use should have recently increased quite significantly. There has been a new focus on test tools like never before.

While I have stated testing tools often lag behind development tools, the lag is getting smaller. With business seeing the need for speed and testing often being the bottle-neck due to lack of good tools, skills, test environments and access to builds and code, there has been huge change in tooling for test teams. Today, the availability of significantly better test automation frameworks supporting more platforms and devices and integrating with more ALM tools is not only commonplace, but essential. And it's becoming more common for test teams to take over, or bring key members in to own the continuous integration tool.

In addition to ALM and Continuous Integration, as stated earlier, virtualization tools are widely used today and are becoming an IT necessity.

You will probably need tools specific to a technology (example: Google's Go! or F#), platform, or device (example: input devices, simulators and emulators) as well.

Significant, high-volume test automation is essential, not optional. Old style test automation will not meet project needs today. Fast, efficient, low-maintenance test design

and automation is mandatory. Tools that support action-based testing can help you achieve the highest production and more effective levels of automation. If this includes more programmers focused on designing code for testability and automation stability, so be it—the testing landscape is undergoing revolutionary change. Leverage technology wherever possible!

Summary

Be prepared! The landscape of software testing has changed forever.

The world, business, the marketplace, and society have moved toward ubiquitous computing. *Things that think* are everywhere. And what we test changes how we test. The flood of new, non-traditional products and services has greatly increased the skills needed to test. And, as always, *test everything*, but do it faster and cheaper.

If you are not mobile you will be. If your product has no mobility, it will.

If you are not fully integrated with your programmers and everyone on the development team, you will be.

If you do not currently have an expanded role in security, scalability, load and performance testing- you will.

If you are not doing high-volume automation, you will be.

If you are not virtualized, you will be.

If you are not fully distributed, you will be.

If you are not thinking/re-thinking what you distribute and what you keep in the home office, you will be. ■

About Michael



Michael Hackett, Senior Vice President, is a founding partner of LogiGear Corporation. He has almost two decades of experience in software engineering and the testing of shrink-wrap, client/server and web-based applications in Banking, Securities, Healthcare and Consumer Electronics.

References

Wikipedia

Application testing in the Cloud
By Joe Fernandez, Oracle and Frank Gemmer, Siemens

[CES 2013: State Of The CE Industry And Trends To Watch](#)
By Eric Savitz, Forbes January 2013

[The Law of Accelerating Returns](#)
By Ray Kurzweil

Feature

Deploying the Cloud? Make 2013 the Year to Do It Right

Leverage the benefits of cloud technology across your enterprise.

By Pete Schmitt, Customer Storage Inc.



It's no secret that the cloud is growing at an exponential rate. By 2016, two-thirds of the world's server workloads will exist in the cloud. But according to Cisco's 2012 Cloud Index, less than half of server workloads currently run in the cloud. Closing the gap between current capabilities and future requirements is a mission-critical priority for businesses across a range of industries. Without adequate planning and preparation, the race to the cloud can easily become a long slog through a minefield of missed opportunities, user failures and IT nightmares.

As more and more workloads make their way to the cloud each year, enterprises have a vested interest in expanding network capabilities and evolving critical data center infrastructure to accommodate an ever-increasing array of cloud-based applications and data storage requirements.

Key Trends in Cloud Technology

Several trends are driving the migration of applications and data to the cloud.

- **Agility:** Cloud deployment enables businesses to improve performance and functionality quickly, launching new applications without a corresponding need for additional infrastructure. Agility is especially important in new and young companies, many of which lack the time and resources to deploy a range of diverse applications internally.

- **Consumerization of IT:** The Bring Your Own Device (BYOD) trend is enabling companies to expand the use of technology through the use of employee-owned devices. The cloud is playing an important role in helping organizations keep up with the pace of BYOD and deliver anytime, anywhere access to workers.
- **Cost Drivers:** Financial metrics are also a motivating factor in the race to the cloud. In general, Software-as-a-Service (SaaS) is cheaper, faster and easier than traditional deployment models - reducing the cost of infrastructure, physical space and IT labor.

Preparing for the Cloud

Successful preparation for future cloud workloads requires planning. By strategically adapting your network capacity, data center and other critical IT functions, you can substantially improve your organization's ability to operate in the cloud.

