July/August 2014 www.TechWell.com

BETTER SOFTWARE™

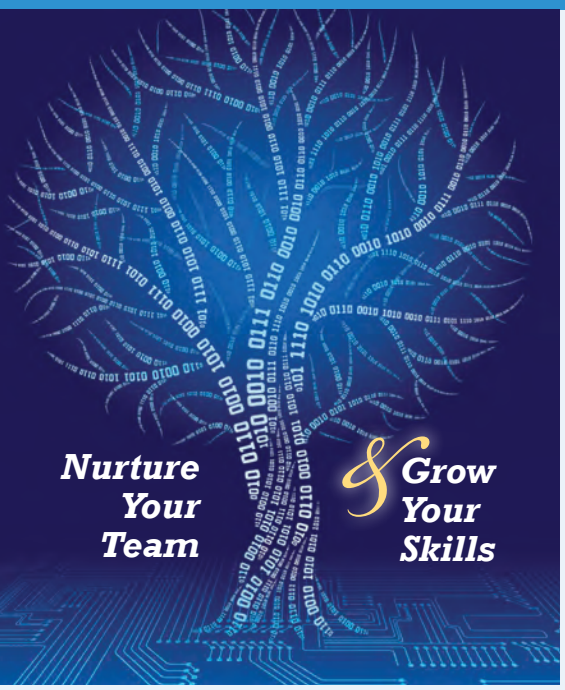
A TECHWELL PUBLICATION

USING PROCESS FRAMEWORKS
Why fixed processes are not enough

MOBILE APP DEVELOPMENT
Build apps users will trust

Wearable Computing Reaches New Heights

GROW YOUR SKILLS WITH TESTING



The more training you take the greater the savings!

Maximize the impact of your training by combining courses in the same location. Combine a full week of training for the largest discount!

2014 FALL SCHEDULE

TESTING TRAINING WEEKS

September 22–26, 2014
Washington, DC

October 20–24, 2014
Tampa, FL

November 3–7, 2014
San Francisco, CA

WASHINGTON, DC TRAINING WEEK
September 22–26, 2014

Green background Indicates courses pre-approved for Project Management Institute PDUs.

MONDAY	TUESDAY	WEDNESDAY	THURSDAY	FRIDAY
Software Tester Certification—Foundation Level			Mastering Test Design	
Testing Under Pressure	Test Estimation and Measurement	Security Testing for Testing Professionals		Exploring Usability Testing
How to Break Software: Robustness Testing Unleashed			Mobile Application Testing	
Fundamentals of Agile Certification—ICAgile		Agile Tester Certification—ICAgile		
Systematic Software Testing			Performance, Load, and Stress Testing	
Essential Test Management and Planning		Leadership for Test Managers	Test Process Improvement	

WAYS to SAVE

Take advantage of the different "Ways to Save" on training using our discount programs listed below. Purchase valuable software quality training for your whole team and save.

EARLY BIRD

Register 6 weeks prior to any training week course and receive $50 off per registered course day. Take a full week of training and save $250!

TRAINING WEEK AND SAVE

Combine specialized training courses in the same location and save. Discounts vary depending on the number of training days combined.

GROUP DISCOUNTS

Have a group and want to save more? Get details on our discount policy by contacting our Client Support Group.

ON-SITE ADVANTAGE

Bring any course to your location for team training. On-site training is both cost-effective and convenient for your team of six or more.

CONFERENCE AND SAVE

Save $300 when you combine any of our pre-conference training courses with your conference registration.

For more details on our discount policy, contact the Client Support Group at ***sqeinfo@sqe.com*** *or call 888.268.8770 or 904.278.0524.*

TRAINING FROM SQE TRAINING

TAMPA, FL TRAINING WEEK
October 20–24, 2014

Green background Indicates courses pre-approved for Project Management Institute PDUs.

MONDAY	TUESDAY	WEDNESDAY	THURSDAY	FRIDAY
Software Tester Certification—Foundation Level			Mastering Test Design	
Testing Under Pressure	Test Estimation and Measurement	Security Testing for Testing Professionals		Exploring Usability Testing
How to Break Software: Robustness Testing Unleashed			Mobile Application Testing	
Fundamentals of Agile Certification—ICAgile		Agile Tester Certification—ICAgile		
Risk-Driven Software Testing		Testing with Use Cases	Performance, Load, and Stress Testing	
Essential Test Management and Planning		Leadership for Test Managers		

SAN FRANCISCO, CA TRAINING WEEK
November 3–7, 2014

Green background Indicates courses pre-approved for Project Management Institute PDUs.

MONDAY	TUESDAY	WEDNESDAY	THURSDAY	FRIDAY
Software Tester Certification—Foundation Level			Mastering Test Design	
Testing Under Pressure	Test Estimation and Measurement	Security Testing for Testing Professionals		Exploring Usability Testing
How to Break Software: Robustness Testing Unleashed			Mobile Application Testing	
Fundamentals of Agile Certification—ICAgile		Agile Tester Certification—ICAgile		
Risk-Driven Software Testing		Testing with Use Cases	Performance, Load, and Stress Testing	
Essential Test Management and Planning		Leadership for Test Managers	Test Process Improvement	

LEARNING OPTIONS:

Public — Instructor-led training in a city near you

LIVE VIRTUAL TRAINING — Live, instructor-led classes via your computer

eLearning — Self-paced learning, online

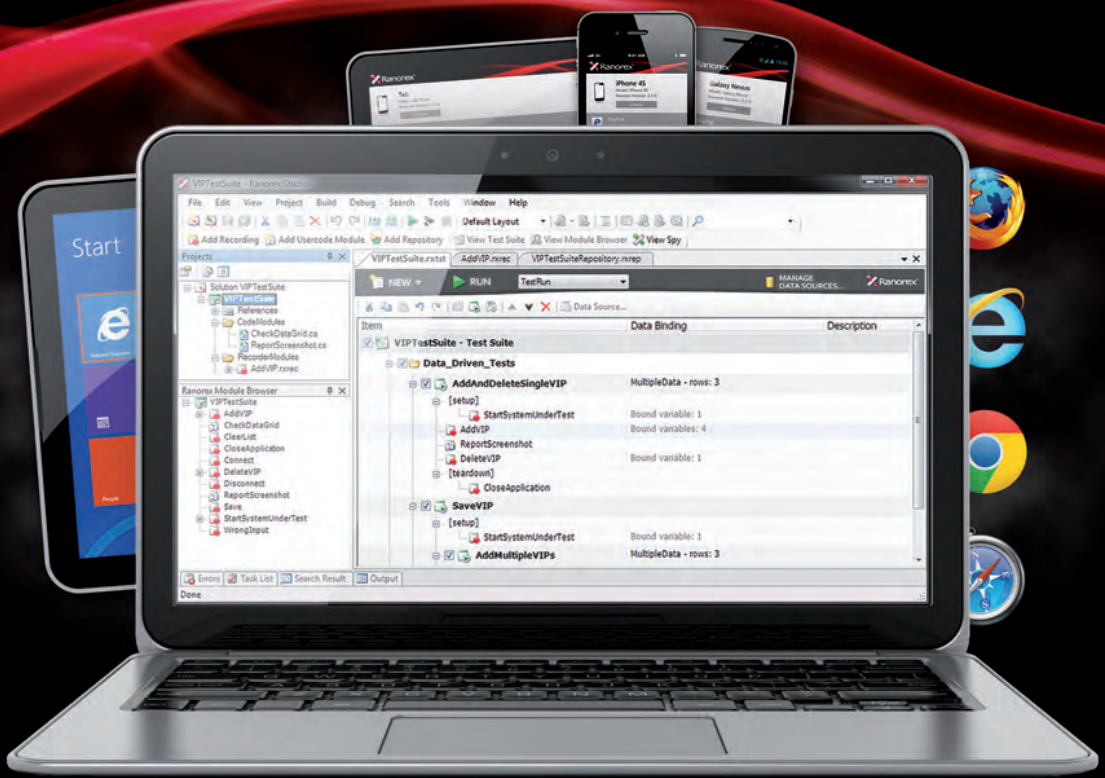
ON-SITE ADVANTAGE — Instructor-led training at your location

For information on our 60+ Public and 40+ Live Virtual Course Dates
visit www.sqetraining.com

SQE TRAINING

Ranorex®

Automated Testing of Desktop. Web. Mobile.



- Robust Automation
- Broad Acceptance
- Seamless Integration
- Quick ROI

Why Use Ranorex
www.ranorex.com/why

Award-winning test automation tools provide seamless testing of a wide range of desktop, web and mobile applications.
Android | iOS | HTML5 | IE | FF | Chrome | Safari | WPF | Flash/Flex | Silverlight | Qt | SAP | .NET | MFC | Delphi | 3rd Party Controls | Java

BETTER SOFTWARE™
A TECHWELL PUBLICATION
Volume 16, Issue 4 • JULY/AUGUST 2014

CONTENTS

in every issue

features

columns

MARK YOUR CALENDAR

SQE TRAINING

software tester certification

http://sqetraining.com/certification

Foundation Level Certification
August 26–28, 2014
Boston, MA
Chicago, IL

September 9–11, 2014
Indianapolis, IN

September 16–18, 2014
San Jose, CA
Toronto, ON

September 22–24, 2014
Washington, DC

September 23–25, 2014
Austin, TX
Sacramento, CA

Advanced Tester Certification
August 25–29, 2014
Chicago, IL

October 27–31, 2014
Washington, DC

training weeks

http://sqetraining.com/trainingweek

Testing Training Week
September 22–26, 2014
Washington, DC

October 20–24, 2014
Tampa, FL

November 3–7, 2014
San Francisco, CA

Agile Software Development Training
November 9–11, 2014
Orlando, FL

conferences

STAR*WEST*
http://starwest.techwell.com
October 12–17, 2014
Anaheim, CA
Disneyland Hotel

Agile Development Conference East
http://adceast.techwell.com
November 9–14, 2014
Orlando, FL
Hilton Orlando Lake Buena Vista

Better Software Conference East
http://bsceast.techwell.com
November 9–14, 2014
Orlando, FL
Hilton Orlando Lake Buena Vista

Mobile Dev + Test Conference
http://mobiledevtest.techwell.com
April 12–17, 2015
San Diego, CA
Manchester Grand Hyatt

STAR*EAST*
http://stareast.techwell.com
May 3–8, 2015
Orlando, FL
Gaylord Palms Resort

Agile Development Conference West
http://adcwest.techwell.com
June 7–12, 2015
Las Vegas, NV
Caesars Palace

Better Software Conference West
http://bscwest.techwell.com
June 7–12, 2015
Las Vegas, NV
Caesars Palace

STAR*CANADA*
http://bscwest.techwell.com
June 2015
More details coming soon!

BETTER SOFTWARE™
A TECHWELL PUBLICATION

Publisher
Software Quality Engineering Inc.

President/CEO
Wayne Middleton

Director of Publishing
Heather Shanholtzer

Editorial

Better Software Editor
Ken Whitaker

Online Editors
Cameron Philipp-Edmonds
Josiah Renaudin
Beth Romanik

Production Coordinator
Donna Handforth

Design

Creative Director
Catherine J. Clinger

Advertising

Sales Consultants
Daryll Paiva
Kim Trott

Sales Coordinator
Alex Dinney

Marketing

Junior Marketing Manager
Cristy Bird

Marketing Assistant
Kelly Radell

CONTACT US
Editors: editors@bettersoftware.com
Subscriber Services:
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

Explosion of Mobile and the Internet of Everything

Now that we're into the warmth of the summer and most of us are enjoying being in the great outdoors, what better way to stay connected than for us to be attached to small computing devices?

Yikes!

