

# BETTER<sup>TM</sup> SOFTWARE

A TECHWELL<sup>TM</sup> PUBLICATION

A NEW REALITY  
Explore Service  
Virtualization

USE IT OR LOSE IT  
Build Your Discovery  
Muscle

*A Look*

BEHIND

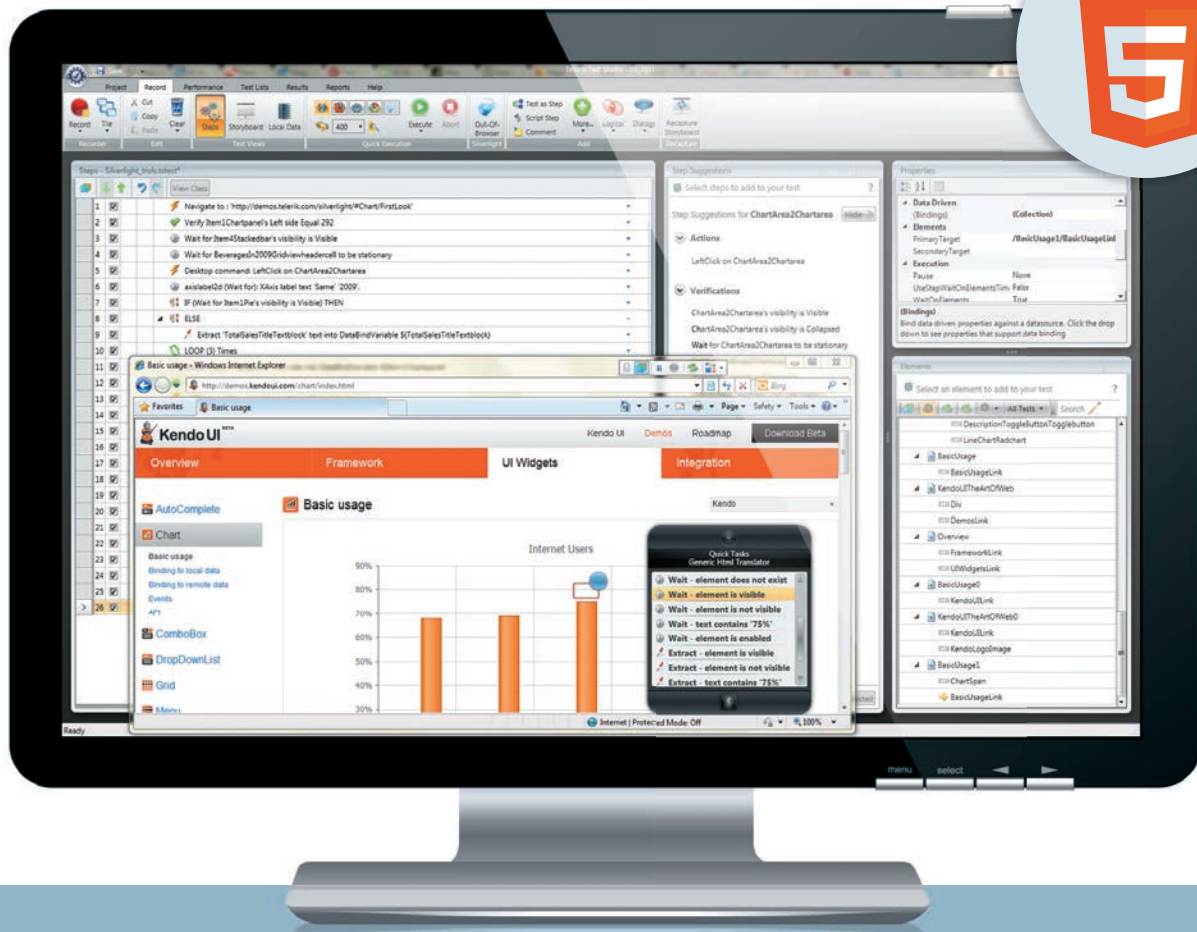
THE

CURTAIN

*Debunking 12 Myths of  
Agile Development*

# Test Studio

Easily record automated tests for your modern HTML5 apps



Test the reliability of your rich, interactive JavaScript apps with just a few clicks. Benefit from built-in translators for the new HTML5 controls, cross-browser support, JavaScript event handling, and codeless test automation of multimedia elements.

[www.telerik.com/html5-testing](http://www.telerik.com/html5-testing)



*Joe says it works  
on his machine.  
Great. He's a one hit  
wonder.*



Looking for cleaner ways to manage your build and deploy process?  
Learn more about our Dynamic DevOps Suite at  
[www.Openmakesoftware.com/DevOps-Answers](http://www.Openmakesoftware.com/DevOps-Answers)

**OM** Open Make **Software**

[www.openmakesoftware.com](http://www.openmakesoftware.com) | 312 440-9545

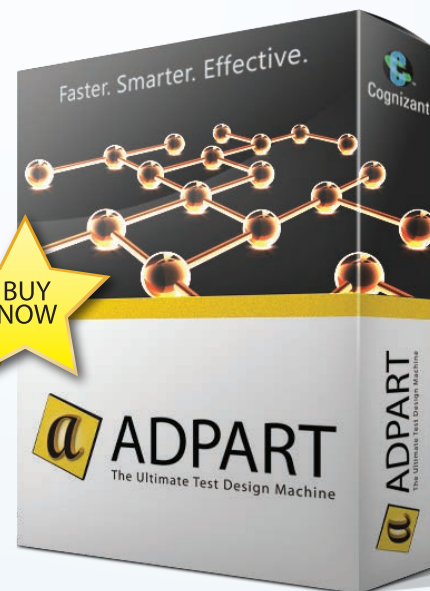


# Suite to suit all your quality needs !

Our experiences over the last decade of providing quality engineering & assurance to Fortune 500 companies, enabled us to build an innovative Quality Assurance Suite that addresses the ever-changing industry landscape and its demands.

"Our Quality Assurance Suite customers are realizing significant cost savings and are able to go-to-market with certainty and speed"

ADPART is an innovative model based test design solution that generates optimized test cases from real-time business requirements at a touch of a button! It gets even better as you save up to 50% of your effort in manual test design and improve overall quality. That means you work less and save more time and money!



You may also be interested in,



## Automating Automation

Intelligent automated test script scheduling and quality engineering solution



## Get Authorized

Validated electronic signature for authorizing testing artifacts stored in HP-QC/ALM, ensuring regulatory compliance



## Mobile Automation

Cross Platform Test Automation tool for both native and web applications across multiple mobile OS platforms



## Make Scripts Perform Better

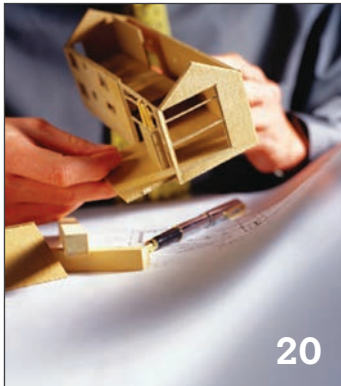
One-click generation of performance scripts based on your standards while facilitating easy reusability and review

For more details on our products and their features, please reach us at [QASuite@cognizant.com](mailto:QASuite@cognizant.com)



Cognizant

## CONTENTS



## in every issue

Mark Your Calendar	4
Editor's Note	5
Contributors	6
From One Expert to Another	13
TechWell Spotlight	14
Product Announcements	34
FAQ	35
Ad Index	36

*Better Software magazine*—The companion to TechWell.com brings you the hands-on, knowledge-building information you need to run smarter projects and deliver better products that win in the marketplace and positively affect the bottom line. Subscribe today at [www.BetterSoftware.com](http://www.BetterSoftware.com) or call 800.450.7854.

## features

### 16 COVER STORY A LOOK BEHIND THE CURTAIN: DEBUNKING 12 MYTHS OF AGILE DEVELOPMENT

When it comes to agile development, Allan Kelly has noticed a lot of misinformation being passed off as fact. In this article, he takes a closer look at twelve of the most common agile myths he has encountered while training new agile teams.  
*by Allan Kelly*

### 20 IMPROVE YOUR TEST ENVIRONMENTS WITH SERVICE VIRTUALIZATION

Want to save money and time, expand your testing environment's capabilities, and still get high-quality releases? Service virtualization provides virtual replicas of physical environments and databases. Its earlier-lifecycle defect detection just might change your company's reality.  
*by Scott Aziz*

### 24 STRENGTHEN YOUR DISCOVERY MUSCLE

An organization shouldn't spend all its time building its delivery muscle without simultaneously building its discovery muscle. In fact, successful software teams deliver great products because they invest in discovery. Learn how to expand your innovation and strengthen your discovery muscle.  
*by Mary Gorman and Ellen Gottesdiener*

### 30 WHY NOSQL MATTERS AND WHAT CONFIGURATION MANAGEMENT EXPERTS NEED TO KNOW ABOUT IT

The NoSQL approach to data storage addresses many problems associated with relational databases and crunches large volumes of data. Eugene Dvorkin offers insight into why NoSQL is important and what configuration management experts need to know about it.  
*by Eugene Dvorkin*

## columns

### 7 TECHNICALLY SPEAKING THE \$440 MILLION SOFTWARE BUG

*by Bob Aiello and Leslie Sachs*  
In August, Knight Capital Group lost \$440 million in one day. But there weren't any traders to blame—at least no human ones. The loss was the result of a software system upgrade gone awry. What can we learn from this and other software catastrophes in the financial sector, and how can we prevent them in the future?

### 32 CAREER DEVELOPMENT CORPORATE UNIVERSITY: GROWING SOFTWARE DEVELOPMENT TALENTS

*by Galyna Datsiv*  
A company with powerful technology and costly equipment is still going to be stuck in a rut without the right personnel. Here, Galyna Datsiv spotlights the concept of corporate universities, where IT organizations can support their products, processes, and even customers by ensuring the growth of new talent and existing employees.

# MARK YOUR CALENDAR



## training weeks

[www.sqetraining.com/trainingweek](http://www.sqetraining.com/trainingweek)

### Testing Training Week

June 10–14, 2013

Chicago, IL

September 16–20, 2013

Washington, DC

October 21–25, 2013

Tampa, FL

November 4–8, 2013

San Francisco, CA

### Requirements Training Week

September 16–20, 2013

Boston, MA

### Agile Software Development Training Week

June 2–4, 2013

Las Vegas, NV

## software tester certification

[www.sqetraining.com/certification](http://www.sqetraining.com/certification)

### Foundation Certification Training

May 7–9, 2013

Seattle, WA

May 14–16, 2013

Baltimore, MD

May 21–23, 2013

Nashville, TN

June 2–4, 2013

Las Vegas, NV

June 10–12, 2013

Chicago, IL

June 18–20, 2013

Tampa, FL

## conferences

### Better Software Conference West

[bsc-west.techwell.com](http://bsc-west.techwell.com)

June 2–7, 2013

Caesars Palace

Las Vegas, NV

### Agile Development Conference West

[adc-west.techwell.com](http://adc-west.techwell.com)

June 2–7, 2013

Caesars Palace

Las Vegas, NV

### STARWEST

[starwest.techwell.com](http://starwest.techwell.com)

September 29–October 4, 2013

Disneyland Hotel

Anaheim, CA

### Better Software Conference East

[bsc-east.techwell.com](http://bsc-east.techwell.com)

November 10–15, 2013

Sheraton Boston Hotel

Boston, MA

### Agile Development Conference East

[adc-east.techwell.com](http://adc-east.techwell.com)

November 10–15, 2013

Sheraton Boston Hotel

Boston, MA

**BETTER  
SOFTWARE**  
A TECHWELL PUBLICATION

Publisher

**Software Quality Engineering Inc.**

President/CEO

**Wayne Middleton**

Vice President of Communications

**Heather Buckman**

Publications Manager

**Heather Shanholtzer**

#### Editorial

*Better Software* Editor

**Joseph McAllister**

Online Editors

**Beth Romanik**

**Jonathan Vanian**

**Noel Wurst**

Production Coordinator

**Donna Handforth**

#### Design

Creative Director

**Catherine J. Clinger**

#### Advertising

Sales Consultants

**Daryll Paiva**

**Kim Trott**

Production Coordinator

**Minda Hatfield**

#### Marketing

Manager

**Phil Parker**



#### CONTACT US

Editors: [editors@bettersoftware.com](mailto:editors@bettersoftware.com)

Subscriber Services:  
[info@bettersoftware.com](mailto:info@bettersoftware.com)

Phone: 904.278.0524, 888.268.8770

Fax: 904.278.4380

Address:

*Better Software* magazine  
Software Quality Engineering, Inc.  
340 Corporate Way, Suite 300  
Orange Park, FL 32073



## HARDER, BETTER, FASTER, STRONGER

When you begin a new workout routine, you ought to have certain goals in mind. Do you want to lose weight? Tone or build your muscles? Increase your endurance or energy level? Maybe you just want to feel a little more confident about how your body burns the calories contained in your favorite dessert.

This month's *Better Software* magazine features many insights into development and change. The authors invite you to review your current processes, improve on the best of them, and discard the worst.