Today's networks must be capable of handling constant interactions characterized by rich media and heavy content, particularly as users migrate away from email toward interactions via social media and other channels. Consequently, networks and data centers must expand to support instant access to many different types of content beyond email.

The first step of network expansion is a comprehensive assessment of your organization's app portfolio. In most cases, executive decision-makers are unaware of the scope of applications that are running in the organization. Once all of the applications that are currently running in your organization have been identified, they need to be ranked and categorized according to future requirements. While some applications may need to remain in-house, others can be transferred to a public cloud environment. From there, the organization can begin to evaluate how to expand the network to manage future workloads.

As virtualization becomes more prevalent in data centers and companies adopt a cloud-based strategy, network architects need to rethink and redesign their current infrastructure to adapt to the new traffic patterns. What once used to be primarily "North-South" network traffic flow is now becoming "East-West." Environments are becoming highly dynamic; workloads are moving to different physical locations on the network as virtual servers are migrated and clients move about the building. The architectures and networking techniques of yesterday are not necessarily well suited to the architectures and applications of today and tomorrow.

A thorough understanding of networking, the infrastructure that is its foundation, and its relationship to applications is necessary when architecting a data center network that is capable of supporting and adapting to future challenges that arise as a result of virtualization and cloud computing. The solution must address all aspects of application delivery - security, availability, performance, and visibility - while exhibiting the qualities and characteristics that define cloud architectures including affordability and elastic scalability. The data must be protected against attacks, intrusions, breaches, and leaks and categorized based on its importance and network resource needs with Quality-of-Service (QoS) capabilities.

Storage and Backup is another key to preparing for cloud migration. Security is a top-of-mind issue in the cloud. Although cloud deployments offer real benefits, your organization needs to know that sensitive data will remain secure. The process for preparing for cloud-based data storage and backup mirrors the process for evaluating network expansion requirements. Starting with an assessment of current data sources and storage routines, the organization needs to evaluate what types of data can eventually either integrate with or be completely transferred to the cloud. Equipped with this information, the organization can begin to identify the technology gaps that need to be addressed to meet future cloud storage and backup requirements.

Purpose-built appliances provide fast backup and restore and deliver local-like performance while using the cloud for secure off-site storage. This helps avoid the need to provision and manage a secondary site for Disaster Recovery (DR) or long-term storage. This can dramatically reduce capital spending, streamline IT infrastructure, and enable payback periods that are measured in months, not years. Appliances can be combined with existing data protection applications and private or public clouds, creating a low cost, highly scalable storage tier for old or infrequently accessed data. Appliances also allow organizations of all sizes to modernize their data protection architecture, eliminate tape, improve scalability, and improve Disaster Recovery readiness. Cloud storage allows organizations to leverage a pay-for-use pricing model and anywhere availability.

Increased virtualization will alleviate some of the cloud migration challenges upfront. Cloud technology enables organizations to move servers to the cloud and back in an integrated and strategic manner. In fact, the use of virtualization can also play an important role in preparing the organization's culture and stakeholders for future cloud deployments. By increasing the use of virtualization now, you can encourage greater acceptance of the cloud across your enterprise. In essence, virtualization can serve as a bridge to the cloud.

It is the key technology that enables the cloud, and without it there is no cloud. The ability to separate the OS and application from the hardware allows it to be the foundation required for on-demand cloud services. The encapsulation offered in virtualization and the mobility that enables a live virtual machine to be moved with no downtime for the application is what the cloud is built on. If you look at virtualiza-



tion / cloud computing as a whole, it really is not about a product, but a journey. Companies initially enter the world of virtualization because they just can't keep up with the increased scale, complexity, and management requirements while maintaining their current traditional infrastructure.

This leads them to the first step in the virtualization journey, which is to consolidate their resources/infrastructure to get better utilization of their servers and to reduce their energy cost. Higher levels of abstraction allow companies to take advantage of the intelligence built into the virtualization software. Intelligent software allows High Availability (HA) and Replication, load balancing, pooled resources, self-automation/orchestration, service definitions or service profiles, templates, policies, self-service portal, service catalog(s), security and identity management, system monitoring/management, capacity planning, billing and charge-back, and licensing.