Breaking away from our typical article content, two of the four feature articles in this issue explore the next wave of computing: mobile and wearable intelligent devices. Rather than simply reporting on these trends, our authors write about how mobile computing and the wave of the Internet of Things bring new experiences and challenges to typical software development. In fact, I'd go so far as to say that our approaches to delivering quality products are going to be tested with a broader Internet of *Every*thing! Just because these devices are small in size doesn't mean that the software design needed to drive their operation is simple. In fact, the planning, development, risk management, and validation effort required of project teams may become even more complicated. You're going to see more articles coming in future editions of *Better Software* that should help you prepare for this exciting technology revolution.

Sometimes I get asked by subscribers, "How can I access past issues of *Better Software* magazine?" To register for a free subscription, point your browser to http://www.stickyminds.com/better-software-magazine. It takes only a minute or two to sign up. To quickly view the latest issue's PDF, simply click the Download link. All current and past issues can be viewed on our archive page: https://well.tc/Gt6.

Occasionally, readers want a printed issue to pass around the office. TechWell has a relationship with MagCloud so that you can order a hard copy of *Better Software* magazine by browsing to http://www.magcloud.com/user/bettersoftware. It may take a couple of weeks after the online issue is posted for the printed version to become available on MagCloud.

We truly value your feedback. Let us and our authors know what you think of the articles. I sincerely hope you enjoy this issue!

Ken Whitaker
kwhitaker@sqe.com
Twitter: @Software_Maniac

Contributors

Alan Crouch is a senior software security specialist with Coveros, a Virginia-based firm focused on agile, software quality, and application security. Alan has worked closely with federal agencies and private companies to advise, audit, and support IT security and governance teams. In addition to his cybersecurity experience, he has a strong background in software engineering, test analysis, test automation, and security testing. Alan has focused his career on building secure software and developing better software security practices. You can contact Alan at alan.crouch@coveros.com.

The managing consultant at Excelon Development, Matt Heusser is probably best known for his writing. In addition to currently serving as technical editor of StickyMinds.com, Matt was the lead editor for *How to Reduce the Cost of Software Testing* (Taylor and Francis, 2011). He has served both as a board member for the Association for Software Testing and as a part-time instructor in Information Systems for Calvin College. Matt can be reached at matt@xndev.com.

As an agile coach, consultant, and trainer, Manoj Khanna helps companies navigate the sometimes choppy waters of agile product development through strategic consulting, training, coaching, and mentoring. He shows project teams how to avoid pitfalls, maximize productivity, and develop effective solutions to deliver maximum business value. Manoj is the cofounder of customer experience management consulting firm Sntio and can be reached at mk@sntio.net.

After working in various management and technical positions in software development and product management, Philip Lew leads XBOSoft's direction and strategy as CEO. Philip has been published in several IEEE and ACM publications and in various trade journals. A speaker at numerous trade and academic conferences, he has worked with hundreds of organizations to assess the quality of their software, examine software quality processes, and set forth measurement plans to improve software quality using systematic methods. Find out more about Philip at xbosoft.com.

Ryan Olivett is an agile consultant at LitheSpeed and is skilled in business process optimization, data analysis, lean management, and Scrum. A certified ScrumMaster and Scrum product owner, Ryan leverages his diverse background and skill set on client engagements to identify problems and provide creative solutions. He primarily works on organizational assessments and process documentation at companies that range from small startups to large enterprises. Contact Ryan at ryan.olivett@lithespeed.com.

With twenty-five years of project management, software development, engineering, and business experience, Bob Payne is the VP of coaching services at LitheSpeed and a leading proponent of agile methodologies and agile engineering practices. As host of the AgileToolkit podcast, Bob has produced more than 120 podcasts, recording a variety of industry leaders and agile practitioners. You can contact Bob at bob.payne@lithespeed.com.

When not working on his theory of time travel, Cameron T. Philipp-Edmonds is writing for TechWell, StickyMinds, and AgileConnection. With a background in advertising and marketing, Cameron is partial to the ways that technology can enhance a company's brand equity. In his personal life, Cameron enjoys long walks on the beach, romantic dinners by candlelight, and playing practical jokes on his coworkers. He can be reached at cphilippedmonds@sqe.com.

Mukesh Sharma created QA InfoTech with a vision to provide quality assurance solutions for business partners worldwide. He is responsible for the company's global operations, marketing, sales, and development efforts. Under Mukesh's leadership, QA InfoTech has grown to five centers of excellence, six hundred employees, and over $15 million in revenue. His career spans DCM Data Systems, Quark Inc., Gale Group, and Adobe Systems in software quality engineering and test management roles. Mukesh is an active test evangelist and can be contacted at mukesh@qainfotech.com.

Tom Wessel has played a number of roles in the IT industry, including technical support, quality assurance, test automation, software development, and project management. Recently, Tom's focus has been on his passion for working with teams in large corporations as a ScrumMaster and agile coach to help evolve their software development practices. Tom is a managing director and agile coach and trainer with Davisbase Consulting and he is the president of Southern Fried Agile, a nonprofit focused on hosting agile conferences in various cities. You can follow Tom's musings on software development at http://www.davisbase.com.

Quick, Before We Release ...

Testing, a key component of the software development lifecycle, requires a strategy in order to reduce product delivery time to market.

by Matt Heusser | matt@xndev.com

When it comes to testing software, most project teams think primarily of two things: the testing of an individual piece of functionality and the testing of the software as a whole system.

We usually refer to the testing of the whole system as *regression testing*. This implies a search for problems that did not exist in the previously released product or did not exist when specific components were tested in isolation. Instead, let's use the term *release testing* to refer to the process of looking for any emergent risk before releasing the product.

The "testosphere" is full of advice about how to do release testing, from methods like "push a button, get a green bar, and deploy automatically" to an entirely manual process that can take weeks or even months to perform. Long-term teams tend to develop a release-test strategy—a cadence—they practice from build to deployment. Once the team has a cadence in place, they can focus on tweaking it to be the most efficient use of time and, in many cases, shrinking the cadence.

"Long-term teams tend to develop a release-test strategy—a cadence—they practice from build to deployment."

So, What Is a Cadence?

A cadence is not a contract, but, instead, a promise that from when testing begins, the software can be released within a fixed amount of time. After all, bugs can be found late in a cycle, and they may be big enough to block release. A cadence is more like an expectation for how long it generally takes to get a build to production. The cadence is a schedule; it can be dialed up, with the team taking more time on a risky release, or dialed down if the risk is low, the consequences of failure are mitigated, or timeframes are tight.

For example, a team that continually monitors production for problems and can roll back changes quickly might have a much shorter cadence than a team testing an embedded pacemaker with no update ability. I'll call these risk factors the reasons behind deciding on a strategy. They are especially important when talking to someone on a different team. Are the risk factors the same? If they aren't, how should that impact strategy?

Before we get into strategy, then, we should at least talk about these factors. Here are ten to consider.

1. ***How often do you want to go into production?*** A team that wants to deploy frequently will employ a different approach from a team producing software that is loaded into a car's embedded computing system once a year.
2. ***How modular is the code?*** If you can make a change in module A and only deploy module A—or at least have confidence that modules B through Z do not break—then your test strategy can be much less comprehensive than an integrated suite for a highly coupled legacy application.
3. ***Does the code have unit tests, developer tests, or other low-level automated change detectors?*** If so, do they catch failures? Teams that have extensive change detection may shift the focus of release testing away from fundamental business process to user-interface testing, unintended consequences, and black swan risks—the kind that require creative thinking to find. Often, with unit tests in place, teams find they can cover these other risks in more depth and in less time than before.
4. ***How bad is the customer impact if there is a major bug in production?*** If customers have a high tolerance for defects and those defects will not cause financial, safety, or legal problems, you may be able to take more risks and test less.
5. ***How expensive is it to retest and redeploy?*** A quick fix-test-redeploy cycle might mean less release testing—especially when combined with #4 above and #6 below.
6. ***Who finds the bugs first?*** If a bug finds its way into production and only a handful of customers encounter it because you fixed it so quickly, does it really make a sound?
7. ***Do you have an event-driven, customer-facing GUI or is it a batch application?*** Batch applications may be much more amenable to automation, including high-volume, randomized testing. Consider creating seams that separate a REST API (which is similar to a batch application) from a thin GUI layer.

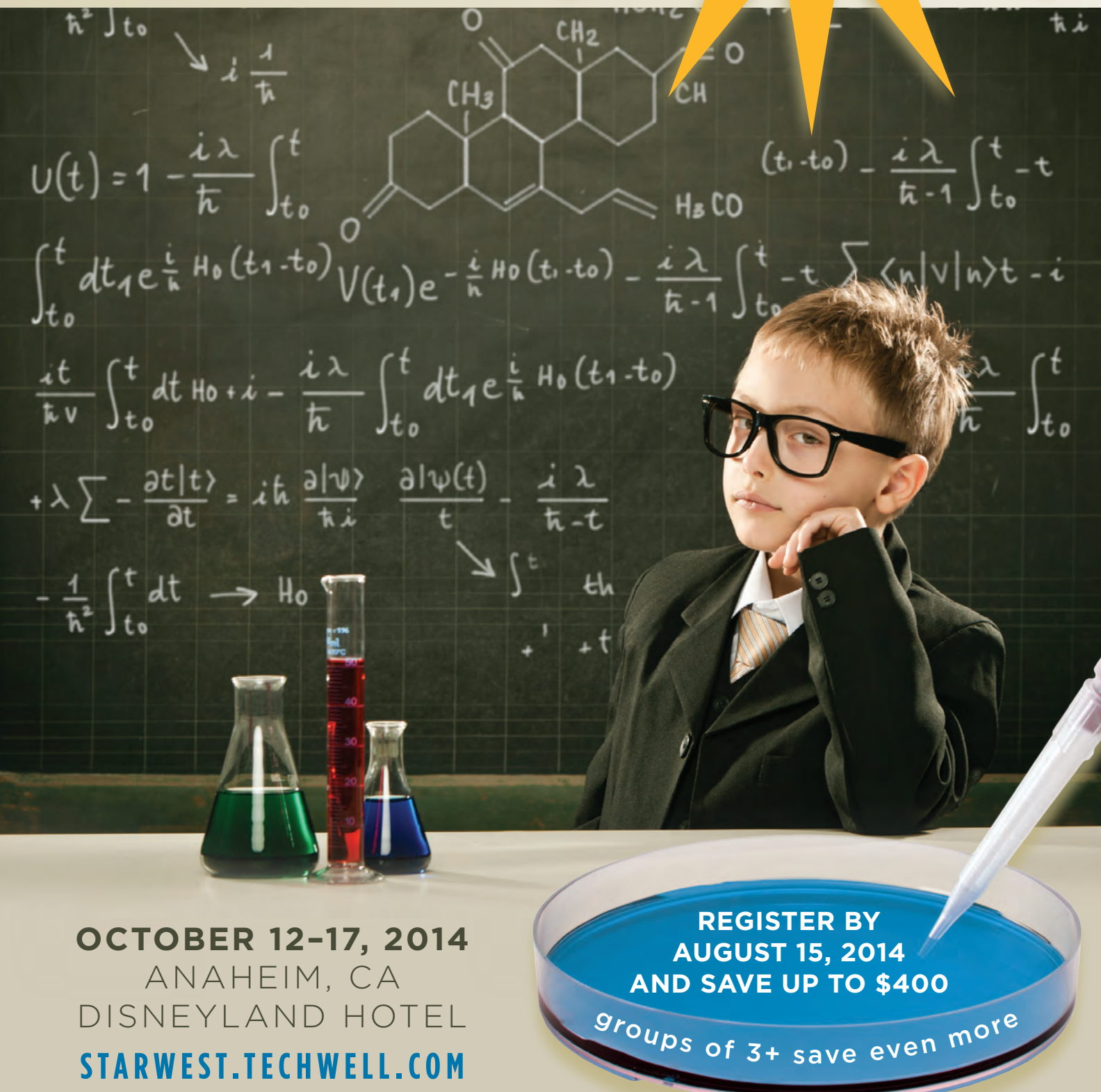
8. ***How expensive is rolling out a fixed build?*** For web-based applications, rollouts may be relatively inexpensive to deploy and transparent to the user. For a desktop application, however, a forced update that requires user intervention may be annoying to customers and imply quality problems.
9. ***Is user acceptance testing (UAT) required? Is it even relevant?*** Some teams enlist the assistance of business users to test the software late in a project's lifecycle. This serves as a final fit-for-use check, but also as a familiarization exercise for the end-users. In many cases, the goals of UAT can be accomplished in different ways and should be performed earlier. Performed as an end-of-project activity, UAT pushes out the cycle time, reduces the general pace of software delivery, and adds risk of a rejection just before going live.
10. ***Do we have any third-party requirements?*** In addition to regulations and audits, does your team need to comply with rules that impact testing? This is also the time to consider other business units, suppliers, and vendors who may care about downtime at the end of the month or a change in a calculation the day before the books are closed for the quarter.