In the cover article, "A Look Behind the Curtain: Debunking 12 Myths of Agile Development," Allan Kelly reviews some not-very-agile practices he's found in companies claiming to be agile. Many of these practices have grown into bad habits that, left unchecked, can damage not only the companies practicing them but also the way others view the agile methodology.

For most people, exercise is something done equally across the various muscle groups—i.e., you wouldn't necessarily want to focus too much on your lower body without also giving some attention to your upper body. But some software companies build their "muscles" disproportionately, choosing to put more emphasis on how they deliver a product than on how they discover or define it. In their article "Strengthen Your Discovery Muscle," Mary Gorman and Ellen Gottesdiener write about the importance of exercising both the delivery muscle and the discovery muscle.

Sometimes, change can come more quickly or have an improved chance of sticking around longer if you use the right equipment. Likewise, in software, the breakthrough you need might be in the form of a new tool or technique. Scott Aziz and Eugene Dvorkin look at two such tools in their respective articles, "Improve Your Test Environments with Service Virtualization" and "Why NoSQL Matters and What Configuration Management Experts Need to Know About It."

If you're going to dedicate the time and effort required for change—whether it's a change to your physique or your work process—make sure that investment isn't wasted on bad habits. Instead, invest in techniques that really help, and strive for an element of balance.

Yours mercurially,

Joey McAllister  
jmcallister@sqa.com



## Contributors



**BOB AIELLO** is a consultant, technical editor for CMCrossroads.com, and coauthor of *Configuration Management Best Practices: Practical Methods that Work in the Real World*. He has more than twenty-five years' experience as a technical manager in several top New York City financial services firms. Bob served as vice chair of the IEEE 828 Standards working group (CM planning) and is a member of the IEEE Software and Systems Engineering Standards Committee management board. Contact Bob at [bob.aiello@ieee.org](mailto:bob.aiello@ieee.org) or [linkedin.com/in/bobaiello](https://www.linkedin.com/in/bobaiello).



**SCOTT AZIZ** currently serves as the quality engineering and testing practice leader for UST Global. His team provides QA consulting and outsourcing services worldwide to Global 2000 customers, including many in the Fortune 50. Scott's expertise is in the application of quality engineering and testing technology to serve the needs of software-enabled businesses through high-quality, fast-performing, secure deliveries that emphasize maximum uptime and minimal end-user disruptions.



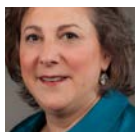
**GALYNA DATSIV** is a corporate university director at SoftServe Inc., a leading global provider of software development, testing, and technology consulting services. With more than ten years of industry experience, Galyna is currently responsible for professional development of IT specialists, ensuring effective training processes and methodologies. Galyna received her PhD in computational mathematics at Ivan Franko Lviv National University, Ukraine. Contact her at [gdatsiv@softserveinc.com](mailto:gdatsiv@softserveinc.com).



**EUGENE DVORKIN** is a software engineer with more than eighteen years of experience in the field. Currently, he is working for WebMD, where he creates real-time event-processing applications using MongoDB and big-data architecture. He is also building a DevOps culture around the project. He spends his free time researching the latest software trends and tools, attending technology meetups, and taking pictures with his beloved Nikon camera. He learns by doing and loves to share his discoveries with other engineers.



**MARY GORMAN** is an expert in business analysis, facilitation, coaching, and training with deep expertise in business systems and product development. She is vice president of quality and delivery at EBG Consulting Inc. Mary works with global clients, speaks at industry conferences, and writes on requirements and planning topics for the agile, business analysis, and project management communities. She is coauthor with Ellen Gottesdiener of the recently released book *Discover to Deliver: Agile Product Planning and Analysis*. Contact Mary at [mary@ebgconsulting.com](mailto:mary@ebgconsulting.com).



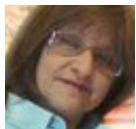
**ELLEN GOTTESDIENER** principal consultant and founder of EBG Consulting Inc., is an internationally recognized facilitator, coach, trainer, and speaker. She is an expert in requirements and business analysis, agile product and project management practices, product envisioning and roadmapping, retrospectives, and collaboration. Ellen's latest book, coauthored with Mary Gorman, is *Discover to Deliver: Agile Product Planning and Analysis*. She also is the author of acclaimed books *Requirements by Collaboration* and *The Software Requirements Memory Jogger*. Ellen works with global clients and speaks at numerous industry conferences. Reach her at [ellen@ebgconsulting.com](mailto:ellen@ebgconsulting.com).



**DAWN HAYNES** is a principal trainer and consultant for PerfTestPlus. As a highly regarded trainer of software testers, she blends almost thirty years of experience with her humorous and practical perspective to provide testers of all levels with tools and techniques to help them generate new approaches to common and complex software testing problems. Dawn also shares her insights at testing conferences and with testing communities worldwide.



**ALLAN KELLY** has held just about every job in the software world, including system admin, tester, developer, architect, product manager, and development manager. Today, he is based in London and works for Software Strategy Ltd. helping companies adopt and deepen agile and lean practices through training, consulting, and coaching. He specializes in working with software product companies, aligning company strategy with products and processes.



**LESLIE SACHS** is a New York State-certified school psychologist and COO of Yellow Spider Inc. She is coauthor of *Configuration Management Best Practices: Practical Methods that Work in the Real World*. Leslie has more than twenty years of experience in psychology and has provided many interventions designed to improve social and educational functioning of individuals and groups. Reach Leslie at [LeslieASachs@gmail.com](mailto:LeslieASachs@gmail.com) or [linkedin.com/in/lesliesachs](https://www.linkedin.com/in/lesliesachs).



**DANIEL WELLMAN** is the director of technology at Cyrus Innovation, a leading New York-based agile software consultancy, where he helps build great teams, projects, and software. He loves the people, interactions, and technical aspects of creating software, with a special interest in building flexible, maintainable systems and practicing test-driven development. Contact Daniel at [daniel@danielwellman.com](mailto:daniel@danielwellman.com), follow him on Twitter at [@wellman](https://twitter.com/wellman), or visit [danielwellman.com](http://danielwellman.com).



# The \$440 Million Software Bug

It's going to take a lot of lucky pennies to make up for losing \$440 million to a system upgrade issue.

by **Bob Aiello** | bob.aiello@ieee.org and **Leslie Sachs** | LeslieASachs@gmail.com

Software and systems glitches have had a major impact on the entire technology industry. The most dramatic incident occurred on August 1, 2012, when Knight Capital Group lost a whopping \$440 million in one day due to a software glitch that was reported to be the result of a trading system upgrade that went awry. In another incident, automated trading firm Infinium was fined \$850,000 for a series of software and systems malfunctions. Even more alarming is that three international trading exchanges—NYSE Euronext, NASDAQ, and the Tokyo Stock Exchange—have experienced software and systems glitches that resulted in penalties, warnings, and sanctions from regulatory authorities. This is particularly shocking because exchanges have long been held to a higher standard, as service disruptions on trading exchanges can have a devastating impact on the world economy.

In addition to these incidents, ten US banks have suffered systems outages due to attacks by hackers, including denial-of-service attacks. The Royal Bank of Scotland, as well as its NatWest and Ulster Bank subsidiaries, faced disruptions for weeks that cost the bank more than £125 million in payments to customers who could not access their accounts or lost money as a result of these outages. In one particularly shocking incident, a man was held in prison after posting bail because Ulster Bank failed to confirm receipt of the bail money that had been paid. Despite receiving compensatory payments, thousands of customers voted with their feet by leaving the bank and taking their money with them.

The world economy depends on these complex financial systems and can be impacted dramatically by service disruptions. These staggering losses clearly show there are severe repercussions for firms that do not ensure that their systems are built and, equally important, supported reliably. In fact, this is exactly why there are federal regulatory requirements within the US and similar international laws that compel financial services firms to establish appropriate IT controls to evaluate and

mitigate risk. Industry standards (e.g., IEEE, ISO, and EIA) and frameworks (e.g., Cobit, ITIL, and CMMI) define the controls that need to be implemented as part of IT governance and compliance.

So, when are these firms going to get serious about compliance and actually implement industry best practices?

Application lifecycle management best practices—including source code management and build, release, and deployment engineering—can enable organizations to reliably support and upgrade software and systems. It is time for corporations to stop viewing compliance as a necessary evil with the primary goal of passing an audit and instead embrace industry best practices as a commitment to truly improving software productivity and quality.

According to published reports, Knight Capital failed to retire obsolete software components that were, in part, responsible for the trading

glitch that ultimately caused the company to cease to exist as an independent corporate entity. There really is no reason for companies to be sidelined by these disastrous system glitches. Even so-called denial-of-service attacks can be addressed through implementation of what is known as the *trusted base*.

The trusted base is a secure platform that is fully verifiable and can be implemented by building the software components using identified baselines and embedding immutable version IDs into each configuration item (CI). Cryptographic hashes can be used to verify that every CI is deployed as intended and also to ensure that there are no unauthorized changes. Deployment frameworks can be used to rapidly build and provision servers. (This is usually the best recourse when a system has been compromised.) The same methods can also be utilized to monitor the environment and ensure that services are fully operational and free of service interruptions.

Recently, there was much industry buzz after the National Institute of Standards and Technology (NIST) determined that its own system had been compromised, with malware discov-

**“Intrusions can happen, but techniques exist that can immediately identify risks and address and remove threats.”**

ered on two of its servers. This was particularly ironic because the servers contained the US National Vulnerability Database, which is used by organizations throughout the world to prevent and address attempts by hackers to attack and penetrate systems. While much was made of the fact that a system containing a security database had been hacked, the fact is that the NIST apparently discovered on its own that there was suspicious activity and was able to identify that its servers had been infected with malware.

This is precisely what the leadership of our banking and financial services firms needs to understand: Intrusions can happen, but techniques exist that can immediately identify risks and address and remove threats. Deployment frameworks and provisioning best practices allow organizations to know exactly what software should be on their machines and to identify unauthorized changes that may indicate that systems have been compromised.

When the banks, trading firms, and exchanges finally learn that application lifecycle management can secure software and systems development, then these corporations can implement practical industry best practices that help improve productivity and quality. Even the most comprehensive implementation of IT controls, including application baselining, separation of controls, and well-defined IT operations, will cost less far than \$440 million and will actually enable organizations to achieve greater success and profitability! **{end}**

### Newsletters for Every Need!

Want the latest and greatest content delivered straight to your inbox every week? Have we got a newsletter (or three) for you!

AgileConnection To Go covers all things agile. CMCrossroads To Go is a weekly look at featured configuration management content. The What's New Gram sends you a weekly listing of all the new testing articles added to StickyMinds.com each week. And, last but not least, TechWell to Go features updates on the curated software development stories that appears each and every day at TechWell.com.

Visit [TechWell.com](http://TechWell.com) or join [StickyMinds.com](http://StickyMinds.com), [AgileConnection.com](http://AgileConnection.com), or [CMCrossroads.com](http://CMCrossroads.com) to sign up for our weekly newsletters.

# CODE LESS TEST MORE



Test Automation for Web & Cloud Apps

- ⊗ Keyword & Data Driven Testing
- ⊗ Cross Browser Support with Chrome, FF, IE
- ⊗ Database Testing
- ⊗ Defect Tracking Integration
- ⊗ WhizGrid™ – Distributed Test Execution
- ⊗ Enterprise Ready



**AGILE**  
DEVELOPMENT  
CONFERENCE  
**WEST**

THE  
COMPLETE  
PICTURE

**BETTER**  
SOFTWARE  
CONFERENCE  
**WEST**

June 2-7, 2013  
Las Vegas, NV  
Caesars Palace

## Two Conferences in One Location

*Register and attend sessions from both events!*

**Early Bird Savings!**

REGISTER BY MAY 17  
WITH CODE **BSAB** AND  
**SAVE UP TO \$200**  
GROUPS OF 3+ SAVE EVEN MORE!

EXPLORE THE FULL PROGRAM AT  
**[adc-bsc-west.techwell.com](http://adc-bsc-west.techwell.com)**

PMI® members can earn  
PDU's at both events

 Project  
Management  
Institute



# Conference Schedule

Build your own conference—multi-day training classes, tutorials, keynotes, conference classes, Summit sessions, and more—packed with information covering the latest technologies, trends, and practices in agile methods and software development.

94 sessions  
offer PMI® PDUs

## SUNDAY

Software Tester Certification—Foundation Level Training (3-days)  
Certified ScrumMaster Training (CSM) + PMI-ACP (3-days)  
Product Owner Certification (2-days)

Agile Tester Certification—ICAgile (2-day)  
Fundamentals of Agile Certification (2-days)  
**Bonus Session:** From Practitioner to Published Author: A Workshop About Writing About Software

## MONDAY–TUESDAY

35 In-depth half- and full-day Tutorials  
Multi-day training classes continue



## WEDNESDAY–THURSDAY

4 Keynotes  
48 Conference Classes  
Networking EXPO  
Special Events  
...and More!



## Who Should Attend?

- Software managers, directors, CTOs, and CIOs
- Project managers and leads
- Measurement and process improvement specialists
- Requirements and business analysts
- Software architects
- Security engineers
- Test and QA managers
- Developers and engineers
- Technical project leaders
- Testers
- Process improvement staff
- Auditors
- Business managers



## The EXPO

June 5–6, 2013

Visit Top Industry Providers Offering the Latest in Software Solutions

### TOOLS • TECHNIQUES • SERVICES • DEMOS • SOLUTIONS

Looking for answers? Take time to explore the Agile Development Conference & Better Software Conference West Expo, designed to bring you the latest solutions in technologies, software, and tools covering all aspects of software development.