It's important to understand that the ability to handle future cloud-based workloads will present different challenges and concerns for the stakeholders in your organization; users are motivated by ease-of-use and increased access to applications and data, CIOs are focused on control, ownership and security and CFOs are primarily concerned about the cost savings, rate of return and OpEx versus CapEx, etc. By thoughtfully and strategically preparing for future cloud opportunities, your organization can address these concerns and fully leverage the benefits of cloud technology across the enterprise. ■

About Pete

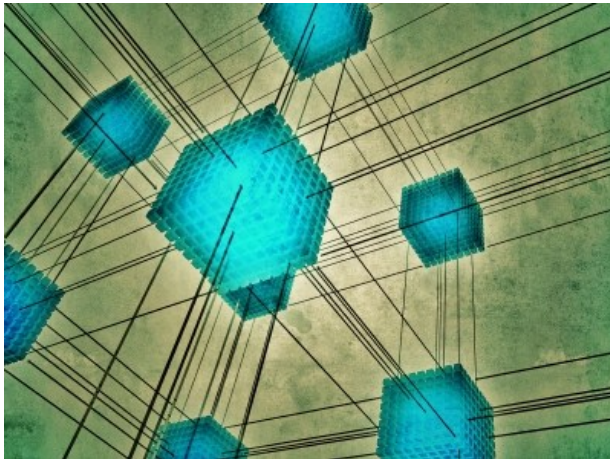
Pete Schmitt is Vice President of Engineering at Customer Storage Inc. Since 2002, cStor has helped companies strategize, create, and implement best in class data center solutions that address business needs. cStor's proven capabilities with key data center technologies provides clients with a fresh perspective, the ability to collaborate with recognized data center experts, and the confidence that goals will be met.

Feature

Service Virtualization Rises to Cloud App Testing Challenge

Service virtualization provides a way to get around that scale issue that is not only practical, but more importantly should lead to better code being deployed the first time every time.

By Michael Vizard, IT Business Edge



One of the challenges with building an application these days is the number of dependencies that application will actually have on other applications. Ideally in order to know how that application will actually perform, application developers would be able to test their application against the application it depends on running in production. The odds of that happening, however, are slim to none, especially if that other application is running as a cloud service that has to be always available to end users.

To solve this problem developers are increasingly turning to [service virtualization](#), which is an emerging set of application testing technologies that allows a developer to create a replica of another application in a testing environment. In fact, a [recent survey](#) of 200 in-house software development executives and managers from enterprises with revenues of more than US \$1 billion dollars in North America - the majority (71%) with over \$2 billion annual revenues - conducted by the market research firm Coleman Parkes Research on behalf of [CA Technologies](#), found that not only does the inability to adequately test applications result in missed deadlines, entire functions wind up being eliminated and the development team as whole lacks confidence that the application will work as advertised.

Given the often limited scope of most application testing that may not be all that surprising. In fact, interest in agile development methodologies aside, as the amount of liability

attached to an application increases the more cautious an organization becomes. What's definitely needed, says Shridhar Mittal, general manager for service virtualization at CA Technologies, is a new approach to testing applications that for the most part are mashups of any number of existing applications that often have dramatically different service characteristics. The challenge, of course, is figuring which one of those applications might adversely affect the performance of your application before your application discovers that issue in a production environment, says Mittal.

Otherwise, says Mittal, all any organization is doing is releasing code on a little more than hope and a prayer that it will actually work.

As [applications become increasingly borderless](#) thanks mainly due to the proliferation of APIs that serve to make applications more accessible, the more tempting it becomes to invoke third-party APIs. But as we all know, the quality of APIs tends to vary widely across the Web. Right now many organizations are using agile development methodologies that in many instances amount to little more than trial and error when it comes to invoking APIs.

As the number of application releases and updates that organization are rolling out in a given year steadily increases it's pretty clear that existing approaches to testing applications won't scale in the age of the cloud. Service virtualization provides a way to get around that scale issue that is not only practical, but more importantly should lead to better code being deployed the first time every time. ■

Originally published on ProgrammableWeb (programmableweb.com)

About Michael



Michael is the Editor in Chief of InfoWorld Media Group where he has been covering computer technology for more than 14 years. He is also a member of the senior leadership team, which provides the strategic vision for InfoWorld Media Group.