Now What?

Take this list and use it as a guide to describe your team's testing approach. If you are ambitious, find an existing customer and ask him to evaluate your team based on this list. Notice the differences between the two evaluations. Now take those evaluations and review with your cadence. Does your existing release strategy map well to those variables? Where is the opportunity for improvement? Suggest that your team try something new to fill those gaps. **{end}**

Breaking Software

STAR WEST



OCTOBER 12–17, 2014
ANAHEIM, CA
DISNEYLAND HOTEL
STARWEST.TECHWELL.COM

REGISTER BY AUGUST 15, 2014 AND SAVE UP TO $400

groups of 3+ save even more

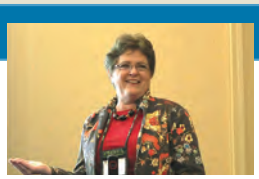
CONFERENCE OVERVIEW

Build your own conference—training classes, tutorials, keynotes, concurrent sessions, the Leadership Summit, and more—packed with information covering the latest technologies, trends, and practices in software testing.

SUNDAY

Agile Tester Certification—ICAgile (2 days)
Fundamentals of Agile Certification—ICAgile (2 days)
Software Tester Certification—Foundation Level (3 days)
Real-World Software Testing with Microsoft Visual Studio® (3 days)
Mastering HP LoadRunner® for Performance Testing (3 days)
Mobile Application Testing (2 days)

MONDAY–TUESDAY



Requirements-Based Testing (2 days)
Multi-Day Training Classes Continue
34 In-depth Half- and Full-day Tutorials

WEDNESDAY–THURSDAY

5 Keynotes
42 Concurrent Sessions
The Expo
Networking and Special Events
Test Lab
…and More!

FRIDAY

Testing & Quality Leadership Summit

Attend the Testing & Quality Leadership Summit Thursday evening and Friday. Join senior leaders from the industry to gain new perspectives and share ideas on today's software testing issues. (Summit registration required)

Workshop on Regulated Software Testing (WREST) (Free, but pre-registration required)

CHOOSE THE CONFERENCE PACKAGE THAT WORKS BEST FOR YOUR BUDGET

SUPER SAVER
★
1 Tutorial Day
$895

BIG SAVER
★★
Conference Only
$1,795

SMART SAVER MOST POPULAR
★★★
Conference + 2 Days
$2,195

For a complete list pricing options, go to: **http://starwest.techwell.com/conference-pricing-2**

Prices valid when you register by August 15th

KEYNOTES TESTING EXPERTS SHARE THEIR FORMULAS

WEDNESDAY, OCT. 15

8:30am

Quality Principles for Today's "Glueware"—Testing Web Services, Libraries, and Frameworks

Julie Gardiner, Redmind

WEDNESDAY, OCT. 15

10:00am

Balancing the Crusty and Old with the Shiny and New

Bob Galen, Velocity Partners

WEDNESDAY, OCT. 15

4:30pm

Lightning Strikes the Keynotes

Lee Copeland, Software Quality Engineering

THURSDAY, OCT. 16

8:30am

Softwarts: Security Testing for Muggles

Paco Hope, Cigital

THURSDAY, OCT. 16

4:15pm

Why Testers Need to Code: Facebook's No Testing Department Approach

Simon Stewart, Facebook

Visit THE EXPO!

Visit Top Industry Providers Offering the Latest in Testing Solutions

TOOLS ★ SERVICES ★ TECHNIQUES ★ DEMOS

WHO SHOULD ATTEND?

Software and test managers, QA managers and analysts, test practitioners and engineers, IT directors, CTOs, development managers, developers, and all managers and professionals who are interested in people, processes, and technologies to test and evaluate software intensive systems

CONFERENCE BONUS

Digital Subscription to *Better Software* Magazine!

STAR*WEST* conference attendees receive a digital subscription to *Better Software* magazine. Delivering relevant, timely information, *Better Software* magazine helps you tackle the challenges of building high-quality software, regardless of your role in the software development lifecycle. **www.BetterSoftware.com**

starwest.techwell.com GROUPS OF 3 OR MORE SAVE BIG!

SUPER EARLY BIRD OFFER

Receive up to $400 off the regular conference registration fee if payment is received on or before August 15, 2014. (depending on the conference package selected)

GROUPS OF 3 OR MORE SAVE UP TO 30% OFF

Register a group of three or more at the same time and save up to 30% off each registration. To take advantage of this offer, please call the Client Support Group at 888.268.8770 or 904.278.0524 or email them at sqeinfo@sqe.com and reference promo code GRP3.

ALUMNI DISCOUNT

STAR alumni receive up to an additional $200 discount off their registration fee. (depending on the conference package selected) If you are a STAR alumni and unable to attend STAR*WEST* this year, you may pass your alumni discount on to a colleague!

MULTI-DAY TRAINING CLASS + CONFERENCE

Save an additional $300 when you attend any of the multi-day training classes and the conference. (discount already reflected in the conference pricing)

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

THE EXPO October 15–16

Visit Top Industry Providers Offering the Latest in Testing Solutions

- Explore this one-of-a-kind Expo designed to bring you the latest solutions in testing technologies, software, and tools.
- Support your software testing efforts by participating in technical presentations and demonstrations conducted throughout the Expo.
- Meet one-on-one with representatives from some of today's most progressive and innovative organizations.

For Sponsor/Exhibitor news and updates, visit **starwest.techwell.com**
To become a sponsor/exhibitor, please contact **sales@sqe.com**

Paco Hope

Years in Industry: **15**
Email: **paco@cigital.com**

Interviewed by: **Cameron Philipp-Edmonds**
Email: **cphilippedmonds@sqe.com**

"I went to a STAR conference probably the first time, I want to say, 2006, maybe 2007. I was just really blown away by the level of professionalism and diligence that people were applying to what I as a nontester had always thought, 'Just make sure the thing works.'"

"I work with a lot of folks who seem to have the opinion that security testing is this magical thing and you need to have a pointy hat and half-moon spectacles and a magic wand if you're going to do something productive in security. Actually, it turns out it's pretty far from the truth."

"I'm no chef, but you know that there are some rudiments like making a roux, which is a very basic white sauce, and everybody knows the ingredients to a basic white sauce are just some butter, some flour, some milk, and cook it in a certain way. So many security testing payloads are exactly that."

"I feel like most organizations that are employing a significant number of software testers are actually almost squandering this opportunity to use highly qualified, highly trained people who know what the heck they're doing to actually go look for security issues."

"I'll just say that if you think of a mobile device … A tablet, a phone … If you conceive of that as a Windows XP PC chock-full of malware, then you have the right mindset. That's the kind of security that you need to associate with a mobile device, and a lot of people don't."

> "The biggest thing I need people to do is to stop thinking of security as magic, and think of security as perhaps a special case of what they're already doing."

"I fully intend to come on stage wearing the full-on pointy hat, magic wand, beard, and half-moon spectacles. That's part of the attraction. Yes, to show that in fact, underneath all of the wizarding costumery, there's actually just a regular guy."

Wearable Computing Reaches New Heights

by Mukesh Sharma

THINKSTOCKPHOTOS.COM

Computing devices worn by users have seen a significant surge in market awareness since 2013. The market for wearables is estimated to hit $6 billion by 2018. [1] Forrester Research forecasts that by 2020, wearables will become a core information device for several health care and personal systems. [2]

While wearable devices such as smart watches have existed for some time now, the data processing they could take on was not very sophisticated. Over the past year, wearable computing has become the latest craze, especially for fitness, health care, and entertainment, and it is expected to be the key consumer technology to watch for in 2014. [3] All of these predictions are certainly heading in the right direction, with companies building products fueled by active venture funding. Wearable computing, like mobile computing, is highly universal and versatile in nature in that any discipline can create custom devices and applications to align with customer needs.

Exciting Wearable Devices Galore

Some exciting wearable devices that hit the market or have announced their plans in this past year include Google Glass, sensor-infused socks (HeapSylon), and baby monitors (Sproutling). [4] These devices are able not only to collect user data in real time but also connect to several other synchronized cloud data sources, offering huge potential for enterprises to create a strong bond with end-users. Business Insider predicts dramatic market share growth [5], specifically for wearable gadgets, as shown in figure 1.

Even with these forecasts, there have been considerable online discussions about the common belief that wearable computing has still not gotten the attention it deserves. While there are reasons that can be attributed to this (discussed in the mitigation strategies section below), we are beginning to see wearables more in the limelight in the past year. For example, wearable devices were a major focus earlier this year at CES 2014, with giants such as Intel showing off their prototypes and plans in this space. [6]

The industry is recognizing wearable computing as the next big bet for inventions and innovations to benefit all stakeholders involved—especially the end-user. It is not sufficient for these gadgets to just gather data; they must also interpret the data into meaningful actions and insights for the end-user and synchronize the data across various platforms and displays, including mobile devices.

If this is not done effectively, it does not benefit the end-user and, more importantly, managing such volumes of data becomes overwhelming and expensive.

Developer and Tester Readiness for Wearable Computing

Bluetooth connectivity, mobile, cloud computing, and artificial intelligence all serve as enabling technologies for wearable computing. Product designers and developers have been working through the varied aspects of wearable hardware and software development over the past few years to achieve the best product strategy and implementation practices.