# KEYNOTES *by International Experts*



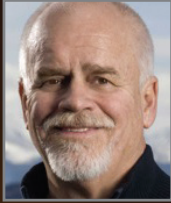
## **The Art of Change: Influence Skills for Leader**

*Dale Emery, DHE*



## **Magnificence: Culture Hacking, the Common Platform, and the Coming Golden Era**

*Jim McCarthy, McCarthy  
Technologies, Inc.*



## **Know the Way, Show the Way, Go the Way: Scaling Agile Development**

*Dean Leffingwell,  
Leffingwell, LLC*



## **Lean Software Delivery: Synchronizing Cadence with Context**

*Mik Kersten, Tasktop*



## **Two Great Conferences**

*One Amazing Destination*

*One registration gets you into all sessions!*

***“From beginner to expert there was something for everyone.”***

*—Rob Frisbie, Software Project Engineer, Gentex*

### **THURSDAY–FRIDAY**

#### **Agile Leadership Summit**

Join your peers and agile industry veterans to explore the unique challenges facing software development leaders as they transform organizations to support agile methods. You'll hear what's working—and not working—for them and have the opportunity to share your experiences and successes

#### **THURSDAY:**

#### **How to Sustain Agile Teams and Organizations**

*Lisa Shoop, Sabre-Holdings*

#### **FRIDAY:**

#### **Design Studios: A Powerful Tool for Culture Change**

*Ruth Frank, Pitney Bowes*

#### **Agile Reality Bites: The Stories of the Struggle**

*Robert Begg, Bluecat Networks*

#### **Integrating Software by Integrating People**

*Todd Little, Landmark Graphics*

#### **Think Tank Discussion**



***Pollyanna  
Pixton***  
*Program Chair*

# Ways to Save on Your Conference Registration



## Your Best Value—The Full Conference Package (5 Full Days), including:

- 2 Days of Pre-Conference Tutorials
- 2 Days of Concurrent Sessions
- 1 Day of the Agile Leadership Summit
- 4 Industry-Leading Keynotes
- The Expo & Bonus Session
- All Networking & Special Events
- All Continental Breakfasts, Lunches, and Refreshment Breaks
- Combine with the other ways to save below for even more value!

## Early Bird Savings!

Remit using code BSAB by May 17 to extend your early bird savings, and save up to \$200 off your registration fees (depending on conference package selected). Call the Client Support Group at 888.268.8770 or 904.278.0524, email them at [sqinfo@sqe.com](mailto:sqinfo@sqe.com), or register now online.

## Training + Conference

Attend any of the training courses + the conference and save \$300 (already reflected in conference pricing).

## The Larger the Group the More You Save!

See the chart below for an example of how much savings groups of 3+ can enjoy on one of our most popular conference packages—Conference + Two Tutorial Days. To take advantage of this offer, please call the Client Support Group at 888.268.8770 or 904.278.0524 or email [sqinfo@sqe.com](mailto:sqinfo@sqe.com), and reference promo code **GRP3**.

Number of Team Members	Regular Pricing	Early Bird Pricing (By 5/17/13)*	Group Savings
1-2	\$2,495	\$2,345	
3-9	\$1,996	\$1,876	<b>20%</b>
10-19	\$1,871	\$1,759	<b>25%</b>
20+	\$1,746	\$1,641	<b>30%</b>

\*Full payment must be received by deadline date

Please Note: We will always provide the highest possible discount and allow you to use the two largest discounts that apply to your registration.

Platinum Sponsor:



Gold Sponsors:



Silver Sponsors:



[adc-bsc-west.techwell.com](http://adc-bsc-west.techwell.com)

REGISTER BY USING CODE BSAB BY MAY 17 AND SAVE UP TO \$200!



## Steve Berczuk

Years in Industry: **24**

Email: [steve@berczuk.com](mailto:steve@berczuk.com)

Interviewed by: **Daniel Wellman**

Email: [daniel@danielwellman.com](mailto:daniel@danielwellman.com)

There's a definition of software configuration management (SCM) from an IEEE standard that says that it involves things like configuration identification, control, review, build management, process management, and teamwork. But that doesn't explain why people might care about SCM. I like to think about SCM as the things people do to make it easy to collaborate on a software project. It includes things like source code management and build management, but also aspects of the software development process like testing that validate that the code does what it is expected to.



It's best to start with figuring out what development approach works best for you and having that drive the SCM approach you choose, but your SCM approach can also encourage a certain work style.

The more frequently you want to release, the fewer code lines you want to be working on, since each integration step between code lines can take time.

There's a balance between checking in code frequently and keeping the code line stable. Frequent check-ins mean more frequent integration, but every time you change something, you risk breaking something. Then again, waiting too long to integrate also has risks. If you are disciplined about testing, you reduce the risks that any change will break the code line.

Communication among team members is the best way to avoid issues. If you know what others are working on and you think that you might step on someone else's work, say something. The code is a form of communication, but not the only one you have.

A monolithic architecture makes testing more difficult, making it harder to be agile. It's difficult to have frequent commits to fewer code lines if every change can have a broad impact on the work of others. A more modular code line means that people can more easily work independently with greater confidence that their changes won't have an unexpected impact on other code.

This is slowly changing, but there still are many teams for whom release management is seen as being over the wall from development, yet the SCM and release-management process affects so much of how developers work. This is analogous to how test and development were considered distinct before people started thinking about agile. While SCM and release management are skills in their own right, they need to be considered part of the responsibility of the development team and integrated into the development process.

**STICKYMINDS.COM™**

For the full interview, visit  
[well.tc/FOETA15-3](http://well.tc/FOETA15-3)

Featuring fresh news and insightful stories about topics that are important to you, TechWell.com is the place to go for what is happening in the software industry today. TechWell's passionate industry professionals curate new stories every weekday to keep you up to date on the latest in development, testing, business analysis, project management, agile, DevOps, and more. Here is a sample of some of the great content you'll find. Visit TechWell.com for the full stories and more!

### How to Choose the Right Communication Tool for an Agile Team

by Steve Berczuk

Email is a thing of the past, according to *The Boston Globe*, with information overload and large amounts of spam-induced message volume as the culprits for its demise. The people interviewed for the story say that in place of email, there is now a preference for using instant messaging, Twitter (following and direct messages), Facebook, text messaging, phone calls, and face-to-face communications. The headline might be overstated, but the article raises some important issues about how we interact with others.

The problem with email isn't the technology itself but how it's used. When you use email as a default form of interaction, you can fall victim to the traps of the law of the instrument, where (to paraphrase) "every interaction looks like an email message." Each of the alternatives to email has advantages for some forms of interactions, such as group conversations. However, each also has limitations.

Continue reading at <http://well.tc/Communication-Tool>.

### Bridging the Public/Private Cloud Chasm

by Beth Cohen

The cloud originated primarily as a medium for consumer services, such as email and online backup. As it grew and gained popularity, cloud bursting was soon touted as a way to deliver highly elastic, more cost effective, shared IT resources to enterprises that wanted to preserve their existing IT systems investments. On paper, the concept is simple.

A company is able to take advantage of public clouds to handle spiky usage patterns and peak workloads without needing to invest in expensive hardware and infrastructure to meet the demand. Some examples include monthly financial reporting, Cyber Monday, and a one-time data conversion project. How can the enterprise leverage its investments in private cloud, yet take advantage of the low cost and elasticity of the public cloud?

Continue reading at <http://well.tc/Cloud-Chasm>.

### The Power of Storytelling for Software Testers

by David Greenlees

The power of storytelling spreads across many different industries, marketing being one of the more prevalent ones. Capturing the attention of your intended audience—and not forgetting about your unintended audience—is critical when trying to get a message across.

Storytelling in software testing is one of the more important aspects of the craft. As testers, test leads, and test managers, you do it all the time. The trick is to make the story a powerful one. The links between software testing and journalism are also strong indicators of this importance. Delivering bad news is what testers do on a daily basis.

Continue reading at <http://well.tc/Storytelling-Testers>.

### Are You a Workaholic?

by Naomi Karten

If you work long hours, take work home with you, and think constantly about work, does that make you a workaholic? Not necessarily. It's easy to confuse working hard with being a workaholic, but workaholism means that you value work over any other activity, even when it negatively affects your health, your family, and the quality of your work.

At the core of workaholism is preoccupation. As described by a psychotherapist, the difference between a workaholic and a hard worker is that the workaholic is on the ski slopes dreaming about being back at work, and the hard worker is in the office dreaming about being on the ski slope! This is the distinction, perhaps, between living to work and working to live.

Continue reading at <http://well.tc/Workaholic>.

### A New Software Development Manifesto for Building the Right Things

by Kent J. McDonald

One thing that distinguishes agile software development approaches from other approaches is their foundation on a set of values and principles. The foundation was originally created in February 2001 with the Manifesto for Agile Software Development, followed in 2005 by the Declaration of Interdependence and in 2009 by the Manifesto for Software Craftsmanship.

Each of these statements of values and principles was created by a group of people who were tackling the same type of problem and thought it may be interesting to see if there were any similarities between their outlooks. The seemingly inevitable outcome of each of these discussions was the conclusion that the development community was missing something and new principles were needed.

Continue reading at <http://well.tc/Right-Things>.

### Is New Technology Taking Us Back to Grassroots Software Testing?

by Rajini Padmanaban

Advancements in technology and what they continue to bring



on a day to day basis are mind boggling. In the last decade, there has been unprecedented growth in almost every domain thanks to key technology advancements in the Internet, mobile and cloud computing, and social networking. This growth touches all walks of life globally—and the current decade will be no exception.

It is amazing to see high-end gesture-based gaming consoles such as Kinect—and it is equally exciting to see the mobile devices penetration in developing countries such as India, where just a few years ago even a wired phone wasn't something that everyone had.

It is very exciting to read news items on some of the upcoming products and features—touch screen walls in hotels, potential growth avenues in the mobile space, hi-tech glasses, and driverless cars. The list is endless.

Continue reading at <http://well.tc/Grassroots-Testing>.

## Is the FDA Ready to Regulate Mobile Medical Apps?

By Pamela Rentz

With the growing adoption of mobile technologies for health care delivery, the US Food and Drug Administration (FDA) has proposed regulating mobile medical applications and, in July 2011, issued preliminary draft guidance.

Fast forward to 2013—final guidance has not yet been delivered. The delay has left many in the mobile health arena to speculate on exactly what will be regulated and how encompassing those regulations will be when they finally arrive.

Continue reading at <http://well.tc/Mobile-Medical>.

## ISP Injects Ads into Paying Subscribers' Web Content

by Beth Romanik

Customers of a small cable operator in the southern US recently noticed banner ads appearing at the bottom of popular websites such as Bing, Target, eBay, and Amazon, which normally don't place ads there. After some detective work, two clients concluded that the Internet service provider was injecting the ads into the sites through JavaScript from an outside company.

Robert Silvie and Zack Henkel noticed the suspicious ads while each was using CMA Communications, which provides Internet, TV and phone services to rural areas of Mississippi, Louisiana, Texas, and Nevada. In addition to the ads appearing on sites that typically had none, they also sometimes replaced existing ads on webpages.

Continue reading at <http://well.tc/ISP-Injects-Ads>.

## The Outside-In Approach to Product Positioning

by Scott Sehlhorst

In his most recent article at *On Product Management*, Steve Johnson puts positioning in perspective and brings a ton of clarity to the top-down, outside-in view of a product. Steve shows how, by using a simple formula or template, you can describe your product very concisely—from the outside-in.

This early-on positioning exercise helps drive focus and clarity for the team and allows a stakeholder to approach funding decisions from a strategic perspective.

In a nutshell, the top-down approach to product strategy is to identify how each product's independent strategy contributes to the overall company strategy and if the combined product strategies will fulfill the overall company strategy. This is a perfect analogy to analyzing requirements correctness and completeness. Are the product strategies aligned? And have we covered everything?

Continue reading at <http://well.tc/Outside-In>.



TECHWELL™

## So, You Want to Be a TechWell Curator?

### What Is a TechWell Curator?

TechWell curators are software professionals who are knowledgeable, enthusiastic, and engaged in the latest industry trends, tools, and technology. Using content sourced from around the Internet, our curators compose short stories that are interesting, entertaining, sometimes thought provoking, and occasionally opinionated.

### What Do I Have to Do?

Each curator is responsible for submitting a minimum of five to ten stories a month. Stories should run 300-500 words, with 400 words being ideal. Stories are built around and should link to articles, videos, blog posts, or other online content—both from our TechWell Community sites and anywhere in the Internet—that the curator considers interesting and applicable to our audience. You should expect to spend one to two hours developing and writing a story. Because audience engagement is key to the success of a curated site, we ask curators to respond to reader comments and questions.