Glossary: Software Testing Landscape

Virtualization: Virtualization (or virtualisation) is the simulation of the software and/or hardware upon which other software runs. This simulated environment is called a virtual machine. There are many forms of virtualization, distinguished primarily by computing architecture layer. Virtualized components may include hardware platforms, operating systems (OS), storage devices, network devices or other resources.

Big Data: A collection of data sets so large and complex that it becomes difficult to process using on-hand database management tools or traditional data processing applications. The challenges include capture, curation, storage, search, sharing, analysis, and visualization.

ALM: A continuous process of managing the life of an application through governance, development and maintenance. ALM is the marriage of business management to software engineering made possible by tools that facilitate and integrate requirements management, architecture, coding, testing, tracking, and release management.

Agile: Characterized by quickness, lightness, and ease of movement; nimble. Not necessarily characterized by fast speed.

Agile software development is a software development practice based on iterative and incremental development where requirements and solutions evolve through collaboration between self-organizing, cross-functional teams. It promotes adaptive planning, evolutionary development and delivery, a time-boxed iterative approach, and encourages rapid and flexible response to change.

Kanban: Kanban is a method for developing software products & processes with an emphasis on just-in-time delivery while not overloading the software developers. It emphasizes that developers pull work from a queue, and the process, from definition of a task to its delivery to the customer, is displayed for participants to see.,

Offshoring/GSD/DSD: Global Software Development (GSD) is "software work undertaken at geographically separated locations across national boundaries in a coordinated fashion involving real time (synchronous) and asynchronous interaction".

Distributed development is a software development model in which IT teams spread across geograph-

ical lines collaborate on applications or various software. These teams are often separated by mini-projects that are brought together for a final software buildout.

Distributed development is a familiar IT approach, but source code control and other issues of the recent past make it less than ideal. However, modern and advanced Web-based tools and collaborative techniques allow teams to work effectively in a distributed fashion.

Cloud computing: The use of computing resources (hardware and software) that are delivered as a service over a network (typically the Internet). The name comes from the use of a cloud-shaped symbol as an abstraction for the complex infrastructure it contains in system diagrams. Cloud computing entrusts remote services with a user's data, software and computation.

Testing in the Cloud: Cloud Testing uses cloud infrastructure for software testing. Organizations pursuing testing in general, load, performance testing, and production service monitoring in particular are challenged by several problems like limited test budget, meeting deadlines. High costs per test, large number of test cases, and little or no reuse of tests and geographical distribution of users add to the challenges.

Testing cloud apps: Cloud testing is often seen as only performance or load tests, however, as discussed earlier it covers many other types of testing. Cloud computing itself is often referred to as the marriage of software as a service (SaaS) and utility computing. In regard to test execution, the software offered as a service may be a transaction generator and the cloud provider's infrastructure software, or may just be the latter. Distributed Systems and Parallel Systems mainly use this approach for testing, because of their inherent complex nature. D-Cloud is an example of such a software testing environment.

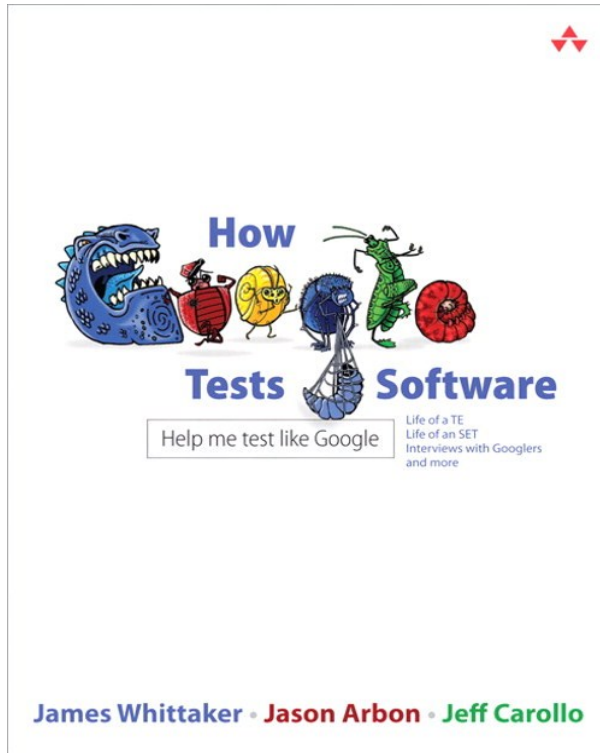
SaaS: Sometimes referred to as "on-demand software", is a software delivery model in which software and associated data are centrally hosted on the cloud. SaaS is typically accessed by users using a thin client via a web browser.