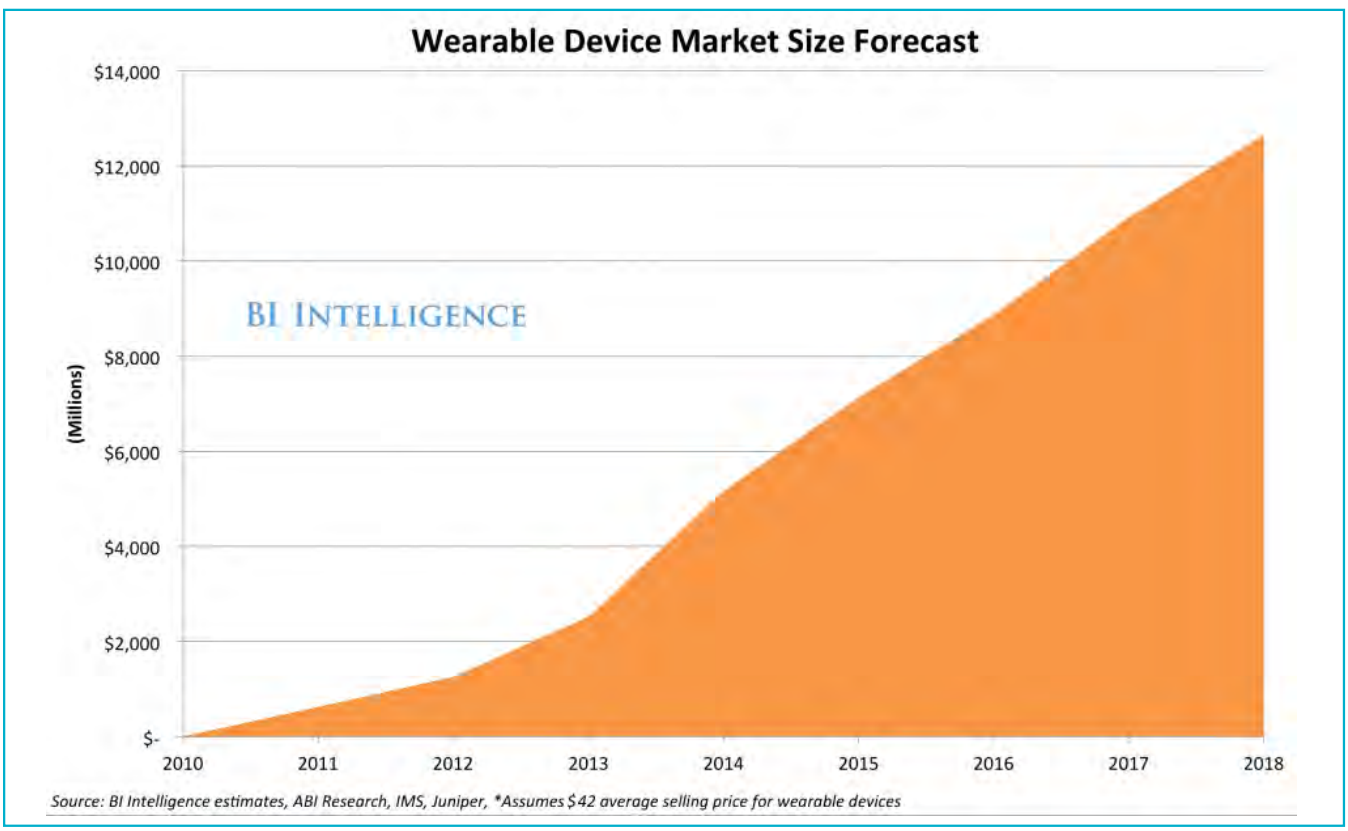


Figure 1: The wearable device market is ready for high-growth market acceptance

Some core elements that need focus include:

- Designing a device that is not just smart but also customizable and context-sensitive to the needs of the user
- Building a strong unique selling proposition given the number of devices that are entering the market
- Focusing more on software apps' richness to enhance end-user appeal
- Exposing APIs to app developers to make the overall packages more engaging for end-users in the future

The last item in the list needs further elaboration. As an example, Google provides a wearables SDK to assist app developers who can now stretch their reach beyond smartphones and tablets. [7] It will take time for developers to gain the right skills for wearables software development. This includes adapting the presentation of information to very small screen sizes, handling some of the existing challenges around complex data and delivering it to users in actionable formats, and developing a suite of supported applications to build an overall holistic experience for users.

Wearable app development is in a similar situation today that mobile app development was in about seven years ago. Google is enabling the industry to embrace this new technology with its open SDK to support developers. With investments from venture capitalists, the right support from some of the larger players already in this space (like Google), and the willingness from developers to take on new and exciting challenges, there is no stopping this wave of innovation getting ready to surge.

Besides just developers' understanding what it takes to power robust and relevant wearable gadgets, software testers also need to build their expertise in this space. [8] What kind of tests are needed, how many of them can be automated, and whether field tests will become more important than in-lab tests are all new considerations for testers in the field of wearables.

Let's Not Forget about Fashion Appeal

So far, we've discussed the market and the development opportunities for wearable products that are being developed, but there's a huge consideration for fashion appeal. At conferences and forums that discuss wearables, one of the challenges discussed is how the devices can equally make a fashion statement to drive mass user appeal. Organizations have already started paying heed to these factors, keeping fashion and aesthetics essential for products they design. As a result, wearable computing may give rise to partnerships with not just the hardware and software makers but also with fashion designers. Sony Ericsson has partnered with the London College of Fashion to hold contests in creating fashion-focused wearable device designs. [9] Not to be outdone, Intel is collaborating with the Council of Fashion Designers of America to design *intelligent* jewelry. [10]

Strategies to Enable Wearables Success

With so much potential that this domain holds, what kind of an influence can this be expected to have for those who wear these computing devices? At this time, there still isn't complete acceptance among end-users in the marketplace. Disbelievers are still working their way through the idea that an external device will be physically placed on them to track various actions. However, the industry expects this to change very soon and to see increased acceptance once users see the benefits and the kinds of results these devices provide.

There are also a few downsides that a user will need to watch for in the wearable computing adoption. Wearables are still relatively expensive. Like any other new technology that is initially pricey, it will take research and development to bring wearables into a more affordable price bracket to help increase overall market share. Some devices are still a little bulky compared to their ancient, non-intelligent counterparts, making it difficult to handle them. These are all typical challenges that the industry would face as any new category of products enters the market. Another problem that the industry will face is user security and privacy. For example, Google had some of these initial hiccups when it introduced Glass and specific places such as bars and restaurants banned its use. [11] Consider the example of an educational institution. Allowing students to enter test areas with their wearables on is certainly going to adversely impact the results of the test. On the other hand, having students remove their wearables and securing them might result in other logistical and security issues.

Despite these challenges, the value of wearables cannot be discounted. To realize the true potential of wearables, these issues will need to be sufficiently addressed.

Balancing between Value and Ignorance

Another important factor to consider is distraction, where the end-user may be preoccupied while sporting wearable gadgets. [12] A simple example is the legal restrictions in the use of a cell phone while driving in the US—these laws primarily are focused on user safety by minimizing distractions. One can only imagine the distraction levels that could result from the scores of wearable devices that may adorn a user in the future. Consider the baby monitor by Sproutling, which will be attached to the baby's body to track her vital signs and help a parent make more informed decisions. While the value can be immense, it can also create a detrimental effect of creating unwanted distractions for the parent with constant worry about how to interpret the information reported by the device.

While the industry gears up and continues to work on building smart devices to make people's lives better, the end-user has a very important role in determining a balance between the benefits that wearables bring and distractions to stay away from. Let's hope that wearables work in a nonintrusive way where "ignorance is bliss" becomes the theme, letting users enjoy life in all its simplicity without completely surrendering to the technology. **{end}**

mukesh@qainfotech.com

Click here to read more at StickyMinds.com.

- References

SOFTWARE DEVELOPMENT IN THE FAST LANE

AGILE DEVELOPMENT CONFERENCE EAST

BETTER SOFTWARE CONFERENCE EAST

REGISTER BY SEPT. 12 USING PROMO CODE **BSME14** AND **SAVE UP TO $600**

GROUPS OF 003+ SAVE BIG!

November 9–14, 2014

ORLANDO, FL | HILTON ORLANDO LAKE BUENA VISTA

Two Conferences in One Location

REGISTER AND ATTEND SESSIONS FROM BOTH EVENTS!

Explore the Full Program at

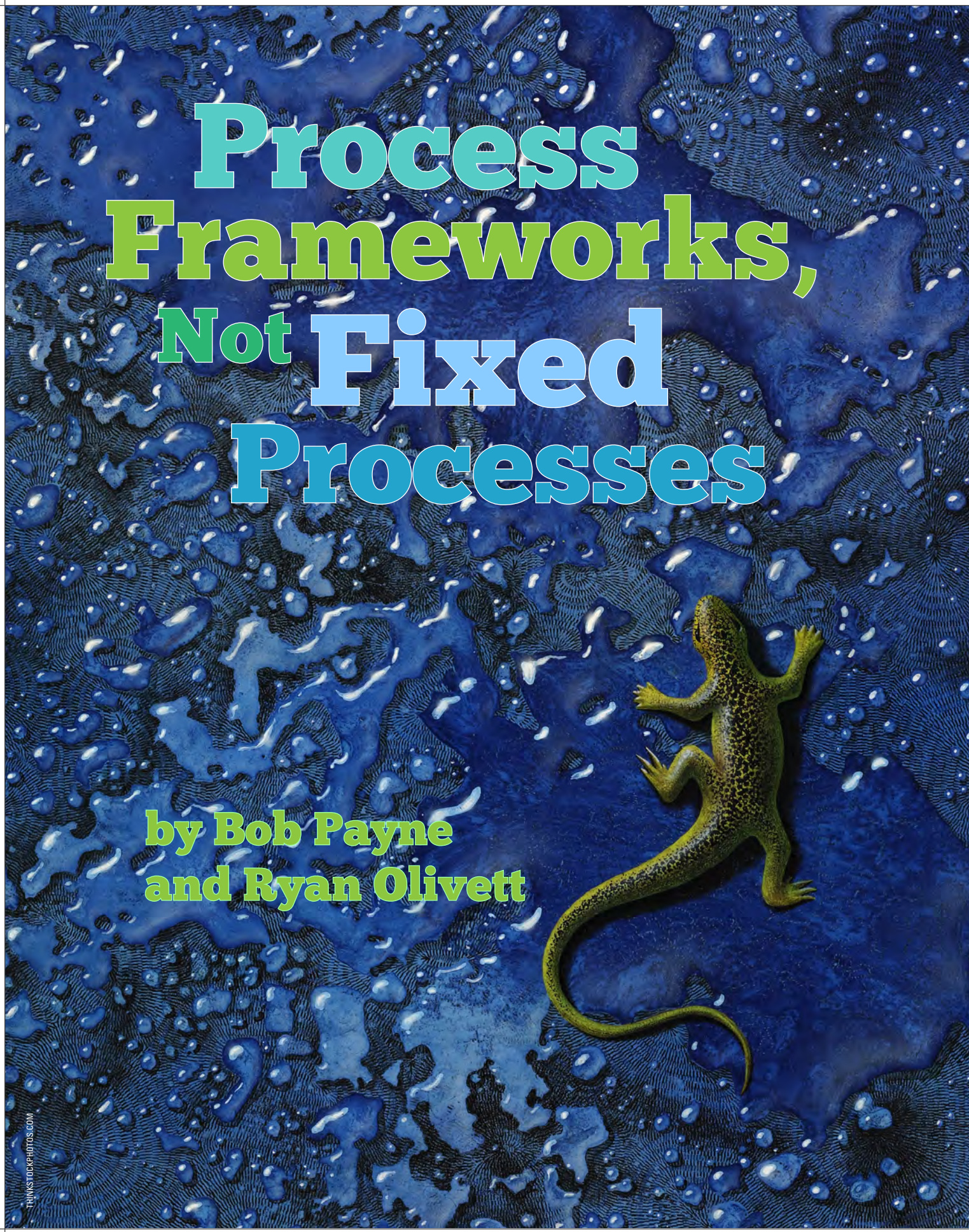
ADC-BSC-EAST.TECHWELL.COM

PMI Project Management Institute REP

PMI® members can earn PDUs at this event

Process Frameworks, Not Fixed Processes

by Bob Payne and Ryan Olivett



THINKSTOCKPHOTOS.COM

Practitioners in the software development industry hold a vast range of opinions on frameworks and processes. Their voices range from pragmatic to dogmatic, from polite to shrill.