### What's in It for Me?

Stories you write will feature your byline with a link to a profile page containing your photo, bio, and links to your blog, Twitter, LinkedIn, etc. Readers will come to know you, your stories, and your personality. Thought leaders are born this way. TechWell curators receive \$100 per story published, up to a total of ten stories (\$1,000) per month. In addition, active TechWell curators receive free Wednesday–Thursday conference passes to any SQE conference and half price on pre- and post-conference event sessions (tutorials and summit).

### What Is the Publishing Process?

Curators submit stories to the TechWell editors, who check them for grammar, style, and punctuation, and then publish them to the site—usually within two business days.

### What If I Can't Write for a While or Want to Stop Curating?

We understand life can get hectic. So, if you need to take a temporary break from curating, we ask that you give us two weeks' notice. In the event you decide curating is not for you, please let us know thirty days in advance so we can look for a replacement.

### How Do I Get Started?

To apply for a TechWell curator position, please contact Noel Wurst at [nwurst@sqe.com](mailto:nwurst@sqe.com) with the following information:

Name  
Company affiliation  
Interest area(s)  
Approximate stories per month you are available to curate

340 Corporate Way | Suite 300 | Orange Park, FL  
32073 | 904.278.0524 | [www.TechWell.com](http://www.TechWell.com)

The background of the entire page is a stage with a wooden floor. Three spotlights are shining from above, creating a bright area in the center. The text is centered on this stage.

*A Look*  
**BEHIND**  
**THE**  
**CURTAIN**

*Debunking 12 Myths of  
Agile Development*

*by Allan Kelly*



I deliver a lot of agile training courses, and I give a lot of talks about agile. It sometimes surprises me that there are still people out there who haven't heard of agile or who only vaguely understand what it is. I guess that just demonstrates the size of the software development world.

Some of the questions that come up again and again are the result of myths people have come to believe about agile. Consequently, I spend a lot of my time debunking these myths.

I've made a list of a dozen of these recurring myths. I say "myths" because they all seem to come from either a partial understanding or a complete misunderstanding of agile. Of course, if you've worked in agile, knocked around the agile community for a while, or been to an agile conference or two, you might be surprised that some of these problems exist. Yet, these are the misunderstandings I hear frequently.

### 1. AGILE IS NEW.

The Agile Manifesto was published in 2001. The Scrum pattern language was presented in 1995 during the Object-Oriented Programming, Systems, and Languages (OOPSLA) conference. The Episodes pattern language (the forerunner of Extreme Programming [XP]) was described in *Pattern Languages of Program Design*. [1] Tom Gilb's Evo method [2] dates back to 1976, and there are some who trace agile's roots back further still.

In fact, a close look at the report from the 1968 NATO conference on software engineering [3] reveals some familiar ideas. Professor Alan Perlis closed one debate saying:

Through successive repetitions of this process of interlaced testing and design, the model ultimately becomes the software system itself. I think that it is the key of the approach that has been suggested, that there is no such question as testing things after the fact.

That sounds like a pretty good description of much of agile to me—thirty-three years before the Agile Manifesto.

### 2. AGILE MEANS HAVING NO DOCUMENTATION.

While some people believe that being agile means you don't need any documentation, that's hardly the truth. You can have as much documentation as you like in agile. Documentation is just another deliverable. If it brings you value, then schedule it and produce it like anything else.

This myth may have started with Kent Beck, the originator of Extreme Programming, who has questioned the need for documentation in rather extreme language. (See Nico Josuttis's interview with Beck for an example. [4]) Over time, this myth seems to have grown to encompass all documentation in all cases.

However, there is both unnecessary documentation and valuable documentation. The trick is to decide which is which.

Still, let's not hide it: Agile probably does mean less documentation. In his 2008 book, *Applied Software Measurement*, Capers Jones writes, "For large software projects, the cost of producing paper documents is more expensive than the code itself." [5]

Unfortunately, sometimes documentation is not just a cost but also a block to communication, as people might hide behind documents when a conversation or an example might be

much more informative.

If someone is willing to pay for a document, then it has value and should be scheduled and developed the same way that code is.

### 3. AGILE MEANS NO DESIGN.

To practice agile, one probably needs *more* design. Design is inherent all the way through development, at every planning meeting and more. Regarding this myth, however, agile does mean that you don't need a big, up-front design that will be invalidated five minutes after someone starts coding.

Big, up-front design has little place in agile. All too often, these designs delay the start of coding and fail to deliver the promised benefits. Agile should instead embrace an emergent design principle that appears most clearly in the refactoring process, where developers improve the design of working code. Design is omnipresent in agile, and it's not just relegated to the beginning of the project.

### 4. AGILE MEANS NO PLANNING.

Agile probably requires more planning on the part of a team. Again, planning is spread throughout the whole development exercise rather than at the front, and it involves the work of everybody rather than one or two anointed individuals.

As with design—which is, itself, a form of planning—agile teams don't need big, up-front planning. In my experience, such plans are hostages to fortune as they quickly become dated and require updating. But customers, sponsors, and other stakeholders hold plans against teams. For example, a plan might require that X needs to be delivered on a particular date. Stakeholders, governance committees, and project management officers may see any deviation from this plan as a failure on the part the team, not a failure of the plan itself.

Nor does lacking a big, up-front plan mean that you don't need long-term planning. Most agile teams get proficient at doing just enough design during the two-week sprint- or iteration-planning stages. The development team produces these collective plans in cooperation with other stakeholders.

In my experience, many teams struggle to plan for the future. I like to see a rough sketch from the product owner, business analyst, or product manager showing what he hopes to ask the team for during the next quarter—typically six iterations—but no further. Often called "release plans," these plans are probably better termed as "quarter plans" or "product plans." Such plans can utilize rough estimates and velocity calculations but do not represent firm commitments.

### 5. THERE IS A RIGHT SIZE FOR A USER STORY.

Every team is different. Get over it.

I have two rules for a user story or any other kind of development story: First, it should be small enough to be delivered sometime soon—i.e., it will not take months and months. Second, it has to have business value in its own right—i.e., someone somewhere can say why it is valuable and, ideally, can write a dollar amount on it.

These two rules often serve to drive the story size in different directions. This only highlights the important role that the requirements people have in finding truly valuable work.

The size of a story is often a function of the experience and knowledge of a development team. When team members lack domain knowledge, they request more and more detail. While this may increase the number of words on a story card, it often serves to decrease the functionality of the item under development.

Conversely, team members with experience and knowledge in the domain don't tend to need too many words of explanation—on the card, in a conversation, or as acceptance criteria—and can deliver large chunks of work.

## 6. WORK MUST FIT IN A SPRINT.

If you are doing hardcore Scrum, then yes, each story should start and end inside the same sprint. But not all teams practice hardcore Scrum. Many teams allow stories to overlap or break the stories down in some way. Personally, I advise teams to break business-focused stories down to development tasks.

In the way I play agile, which I call “Xanpan” (a fusion of XP and kanban), I encourage sprint-spanning stories in order to improve flow. I allow for stories that span sprints, but I won't let them continue forever (three sprints and you're out!). Additionally, my teams try to break down sprints into smaller pieces of work.

All that said, small stories are far better than large stories, but you need to get good at writing small stories that fit within two-week sprints. The complete-within-the-sprint rule can be an effective way of forcing teams to get away from stories that are too large. While this rule is fine to aspire to, it is something that requires practice and learning on your part. It isn't something you can do on your first day of agile or even after a few days in an agile course.

## 7. DEVELOPERS GET TO DO WHAT THEY LIKE.

If this myth is true for you, then you are doing it wrong. Agile needs well-disciplined teams. What gets done should be led from a specific role, usually the customer or product owner. If developers are doing what they like, then there is a good chance that something is wrong with that role, the person playing it, or the authority vested in it.

It seems to me that people on the business side sometimes believe this myth because they fail to grasp the nature of agile, which is a collaborative process that involves discussion and negotiation between those who are building the system and those who requested it.

## 8. SCRUM AND KANBAN ARE SWORN ENEMIES.

Some individuals acquire a lot of eyeballs by saying that Scrum and kanban are sworn enemies. A cynic might even say that the marketing efforts behind both methods view the image of competition as a useful publicity tool.

In my experience, many teams are combining ideas from the iterative methods (Scrum and XP) with ideas from kanban. Corey Ladas described one combination, ScrumBan, four years ago, and I continue to evolve my own Xanpan method.

## 9. AGILE DOESN'T WORK FOR FIXED-DEADLINE PROJECTS.

Agile works best in fixed-deadline project environments. When team members use agile, particularly the Scrum and XP

versions, they receive much of their power from the harnessing deadline effects: sizing work to the deadline, individual motivation, and willingness to renegotiate just what is being built to meet a date.

A corollary to this is that agile methods are also very good for fixed-price work, because costs are overwhelmingly wages multiplied by time.

## 10. AGILE DOESN'T WORK ON BROWNFIELD PROJECTS.

Agile works best in environments where something already exists—so-called brownfield work. On a greenfield effort (one with no existing codebase), first base is a walking skeleton, a barebones system that works end to end. Brownfield projects start with a working system in place.

Granted, it's harder to retrofit automated tests to an existing system than to create tests for a new system, but it is far from insurmountable.

## 11. AGILE DOESN'T WORK ON GREENFIELD PROJECTS.

While it's a myth that agile doesn't work on greenfield projects, remember that your first objective is to get yourself to a steady state where you can think like you are operating on a brownfield project. Believe it or not, for every two people I meet who say, "I can see how agile will work for greenfield projects, but we have an existing system," I meet someone who holds the reverse point of view.

Automated testing in a greenfield environment is easier, but the fact that no codebase or working product currently exists presents its own challenge. There is often an all-or-nothing mentality among developers and managers that needs to be overcome. Additionally, many developers are concerned with the need to design and architect. It can be hard to accept evolving, emergent design until one has experienced it.

## 12. NOW IS NOT THE RIGHT TIME.

I regularly hear excuses like "We like the sound of agile, but we are in the middle of ..."; "Agile will not work for us because ..."; and "Agile is a good idea, but ..."

You can always talk yourself out of agile or find a good reason for not doing it today. I believe the time to change is as soon as you decide that it genuinely is better to do so. In my mind, the ideal project with which to start an agile initiative is a brownfield system with a fixed deadline of about three to six months, when development has started but requirements are still unclear.

These are the twelve myths I keep encountering. Maybe you've seen some of them, or maybe you've seen some of your own. Now that I've debunked them publicly, hopefully we will not have to spend quite so much time debunking them in private. **{end}**

allan@allankelly.net



For more on the following, go to [StickyMinds.com/bettersoftware](http://StickyMinds.com/bettersoftware).

■ References



# IT'S YOUR TIME TO SHINE



Distinguish yourself from your peers and gain a competitive edge

Attend class  
in-person or  
virtually from  
office/home

## ALPI'S TRAINING OFFERS:

### Technology and Methodology Courses

- ✓ **HP:** Quality Center ALM, QuickTest Pro, and LoadRunner
- ✓ **Microsoft:** Test Manager, Coded UI, and Load Test
- ✓ **Process Improvement:** ISTQB Certification, Managing Change, Requirements Definition, Enterprise Quality

### Interactive Learning Method™

Bring your Workplace to the Classroom & Take the Classroom to your Workplace™

### Post Training Support

Free refresher courses  
Process and tool  
implementation services

### Bulk Training Program

Savings of up to 40% with credits  
good for a year



Since 1993, ALPI has empowered clients with innovative solutions delivered by our staff of flexible and creative professionals. Trainings are held virtually, at our state-of-the-art facility located just outside of the Nation's Capital, or onsite at your company location.

Contact [training@alpi.com](mailto:training@alpi.com) or 301.654.9200 ext. 403  
for additional information and registration details

**WWW.ALPI.COM**



## IMPROVE YOUR TEST ENVIRONMENTS WITH SERVICE VIRTUALIZATION

BY SCOTT AZIZ



Service virtualization is a proven technology that saves money and greatly expands the capabilities of enterprise software testing environments. It enables testing organizations to have higher-quality releases in a fraction of the time. With its accurate simulation of upstream and downstream environments, service virtualization enables earlier lifecycle testing for manual, automated, mobile, performance, and many other categories of tests that need to be executed.

Today's interconnected enterprise environments typically comprise complex SOA- and service-based architectures as well as private and public cloud computing components with third-party or partner systems interconnectivity. These environments are expansive, complex, and expensive. The need for a robust testing environment across these architectures is mandatory for testing accuracy, but often a complete test environment is just not available for the testing team.

This limitation introduces a significant risk across an otherwise comprehensively designed and well-intentioned testing strategy. As system complexity increases, the likelihood of a realistic test environment's being available decreases due to factors that include cost and complexity. The testing risks from this limitation include unavailable or inaccessible services or infrastructure to execute critical planned tests against, and missing components (data or services) that limit the amount of testing that can be performed. This often forces a testing team into a situation in which comprehensive test coverage must be reduced based on the limited availability of accurate test environment components.

In recent years, hardware and OS virtualization have provided relief for some of these test environment limitations in the form of stopgap measures. However, given the complexity of today's environments, traditional virtualization is not enough. For example, traditional virtualization technology does little to simulate a large mainframe or ERP system due to the fact that the computing environments of both of those systems are much larger in scale than traditional hardware virtualization technology can offer.