Sources: Wikipedia, Techopedia

Book Review

How Google Tests Software

By Adrian Woodhead, Last.fm



Having developed software for nearly fifteen years, I remember the dark days before testing was all the rage and the large number of bugs that had to be arduously found and fixed manually. The next step was nervously releasing the code without the safety net of a test bed and having no idea if one had introduced regressions or new bugs.

When I first came across unit testing I ardently embraced it and am a huge fan of testing in various forms — from automated to smoke tests to performance and load tests to end user and exploratory testing. So it was with much enthusiasm that I picked up *How Google Tests Software* — written by some of the big names in testing at Google. I was hoping it would give me fresh insights into testing software at "Google Scale" as promised on the back cover, hopefully coupled with some innovative new techniques and tips. While partially succeeding on these fronts, the book as a whole didn't quite live up to my expectations and feels like a missed opportunity.

The book is written in an informal, easy to read manner and organized in such a way that readers can read chapters in any order or just choose to focus on the parts that

interest them. One annoying layout choice is to highlight and repeat certain key sentences (as is often done in magazines) resulting in one reading the same thing twice, often only words away from the original sentence. Thankfully this is only the case in the first two chapters, but it highlights the variable quality of this book — possibly due to the authors having worked separately on different chapters. *How Google Tests Software* isn't a book for people new to testing or software development. The authors assume you know a fair amount about the software development lifecycle, where testing fits into this and what different forms testing can take. It is also largely technology neutral, using specific examples of testing software that Google uses only to illustrate concepts.

After a brief introduction as to how testing has evolved over time at Google, the book devotes a chapter to each of the key testing-related roles in the company: the 'Software Engineer in Test' (SET), the 'Test Engineer' (TE) and the 'Test Engineering Manager' (TEM). SETs are coders who focus on writing tests or frameworks and infrastructure to support other coders in their testing. The TE has a broader, less well-defined role and is tasked with looking at the bigger picture of the product in question and its impact on users and how it fits into the broader software ecosystem. These two sections form the bulk of the book in terms of pages and interesting content. The TEM is essentially what the name says — someone who manages testers and testing and coordinates these activities at a higher level within Google.

The descriptions of each of these testing roles highlights the ways Google's thinking about testing has matured and also shows how some of these approaches differ from other companies'. There are also explanations of the tools and processes that people in these roles use and follow and this for me was the most interesting part of the book.

Topics covered include: specific bug tracking and test plan creation tools; risk analysis; test case management over time; and automated testing. Particularly of note are discussions on using bots to perform testing of web pages to detect differences between software releases, cutting down on the amount of human interaction required as well as the opposite approach — using more humans via "crowd sourced testing" among first internal and then select groups of external users. The tools that Google utilizes to simplify tester's jobs by recording steps to reproduce bugs and simplifying bug reporting and management sound very useful. Many of the tools described in the book are open source (or soon to be opened) and are probably worth following up on and investigating if this is what you do for a living.

In addition to the main body of text most chapters also include interviews with Google staff on various testing related topics. Some of these are genuinely interesting and give the reader a good idea of how testing is tackled at Google on a practical level. However some of the interviews fall into the "navel gazing" camp (especially when the authors interview one of themselves) and feel more like filler material. I enjoyed the interviews with Google hiring staff the most — their take on how they recruit people for testing roles and the types of questions they ask and qualities they look for make a lot of sense. The interview with the GMail TEM was also good and illustrated how the concepts described in the book are actually performed in practice. The interviews are clearly marked and can thus be easily skipped or skim read but one wonders what more useful text could have been included in their place.

The book wraps up with a chapter that attempts to describe how Google intends to improve their testing in the future. The most valuable point here is how testing as a separate function could "disappear" as it becomes part and parcel of the product being developed like any other feature, and thus the responsibility of all of the people working on the product as opposed to it being a separate thing. Another key point made throughout the book is how the state of testing at Google is constantly in flux which makes sense in such a fast moving and innovative company but leaves one questioning how much of this book will still be relevant in a few year's time.