An Evolutionary Approach to Delivery

Some practitioners prefer sequential frameworks and processes, such as waterfall, while others prefer iterative and incremental methods found in agile. Because individuals and organizations can be resistant to change and have strong preferences for some frameworks or processes, processes become fixed, reducing organizations' ability to deliver value to customers.

As agile proponents, we find that we must constantly explain our positions within the wide spectrum of beliefs and help others navigate contradictory information and opinions. In this article we attempt to describe what we have observed in the world of process and process frameworks, particularly related to agile methodologies. We also identify some of the main problems we have encountered and what works best to help organizations avoid fixed processes. These best practices should build a case for an evolutionary process of improvement to establish the process that fits your current situation.

Why Should Anyone Care About Process?

Even though we are process wonks, it is not the process or the frameworks themselves that we find interesting: Our fascination lies in another dimension of the topic. Clues can be found in the Agile Manifesto and its principles. The manifesto states that people and interactions should be valued over processes and tools. Further, the principles of the manifesto emphasize, "Our highest priority is to satisfy the customer through early and continuous delivery of valuable software." Ironically enough, many in the agile community argue about processes and frameworks when the main focus should be on how to best deliver value to customers.

Processes exist to enable delivery of value through the creative contributions of the people doing the work—no more and no less. As methodologists, we must concentrate on ensuring we use processes to unleash people's ability to deliver value, or change our processes when they do not. This is difficult in practice, to be sure, but the delivery of value must be our principal concern when we judge the value of any process or framework.

The final principle behind the Agile Manifesto states, "At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly." Why, then, do so many in the agile community argue about which process or process framework is best? It would seem irresponsible to suggest that a one-size-fits-all model will always be successful. Yet, many methodology advocates do just that. We all know that organizational dynamics and problem domains are different between projects and organizations, as they usually change over time. As a result, you may not pick the right process or framework to start with. Inevitably, things will change and another process will have to be used to improve from the first one.

Ending Up with Fixed Processes and Dogmatic Behavior

In the 2007 Rails Edge keynote "Cargo Cults and Angry Monkeys," Dave Thomas described the type of dogmatic behavior and resistance to change seen in the programming and agile communities.

Cargo cult: The term cargo cult refers to the phenomenon observed in some post-WWII Pacific island cultures that started mimicking the behavior of western troops to bring cargo to the island. Some of these cultures went as far as building mock runways and airplanes and holding military drills to bring about the same results they had seen. Organizationally, this refers to dogmatically applying the same processes that worked in one situation in a different situation and expecting the same results.

Angry monkeys: In a 1967 experiment by G. R. Stephenson, rhesus monkeys were all sprayed with water when any of them went for a food reward in the cage. Eventually the monkeys would stop any other monkey from going for the food reward to avoid the watery consequences, and that behavior became the cultural norm in the group. Monkeys were replaced slowly until none of the monkeys in the group had ever been sprayed with water, yet they kept displaying the punishment behavior. In organizations, many processes become fixed just because "That is the way we do things around here."

At the time, Thomas's analogies seemed to be a bit over the top. Unfortunately, we have encountered these two problems over and over again.

People become dogmatic about the process and frameworks they prefer. They believe that because a particular method worked well in one situation, it will work for any project. For some, this is self-serving. Consultants and trainers espouse one brand of agile or another because they profit from training and consulting focused on their areas of expertise. For others, they simply believe that one way is better than another, and they don't consider the organizational dynamics within which people are working before recommending solutions.

People and teams in the agile community also are resistant to change. They become committed to doing what they have always done because it has worked in the past or because they have vested interests in the status quo. Peoples' skills may become obsolete with a new approach to doing business. In some cases, executives in an organization rely on more traditional approaches that negate any attempts to modernize and change. The reasons for resistance to change are endless.

Processes and Process Frameworks

Processes and process frameworks have many definitions depending on whom you ask. Business processes are defined to be activities that are structured to achieve particular business goals. Companies tend to structure these activities in ways they believe most effectively and efficiently achieve business goals. Processes are descriptive and concrete, whereas process frameworks tend to be prescriptive and more abstract. A business process framework refers more broadly to the underlying conceptual structure of business processes and may include guidelines, policies, and values.

Oftentimes processes are embedded within a process framework, and yet those performing the process may not even realize the underlying conceptual framework. The specifics of the process could change but the overall framework would remain the same. Alternatively, if the process does not change over time, it becomes a fixed process.

Tension between Process Framework and Fixed Process

Scrum serves as a great example of a process and process framework to explore. Scrum is the most popular agile method right now. It is fairly simple conceptually, and in many organizations it is wrongly implemented as a fixed process rather than as a process framework outlined in figure 1.

Characteristics:
- Short, timeboxed sprints
- Iterative incremental delivery
- Value delivered as a measure of progress

Meetings:
- Sprint planning
- Daily scrum
- Sprint review
- Sprint retrospective

Artifacts:
- User stories
- Sprint burndown
- Product backlog
- Sprint backlog

Roles:
- Product owner
- Team

Figure 1: Key elements of the Scrum process framework

A team or organization that practices Scrum is using its process framework to define and inform their business processes. Scrum provides a macroprocess that includes meetings, artifacts, roles, and values to enable teams to deliver software functionality. Scrum values help practitioners better understand the underlying conceptual framework and guide them in adding valuable microprocesses into their approach (figure 2).

- Retrospectives
- Program inspect and adapt events
- Quality circles
- A3 process
- Value stream maps
- Lean standard work
- Process flow diagrams

Figure 2: Process continuous improvement tools

The Scrum framework provides a starting recipe and a process for changing that recipe to suit your situation. The sprint retrospective is the tool to help achieve continuous process improvement in the Scrum framework. During a retrospective the team works on improving processes. In fact, the Scrum patterns community collected a series of common modifications to Scrum in the industry. If your Scrum processes and team practices stay the same after several cycles of retrospectives, you are not harnessing the benefits of continuous improvement.

Unfortunately, the recipe does remain the same for many teams. They dogmatically implement the starting recipe and then become resistant to change the process because they believe they would no longer be doing Scrum. It may be valid to ask if your modifications have moved you out of the Scrum framework, but at some point, you should not care. If you are consistently improving your delivery and measuring real outcomes rather than process compliance, you are doing the right thing.

How to Select and Evolve a Process

The best strategy is to pick the best-fitting process based on experience, advice, and company strategy. After getting a process running, periodically experiment with changes to see how that affects your team's ability to deliver value. If the change works well, keep it and improve it. Otherwise, try something different.

To select and evolve a great process, try these steps:

1. Start with a base set of processes.
2. Measure the effectiveness regularly.
3. Experiment with small or large changes.
4. Keep what is working.
5. Change or delete the pieces that do not work.
6. Repeat this periodically.

For learning to be incorporated effectively into the process, run small change experiments to generate different outcomes and then act on the results of those experiments regularly. Lean thinkers and agile practitioners know that processes must change over time to continue delivering value.

When you look at continuous improvement processes initiated with the Plan-Do-Check-Act (PDCA) cycle [1,2], two things are clear:

1. Processes must be continually improved.
2. The best people to understand, suggest, and implement those changes are the people doing the work.

This is true in manufacturing, and it is equally true of the work done in software development.

Understanding the Basis for an Evolutionary Learning Cycle

Lean, in particular, stresses continuous improvement through standard or standardized work that sets a basic framework against which teams and organizations can measure progress. A commonly cited quote from Taiichi Ohno, the founder of the Toyota Production System, puts it this way: "Without

standards, there can be no kaizen." In a sense, Scrum, kanban, and other agile process frameworks should be conceptualized as standard work, and we should expect that the framework and processes will change slightly or substantially over time. Truly lean organizations reward their leadership not for how little their processes have changed but for how many improvements have been adopted.

It's important to note that lean standard work is not just top-down but also bottom-up. Teams are expected to baseline, experiment, and improve processes under their purview, while management is expected to support these efforts while also performing this cycle of improvement at a higher level across teams (and potentially across the organization). It requires real empowerment of teams and the sort of lean-thinking leaders many of us have been talking about for years. It also requires that we measure the progress as delivering working software (or more generically, if we are not talking about a software team, delivering customer value). When teams are empowered to make changes that improve delivery, they also become more engaged with the product they are delivering and the process by which they are delivering that product

As an example, *lean standard work* is an approach to set the baseline for continuous improvement by documenting and understanding the current process. Outcomes are measured when changes are made to that process. In this way, improvement can be verified empirically through improvement in the delivery of real value. Despite its name, standard work does not mean fixed processes—it means quite the opposite.

Conclusion

By implementing a learning cycle based in standard work and periodic retrospectives, individuals and organizations can embrace change when it leads to verifiable improvement and overcome strong preferences and dogmatism for particular frameworks and processes. Agile and other process frameworks should be viewed through the lens of standard work, and they should provide a starting recipe to be changed and optimized over time. **{end}**

bob.payne@lithespeed.com
ryan.olivett@lithespeed.com

Sticky Notes

Click here to read more at StickyMinds.com.

- References

BUILDING MOBILE APPS THE USER CAN TRUST

BY PHILIP LEW

THINKSTOCKPHOTOS.COM

With the advent of smartphones and other mobile platforms, users have quickly shifted from conven-tional desktops to these new, highly sophisticated devices in a variety of shapes and sizes, including wearable computing devices. This rapid migration to the Internet of Things includes many specialized objects such as thermostats, refrigerators, and blood pressure monitors that are connected to the Internet, enabling the user to control them remotely while the devices collect mounds of data. This real-time connectivity has led to software applications that allow the user to track data, analyze trends, and control and configure the device. As intelligent and sophisticated as these resources are, they do pose user experience challenges.

With mobile applications, user experience (UX) becomes even more complex due to mobile's natural characteristics: wide audience use, varied environmental conditions, and a user's personal relationship that may develop with the device. As mobile applications inundate the app stores, mobile user experience is gaining serious attention because some software companies are wondering why their apps are not getting acceptance while others with fewer features gain traction. If a user cannot operate a mobile app in thirty seconds or less, he will most likely uninstall it and search for another.

Some of the key factors that should be considered when designing the mobile user experience need to focus on how the mobile device and its software operate, how the buttons work, and how easy it is to perform everyday tasks. However, there's a key factor missing: Few think of the user experience in terms of trust. Yet trust is one of the most critical components in the user expereince design that will lead to continued and long-term usage of an application. It is just as important as the user's efficiency and effectiveness (which are more related to the pragmatic aspects of user expereince) in using the application. Without it, users could easily become dissatisfied and use another application or subconsciously not use the application again. The bottom line is that if users cannot trust that their applications are secure and come from a trusted source, they will lose confidence not only in the application but also in the service provider. [1]

To begin to evaluate and improve trust, there needs to be a model for organizing and characterizing the concept of trust. In other words, what characteristics do your friends and the people around you have that instill you with a sense of trust? What makes you trust them? Some possible characteristics and behavior include:

- They are prompt and on time (reliable).
- They do what they say they will do (commitment).
- They are straightforward, easy to read, and not complicated (transparent).
- They clearly communicate (no hidden agendas).

When you think about a software application, it should include these same basic requirements in order to instill user trust. Therefore, these same trustworthy human characteristics should be mapped when developing an app. The objective is to design an application with trust in mind as a product quality attribute and then evaluate and improve how well the application fosters user trust.

Developing a Trust Model

It is important to define those characteristics of application product quality that influence trust when the user is actually running the application. ISO 25010 models critical characteristics including security, usability, reliability, availability, safety, and privacy. Although not explicitly stated as trust characteristics, these deserve consideration when modeling trust. Aligning with the ISO 25010 specification for systems and software requirements and evaluation enables us to supplement information quality as another characteristic of product quality that has an influence on trust. [2,3] Additionally, some of the characteristics are recategorized as subcharacteristics in the model. The resulting product quality model for trust can be depicted as a requirements tree, as shown in figure 1.