## What Is Service Virtualization?

Service virtualization is a newer test environment technology that can solve the limitations of traditional hardware and OS virtualization in testing environments. It allows for the creation of virtualized physical environments and databases in a test environment that accurately reflects the behavior of a target production environment. The types of environments that can be virtualized are numerous and include mainframe, ERP, web service, ESB, and many other platforms and systems. This diversity of environment virtualization capability allows testing teams to circumvent the limited availability of physical testing environments.

Service virtualization works by capturing or recording transactional traffic between two systems in a production environment and using that transactional information to create a virtual environment with virtual assets that simulates the behavior of a real, physical environment. Service virtualization is different from traditional "stub" technology, which has been

around for many years. Stubs do an adequate job of listening and responding to requests for transactions and data. However, they are generally stateless and unintelligent because in most cases they return a single, canned response.

## Tools of the Trade

Service virtualization tools provide a state-aware and intelligent environment because context-sensitive data can be returned. This is possible because as dynamic variables—session IDs, customer information, date information, and other valuable data—are passed to these tools, the tools return an intelligent and informed response. The service virtualization tools' ability to parse this data and apply predetermined logic to form a response that is exactly like the response from the actual physical environment is a game changer. This allows for the creation of a much more realistic test environment because state-aware transactions accurately simulate environments that have complex business rules and extensive data. The greatest benefit for the tester is that service virtualization tools offer access to realistic virtual environments for increased test coverage earlier in the test cycle compared to a traditional test environment that must be configured and available with all downstream, dependent systems.

Several different service virtualization tools are available in the market today from many of the leading vendors that produce development and testing lifecycle tools. The tools work in a similar manner and generally have robust automated capabilities that allow for the automated capture of services and transactions under test. They have recording capabilities that enable the capture of other assets as well, such as transactional system information that may be available through log files. The tools' automated capability to capture transactions and services and build intelligent virtual assets offers a great advantage over earlier stub technology that offered limited virtualization capabilities. Where stub maintenance was largely a manually intensive process that required constant effort, the latest service virtualization tools offer automated capture mechanisms that greatly decrease the maintenance effort.

This automated transaction-capturing mechanism works in a manner similar to software-based network sniffers. All you need to do is provide the tool with an accurate stream of transactional data across a properly configured environment and let the tool record the needed traffic stream that you are looking to capture for virtual simulation. This usually can be accomplished in a matter of minutes for an active message stream. You may need to generate some of the messages yourself in cases where you are trying to record transactions in a non-production environment that may not have much development or testing activity. You can do this manually or in conjunction with other test engineers by running functional testing scenarios. You can also run an automated functional-testing script against the environment that you are looking to virtualize. You just have to be sure that you have a large and varied transaction mix that accurately reflects expected production traffic. The tools provide the ability to manually create or edit transactions that are not covered by the test scenarios that you've



executed to generate the transactions. For example, there might be special conditions that you wouldn't cover as part of a normal regression test and therefore have not included in the functional testing scenarios that you executed.

Once you have captured and recorded the appropriate number of transactions, you must establish an ongoing maintenance activity to ensure that when applications change, the changed or newly created transactions are updated in the service virtualization tool. This allows for the proper maintenance of the heterogeneous messaging protocols and formats across the request and response stream of the applications under test.

This maintenance is similar to the maintenance needed for automated functional test scripts. As the underlying application changes, the test scripts need to be changed to accurately exercise the application under test. Generally, the ongoing maintenance needed for service virtualization tools will only be a small percentage of the initial effort, but that will depend on how much the applications that you are virtualizing in the environment are changing.

The tools offer features that define how you would like your virtual assets to behave with the application under test. Quality-of-service configurations are available that allow you to define some state-aware behavior attributes, such as performance- or timing-specific response rules. This allows you to simulate the latency of various transactions of the actual production system, which provides a more realistic testing environment for the application under test. To simulate failure conditions, you can also configure data rules and set up error conditions that will replicate system errors during testing. All of these features provide a more realistic virtual test environment compared than first-generation stub technology.

Service virtualization tools automatically generate an accurate test environment simulation model by capturing, customizing, and maintaining transactional messages and data. The tools allow for the isolation of the specific behavior needed to simulate an accurate testing environment. This functionality enables an accurate simulation in a short amount of time, with a reasonable amount of effort, and allows a testing organization to have needed dependent system requirements met in a virtual environment very early in the development lifecycle. This leads to more comprehensive testing earlier in the lifecycle and that will, in turn, lead to detecting and closing defects earlier.

Service virtualization tools automatically generate an accurate test environment simulation model by capturing, customizing, and maintaining transactional messages and data. The tools allow for the isolation of the specific behavior needed to simulate an accurate testing environment. This functionality enables an accurate simulation in a short amount of time, with a reasonable amount of effort, and allows a testing organization to have needed dependent system requirements met in a virtual environment very early in the development lifecycle. This leads to more comprehensive testing earlier in the lifecycle and that will, in turn, lead to detecting and closing defects earlier.

## Improvement Comes at a (Lower) Cost

High costs often deter the procurement of enterprise test environments and dependent, interconnected, downstream systems. Service virtualization is a low-cost alternative that can provide an easily calculable return on investment. Even when dependent and interconnected test environments do exist, the maintenance involved in configuring those environments and the effort necessary to keep up with test data management activities in those environments often are resource intensive and very costly. In most cases, the return on

*continued on page 36*

# GET A TESTING RESULT YOUR CEO WILL LOVE.

Your CEO will love you when you show them how you can achieve an outstanding testing ROI with ISTQB Software Tester Certification.

With the average cost of a software defect in the range of \$4,000 – \$5,000<sup>11</sup>, if ISTQB Certification helps your tester eliminate even just one defect, the result is nothing less than, well, loveable: an ROI of up to 2000%.

ISTQB Software Tester Certification is the most widely recognized and fastest-growing software tester certification in both the U.S. and the world. Discover how ISTQB certification can pay for itself in a matter of days: That's a testing result any CEO will love.

## Want an even better ROI?

Take advantage of our new **Volume Purchase Program.**

**ASTQB**  
American Software Testing Qualifications Board, Inc.  
*ISTQB Certification in the U.S.*

<sup>11</sup> Capers Jones, "A Short History Of The Cost Per Defect Metric", Randall Rice, "The Value of ISTQB Certification"



Learn more  
now at

[www.astqb.org](http://www.astqb.org)



Be a  
**TESTING**  
**ROCK STAR**

SEPTEMBER 29 –  
OCTOBER 4, 2013

Anaheim, CA  
Disneyland Hotel

**STAR  
WEST**

036428  
*Register by*  
**AUGUST 2, 2013 AND**  
**SAVE UP TO \$400**  
GROUPS OF 3+ SAVE EVEN MORE!  
036428



[STARWEST.TECHWELL.COM](http://STARWEST.TECHWELL.COM)



**Strengthen  
Your  
Discovery**

**Muscle**



THINKSTOCKPHOTOS.COM

**by Mary Gorman and  
Ellen Gottesdiener**



In a recent interview in the *New York Times*, Panera Bread co-CEO Ronald M. Shaich talks about the importance of developing an organization’s “discovery muscle” as well as its “delivery muscle.” [1] Most companies have worked hard to perfect delivery—how they get work done—he says, because delivery “feels rational, people feel much safer with it, and you can analyze it.” But discovery—the activities you undertake to define or change your product, service, or market—is about “leaps of faith. It’s about trusting yourself. It’s about innovation.” The key, Shaich says, is for the discovery muscle to be at least as strong as the delivery muscle.

He took the words right out of our mouths. This need for balance between discovery and delivery applies in spades to software development. Our new book, *Discover to Deliver: Agile Product Planning and Analysis* [2], explicitly makes the case for equally balancing your commitment to these key activities. We define the relationship between them: In lean/agile software development, discovery and delivery are interwoven, interdependent, continuous activities (see figure 1). Each feeds the other.

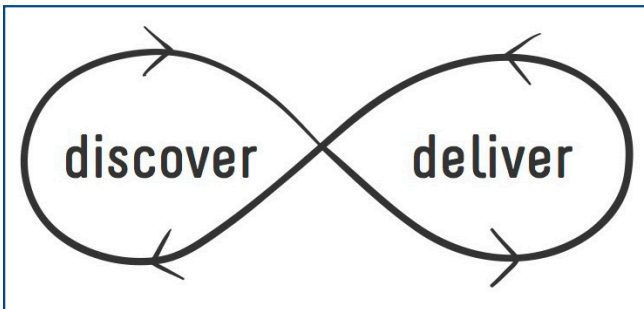


Figure 1: Discovery and delivery are a continual process.

Traditionally, software teams have been out of balance—strong on delivery but weak on discovery. As a result, they tend to deliver technically excellent software that, unfortunately, may solve the wrong problem, possibly lags the market, or otherwise falls short of meeting the stakeholders’ real needs.

If your team needs to develop its discovery muscle, it’s not really about making a leap of faith. It’s more about making a leap of learning. Let’s look at why and how.

## Context Counts

Many teams set themselves up for failure by not including discovery in their process from the very beginning. Before you begin talking about product options (aka requirements), the stakeholders need to hammer out a product vision and goals.

One of your most useful discovery tools is constructing and asking good questions to set the context and determine your mark. What is your company strategy, and how does your vision for the proposed product align with it? Should the strategy be revised in light of recent developments? What is your problem or opportunity? What is the competition doing or not doing? What is your competitive advantage? Are you being too safe? Might there be customers beyond your current market? Who cares?

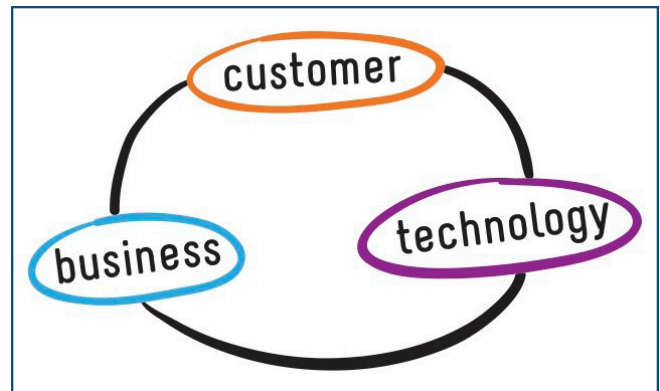


Figure 2: The product partners need to represent diverse viewpoints.

The vision activity should include people who represent three realms: the business, the customer, and the technology group—all of whom we call the product partners (see figure 2). They need to collaboratively contribute to the vision.

The lean/agile practices we use position value as the chief criterion for planning which product options to develop next. Of course, you’re not working at the level of product options yet, but now—while you’re developing the product vision—is the time for the partners to define their value considerations. A few examples:

- **Customer Value Consideration:** Save time, money, and frustration.
- **Business Value Consideration:** Support our growth strategy.
- **Technology Value Consideration:** Ensure a scalable technology platform.

## Working Together

Discovery is a team sport, not a spectator sport. Your discovery team must include diverse voices and perspectives. The partners collaboratively expand their thinking, look at the problem with fresh eyes, and consider new or unique possibilities. They reach far and wide. They listen to differing perspectives and reach agreement. You gain two benefits: you exploit the wisdom (attributed to Aristotle) that the whole is greater than the sum of its parts, and the product partners collectively own the discoveries.

As you collaborate in discovery, it’s important not to bow to conventional wisdom. It’s easy for your discovery muscle to shorten and tighten if you don’t stretch it. Instead, critique the way you’ve always done things. Be courageous, creative, and curious. Find out why. Moreover, as Shaich notes, defying conventional wisdom may help you discover something that gives you a competitive advantage.

## Holistic Discovery Practices

Discovery engages the partners in learning and possibilities—it demands problem-seeking and solution-finding. To that end, your discovery process should address the product needs for each of what we call the “7 Product Dimensions” (see figure 3).


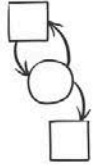

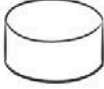



						
User	Interface	Action	Data	Control	Environment	Quality Attribute

Figure 3: Discover all 7 Product Dimensions.

Discovering all 7 Product Dimensions gives you a holistic understanding of the product’s functional and nonfunctional needs. You explore options within and across each dimension. This cross-dimension perspective helps inform and balance your discovery.

During a recent discovery workshop at a health care provider, the partners quickly identified “obvious” users such as health care members and health care providers (e.g., physicians). Digging deeper, they discovered the health care medical director, who has unique product needs. Then they considered the users in light of the other dimensions and discovered new interfaces and environments, along with related actions, data, control (business rules), and quality attribute options. This discovery work provided a number of benefits. The partners’ shared understanding of the broad range of product possibilities helped shape the overall architecture, saving future rework. It guided their research on usability needs for complex data-visualization interfaces. And they reduced risk by collectively selecting the high-value options for their first release.

In conjunction with the 7 Product Dimensions, you use the “structured conversation” to frame your discovery sessions into three activities: explore, evaluate, and confirm (see figure 4).