How Google Tests Software isn't a bad book but neither is it a great one. It has some good parts and will be worth reading for those who are interested in "all things Google." For everyone else I'd recommend skimming through to the parts that grab your attention most and glossing over the rest. ■

About Adrian



Adrian is a hands-on technical team lead at Last.fm in London where he works with a team focusing on the services behind the scenes that power the popular music website. Prior to this Adrian worked in Amsterdam for 2 mobile startups (both of which failed), a bank (good money but dull), a content management system (even duller), a digital rights company (good

in theory, evil in practice) and in South Africa for a multimedia company. You can visit his website at <http://massdosage.co.za>

TestArchitect™

**YOUR
HIGH
VOLUME
TEST
AUTOMATION
SOLUTION**

TestArchitect™ features:

- All-In-One Solution: Test Management, Test Development and Test Automation
- Action Based Testing™ Methodology
- Built-In Customizable Automation
- Remote Test Execution
- Customizable Dashboard

Vietnam View

What the Pho? A look into Vietnam's Signature Dish

Pho [fuh] *noun* - a Vietnamese soup made of beef or chicken stock with rice noodles and thin slices of beef or chicken.

By Brian Letwin, LogiGear Corporation



Vietnam draws thousands of foodies each year who are eager to explore the country's large selection of fresh and simple food. The options are nearly endless, and it's no exaggeration to say one would need to invest a few years in order to sample everything.

Vietnamese cuisine in general has clear regional distinctions. Hanoi is famous for *xoi xeo*, the dish with sticky rice, fried onions, and ground mung beans. Hue features a plethora of gelatinized rice and tapas style dishes, and Hoi An is known for *cau lau*, a soup with noodles made from the ashes of local trees. But there's one Vietnamese soup dish that transcends not just regional geography, but has become popular all over the world – *pho*.

Pho is a relatively recent food creation, making its first appearance in a Hanoi textile market at the turn of the 20th century. But the dish has come a long way from its regional beginnings and today, *pho* is available everywhere – on the street, in small, family run restaurants, and even in franchise-style restaurants.

Pho is a soup made from simple ingredients: Beef, marrow-rich beef bones, star anise, roasted ginger, roasted onion, black cardamom, coriander seed, fennel seed and clove. The simplicity of the dish, along with the flavor, contributes to pho's widespread appeal.

There's no denying that the broth is the most important element of Vietnamese pho. It is in fact the key to a successful bowl of pho. Once the broth is ready, it is served with a generous portion of rice noodles and garnishes such as green onions, white onions, Thai basil, fresh chili peppers, lemon or lime wedges, bean sprouts and coriander or cilantro. Fish sauce, hoisin sauce and chili sauce are all common options for adding more flavor to the broth.

Pho lovers judge a bowl served to them in a restaurant by sipping the broth first without putting in any seasoning or garnishes, but the Vietnamese always say that the best pho you will ever taste is the one cooked by your own mother.

While it's not uncommon to see women starting the broth before sunrise to accommodate the impending breakfast rush—though *pho* can be eaten at any time of day—many young adults say their own mothers have given up making pho because of the time needed to prepare the broth. Now if they want pho, they go out to fulfill their craving for the dish. Thanks to its popularity, it's not difficult to enjoy a delicious bowl of soup rain or shine, day or night. ■



TestArchitect™

Action Based Automated Testing Goes Mobile



testarchitect.com | 800.322.0333

LogiGear®



LOGIGEAR MAGAZINE
FEBRUARY 2013 | VOL VII | ISSUE 1

United States	Viet Nam, Da Nang	Viet Nam, Ho Chi Minh City
2015 Pioneer Ct., Suite B	7th floor, Dana Book Building	1A Phan Xich Long, Ward 2
San Mateo, CA 94403	76-78 Bach Dang	Phu Nhuan District
Tel +1 650 572 1400	Hai Chau District	Tel +84 8 3995 4072
Fax +1 650 572 2822	Tel: +84 511 3655 333	Fax +84 8 3995 4076
	Fax: +84 511 3655 336	