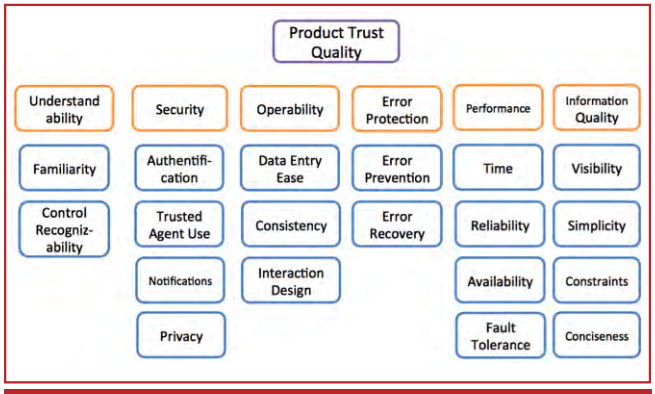


Figure 1: Requirements tree for product quality trust

The goal is to instantiate the product quality model with only those characteristics that influence the user experience from a trust point of view. Each of the characteristics in the model can be decomposed to subcharacteristics and attributes that can be designed, measured, and then evaluated.

Implementing Trust in Mobile Apps

Examining the definitions of the security characteristic and its related subcharacteristics shows that these concepts cover important and common aspects of trust, such as privacy and confidence in using a particular mobile application. Figure 2 shows one potential implementation of the security characteristic: Kaiser Permanente's mobile app displays the entire user name, but Fidelity's displays only a portion of it.

In a recent survey [4], most respondents said they believe privacy is a critical issue when using mobile applications and that there is a need for transparency and choice about what personal information mobile apps and websites collect and share, especially when it comes to advertising and geolocation data. This continues to be a sensitive issue, as mobile device users usually don't know what is collected or how to control it. In some instances, this is actually hidden through unclear language and menu structures and therefore unintentionally approved by users, as shown in figure 3.

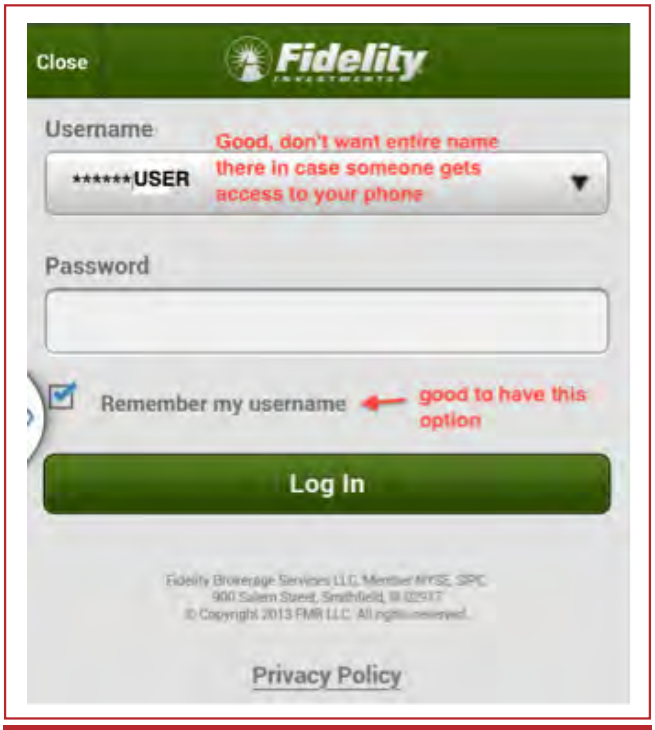


Figure 2: Security and privacy implementation on mobile

Figure 3: An obscured option to limit ad tracking

Additionally, more than one-third of survey respondents identified privacy as their number one concern when using mobile applications, followed by security (26 percent) and identity tracking (19 percent). These trust concerns show a significant need to characterize and evaluate these attributes for mobile applications. Many of the identified subcharacteristics are represented as a high level of abstraction and, as the ISO states, "need further breakdown/tailoring according to the specific needs and objectives." [5]

As a result, there is a need to specify attributes for trust from the product quality perspective. Here are some possible implementations for various security subcharacteristics:

Unique account creation availability: This mechanism is well accepted and should be implemented in all applications with any personal data.

Password retrieval availability: This mechanism enables a user to request a new password provided there is a specified retrieval method, such as an email address.

One-time password or security token availability: One-time passwords expire after a single usage, thereby preventing hackers from attempting to use a password after it has already been used once. Such passwords are usually deployed using tokens, such as hardware dongles from vendors like RSA or software applications that issue a unique password every time. [6]

Implicit authentication availability: Mobile devices, being small in size, pose difficult interaction when it comes to the keypad interface. One study found that 56 percent of smartphone mobile users find password entry annoying and mistype a password at least once in ten attempts. [7] Implicit authentication makes use of a user's behavior model as a unique feature for identification, such as gestures. [8]

Biometric authentication availability (for unlocking device only): This allows the user authentication based on personal physical attributes, including facial and fingerprint recognition.

Terms and conditions: These should be clear and easily available for the users to see, read, and acknowledge.

Some of these characteristics and their implementations change over time, as do user expectations. Take, for example, the terms and conditions written in small print on the back of most of the credit cards. No one reads them because they are written in very small font and are too complex to be understood. For software, most installations require an acceptance of the terms and conditions. Most users check the "I agree" box to complete the installation without reading what they're agreeing to. This is especially true for mobile applications due to small screen sizes and task urgency. Product characteristics such as these would have led to distrust, but because of user expectations today, disclosures are taken for granted as not being trustworthy. Most people assume they are signing their lives away and will give all their data away as well. This is just one example of how the environment and context is constantly changing and influencing product design, the end-user's perception, and user experience in the mobile age.

From Product Trust to End User Trust

The goal is to be able to generate trust while the software app is being used. Borrowing from ISO 25010's concept of product attributes influencing quality, figure 4 shows how to conceptualize this effect.

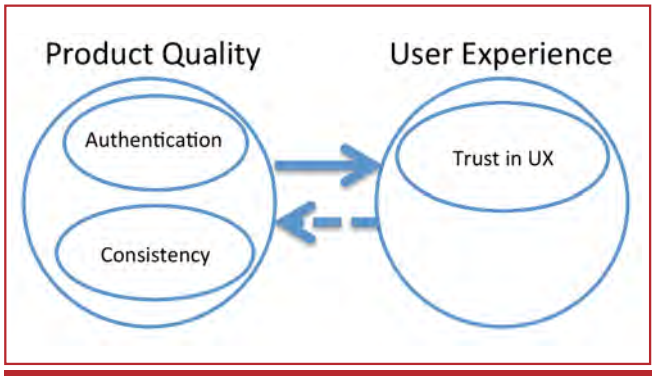


Figure 4: Product quality leading to trust in user experience

Because mobile phones are personal communication devices, there is an extra element involved in using and trusting mobile software applications. If someone asks to borrow your mobile device, what trust does this simple act of kindness imply? If you loan your mobile phone to a person at the airport to call a ride, are you worried he accessed or saw some of your private information and other apps you may have? Ultimately, there is a relationship between an individual and his mobile device that extends far beyond that of a pen or pencil. Therefore, the mobile user experience includes not only privacy when you don't want others to see your data, but also primary ownership when you don't even want others to use your phone. You won't find mobile phone bars similar to Internet cafes, where anyone can rent a mobile phone in a big room sitting down next to someone else.

Such inherent characteristics involving primary ownership and personalization influence satisfaction where trust is a subcharacteristic, as shown in figure 5. [5]

ISO 25010 further defines trust as the "degree to which a user or other stakeholder has confidence that a product or system will behave as intended."

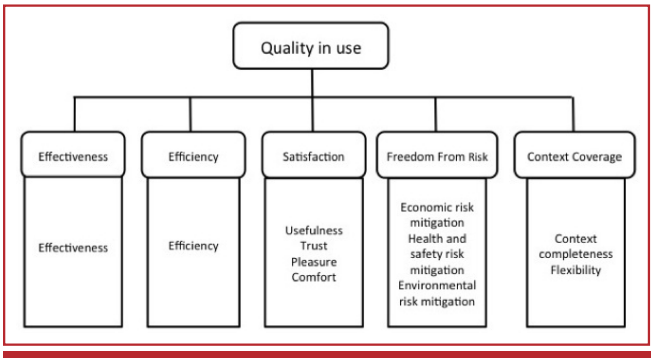


Figure 5: ISO 25010 quality-in-use characteristics

Quality in use and any characteristics must be evaluated with contextual factors of an actual task. In the real world, one of the most critical factors influencing trust is the risk tolerance of the user who is dependent on the task he is trying to execute. For a financial application that requires entering data such as a credit card number, risk tolerance is much lower than for, say, a news application. The actual trust experienced by the end user is dependent not only on the product's trust characteristic implementations, but also on the user in a specific context. Evaluating the relationship between the product characteristics and the user experience through experiments and taking behavioral measurements can be done through tracking specified resulting behaviors and through survey instruments. [9]

Summary and Conclusion

Globalization was catapulted into the mainstream by the Internet in the late 1990s and the first decade of the millennium. Now, mobile devices have become ubiquitous and smartphones and cloud computing are available to anyone, everywhere. As the mobile world continues to expand into wearables and the Internet of Things, we find that our lives, while becoming more convenient, also come with implicit connections and risks. Indeed, we are tied to our mobile phones and their software, and we must trust these ties that bind. Designing, evaluating, and improving an end-user's trust in these mobile apps requires systematic methods. To that end, an adaptable and flexible model borrowing from generally accepted ISO 25010 modeling concepts while instantiating specific product characteristics can be used for evaluation and improvement of trust with mobile applications. **{end}**