The structured conversation is a lightweight framework

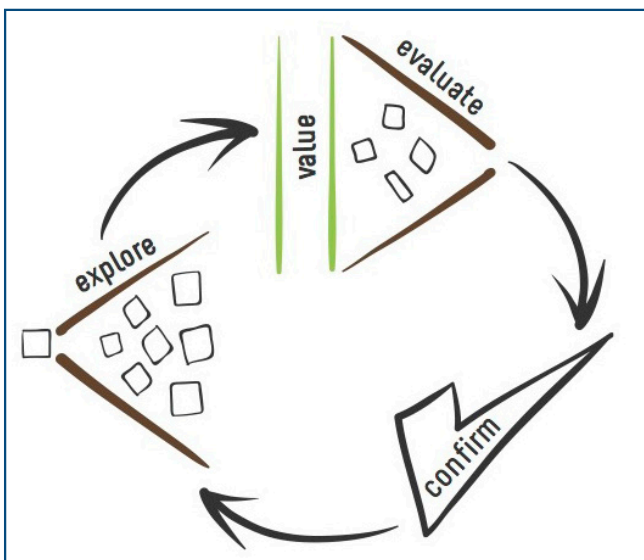


Figure 4: The structured conversation guides your discovery work.

that guides the product partners as they learn about the product’s possibilities and decide what to deliver. You *explore* options for all 7 Product Dimensions. As you *evaluate* the benefits and risks of each product option, you use the partners’ value considerations to identify high-value options. You also *confirm* the partners’ understanding of the options.

And importantly, at each new planning session you’re open to exploring, evaluating, and confirming new product options, using your learning from prior deliveries. Thus, the structured conversation helps you balance your discovery and delivery muscles by moderating between the expansive thinking of discovery and the more focused thinking of confirmation and delivery.

## Engage Creatively

Discovery is the rich interplay of creativity, new or expansive thinking, and human-centered design. You stretch your discovery muscle, making way for purposeful serendipity.

This aspect of discovery is often referred to as “design thinking,” a newer term for diverge-then-converge practices that inspire, boost, and challenge the partners. Design thinking draws from an amalgamation of disciplines such as visual design, professional facilitation, architecture, industrial engineering, contextual inquiry, participatory design, improvisation, learning games and simulations, and the burgeoning innovation and creativity movement.

Discovery goes beyond writing stories. Design matters. A lot. Your product’s look, feel, color, and aesthetic appeal—its visceral likeability—matter. Space and tools matter, as well. You “uncube” physical space so that people can conveniently and extemporaneously interact as they converse, sketch on posters, walls, or whiteboards, and select from and play with colored posts, markers, and glue to find possibilities and explode problems.

To get there, you can draw from the family of facilitated workshop activities—variously called collaboration activities, serious games, Innovation Games, and more—to spark possibilities and yield serious outcomes. You can also use sketching, brain writing, affinity mapping, or card sorting activities. Techniques such as personas, role play, contextual inquiry, empathy maps, and journey maps help you gain affinity with your users, and you can employ analysis models such as prototypes, state




VIEW	TOOLS & TECHNIQUES	
	TEXTUAL	VISUAL
 <p><b>BIG-VIEW</b></p>	Business policies, data domains, features, events, environment, architecture, personas, processes, scenarios, terms, high-level quality attributes.	Architectural drawings, business process diagrams, capability maps, conceptual data models, empathy maps, relationship maps, state diagrams, value stream maps.
 <p><b>PRE-VIEW</b></p>	Business policies, data entities and relationships, events, examples, features, glossaries, interfaces, locations, minimum marketable features, personas, scenarios, stories, use cases, and user roles.	Business process diagrams, context diagrams, data models, dependency graphs, prototypes, state diagrams, story maps.
 <p><b>NOW-VIEW</b></p>	Acceptance criteria, atomic business rules, data entities, data relationships and attributes, detailed interfaces, environment specifications, examples, personas, scenarios, stories, and measurable quality attributes.	Decision tables, decision trees, detailed prototypes, logical data models.

Figure 5: You can use various discovery tools and techniques according to the planning view.

diagrams, data models, and dependency graphs to tap into visual thinking.

## Calibrate Your Discovery

You need to adjust your discovery scope depending on your planning horizon—what we’ve termed the *Big-View*, *Pre-View*, and *Now-View* (see figure 5). You might not necessarily start from the top and work down, as long as you have a clear definition of your discovery scope.

In the Big-View (the longest planning horizon, up to two years), your discovery muscle needs to be loose, flexible, and far reaching. At this planning horizon, you discover high-level, generalized possibilities for the product, across all 7 Product Dimensions. You might want to start discovery with a vision statement for the product, or your discovery might lead you to craft the vision.

In the Pre-View (the middle distance, or release view, possibly a few weeks to a month), your discovery muscle is toned and controlled. Your discovery is framed within a clearly defined space to explore, evaluate, and select high-value product needs to enable planning for the next release.

In the Now-View (the shortest view, typically the next iteration: days or weeks), your discovery muscle is taut, ready to spring or sprint. You are focused on a narrowly defined space. You need to discover and define high-value product require-

ments in sufficient detail to enable immediate development and potential delivery.

## Training Your Discovery Muscle

To get the most from discovery, be prepared to stretch, reach, bend, and twist the way you think about your product. At first, you may find training your discovery muscle uncomfortable, even unpleasant. Or the partners may not be willing to discover together. Yet, regularly exercising your discovery muscle improves flexibility and range—you will see your business through different eyes. You will develop skills in discovery and innovation, building muscle memory so that discovery becomes natural, seeming to progress without conscious effort.

**Warning:** As with any routine, your initial enthusiasm for discovery may dim with time and repetition, and your discoveries may be less powerful. If that happens, look for new ways to spark creativity. Vary your techniques, inject new ones, and play innovation games. Switch roles, and take on a different perspective. Get out of your comfort zone.

## Reaching a Shared Understanding

The 7 Product Dimensions and the structured conversation are essential tools that enable the people in the three partner realms to agree on which discovered product needs are high value. The 7 Product Dimensions construct, for example, em-



“Your discovery muscle needs to be fit and ready to move, explore, and innovate to exploit opportunities.”

ploy a visual language that all the stakeholders can use in talking about the product. And because of the structure in the structured conversation, the partners become intimately acquainted with the incremental nature of the lean/agile process.

So, what’s the biggest mistake we’ve seen that obliterates the benefits of these powerful tools? Failing to include the right people from beginning to end. The resulting siloed conversations cause people to develop different—often conflicting—definitions of the product to be. Discovery becomes haphazard and undisciplined, scope changes often, and people retreat into the apparently safer, saner world of delivery. As a result, opportunities to make a great product fizzle out.

### It's a Process, Not an Event

At any given time—this year, this release, this iteration—you may discover new product possibilities. That’s the nature of discovery. This means that your discovery muscle needs to be fit and ready to move, explore, and innovate to exploit opportunities.

In the process we advocate, you move your discoveries into delivery so you can validate the value of the product options you chose. You assess what you deliver, and you validate what you have learned. Did the result deliver the anticipated value? If so, keep doing what you’re doing. If not, then loop back using your validated learning to feed subsequent exploration and experimentation.

Successful software teams deliver great products because they invest in ongoing discovery as well as delivery. With frequent exercise, their discovery muscle becomes stronger and more limber and works smoothly in tandem with their delivery muscle. **{end}**

mary@ebgconsulting.com  
ellen@ebgconsulting.com

Sticky  
Notes

For more on the following, go to [StickyMinds.com/bettersoftware](http://StickyMinds.com/bettersoftware).

- References
- Figure Sources

### Contribute to the Agile Community!

We are looking for contributions from the community for publishing on AgileConnection.com. If you have a technical publication, story, or even just an idea that you would like to share, email AgileConnection technical editor Johanna Rothman at [jr@jrothman.com](mailto:jr@jrothman.com) to get started or, if you prefer, click the link below to submit your idea directly from the website.

Here’s more on our editorial process and what we’re looking for in an article submission:

- 1. Tell a Story:** Our readers love articles that tell a compelling story! Tell us a story of success or failure. Sure, change the names to protect the innocent or guilty. If you’re not sure how to do that, we will work with you to help. Tell us how things were terrible at the beginning of the project. Tell us what happened when you used agile or lean approaches. If you have data or metrics, that’s even better.
- 2. Word Count:** Articles that run about 1,000 words are perfect. However, if they are longer or shorter, we’ll work with you to get them published.
- 3. Editorial Process:** Submit an abstract or story idea, and we’ll review it to make sure it’s a good fit for our audience. If it is, we will work with you to make sure you are saying what you want to say and help you craft your story into something you can be proud of. We will copy edit your prose for grammar, style, and flow. Then, we add your article to the publishing schedule.

Your article will be spotlighted in a weekly AgileConnection To Go newsletter, so expect some great exposure.

Can’t wait to get started? Simply email Johanna at [jr@jrothman.com](mailto:jr@jrothman.com) or use this Contribute to the Community link and complete the submission form:

[agileconnection.com/webform/contribute-community](http://agileconnection.com/webform/contribute-community)



# TECHWELL™

visit us  
online at  
techwell.com

TechWell features insightful, software-focused stories curated by passionate industry professionals. Updated daily, TechWell keeps you informed about what's happening in software development and delivery.

The screenshot shows the TechWell website interface. At the top, there's a navigation bar with the TechWell logo and links for 'About', 'Subscribe', and 'Contact Us'. Below this is a secondary navigation menu with categories: 'Develop', 'Test & QA', 'PM', 'Requirements', 'Agile', 'CM & ALM', 'DevOps', 'Cloud', 'Mobile', and 'People & Teams'. The main content area is divided into several sections:

- TRENDING:** A row of four featured articles with thumbnails: 'Creating an Agile Mindset', 'Should Agile Teams Ban Telecommuting?', 'How to Deal with Distractions', and 'Evernote's Security Breach Leads to Locked Passwords'.
- LATEST STORIES:** A list of recent articles, including 'How to Wake Up Earlier', 'Clean Language in Business Analysis', 'How to Say "No" When Asked for Help', 'Why is the Configuration Management Database Both Praised and Reviled?', 'In-the-Wild Testing', 'The Right Way to Split User Stories', and 'Developers Get the Chance to Succeed on the Big(er) Screen'.
- TECHWELL TO GO:** A section for receiving stories via email, with a subscription form.
- PARTNERS:** Logos and descriptions for partner companies like HP Solutions, Seapine Software, LogiGear, uTest, and CollabNet.
- CONNECT WITH US:** Social media icons for LinkedIn, Twitter, Google+, Facebook, and RSS, along with an 'Open Well' button for sharing ideas.
- UPCOMING EVENTS:** A list of events such as STARCANADA Conference, Software Tester Certification, and STARSEAST Conference.

Visit Techwell.com today  
and join the conversation!

Read

React

Participate





# WHY NoSQL MATTERS

**AND** WHAT CONFIGURATION MANAGEMENT  
EXPERTS NEED TO KNOW ABOUT IT

BY EUGENE DVORKIN





NoSQL is a new approach to data storage that addresses many problems associated with relational databases. Many popular websites—including Facebook, Netflix, and Digg—are using NoSQL to crunch large volumes of data.

Information and applications are now often delivered via different channels, such as web applications, mobile devices, and complex social networks. Many technology professionals refer to “big data” as the approach for handling these channels that are required to crunch huge amounts of information that enterprises want to capture and use. NoSQL is a good fit for storing and accessing such huge volumes of data.

Technology professionals may find that it can be challenging to tackle new technologies and approaches. This article will give you some insight into why NoSQL is important and what configuration management experts need to know about it.

## Relational Databases vs. NoSQL

Developers often find that new challenges require new approaches and creative technical solutions to help their organization achieve its goals. Most systems rely on time-proven relational database management systems—Oracle, Sybase, Microsoft SQL Server, MySQL, PostgreSQL, etc.—to manage large stores of data, and for many years we software developers have developed applications using relational databases. For the most part, relational databases have served us well, and they will be with us for many years to come.

So, what happens when people suddenly develop and use an alternative solution for database storage? To answer this question, we need to consider what’s wrong with the relational database model.

## Scalability

Let’s take a look at scalability. With the explosive growth of applications on the web, mobile devices, and social networking websites, we have seen a corresponding growth in traffic and users’ expectations for instant information. Scalability has become a very big challenge to application developers and architects, and relational databases often cannot meet speed and performance requirements.

Relational databases generally require that you purchase bigger hardware when the database size exceeds a certain limit. As the database grows, we have to add additional resources to the database server—usually CPUs and memory. Sometimes, we move the whole database to bigger, more expensive hardware.

We usually call the relationship between the size of the database and the hardware requirements “vertical scalability,” and it can be very expensive. It’s called vertical scalability because we are scaling up by adding resources to a single node. It has been my experience that NoSQL solutions can scale horizontally via “sharding,” where each node processes and stores only a portion of the huge data set. You can think of this approach as a distributed database that allows companies to collect large amounts of data just by adding new nodes to the cluster.

## Performance

While scalability is essential, performance is an even more

important issue that affects the relational database model. Today, users expect everything to work instantly. Most users feel that it’s unacceptable for a web page to take three seconds to load.