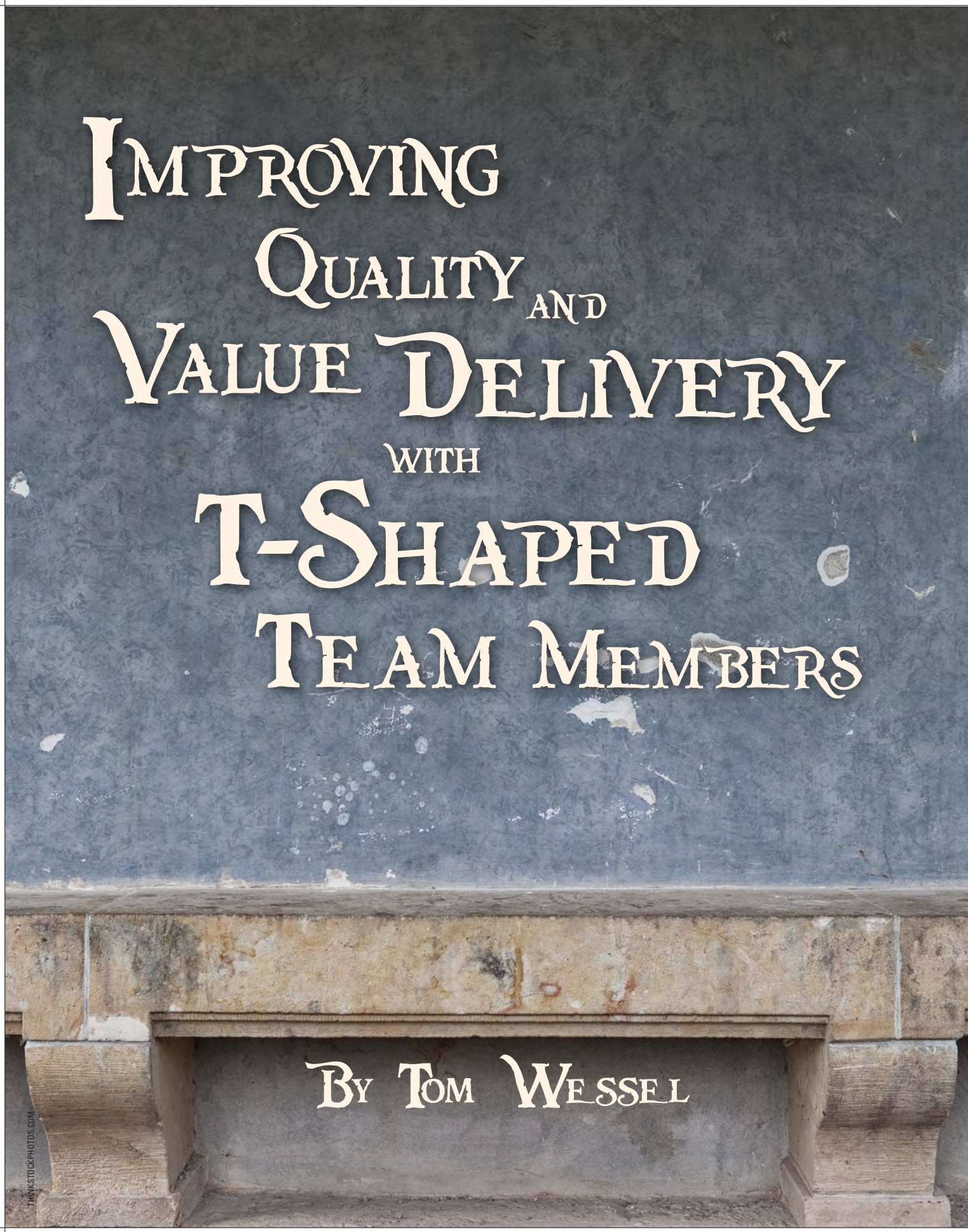
philip.lew@xbosoft.com

Sticky Notes

Click here to read more at StickyMinds.com.

- References

Improving Quality and Value Delivery with T-Shaped Team Members



By Tom Wessel

THINKSTOCKPHOTOS.COM

Customers ultimately define quality of a software solution by determining whether the solution meets their needs. One of the greatest challenges associated with traditional waterfall-like software development approaches is how to improve product validation from the beginning of the development lifecycle so that the software solution quality also delights the customer. In order to increase the likelihood of this outcome, the software development team must take the necessary steps to prevent defects from occurring through the acceleration of the detection and correction testing cycles so that issues are identified and resolved sooner.

An effective means to improve quality is to embrace a whole-team approach to software development centered around each team member's knowledge, ability to collaborate, and specialization. A cross-functional group brings all the necessary disciplines together to deliver a solution to market and they should remain together throughout the software development lifecycle, from beginning to release.

The Need for Cross-Functional Teams

With the emergence of agile as a recommended approach to software development, the cross-functional team is at the core of what makes agile so effective. A cross-functional team is composed of those skill sets necessary to create incremental value for the customer. The skills required from one initiative to the next may vary, but in general most software efforts include the roles of ScrumMaster, business analysts, quality assurance, front-end developers, back-end developers, user experience designers, technical writers, architects, and database administrators.

The power of the cross-functional team approach is that each member of the team is going through the development lifecycle process as a unified group. Collectively, they take on the role of interpreting customer needs by collaborating, discussing, designing, coding, and testing as a cohesive unit. This approach reduces wasteful information handoffs and requirements ambiguity because each participant comes to the table with a different perspective. Each member of the cross-functional team asks different questions in order to create a shared understanding of what the customer really needs. Most importantly, the customer becomes an active participant of this cross-functional effort starting with the initial conversations about requirements and influencing the priority that those requirements are delivered. This helps the team understand the acceptance criteria as the product evolves through working code.

Defining the T-Shaped Skill Set

In order for an agile team to truly become a cross-functional one and achieve high performance, it must be composed of members with T-shaped skills. A team member with a T-shaped skill set is one who possesses deep expertise in a single skill, such as business analysis or front-end development. This makes up the vertical bar of the T in figure 1.

The horizontal bar at the top indicates a member who has a broad knowledge base and represents the ability to collaborate with other roles on the team. The origins of the T-shaped skill set arose from IDEO, a design and innovation consulting firm that realized that hiring employees with I-shaped skill sets (the upright bar in the figure alone) with only specialized skills meant they were not able to collaborate, and it ultimately reduced product innovation.

Team members who take on multiple roles during an iteration or time boxed period of development increase the probability that the team will meet its goals. The classic example is of a QA tester who is falling behind in performing functional testing and can be assisted by a developer.

Evolving to a Pi-Shaped Skill Set

As the team matures, there may be a need to grow each team member's capabilities beyond the T-shaped skill set. The next evolution is to that of a pi-shaped skill set (figure 2).

Figure 1: T-shaped skill set

Figure 2: Pi-shaped skill set

This shape offers a better way to represent team members who embrace change and need to constantly adapt. Rather than possessing a single area of expertise, a team member with a pi-shaped skill set possesses two areas of expertise, such as developer and tool smith, business analyst and UX designer, or QA analyst and test automation engineer.

Team members with multiple areas of expertise provide even greater flexibility to the team, especially in organizations that have limited resources. Team members who are evolving their skill sets to become more T-, pi-, or even comb-shaped—possessing three or more areas of specialization—embody the spirit of agile: that you are never truly done being agile. It is a journey of continuous self-improvement. It is this sequence of small improvements that progresses you toward greater value creation, both personally and as a member of a high-performing team.

How Do We Get There from Here?

Growing team members while they are busy generating value is an ongoing challenge for learning organizations. There are various approaches to developing expertise in more than one area and proficiency in multiple areas.

Identify Skill Gaps

Using Scrum as an example, the ScrumMaster, or servant-leader of the cross-functional Scrum team, should assess the team to determine the skills that are either lacking or could be areas of growth in order to mature and evolve the team to a higher level of performance. Once any gaps are identified, the ScrumMaster should have a conversation with each team member and the functional manager to discuss opportunities for growth.

For example, if the ScrumMaster or Scrum team determines that they need to develop an automated regression suite to automate functional tests at the service layer in order to accelerate development and execution, then the team will require skills in frameworks that support this effort. One or more team members may prove to have the aptitude and desire to pursue this learning and will develop the needed skills to execute this approach.

Develop a Training Plan

The functional manager and team member are responsible for creating the training plan, but ultimately the team member is accountable for its execution. The ScrumMaster provides guidance and an objective assessment of the results of the plan. The plan should be reviewed periodically with the team member so the necessary assessments and adjustments can be made.

The plan itself can be composed of some of the following action items:

- ***Formal training:*** Formal training can take the form of live classroom or online learning sessions. This is a great way to get more in-depth training regarding a skill set, and most incorporate exercises that reinforce the concepts learned. The more hands-on the class is, the better, because most passive learning associated with training is lost if not applied shortly after the class.
- ***Informal training:*** Informal training is a cost-effective approach to share knowledge among members of the same team or across teams. This can be as simple as conducting lunch-and-learn events or participating in a community of practice that is pertinent to a specific skill set needing attention. The Internet can also provide an incredible assortment of valuable, free web seminars the team can view and later discuss.
- ***Pairing:*** Pairing is a powerful way to share knowledge among team members by matching an experienced team member with a less experienced one to work together in learning a new skill. This is a common approach in software development to bring a team member up to speed with a specific area of code. The two work together to either fix a known defect, enhance the code, or refactor it. The key is that the learning is hands-on, resulting in a greater retention of the new skill, instead of using more passive approaches such as code reviews.
- ***Self-study:*** Self-study is a key component of any training effort. Ultimately, it is up to the team member to develop proficiency in a new skill, and the more information the individual seeks out on his own, the more it helps to improve his understanding. There are a plethora of websites, wikis, and books on every subject pertinent to any desired skill set. Self-study should be incorporated into every training plan.
- ***Mentoring:*** Establishing a mentor for the team member can greatly increase the chance of success. The role of the mentor is to assist the team member on his journey because he may be more knowledgeable and accessible than the team member's functional manager.

Regardless of the approach used to develop a new skill, it is imperative that the team member is held accountable for the growth in the defined skill and that his available capacity for project work is adjusted to accommodate the impact of his learning on the team's productivity.

As a manager, growing team members to a T-shaped skill set is a tremendous challenge in and of it itself. Evolving a team member from a T-shaped skill set to a pi-shaped one is not for the faint of heart, and it may not be necessary for all teams or team members. However, in the spirit of agile and kaizen, software development organizations must strive to make incremental improvements in order to remain viable in a world of constant change. This journey of continuous improvement should increase the likelihood that software solutions will not only meet product requirements but also delight your customers.

As a team member, continue to pursue learning within your primary discipline by carving out time to develop proficiency in another discipline within software development. This not only will make you a more valued asset to your team, but also will help you improve within your specialization because you will gain greater context for future roles. The continuous pursuit of learning will help your team and your project and will also make a huge difference to your career. **{end}**

tom.wessel@davisbase.com

Innotas Announces Agile Portfolio Management Solution

Innotas announces a cloud-based agile portfolio management solution that delivers portfolio management across agile projects. Innotas Agile Portfolio Management integrates with a range of agile development tools, such as Atlassian JIRA, giving organizations the flexibility to continue using their existing agile tools for team and project execution, while benefitting from portfolio and program management to meet business objectives.

Innotas Agile Portfolio Management delivers:

- Portfolio Visibility and Planning: Configurable reports and dashboards to analyze development team and project work, such as new features versus maintenance or strategic versus operational.
- Financial and Cost Tracking: Enables management to measure the true cost of agile projects by translating agile terminology to business terms used to run your business.
- Resource Management and Time Tracking: Provides capacity and demand-planning capabilities, including "what-if" scenarios for insight into cross-team dependencies.
- Strategic Alignment: Tracks performance against initiatives and goals. Translates team velocity into predictable release planning and accurate product roadmaps.

Innotas Agile Portfolio Management supports the popular scaled agile framework (SAFe) methodology for scaling enterprise agile projects.

http://www.innotas.com/solutions-agile

ITTIA DB SQL Announces In-Memory Data Management for Embedded Systems

ITTIA announces new capabilities of ITTIA DB SQL that empower embedded systems to store a large volume of data in-memory for processing and analysis.

ITTIA DB SQL offers in-memory data storage that enables developers and decision-makers in sectors such as network appliances, enterprise infrastructure, and healthcare to search for key data points in a very large data set. By leveraging systems with tens or even hundreds of gigabytes of RAM, ITTIA DB SQL can process millions of rows in a matter of seconds. Application developers use SQL to express queries that search for relevant records, aggregate the results, and calculate important statistics with a predictable, real-time response.

http://www.ittia.com/products/examples

QASymphony Unveils 3 Tiers of Software Testing Tools

QASymphony announces a major new update to its qTest test management platform. This update provides extensive integration capabilities through the formal release of its APIs, support for test automation tools such as Selenium and testNG, and features that help simplify management of large global teams.

Along with new capabilities, QASymphony is also offering the qTest test management solution as three separate editions:

- qTest Project is geared for small teams of fewer than ten users working on single projects.
- qTest Pro is for small to mid-size teams looking to upscale and improve test management.
- qTest Enterprise is designed to support many users on multiple projects and integrate with other enterprise-level apps.