Software developers should know that joining two or more tables in relational databases can adversely impact performance. In fact, the more database “joins,” the harder the database has to work, which leads to poor performance. NoSQL solutions do not require table joins. Instead, they use either embedded documents, like MongoDB, or columns, such as Hbase.

At WebMD, we leveraged MongoDB to develop a sophisticated, real-time, in-session rule engine that captured click-stream traffic data. Performance, scalability, and data-model characteristics were key considerations in this decision. Given our traffic, this was *a lot* of data. Just writing thousands of events per second is a challenge, but the sharding capabilities of MongoDB allowed us to distribute write operations across several servers. If the need arose to support even bigger traffic, we could distribute our database operations even more.

NoSQL is commonly used for special purposes, such as modeling graphs (like a family tree) or displaying connections on Facebook. Relational databases can be used for the purpose of modeling graph relationships, but they are much more complicated to model and code. On the other hand, NoSQL database Neo4j is commonly used to store and process graph relationships. Other unstructured data storage needs can sometimes be best handled by a document-based solution such as MongoDB.

Many of these technologies are new and specialized, but you will find that they are often exactly what you need to handle large amounts of data efficiently. It is common these days to architect an application that talks to different data storage based on a specific user case. For example, a product catalog for an e-commerce site can be stored in MongoDB and session data stored in Redis, while all transactions are in a MySQL database. We call this “polyglot persistence.”

## Advantages to NoSQL

NoSQL is more flexible than relational databases, partly because there is no dependence on a relational-model layout known as a *schema*. For example, the document-based solution MongoDB does not enforce document structure. A document (which corresponds to a row in the relational database management system) in the same collection (or table) does not need to have the same set of fields or structure, and common fields in a collection’s document may hold different types of data.

NoSQL is easy for developers to work with because there is no need to have complex mapping between tables and rows from a relational model to object and classes in object-oriented languages. In fact, applications can simply save objects directly into the NoSQL solution using the provided drivers.

NoSQL databases are a perfect fit for cloud-based deployment. Amazon and other cloud providers make it very simple and economically feasible to provision virtual machines and install databases. There are even prepackaged installations for NoSQL databases within Amazon AWS Marketplace. This gives

*continued on page 36*

# Corporate University: Growing Software Development Talents

Don't throw money at technology and tools without also investing in people.

by Galyna Datsiv | [gdatsiv@softserveinc.com](mailto:gdatsiv@softserveinc.com)

Finding skilled people has become a key requirement of business success, especially in the growing field of information technology. A company might have the most powerful technologies and costly equipment but still lack the personnel to apply these tools efficiently.

The industry constantly changes, and the requirements for candidates do, too. In just a couple of years, today's skills and knowledge won't be enough. Workers must continuously improve their qualifications, acquire new skills, and adjust to modern industry trends.

## Getting New Software Development Professionals on Staff

Headhunting, one of the most popular ways of getting new people to join a team, may seem like a great solution to a shortage of talent. But, examined more closely, it is seldom more than a salary-boosting approach to attract an employee and fill a gap. Your company gets a one-time benefit, but headhunting quickly becomes a never-ending process of employees going from one company to another in search of more attractive offers. Once that cycle begins, where do *new* skilled professionals come from?

There are better ways to add talent to your organization. With the IT industry growing at such a fast pace, one of the best options is to educate professionals inside your company. It also helps to guide your business in the right direction, making it possible to invest in human capital development. In response to this, some organizations have launched corporate training centers and corporate universities, which are special educational entities focused on the training of employees and potential candidates.

## Corporate University—The Best Place to Train Your Workforce

The purpose of a corporate university is to use the existing skills and expertise of trainees for their further development and to add all the necessary qualifications an organization requires. The training process at corporate universities is ad-

justed to the business strategies and needs of the parent company and its customers to prepare the potential candidates for a future job, as well as to introduce them to corporate practices and culture. In software development, a training process often resembles work on a real project, which provides trainees with valuable practical skills and a thorough understanding of the work process and requirements.

No one can provide you with the understanding of industry specifics better than a seasoned industry expert, so it is an excellent practice to involve professional teachers together with skilled industry specialists in the training process at corporate universities. They will be able to share their experiences, inspire candidates with good contextual practices, and also warn

them against possible risks on the way toward accomplishing their professional goals.

In order to educate new and existing employees, corporate universities develop individual training plans that embrace the skills and knowledge that em-

ployees should possess in order to fulfill their duties with excellent results. Such plans will provide them with a clear idea of which techniques and skills they need to master to be ready for the job and the company environment.

## A Training Course or Unique Education Process?

Like any other training process, a corporate university system would not function properly without progress evaluation for trainees. Companies with a systematic approach toward the career and professional growth of each employee provide such evaluations on a regular basis. The results of these evaluations are a starting point for making a personal developmental plan for each employee. As you can see in figure 1, the process of professional growth is iterative. Training is always followed by performance evaluation and an update of individual career plans.

## The Benefits of a Corporate University for Software Development

There are more ways to add value to corporate universities.

“A company's success depends on the performance of its workforce.”

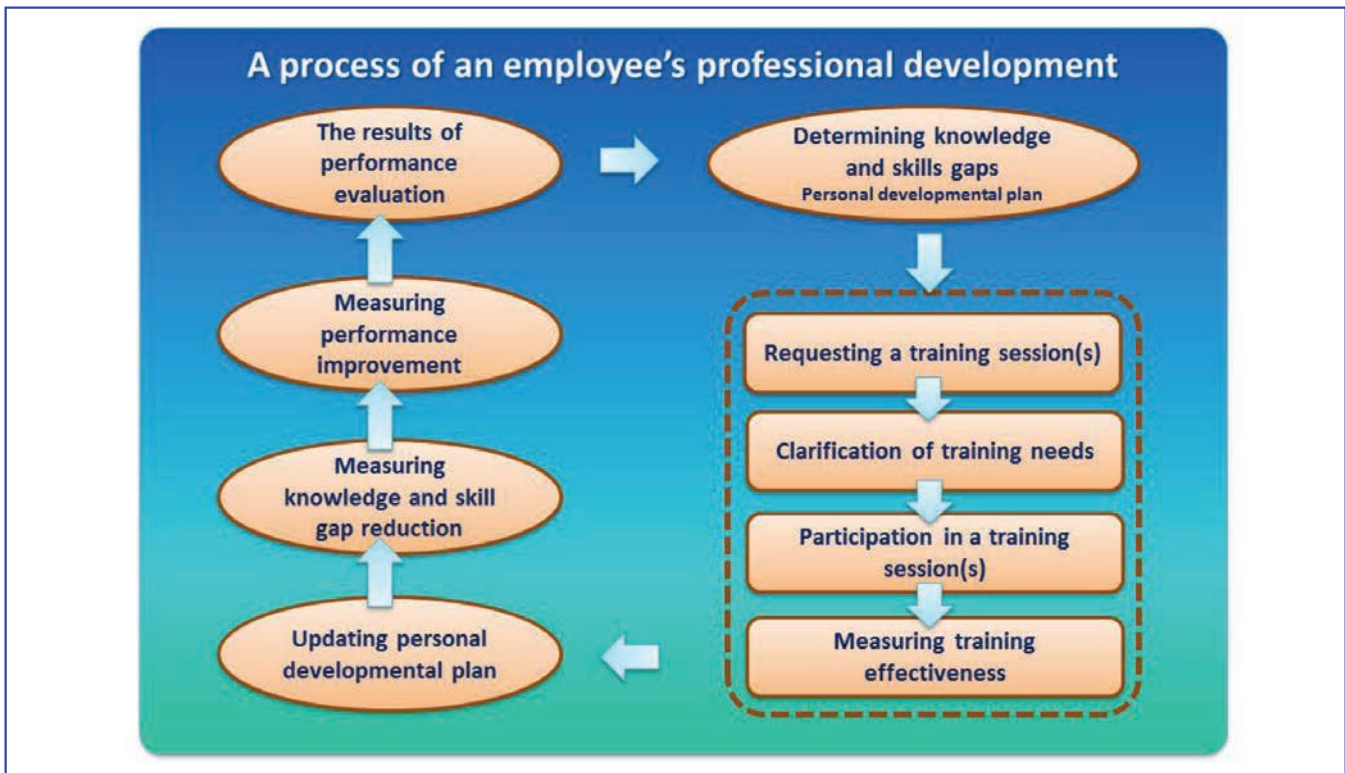


Figure 1: The iterative process of professional growth

For instance, introducing new employees to the corporate culture and practices will make it easier for new people to adjust, to become a part of the team, to get used to the new working environment, and to organize their work more efficiently.

Corporate universities have become one of the most effective solutions to the problem of training and hiring an industry workforce, despite the fact that organizing such entities involves considerable effort and resources. These investments are justified by the long-term results, as companies are now able to hire loyal employees who are motivated to improve their own qualifications and provide breakthrough ideas for business growth.

Most corporate universities not only train young specialists but also develop training plans for existing employees to improve their qualifications and invest in their career growth. As for IT, upon completion of a training course, the trainees often get the chance to become certified industry experts by receiving an international certificate in their related field (Microsoft, Cisco, etc.), which serves as another motivating factor. Requalification opportunities can be provided, as well. At international corporate universities, employees are often required to improve their skills of English, a global language of the IT industry, with a strong focus on their professional needs and company specifics.

Customers also receive significant benefits if a vendor has in-house training facilities. The professional level and practical skills of junior developers who undergo training at a corporate educational institution are considerably higher than those of the newcomers. Any knowledge gaps can be identified and

corrected quickly, and additional certifications can be obtained without taking staff out of the work process. Plus, corporate training opportunities increase motivation and loyalty of personnel, thus reducing staff rotation on projects.

## Corporate University vs. Traditional University

It is important to remember that corporate universities will never substitute for traditional universities. Their primary goal is to offer education and training adjusted to the business needs of the parent company. However, a number of companies integrate their corporate university activities into traditional universities by adding special training courses to the university timetables. This is another good way to make sure that young professionals are introduced to the needs, new trends, and practices of the ever-changing world of IT.

## Conclusion

Corporate universities are among the most promising investments required for the growth of an IT organization. A company's success depends on the performance of its workforce, so you have to ensure the growth of new talents and existing employees. By growing a team of seasoned industry experts now, software development companies can be sure that tomorrow they will be able to add even more motivated and skilled employees capable of moving the company to the next level. Growing dedicated technologists with a deep understanding of the business' needs will help a company empower its customers through high-quality services and products. **{end}**



## TestPlant Hatches eggOn

TestPlant, an automated software testing company, released eggOn, a new development in mobile app testing technology. The eggOn application can be installed on any Android or iOS phone, including iOS 6, and provides full control of the device from eggPlant without jailbreaking. With eggOn, developers do not need to install third-party software or deploy complicated software development kits to allow full testing on real devices.

EggPlant for mobile is an automated solution for mobile application testing across all operating systems and end-user devices (including BlackBerry and Windows devices), allowing full control of real devices to run tests that other tools cannot. Requiring no special hardware, eggPlant is the only automated mobile app testing solution that uses image-recognition technology to allow it to see and interact with any device display, whether testing native or web-based mobile applications.

[www.testplant.com/products/eggplant/mobile](http://www.testplant.com/products/eggplant/mobile)

## SmartBear Software Releases Collaborator

SmartBear Software, a software quality tool company, released Collaborator, an easy-to-use tool that expands upon the review features of CodeCollaborator, the first enterprise-class commercial code review tool. Collaborator delivers the proven code review experience to documents authored by members of the extended development team, including user documentation, specifications, and test plans.

Collaborator allows reviewers to provide their feedback in a centralized tool, meticulously tracking each comment until resolution. This allows all reviewers to see each other's comments and to interact using a chat-style interface if they happen to be reviewing the artifact at the same time.

Collaborator includes all of CodeCollaborator's features for code review, including out-of-the-box support for sixteen SCMs, integration with major IDEs, customizable workflows, defect management, and extensive reporting through its audit trail. Collaborator adds support for reviewing Microsoft Word and PDF documents.

[smartbear.com/products/software-development/code-review](http://smartbear.com/products/software-development/code-review)

## Compuware Corporation Unveils Compuware APM AJAX Edition 4

Compuware Corporation, a technology performance company, unveiled Compuware APM AJAX Edition 4, a cross-browser diagnostics tool. This new version enables developers to test and diagnose performance issues across all recent and legacy browser versions in Internet Explorer and Firefox.

With more application logic sitting in the web browser than ever before, it has never been more important to measure and optimize the performance of Web 2.0 pages and applications. Additionally, in a world where the competition is only a click away, highly interactive web experiences with dynamic Web 2.0 pages and applications have become crucial to keeping customers loyal and attracting new ones. This often means introducing AJAX actions that do not trigger a complete page

reload in order to deliver a world-class end-user experience.

AJAX Edition 4 was built to deliver root-cause analysis of performance issues that this increased complexity often creates. It allows developers to pinpoint problems and immediately make front-end web performance improvements on their website and Asynchronous JavaScript and XML (AJAX) web applications.

[compuware.com](http://compuware.com)

## Keynote Systems Integrates the Keynote DeviceAnywhere Platform with Worksoft Certify

Keynote Systems, an Internet and mobile cloud testing and monitoring company, announced the integration of the Keynote DeviceAnywhere platform with Worksoft Certify. Through the Worksoft Certify Mobile Interface Extension, users can now perform automated end-to-end testing of SAP mobile applications on real devices.

The integration of DeviceAnywhere is a natural progression of Keynote's previously announced worldwide partnership to extend Worksoft's test automation software for SAP solutions to the mobile ecosystem. Worksoft now connects to DeviceAnywhere through the Worksoft Certify Mobile Interface Extension, extending users' existing libraries to include a private cloud of mobile devices in conjunction with Worksoft's Certify solution.

The Keynote-Worksoft integration allows SAP customers to use the Worksoft Certify Mobile Interface Extension to test both SAP and non-SAP mobile business applications, enabling operations and QA teams with advanced tools to ensure quality in mobile business systems across industries.

[keynote.com](http://keynote.com)

## Perforce Software Launches Perforce Commons

Perforce Software launched Perforce Commons, a document collaboration tool that ensures that business professionals can more productively work together on files. Commons supports all types of files—from the largest binary objects to the smallest image files—and has merge capabilities for the most common document types, including Microsoft Word and PowerPoint. Because it keeps track of the complete history of any file, it saves business teams significant time and trouble in finding, revising, and collaborating on documents.

Document collaboration issues, such as merging multiple edits into a single document or confusion over the most current version of a file, occur as documents pass through many workers and many versions. These issues occur frequently and take a sizeable toll on productivity. Commons makes it easy for business teams to access and version files alongside the contributions of technical staff. Perforce customers now have a single source of truth for all files and documents enterprise-wide.

[perforce.com/product/commons](http://perforce.com/product/commons)

# FAQ

expert answers to  
frequently asked  
questions

by Dawn Haynes  
dhaynes@perftestplus.com

## What Should I Do When My Bug Gets Rejected?

Software testers often ask me this question, and it takes me back—way back—to my first testing job. I was the sole tester. I took my new job very seriously because I came from technical support and because of my shiny new title, QA engineer. I thought that meant I had to “assure” quality and protect the customers from as many bugs as possible. It turns out I was wrong. Here’s what I wished I had known then.

**1. Don't take it personally. It's not about you.** Really, it's not. Don't pout or get upset. It's not useful. Also, don't make it personal for others. Avoid asking questions like “Don't you care about quality or the customer?” In almost thirty years, I've never successfully persuaded anyone to fix a bug by beating him or her with the Quality Stick!

**2. Don't argue. Start a conversation.** It's easy to get defensive when someone says your bug is invalid because “It Functions as Designed,” “A User Would Never Do That,” “That Will Never Happen in Production,” or the infamous “It's Not a Bug.” Instead, seek to understand why this bug isn't getting traction. Maybe it's an unlikely or noncritical issue. Maybe it's too late or too risky to fix this bug right now. But, you'll never understand or adapt unless you find out how the decision makers came to their conclusion.

If that's not enough for you to sleep well at night, expand the conversation to include impacts of the bug's occurrence. Ask how stakeholders would react if these issues surfaced after deployment. Include what you learn in the bug report, and let it fight for itself! (See Cem Kaner's course on bug advocacy to learn how to write better and more compelling bug reports. [1])

**3. Don't try to do someone else's job.** (You do enough already!) If you've read this far, it's unlikely that your job is to decide what to fix, so seek out the accountable party and ensure that she is aware of the issue. Be prepared to accept her decision, even if you don't agree. Remember, all software ships with bugs—some we know about, some we don't. Fixing this bug is a choice with impacts. Ultimately, it's a business decision to fix or not.

**4. Don't get stuck.** There are always more bugs to find than can be fixed, so once you've done your due diligence, let go and move on. Being a testing “boat anchor” to keep the project from moving forward often is not the best strategy, so go find a bug that will get fixed!

**5. Avoid insanity (doing the same thing repeatedly and expecting a different result).** If you're finding a lot of bugs that the team isn't fixing, find out why. Are you testing the wrong thing? Is your focus out of scope? Find out what the stakeholders' current area of concern is. Ask about testing's mission on this project, at this time. Is it finding defects or verifying functionality?

**5.1. Learn and adapt.** Apply what you've learned, ask what's important before you invest a lot of time doing test activities, and do some research. I did. I always had thought that I was an excellent predictor of bug importance, but empirical evidence from many years and projects showed that I was “right” about half of the time—and those aren't betting odds, folks.

Good luck!

Sticky  
Notes

For more on the following, go to  
[StickyMinds.com/bettersoftware](https://StickyMinds.com/bettersoftware).

■ References

*Aziz, continued from page 22*

investment for service virtualization tools can easily outweigh physical test environment maintenance costs for a testing team.

Service virtualization technology streamlines development and testing efforts through the elimination of defects earlier in the lifecycle. Many of the eliminated defects are integration-related defects that are typically not found until the end of a software development lifecycle due to unavailability of accurate dependent systems in a testing environment. Service virtualization technology benefits a software development organization by enabling quicker delivery of a higher-quality product to the marketplace through earlier-lifecycle defect detection. The tools allow an organization to decrease infrastructure costs, increase developer and tester productivity, and control software development and testing cycle times. **{end}**

scott.aziz@ust-global.com

*Dvorkin, continued from page 31*

you the opportunity to test and deliver your ideas without a huge, up-front investment in hardware and license. Most of the leading NoSQL solutions are open source. Then, as your application grows, just keep adding shards to your database cluster by adding virtual machines and distributing the load.

There are growing numbers of cloud database providers who can take care of all database management day-to-day maintenance operations and provide support in exchange for a subscription fee. Database as a service is a growing field, and we see new players all the time.

There are different types of NoSQL that you may find yourself having to support on your next deployment. For example, you could use a distributed key-value pair (Riak or Redis), a document-based (MongoDB) or column-based (Cassandra) version, or some kind of graph database (Neo4j).

New technology requires you to be creative and open to new strategies. If you are the release engineer on a NoSQL project in the future, I hope this article helps you better understand how this technology works and where you can help your team achieve success. **{end}**

edvorkin@gmail.com

## index to advertisers

ADC/BSC West 2013	adc-bsc-west.techwell.com	9-12
ALP International	www.alpi.com	19
ASTQB	www.astqb.org	22
Cognizant	www.cognizant.com/testing-services	2
HP	www.hp.com	Back Cover
OpenMake Software	www.openmakesoftware.com	1
STARWEST 2013	starwest.techwell.com	23
SQE Training—Live Virtual	sqetraining.com/training/delivery-options/live-virtual	Inside Back Cover
TechWell	www.techwell.com	29
Telerik	www.telerik.com/html5-testing	Inside Front Cover
TestingWhiz	www.testing-whiz.com	8

**Display Advertising**  
advertisingsales@sqe.com

**All Other Inquiries**  
info@bettersoftware.com

*Better Software* (ISSN: 1553-1929) is published six times per year: January/February, March/April, May/June, July/August, September/October, and November/December. Back issues may be purchased for \$15 per issue plus shipping (subject to availability). Volume discounts available. Entire contents © 2013 by Software Quality Engineering (340 Corporate Way, Suite 300, Orange Park, FL 32073), unless otherwise noted on specific articles. The opinions expressed within the articles and contents herein do not necessarily express those of the publisher (Software Quality Engineering). All rights reserved. No material in this publication may be reproduced in any form without permission. Reprints of individual articles available. Call for details.

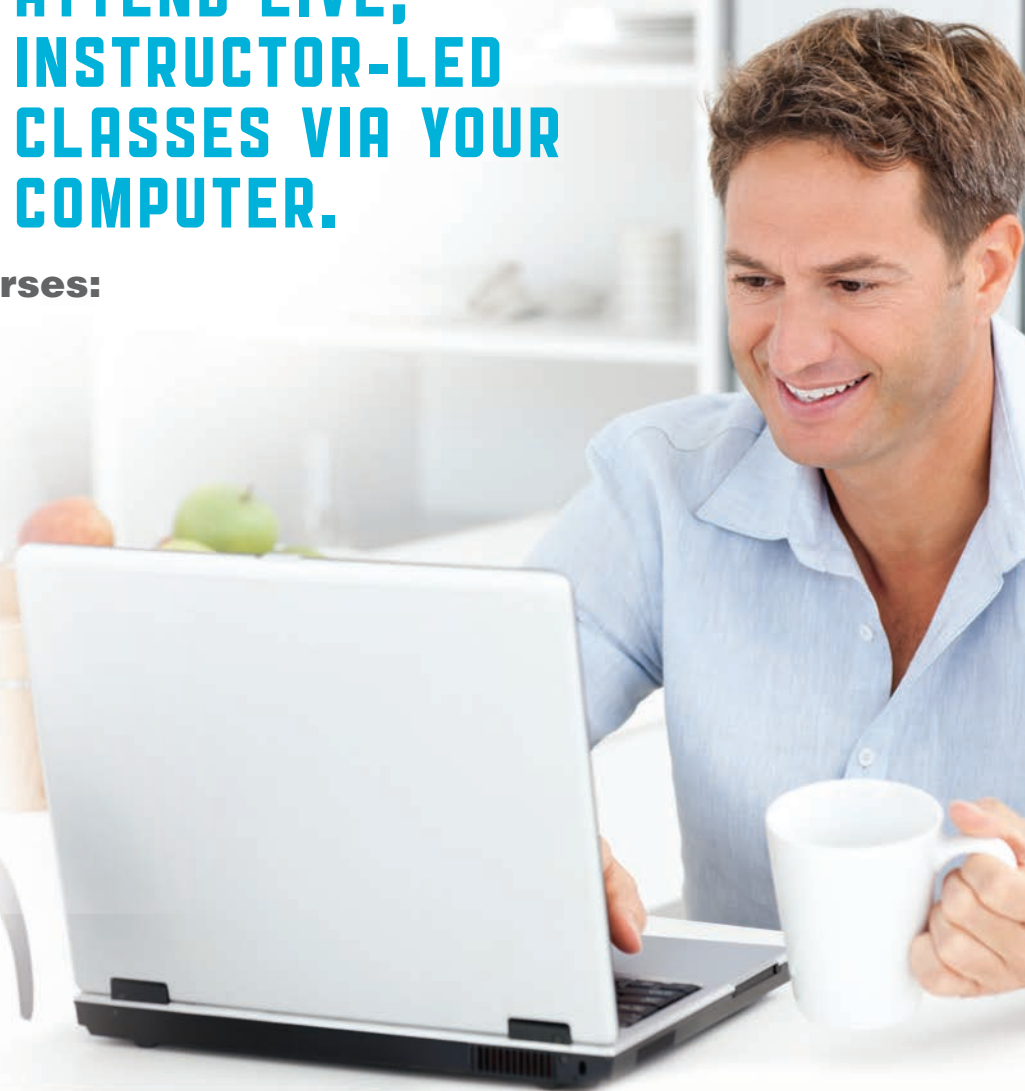




# ATTEND LIVE, INSTRUCTOR-LED CLASSES VIA YOUR COMPUTER.

## **NEW** Live Virtual Courses:

- » Testing Under Pressure
- » Performance, Load, and Stress Testing
- » Getting Requirements Right the First Time
- » Essential Test Management and Planning
- » Finding Ambiguities in Requirements
- » Mastering Test Automation
- » Agile Test Automation
- » Generating Great Testing Ideas
- » High Powered Test Design
- » Configuration Management Best Practices
- » and More



## **Convenient, Cost Effective Training by Industry Experts**

### **Live Virtual Package Includes:**

- **Easy course access:** You attend training right from your computer, and communication is handled by a phone conference bridge utilizing Cisco's WebEx technology. That means you can access your training course quickly and easily and participate freely.
- **Live, expert instruction:** See and hear your instructor presenting the course materials and answering your questions in real-time.
- **Valuable course materials:** Our live virtual training uses the same valuable course materials as our classroom training. Students will have direct access to the course materials.
- **Hands-on exercises:** An essential component to any learning experience is applying what you have learned. Using the latest technology, your instructor can provide students with hands-on exercises, group activities, and breakout sessions.
- **Real-time communication:** Communicate real-time directly with the instructor. Ask questions, provide comments, and participate in the class discussions.
- **Peer interaction:** Networking with peers has always been a valuable part of any classroom training. Live virtual training gives you the opportunity to interact with and learn from the other attendees during breakout sessions, course lecture, and Q&A.
- **Convenient schedule:** Course instruction is divided into modules no longer than three hours per day. This schedule makes it easy for you to get the training you need without taking days out of the office and setting aside projects.
- **Small class size:** Live virtual courses are limited to the same small class sizes as our instructor-led training. This provides you with the opportunity for personal interaction with the instructor.

# May the most agile development team win.

## Try a 30-day free trial of HP Agile Manager.

Automate sticky notes, scrum boards, and burndown charts. Ramp up Agile teams faster. Manage sprints with an intuitive, easy-to-follow UI. It's the purpose-built Agile management tool designed to help you develop complex applications faster. Because making the complex simple matters. Start your free trial today. Go to [hpagilemanager.com](http://hpagilemanager.com)



**Make it matter.**