The qTest platform provides a collaborative work environment for teams to manage requirements, design test cases, plan test execution, track defects, and generate status and quality-metrics reports. With support for both scripted and exploratory testing, qTest lets you add innovative exploratory to manual and automated test management, creating a consolidated solution that accelerates the testing process to keep pace with today's rapid agile software development.

http://www.qasymphony.com

Zephyr Enterprise Edition 4.6

Zephyr announces Zephyr Enterprise Edition 4.6.

Available for download as well as SaaS, the release introduces productivity capabilities with release/sprint cloning, enhanced access to testing data via the new advanced search mechanism, and better planning tools with time tracking.

Additional features also available in this release include; custom field searches, importing test cases from local machines and a Linux installer.

http://www.getzephyr.com

TestPlant announces eggMan and eggBox

TestPlant announces the launch of two new tools for desktop and mobile test engineers. eggMan and eggBox extend the eggPlant range to include a manual testing tool and an appliance to create an instant testing lab, or testing center of excellence.

eggMan is a free tool for manual test engineers. It allows testers to connect with any system, locally or remotely. It is compatible with eggCloud, which provides secure and scheduled connection to devices anywhere on a private cloud. eggMan includes a free test automation tutorial with access to on-line documentation and videos. eggMan allows automation and manual test engineering teams to collaborate and share systems under test.

eggBox delivers a new iOS Gateway in an appliance, which also contains a switchable option list of eggCloud, eggPlant Manager, and eggIntegration. It is a test lab in a box that works in all operating system environments. eggBox is designed both for desktop and mobile testing and deploys tools available to in-house or offshore test resources.

www.testplant.com

FAQ

expert answers to frequently asked questions

How Does Security Testing Fit in My QA Process?

by Alan Crouch
alan.crouch@coveros.com

End-users of your software application expect security and privacy. As technology becomes more sophisticated, so do hackers who will look to exploit your application for financial, political, or other gains. Many testers are being asked to certify the quality of the application and to ensure that it is defect-free—which for more and more organizations means free of security vulnerabilities.

The most common question I receive from testers who are new to security testing is, "How does security testing fit into my QA process?" While security testing should be a part of your QA process, security is not typically an item on your checklist that is supposed to take place as a definitive part of your software development lifecycle (SDLC). Your responsibilities for security testing aren't just limited to the testing phase. Instead, security is a pervasive, underlying principle to be applied throughout the entire SDLC.

In practice, not every implementation of a security testing plan works for every organization or project, and no single implementation will guarantee a successful outcome. In order to improve your chances for success in catching security issues, your organization should, at a minimum, include the following tasks.

1. ***Define security requirements along with functional and nonfunctional requirements.*** Security requirements should be concise and descriptive enough to test, like any other well-written requirement. A good tester will define acceptance criteria and tests just like he does for an application's functional tests. For example:
   - *SEC-REQ-01:* The system must encrypt all user data while at rest (on the data or file system) or in transit (using HTTPS/SSL) to prevent unauthorized access or disclosure to sensitive information. Acceptance criteria include the following:
     A. No user data (passwords, financial information, personally identifiable information, etc.) is stored in clear text in the database or on the file system.
     B. All data entry forms are submitted over HTTPS. When you define your use cases, don't forget to define abuse cases. Unlike use cases, abuse cases look at how your system's current requirements might be used to attack your users, application, or data. These abuse cases define ways an attacker might try to use the application in unintended ways even if it comes into conflict with other requirements. Abuse cases also allow us to identify architectural defects early.
2. ***Perform automated vulnerability scanning early and often.*** Don't wait until product release to perform vulnerability scanning. Like with any other defect, the earlier you find the vulnerability, the cheaper it is to fix. Perform weekly static code analysis and security scanning along the way and mitigate any issues. If you've implemented continuous integration, like the inclusion of regression testing, adding security testing as another gate to a release should be a relatively easy task.
3. ***Eliminate false positives with manual testing.*** No tool is perfect. Automated scanning tools will identify false positives and won't necessarily find every vulnerability in your application. You'll want to consider the use of manual tests as a necessary part of the process.
4. ***Perform thorough security testing in a production-like environment and provide yourself enough time to fix any security defects before a release.*** System architecture, OS, integration, and implementation can have huge impacts on your application. Often a vulnerability found elsewhere in the system in conjunction with an application vulnerability can be far more damaging than either alone. Only testing in a production-like environment will help identify issues your customers would encounter in the production environment. Testing in development is not sufficient enough to provide you an accurate picture of what is vulnerable once your application is deployed.

While there's no silver bullet for security, and even the most thorough security testing does not guarantee a vulnerability-free application, adopting some of the best practices listed above and conducting a reasonable amount of security testing can reduce the risk of being exploited. **{end}**

How Can You Get More Effective with DevOps?

The promise of DevOps is to provide a basis of collaboration between organizations and IT that produces superior customer value.

by Manoj Khanna | mk@sntio.net

One of the most recent changes in the operations and delivery surrounding IT has been a new awareness of how critical support for ongoing operations is, and how the value chain should continue to improve for the various businesses that IT supports. For large and fair-sized corporations, where the core business is anything but IT, popular opinion suggests that IT is perceived to consume company profits—essentially, a cost center. This has led to a high level of scrutiny in recent years, especially with tighter budgets and shrinking profits. The need for tight control of expenditures in IT development and support to keep the network running smoothly is a growing concern for the CIO and CFO. In effect, this has turned into a struggle between the need to maintain tight operations with minimum expenditures and resources while attempting to maximize support and continuing to produce results.

The industry has seen this trend growing over the past few years, and the introduction of lean and agile methodologies has enabled management teams within these organizations to understand how to get more value when spending less.

Gone are the days of an overabundance of IT jobs where IT staff is populated with more personnel than needed to anticipate possible unforeseen emergency situations caused by unknown variables. Times have changed. Today, IT is leaner and forced to be more efficient by virtue of better software written for the enterprises with enhanced development tools and superior support structures.

Introducing DevOps

Information services teams within organizations have certainly matured over the past fifteen years. The bar has been raised from even just a few years ago. The IT industry is also getting to a point where the overall direction for the work that is produced or developed is for self-consumption and self-sustainability of the organization.

Organizations are getting smarter, workplaces are getting safer, and technologies are becoming clearer with every six-month cycle. The concept of accomplishing more with less is eventually driving organizations toward better, more strategic management of resources and people, being more efficient, and generating high business value while maximizing profits.

DevOps, or development operations, is a term used to define a specialized set of resources and people who supply desired processes of efficiency and agility. This is designed to make organizations and their IT departments smarter and more productive while reducing defects. DevOps assists in generating higher business value for the organization while simultaneously lessening costs. This specialized grouping of resources and people wasn't conceived yesterday—rather, this has always existed within the IT realm of application management and support.

> "DevOps assists in generating higher business value for the organization while simultaneously lessening costs."

So, What Has Changed?

DevOps is simply the result of technology's continual quest to find something new and cool to refer to year after year. Certainly, DevOps sounds trendy and interesting. On a serious note, DevOps is in fact gaining a lot of ground within structured IT management and operations circles. DevOps is not a passing fad; it is here to stay. Although the terminology might change over the years, the underlying integrity benefits will not. The prime purpose of creating a structure around DevOps is quintessential. Organizations don't just see development costs as a benchmark by which to indicate the product's quality, value, and profitability. Now, organizations take the perspective that cost saved is of more value when strategically invested toward the betterment of technologies. This results in operational gains, which promote success in business and help attain more customers while successfully moving forward.

The Benefits of DevOps

DevOps provides a more cooperative, productive partnership between development and operations teams by fostering improved communication and efficiency during critical plan-

ning and development stages. This reduces or eliminates potential costs and problems down the road commonly linked with unforeseen changes. Typically, most personnel involved with DevOps can apply agile and enterprise principles that can result in the successful deployment of DevOps processes.

DevOps focuses on typical key product development issues, such as testing and delivery, while stressing the business value of processes beyond release management, including maintenance updates. This can be accomplished through the adoption of a more iterative method and incremental build model of development. Each milestone is carefully evaluated by the product development teams, analyzed, and modified as needed. Only then does the team continue with the build and, ultimately, deployment. This continuous integration might seem tedious, but these frequent checks and balances actually make the entire deployment process smoother and more effective in the long run, as the need for backtracking and correcting mistakes is minimal.

A more iterative approach for DevOps is less traditional than other methods, emphasizing the importance of a strategic partnering of development team members. This promotes communication among key personnel and allows every team member's critical input to be considered, streamlining the development process even further. Communication and feedback are viewed as essential to reducing production costs and delivering business value, IT stability, and efficiency. This path to more effective communication requires an excellent communication infrastructure in place to ensure that nothing is missed and that no team member is out of the loop. This can appear to be daunting, especially with geographically distributed teams that include a diverse number of resources. This is where the value of change management and release management are demonstrated. It is essential that all team members are aware of what is required of them and their full participation is agreed upon. Some typical communication challenges to be managed can include:

- "I didn't know we were supposed to do that."
- "My team doesn't have the expertise, time, or resources to complete this milestone."
- "Team A didn't communicate to Team B what their requirements were."

Any potential problems can be avoided by carefully managing and facilitating communication among team members so there are no lingering surprises to be uncovered.

Going Forward with DevOps

Ultimately, the broad definition of DevOps is simply a method to foster effective communication and collaboration among development and operations team members about delivering more for less; working smarter, not harder; and doing it quickly. The rise of social media and cloud computing necessitates the rapid, effective deployment of new IT systems and addresses the critical need for fewer maintenance releases while recognizing the unacceptability of the word downtime.

IT developers know the importance of business value, and DevOps helps them accomplish that by delivering faster product solutions, eliminating problems, and introducing added value through reduced costs and network and system stability. In addition to fostering communication and trust between departments, DevOps team members also should learn some new skills—all of which has a positive trickle-down effect and ultimately leaves a significant, positive impact on the organization. **{end}**

index to advertisers

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).

Taking mobility testing to the next level

Led by Capgemini's mobile solutions and testing experts, our global Mobile Testing Center of Excellence is equipped with the latest testing tools and infrastructure from industry leading partners plus proprietary tools and accelerators developed by Capgemini. Our CoE acts as a hub for mobile testing services and brings together people, tools and processes to support mobile testing engagements.

What We Do

The CoE delivers performance, functional, compatibility, usability and security testing for mobile environments. With global penetration of mobile devices at 93% and growing*, firms must ensure the mobile enterprise strategy takes testing into account right from the start. Capgemini uses a wide variety of mobile testing tools and addresses the challenge of platform fragmentation and usability testing through a core set of physical devices covering all major device families and operating systems. This infrastructure is combined with a private cloud solution so mobile testing can be run from any Capgemini location across the globe.

With change a constant in the fast-paced mobile industry, we keep a strong focus on research, development and innovation. As new tools and platforms emerge, we create studies, methodologies, and proofs of concept for new mobile testing tools to spur innovation and meet client challenges. Capgemini supports mobile development and testing for some of the largest financial firms in the world. We have strategic alliances with technology leaders including IBM, Microsoft, Apple, Google and SAP.

*Source: Social, Digital & Mobile Around the World, January 2014, Wearesocial.sg

Need More Information?

Learn more about our five mobile testing services, Mobile Testing Center of Excellence and overall mobile and testing solutions:

- Mobile Testing Services
- Testing Services
- Mobile Solutions

About Capgemini

With more than 130,000 people in 44 countries, Capgemini is one of the world's foremost providers of consulting, technology and outsourcing services. The Group reported 2013 global revenues of EUR 10.1 billion. Together with its clients, Capgemini creates and delivers business and technology solutions that fit their needs and drive the results they want. A deeply multicultural organization, Capgemini has developed its own way of working, the Collaborative Business Experience™, and draws on Rightshore®, its worldwide delivery model.

People matter, results count.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING