

BETTERTM SOFTWARE

A TECHWELL™ PUBLICATION

**DESIGN A TEST
STRATEGY**

Critical tips to create a
quality product

CLOUD DEVELOPMENT PM
Adjusting how you lead for
cloud computing projects



LEVERAGE THE POWER
YOU DON'T KNOW
YOU HAVE



TECHWELL EVENTS

Call *for* Speakers

You are invited to submit a speaking proposal for the upcoming STARWEST 2015 conference based on your experience and insight into software development—the software community wants to hear your story! We want to learn more about your case studies, success stories, innovations, and practical lessons that conference attendees will be able to take back to the office and use in their daily professional actions.



STARWEST

Sept. 27-Oct. 2, 2015
Anaheim, CA

DEADLINE FOR PROPOSALS:
FEBRUARY 8

SPEAKER BONUS

Accepted speakers get a complimentary “Conference Only” registration to the conference at which they are selected to speak (a \$1,995 value).

Topics such as test management, testing techniques, test automation, agile testing, cloud testing, mobile testing, and personal excellence are of particular interest to STAR delegates.

SUBMIT YOUR TALK AT:
<http://vlt.me/speaker>

Upcoming Events



MOBILE
DEV + TEST

April 12–17, 2015
San Diego, CA • USA
Manchester Grand Hyatt
mobiledevtest.techwell.com



May 3–8, 2015
Orlando, FL • USA
Gaylord Palms Resort
stareast.techwell.com



June 7–12, 2015
Las Vegas, NV • USA
Caesars Palace
adc-bsc-west.techwell.com



June 21–25, 2015
Vancouver, BC • Canada
Westin Bayshore
starcanaada.techwell.com

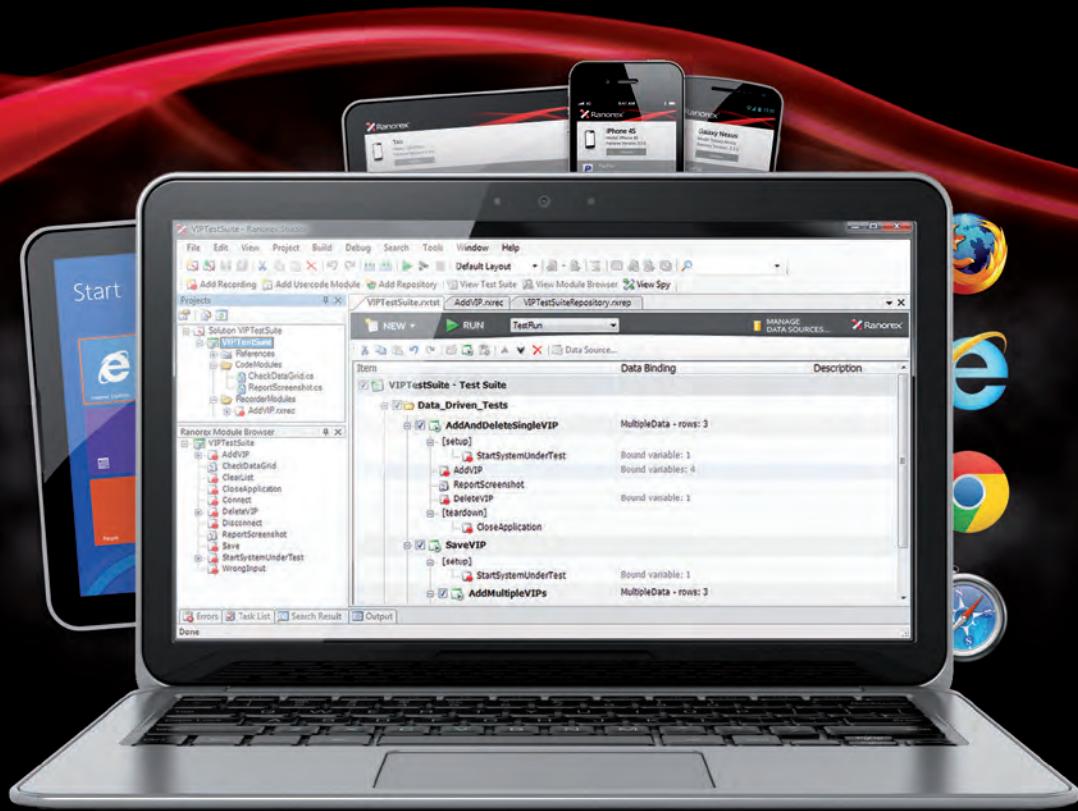


September 27–October 2, 2015
Anaheim, CA • USA
Disneyland Hotel
starwest.techwell.com



November 2015
More Information Coming Soon!
adc-bsc-east.techwell.com

Automated Testing of Desktop. Web. Mobile.



Robust Automation



Seamless Integration



Broad Acceptance



Quick ROI



Why Use Ranorex

www.ranorex.com/why





CONTENTS



16



20

in every issue

Mark Your Calendar	4
Editor's Note	5
Contributors	6
Interview with an Expert	11
TechWell Spotlight	28
Product Announcements	29
FAQ	31
Ad Index	33

Better Software magazine brings you the hands-on, knowledge-building information you need to run smarter projects and deliver better products that win in the marketplace and positively affect the bottom line. Subscribe today at BetterSoftware.com or call 904.278.0524.

features

12

COVER STORY

LEVERAGE THE POWER YOU DON'T KNOW YOU HAVE

Being technically gifted and performing great work may not be enough when you consider a large part of your job is getting others to collaborate with you. Bob McGannon presents some simple ways you can present yourself and your views that can result in greater job satisfaction and career success.

by Bob McGannon

16

PROJECT MANAGEMENT IN CLOUD DEVELOPMENT

Cloud-based software development definitely changes how project managers need to approach their projects and lead their teams. Cloud development is not the same as traditional software product development and requires a unique mix of traditional project management and agility. Project managers considering working on cloud-based projects need to read what Sridhar has to say.

by Sridhar Kethandapatti

20

HOW TO DESIGN A TEST STRATEGY

Speaking from his experience with test-centered design projects, Jon Hagar explores some testing pitfalls that could have been avoided if the right test strategy had been chosen. You won't find a better, easier-to-understand explanation of a practical test strategy.

by Jon Hagar

24

OVERCOMING CHALLENGES THAT IMPACT PRODUCT DELIVERY

Eric Winquist has considerable experience not only leading his company but also consulting with development teams. His advice to "link people to the work" is one of many golden nuggets of information that will help your team deliver projects on time.

by Eric Winquist

columns

7

TECHNICALLY SPEAKING

A TALE OF TWO DOCTORS AND THE VALUE STREAM

Johanna Rothman compares the experience a patient has with doctors to software development. Producing value implies that the customer believes the software you create provides an important benefit by taking into consideration the way a user works along with an exceptional user experience.

by Johanna Rothman

32

THE LAST WORD

ARE YOU DONE OR DONE-DONE?

Probably one of the most frustrating roles a manager has to master is how to know the true status of work being performed. To a developer, completing 80 percent of the work may be good enough, but is it even close to being really done? Masha Nehme shows techniques you can use to verify task completion.

by Masha Nehme

MARK YOUR CALENDAR



software tester certification

<http://www.sqetraining.com/certification>

Foundation-Level Certification

November 9–11, 2014
Orlando, FL

December 2–4, 2014

Phoenix, AZ

agile development training

<http://www.sqetraining.com/agile>

November 9–11, 2014

Orlando, FL

December 2–5, 2014

Austin, TX

conferences

Mobile Dev + Test Conference

<http://mobiledevtest.techwell.com>

April 12–17, 2015

San Diego, CA
Manchester Grand Hyatt

STAREAST

<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

STARCANADA

<http://starcanada.techwell.com>

June 21–25, 2015

Vancouver, BC Canada
Westin Bayshore

STARWEST

<http://starwest.techwell.com>

September 27–October 2, 2015

Anaheim, CA
Disneyland Hotel



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

Marketing Manager

Kim Bryant

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





HAPPY HOLIDAYS!

As we wind up yet another year, let's all give thanks for everything we have: family, friends, and all of the wonderful folks we work with. This issue of *Better Software* is only the eighth I've worked on, and what you probably don't know is there are more people in the Florida headquarters who work tirelessly to deliver a quality magazine to your inbox on schedule every two months.

Here's a quick overview of what goes on behind the scenes. A magazine issue takes about three months of work, from the initial contact with potential authors to receiving the articles, painstakingly collaborating through four stages of editing, laying out the magazine, selecting the art, and polishing up the final product with ads and hyperlinks. The magazine also provides great information from our advertisers, along with advance information about TechWell conferences you'll not want to miss.

I'd like to thank our staff listed on page 4.

This issue has some amazing articles, starting with Bob McGannon's great advice about how you can come out of your shell to work even better in a sometimes stressful project team environment. Sridhar Kethandapatti's article on how to manage a cloud-based application project continues to educate all of us who work in mobile and cloud-based computing. If you weren't sure how to approach testing a software product, you're going to love Jon Hagar's "How to Design a Test Strategy." And, finally, Eric Winquist presents some impactful approaches to overcoming product delivery challenges.

Based on feedback from our readers, online distribution not only is environmentally sound but also allows you to keep all of our issues for reference at your fingertips. I download each issue's PDF to my iPad for viewing in either iBooks or the Kindle app. I also like to keep every issue available for instant viewing on my desktop computer.

We truly value your feedback. Let us and our authors know what you think of the articles by leaving your comments. I sincerely hope you enjoy this issue! And please spread the word about our magazine by tweeting a message with @bettersoftware or #bettersoftware.

Happy holidays, everyone!

Ken Whitaker

A handwritten signature in black ink that reads "Ken Whitaker".

kwhitaker@sqe.com

Twitter: @Software_Maniac

Contributors



ARLEN BANKSTON is an established leader in the application and evolution of agile software development processes such as Scrum, kanban, and Extreme Programming. He is a Lean Six Sigma Master Black Belt, Certified ScrumMaster Trainer, and Certified Scrum Product Owner. Arlen has led the integration of interaction design and usability practices into agile methodologies, presenting and training frequently at industry conferences and to Fortune 100 clients. You can contact Arlen at arlen.bankston@lithespeed.com.



For more than thirty-five years, **JON HAGAR** has been a software tester, thinker, and teacher supporting software product integrity, verification, and validation. Jon works on various standards including ISO, IEEE, and OMG. He consults, publishes, teaches, and mentors regularly and has published the book *Software Test Attacks to Break Mobile and Embedded Devices*. Jon can be reached at embedded@ecentral.com.



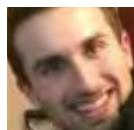
With more than twenty years of IT industry experience, **SRIDHAR KETHANDAPATTI** has worked in a variety of industry verticals focusing on designing and executing effective organizational initiatives. He has diverse experience in IT program management, integration, and enterprise applications implementation. Sridhar can be reached at sriketh16@gmail.com.



BOB MCGANNON is vice president of Mindavation Inc. in the United States and director of Mindavation Pty Ltd in Australia. Both of these consultancy organizations focus on increasing business capabilities through advanced training and coaching. Bob has worked with customers in more than fifteen countries and most recently has created business leadership and project management courses on lynda.com. You may reach him at rmcgannon@mindavation.com.



MASHA NEHME is a project management consultant, freelance writer, and editor based in Austin, Texas. Her approach to helping teams become successful draws on her ten years of experience in managing projects along with her background in journalism and communication. Masha has worked in a variety of organizations across the United States and internationally. She has also served on the board of the PMI Austin chapter. Masha can be reached at masha.nehme@abremod.com.



A longtime freelancer in the tech industry, Josiah Renaudin is now a web content producer and writer for TechWell, StickyMinds, and *Better Software* magazine. Previously, he wrote for popular video game journalism websites such as GameSpot, IGN, and *Paste Magazine*, where he published reviews, interviews, and long-form features. Josiah has been immersed in games since he was young, but more than anything, he enjoys covering the tech industry at large. Josiah can be reached at jrenaudin@sqe.com.



JOHANNA ROTHMAN, known as the “Pragmatic Manager,” provides frank advice for your tough problems. She helps leaders see problems, seize opportunities, and remove impediments. Johanna is working on a book about agile program management. Johanna writes columns for StickyMinds and ProjectManagement.com, writes two blogs on her jrothman.com website, and blogs on createadaptablelife.com. Contact Johanna at jr@jrorthman.com.



ERIC WINQUIST sets the strategic vision for Jama, a company he founded in 2007 after witnessing common frustrations companies face in delivering software products to market. With more than 20 years experience in product development and entrepreneurship, he is passionate about reinventing product delivery in the enterprise by bringing people into the process. Eric's interest and background in the psychology of work and motivation has formed many of the innovations in Jama's collaboration software. Please contact Eric at linkedin.com/in/winquist.

A Tale of Two Doctors and the Value Stream

Have you ever visited a doctor and been completely frustrated? Does your software leave your users just as frustrated and unhappy?

by Johanna Rothman | [jr@jrothman.com](mailto:jrothman.com)

Like some of you, I see more than my fair share of doctors. Some doctors understand what it's like to be a patient. And some, not so much.

Doctor X runs a practice where you go in, pay your copay, and wait. Wait, wait, and wait. First, you see the nurse and provide your health history. Next, you provide your health history to the attending fellow. A fellow is a doctor who has completed a residency but has not practiced medicine alone in this specialty yet and is like an apprentice to an experienced doctor. Finally, you get to meet with the doctor, who was the person you thought you had your appointment with. Doctor X is a great guy, extremely qualified, a super-specialist, and known as one of the "Best of Boston." He is someone you want on your side if you have a tough medical condition.

If you need to see another specialist in Doctor X's office, you need to make an appointment with that specialist. These two specialists are related, but they don't work on the same days. Thus, you have to go on two separate days.

This means double the driving time, additional parking fees, significant aggravation, and lost time from work. And let's not forget the waiting time. It's the same deal for waiting for the other specialist, past your scheduled appointment time.

Let's contrast this with Doctor Y, a different specialist with a different approach to medicine. Doctor Y is also called a "Best in Boston" doctor. She also has a variety of specialists associated with her practice. But she has a different workflow. Oh, you still have to go through similar steps to finally see her: first the nurse, then the fellow, and, finally, the doctor. But if you have to see other specialists, you often see those doctors the same day, at the same appointment. It's like magic. This is known as flow.

Both are great doctors, but the customer experience is dramatically different. With Doctor X, I was frustrated. I measure wait times between doctor visits in weeks or months. I measure wait times with Doctor Y in minutes.

We're familiar with this problem in the software world. Sometimes you have so many projects that you never get to be in flow on anything in one day. You can't finish anything without interruption.

We do this to our users in the software world, too. I bet you've seen user interfaces where you started something but needed information on another page. You couldn't finish everything without interrupting yourself. The user experience moved you out of flow.

At Doctor X's office, they sometimes forget to get my temperature, blood pressure, or other basic patient data until after my appointment is over. We don't need it for my health condition. It's not relevant. But it is for my chart completeness. On one recent visit I had waited for an hour to see the doctor, and on my way out, the nurse asked me to stop so she could take that data. I declined. I explained I was going to miss a client call if I took the time to let her collect that data.

You should have seen the look on her face!

But Doctor Y's office integrates that data gathering with welcoming the patient. It's part of the normal workflow.

I bet you have projects just like Doctor X and Doctor Y. Some projects, like Doctor X, can't get out of their own way. They start, stop, pause, and then continue. You can't establish momentum—you have too many project wait-states. Just as I wonder if I will ever get out of the doctor's office, you may have similar concerns if you will ever get to release your product. Some projects, like Doctor Y, seem to flow without too much trouble. Most projects have challenges, but some teams simply handle unexpected issues a lot easier and without much impact.

What's the distinction?

There are two things that make a difference: understanding that work flows through the people on the project, and what is valuable from the customer's perspective. In our example, understanding the patient's perspective is critical. What are

the most common symptoms that bring patients to visit the doctor? Can you organize the process and procedures so the doctors or specialists are available to minimize separate office visits? (In software, this is the same as understanding the most valuable features.)

That is a *value stream*.

If you know how to recognize your workflow and know what is valuable, your project should avoid rework. In addition, your team spends less time waiting for work. You will

reap another side benefit that is equally important by having less work in progress.

All the wait times I spent at Doctor X's office? That's work in progress. I was in a queue, and I wasn't the only one. Everyone in the queue was as frustrated and angry as I was about having to wait.

You might think we were optimizing the doctor's time. But this doctor is a specialist. This doctor is going to take as much time as he needs. He's not going to feel rushed.

If you feel rushed because you have a ton of work in progress, what might you do? Incur technical debt by not doing such a good job? Create defects? Or do a great job and queue work behind the work you're doing, making everything else wait?

What I've described is working alone as specialists, as in Doctor X's office—which is what many of us do, regardless of our project approach. Contrast that with swarming or mobbing, as in Doctor Y's office.

Swarming is when a team says, "Let's work on a feature together. How can we move this feature across the board?" The team might break into pairs or triads on some piece of the feature and return in about thirty minutes once they complete that piece or task. The goal is always to move the feature across the board.

Mobbing is when the entire team works together to move the feature across the board. Though the team always works together, there is a driver—someone with fingers on the keys. The team tells the driver what to type.

It works for medicine. These approaches can work for software even better.

When you eliminate queues and bottlenecks, you release faster and with higher customer satisfaction because you work on the most valuable work. Can you try it where you work? {end}

All Paths Lead Up



Ready to take your software testing career to the next level? ISTQB Certification will lead the way.

You've proven you have basic software testing knowledge with your ISTQB Foundation Level Certification. Now is the time to take your certification and career to the next level. ISTQB Certification has an advanced career path for every tester. Choose 1, 2 or all 3 ISTQB Advanced Level Certifications:

- Test Manager
- Test Analyst
- Technical Test Analyst

**See the ISTQB Software Tester Roadmap and choose your path to greater career success:
www.astqb.org/advanced**

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

DEVELOP and TEST for the MOBILE FUTURE

MEET *the*
KEYNOTES



Jeff Payne



Jaimee
Newberry



Daniel
Steinberg



Jim McKeith



website now live at

MOBILEDEVTEST.TECHWELL.COM

APRIL 12-17, 2015
SAN DIEGO, CA
Manchester Grand Hyatt

Register by February 13 and save
up to \$400 off your conference

ENLIGHTENED SOFTWARE TESTING STRATEGIES



JUNE 21-25, 2015
VANCOUVER, BC | CANADA
WESTIN BAYSHORE

REGISTER BY
APRIL 24, 2015
AND SAVE UP TO \$300
GROUPS OF 3+ SAVE EVEN MORE

STARCANADA.TECHWELL.COM

Jeff "Cheezy" Morgan

Years in Industry: 10

Email: jeff.morgan@leandog.com

Interviewed by: **Josiah Renaudin**

Email: jrenaudin@sqe.com

"Agile is all about adapting. We know that agile says that we should create this culture of continuous improvement, and that means that whatever the original founders came up with, we're going to continue to improve upon it."

"What has me the most excited [about the future of agile] is that I'm starting to see a reemphasis on technical skills and craftsmanship."

“

I think we have to change the way that teams are treated. We have to change the way that we interact with our customer in order to be agile.

”

"What happens is, if we allow watered-down practices to call themselves agile, those of us who are out there trying to do very serious works using agile ideas, values, and agile principles get tainted along the way as well."

"Agile really is about the people. It is about us, the way that we build software, the way that we craft software, the way we delight our customers."

"For me, agile is the values and principles. If you read the Agile Manifesto, that is what agile is, and that hasn't altered. What has altered is what people have said helped them achieve those things."

"There are a lot of people who are looking for what is next already because they come in discouraged or disillusioned with some of what has happened in the agile community and have separated themselves from it, so they're looking for what is really next."

"I don't believe agile needs to change. I think that we need to make sure that when we relabel something that's agile, we call it agile."

LEVERAGE THE POWER YOU DON'T KNOW YOU HAVE

BY BOB MCGANNON

Many people associate their title or role with the power they possess at work. Although a factor, your role in an organization plays only a small part in the degree of power you can utilize in the work environment. The majority of our power comes from the way we carry ourselves, the integrity of the knowledge we profess to have, and the deliverables we produce that ultimately contribute value to the business.

There are five factors of power that you can master every day if you focus on and leverage them in your regular interactions with your peers, customers, and managers.

Your Voice

Think back to an instance when you received a phone call from someone you did not know. Like most of us, you'd form an opinion about the person from the information you have at hand—specifically, the voice. Does this person sound confident, determined, timid, or simply desperate to make a sale?

Certainly some aspects of our voices are hereditary or dependent on the physical structure of our mouths and throats. However, most of the voice characteristics that leave an impression on others are fully under our control. For example, most people can tell when the person on the other end of the phone connection is smiling. Slight variations in tone and speed of speech usually communicate happiness and confidence.

Being aware of your state of mind and trying to regulate the pace, volume, and tone you are using conveys a lot about your position and power. Understanding how you come across can be extremely important if you have vital information to relay. Sharing that information succinctly, with a tone that is confident but not cocky, demonstrates not only that you have access to data that is important for the business, but also that you are willing and able to convey that information openly. A balanced and confident voice can enhance the power you are perceived to have within the organization.

Posture and Positioning

Picture yourself getting a wonderful piece of art from a five-year-old and congratulating the artist on her beautiful use of colors. When you share your enthusiasm for the work of art, how do you position your body relative to the young artist? Instinctively, you will most likely lean in relatively close, point to the artwork, smile, and share your congratulations. If you stood far back, reached out with the full length of your arms, and did not engage close to the artwork or the artist, you would probably not be perceived as sincere in your enthusiasm for the work.

In contrast to this example, picture yourself being pulled over while driving and having to share your driver's license and auto registration papers with a police officer. How do you typically position your body and pos-

ture in this instance? I suspect you don't lean in, smile, and speak in enthusiastic tones. Rather, you would keep your distance and try not to position your body in any way that may seem provocative.

These two examples demonstrate how we use our body position and posture to convey our comfort or power level when interacting with other people. We want the young artist to see our knowledge and enthusiasm, and we don't want to inspire an undesirable reaction from the police officer.

Applying these concepts to the work environment is straightforward, but it requires you to become aware of your posture and the proximity of your body relative to others you are talking with. Although there are subtleties involved, standing closer to someone conveys confidence and trust, and implies a relationship or a desire to build a working relationship. Standing farther away conveys respect but also can imply a reduced sense of power. How do you position yourself when you are presenting something to a senior manager: closer with confidence, or farther away to show deference? Either way, you are communicating something about your personal power and leaving an impression of your level of confidence and authority.

Eye Contact and Openness

Have you ever had a chat with someone who doesn't look at you during your discussion? That lack of eye contact rarely elicits trust and confidence in what is being shared. At the other end of the spectrum, have you ever had the creepy experience of someone staring at you without blinking? The degree to which you look at someone when you are talking reflects your sincerity and your interest in the reactions of the person you are addressing. You have considerable power in how you are perceived by others simply by focusing on where you direct your eyes.

If you decide to look away from the gaze of a superior and spend time in your discussion looking at the floor or at notes in your hand, you are unlikely to evoke trust in your abilities and the integrity of what you are sharing. Instead, if you glance at your notes periodically and look directly at your superior as you share information, you are likely to convey confidence in yourself and the content of your messages. On the other hand, if you stare, the potential power you would otherwise inspire gets eroded as you draw more attention to your unwavering gaze than to the information you're imparting.

Questions Versus Statements

I once worked for a manager who said his secret weapon was to ask a lot of questions, but never to ask a question he did not know the answer to. Asking questions versus making statements can convey power in a number of ways.

First of all, it is a respectful way to manage other

people and their views. Even when you believe you know the right course of action to take, asking others for their views and perspectives rather than just telling people your plan can be empowering without diminishing your position in the organization. Your questions can both show your desire to confirm what you believe and collect the perspectives of others. A question also can allow you to provide guidance in a subtle but effective way. For example, you could say, "I believe this software module needs to satisfy requirements a, b, and c. Do you agree, or have I overlooked anything? I'd really like your opinion."

The questions you pose in a discussion also can improve a conversation's efficiency and effectiveness. This approach allows your answer to inform and steer the group toward a good conclusion. Alternatively, if new information surfaces, it allows you to propose appropriate actions with updated facts. This is especially true for leading agile meetings, where encouraging total transparency while providing guidance to the team is necessary for a ScrumMaster. Both of these potential outcomes can improve or affirm your positional power in the organization by enabling your influence with facts while actively including the team's feedback.

Content and Context

Of course, having a great deal of focus on your voice, posture, and the mix of questions and statements you make will not enhance your personal power if what you are communicating doesn't connect to the listener.

You have to do your homework. Make sure you have your facts straight, and if your data is limited to a certain business area or applications suite, make that clear during your discussion with the team. You don't want to be dismissed because you have been perceived as making a generalization that undermines your message.

Also ensure you understand the technical capability of the people you are addressing. If you aren't certain, confirm their level of expertise. Avoid the use of jargon and acronyms, and explain terms so your communication is effective to practitioners as well as nontechnical managers. Although it may seem to enhance your power, getting too technical to show off your capabilities actually diminishes your perceived power in the long run.

Tailor your conversation to the organizational level of the person you are addressing. In most organizations there are three different vocabularies you need to be aware of.

1. **Operational:** People at the operational level are interested in process steps, coding standards, and meetings that assist them in what they are supposed to do on a daily basis. People who will respond best to operational language are programmers, business analysts, testers, user interface designers, authors of user documentation, and others who are regularly producing technical deliverables. Communication to operational staff members

should focus on these items and employ the terminology used by the teams who are directly working with these technical components.

2. **Tactical:** People who spend the majority of their time operating at the tactical level are interested in understanding the business and technical trends that impact their departments. They are concerned with how to respond to the demands placed on their technical teams. Managing budgets is also in the realm of tactical managers, as is the estimation process to ensure technical deliverables are produced within cost and schedule guidelines. As a result, communicating with people who have a tactical focus should emphasize things such as training elements, overall business needs, forecasts of demands, and the costs associated with technical proposals.
3. **Strategic:** Architecture elements, new technologies that could be considered over time, and ways to respond to marketing trends or government regulation are topics that are of interest to strategic senior leaders. The vocabulary of strategic management includes return on investment, corporate values, customer satisfaction, and key performance indicators.

What Do I Do Next?

The key to ensuring you use your power effectively is to align your terms and communication focus to the right level. For example, talking about code structures and the format of user documentation to a senior manager who is focusing on customer satisfaction likely will not be received well. Although good user documentation is a piece of overall customer satisfaction, it is very unlikely that the documentation format is an area of primary interest to a strategic level manager. The presence of documentation, the right functionality, agile responsiveness to a rapidly changing business, and skills shortages overshadow the detail of documentation structures.

In Summary

Your position in an organization constitutes a small part of the power you yield in your organization. Focusing on your voice, the way you present and explore facts, your posture and proximity to your conversation partner, using appropriate eye contact, and the vocabulary you use to present your information do more to reflect your power than your position on the corporate organization chart. **{end}**

rmcgannon@mindavation.com

the QUEST to TEST



MAY 3-8, 2015
ORLANDO, FLORIDA
GAYLORD PALMS RESORT



REGISTER BY
MARCH 6, 2015
AND SAVE UP TO \$400
GROUPS OF 3+ SAVE EVEN MORE

STAREAST.TECHWELL.COM

Project Management in Cloud Development

by Sridhar Kethandapatti

In traditional software development, the team follows a project management lifecycle whether it is waterfall, agile, or a combination of both. Following a process helps to reduce the risk so the final product is delivered on time and within budget. By establishing a change management process and configuration management organization, you'll have a better chance at achieving project success.

This has been true of typical desktop software development, but our industry is rapidly changing to just-in-time application updates to the cloud. Cloud-based project management imposes a new set of assumptions and constraints. As a result, there are risks and pitfalls that you'll want to avoid.

Project Management Tools Supporting Cloud Computing

With cloud development, software applications are developed and managed in a virtual space known as the cloud. Cloud development represents a new consumption model where the software is accessed by a user through a web browser instead of the software being installed and executed on a user's computer. This is a different model from desktop, tablet, or phone apps that utilize cloud-based services primarily to exchange data.

Technology companies such as Google, Amazon, and Microsoft are competing to provide their own cloud-based service offerings. As a result of this evolving trend, tools that manage these software projects are now becoming part of cloud solutions offering various project portfolio management (PPM) capabilities.

This leads to the question of whether technology organizations are prepared for the cloud. Do they have better process management to perform the transition from traditional software development and are IT teams ready for the transition?

Having a well-defined and sophisticated cloud-centric PPM will help the client manage and deliver cloud development. A PPM toolkit stored in the cloud will enable all the stakeholders to access the relevant data in real time. Online tool access enables the project to begin quickly with reduced cost during the project lifecycle.

A robust PPM can enable better standardization and continuous improvement of a centrally-located PPM toolkit's services within an enterprise project management office. As a result, any changes or updates can be made available to consumers in an efficient manner. This online deployment capability gives the project manager a much more efficient delivery mechanism. This also introduces widespread customer risk if deployment doesn't go as expected.

Redefining Project Management Roles

Adapting traditional project management may not be the right solution for cloud-based software development, as it involves components that require both traditional and agile methodologies.

A cloud project can encounter several challenges:

- Obtaining security buy-in from the program leadership team to implement the cloud
- Implementing and establishing a proper governance process to manage data security in the cloud environment
- Adhering to vendor protocols so the infrastructure is implemented according to the guidelines and requirements
- Business continuity planning to avoid any failover surprises.
- Assessing proper risks and determining return on investment to cloud sponsors

How the Cloud Benefits Project Management

The benefits of having more effective PPM through cloud computing depends on the capability of organizations to centralize and manage cloud projects that are accessible from a centralized location. By using the cloud, project managers can perform daily functions in the central repository available to all stakeholders. This will significantly reduce the overall time to execute the project management activities.

Agile Methodologies for the Cloud

Not unlike agile methodology used for software product development, each component of a product under development has to be tested and validated in cycles thoroughly until the final product is ready for delivery.

In a rapid-development agile process, customers may have to compromise on certain features that are deemed less important to ensure a timely project completion. Failure to attempt to include every scope, whether important or not, can then cause unnecessary risk of additional cost overrun and can also delay the delivery of the product to the end customer. Figure 1 shows how these processes relate for cloud-based development.

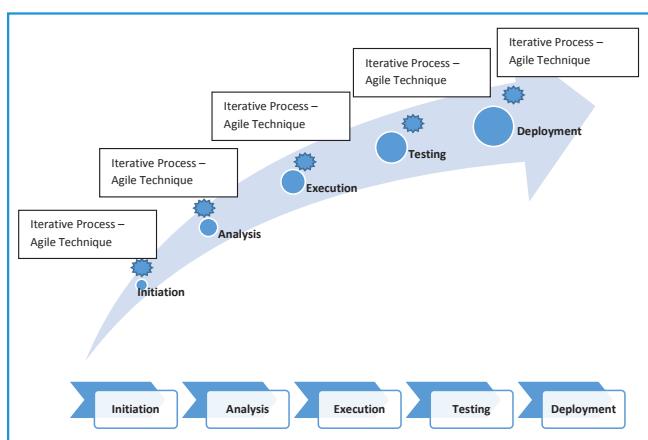


Figure 1: Project management for cloud-based development

For cloud-based solutions, the following project management processes could be implemented to manage better software development as phases.

Identification phase: Engaging the right team and identifying the right team members are critical to the success of cloud development. Similarly, onboarding a technologically savvy project manager is crucial; a non-technical project manager might not be able to identify a risk at the early stage due to a lack of technical understanding.

Scope definition: It is important to break up the scope into several iterations based on the components migrated to the cloud. This includes identification of infrastructure and software application that will be migrated to the cloud.

Analysis phase: This process has to be iterative and is critical for cloud development. During this phase, identification and design of data security methods and the creation of a comprehensive plan to migrate data to the cloud need to be agreed upon for successful cloud development.

Early mitigation of these issues is vital and involves the following steps:

1. Identify the cloud infrastructure risk and address any security constraints.
2. Implement a data security standard and obtain buy-in from project sponsors to ensure they are aware of the components that will be part of the cloud versus custom development.
3. Implement data encryption with strong security features.
4. Update security policies and safeguards so that no private data is ever shared.
5. Determine data to be available on demand in the cloud such as those data elements displayed in the form of a summary dashboard and what should be kept in its database infrastructure.

Build phase: Similar to the analysis phase, build development should be iterative, with the project manager closely monitoring the progress of the development and identifying any early warning of risks to communicate to the stakeholders. The project manager needs to be more vocal, interacting with the technical team to ensure that the proper solution is designed and crafted for cloud development.

Execution phase: This phase is critical for cloud development due to the fact that code may be in constant change until this phase is completed and ready for testing. Agile techniques and the traditional follow-up process should ensure that entry and exit criteria within release management are met for the final product delivery.

Testing phase: Performing automated testing instead of manual testing is better for the final success of the cloud development. Using agile techniques, the cloud

development can be validated to be ready for final integrated testing. Utilizing a niche tool to support the automated testing can ease some of the testing challenges and other dependent risks.

Deployment phase: This is the final handshake on the long-awaited product that will be showcased to the customer during the go-live process. The cloud-based development needs to incorporate configuration management and release management processes so that the product is deployed into production seamlessly.

These processes may have their own challenges specific to each client and, as a result, each process can be used to fit specific client needs.

Cloud development is still in the premature stage, and it has its pros and cons that need to be clearly looked into before implementing the cloud development process, as seen in table 1.

Pros	Cons
Less maintenance from the client as the service will be hosted as part of the cloud service provider	Risk of data security being stored on the vendor's servers
Hosted using the vendor's infrastructure, reducing overhead to the client	In most cases cloud-based software as a service applications are hosted in the cloud, which is not easily reachable by the users as they are located in a different place for the key people to maintain and administer. This introduces latency issues to the environment where the application demands quick response time.
Adapting and converting recently developed applications should be easy, as there are already upgraded capabilities provided by the cloud product vendors.	Customization of large legacy applications to the cloud is not feasible due to older technology and lack of migration to the cloud workspace.
Flexibility for end-users to define their applications due to high availability and that the cloud infrastructure can be built by service providers	Internet bandwidth and speed will be determined based on the service provider. This will potentially cause risk to the project, and mitigation could be a challenge.
Integration with customers' current data is a key feature.	Upgradability is limited to the cloud environment if applications are based on older technology.
Service-level agreements and other support models will be easy to define and be managed by the end-user.	Availability of support team members in the event of cloud failure along with business continuity planning will be difficult.

Table 1: Pros and cons of cloud development

Managing Risks

Risk management plays a vital role in ensuring proper risk governance identifies issues early and to subsequently mitigate those issues. There are several risks that need to be closely monitored and controlled to avoid project failure:

- Risk in determining a private or public cloud to host the service
- Availability of data in the cloud
- Scalability of the application on the cloud
- Infrastructure compatibility to support across the platform in the cloud environment
- Upgrade capability to support new technology trends in the cloud environment
- Lack of technical know-how to configure both private and public clouds
- Availability of large storage space to manage the integration with other applications

Managing Support and Maintenance

The support model and the development maintenance plan need to be identified before the product can be rolled out to end-users. Following the implementation, support and maintenance will need to be provided based on the capabilities of the hosted service. You can decrease the risk by using staff with cloud software development and deployment skills.

Conclusion

It is recommended to use a comprehensive project management approach that addresses a wide number of risks and issues, and can position your team for successful cloud software development and deployment. Early adoption of cloud project management techniques can benefit project portfolio management toolkit standardization with greater program transparency to your team and stakeholders. The benefit of adopting successful internal procedures unique to cloud-based environments should result in effortless deployments with the goal to achieve a happy customer experience. {end}

sriketh16@gmail.com

WANTED! A FEW GREAT WRITERS!

Better Software magazine is always looking for authors interested in getting their thoughts published in a leading online magazine focused in the software development/IT industry. If you are interested in writing articles on one of the following topics, please contact me directly:

- Testing
- Agile methodology
- Project and people management
- Configuration management

I'm looking forward to hearing from you!

Ken Whitaker

Editor, *Better Software* magazine

kwhitaker@sqe.com

How to
**DESIGN A
TEST STRATEGY**
by Jon Hagar

Several years ago, I worked on a test project that had a strategy of verification and validation based on the risks of the system, with specific plans to use models, test attacks, math, requirements checking, inspection, and analysis during a traditional lifecycle development model. It was costly, but then the system involved millions of dollars and even risk to life. As a result, I took it upon myself to learn everything I could about testing, standards, policies, strategies, plans, and doing the actual validation work.

Some of the quality decisions made by the team were perplexing. In one case, the testers found errors in the code, which development said would never happen. A decision was made to not correct the errors. On the third use of the system, one of those errors almost caused the system to crash. The customer was not happy with the developers, and everyone, including the test staff, revisited all the bugs that had been dismissed previously. The team took a “break it” strategy to find new errors before the customer did.

The test group’s renewed focus on test basis and risk mitigation strategy, including multi-prong test approaches, verification, and validation, became quite successful as it continued to evolve over many years of the product’s use and maintenance.

What does test basis, strategy, and having multiple approaches to testing mean? How can any project or tester use test basis and strategy to improve testing?

The Ten-Thousand-Foot View: Strategy and Basis

For starters, consider these definitions:

Strategy: The art of devising or employing plans or strategies toward a goal

Test strategy: Part of the test plan that describes the approach to testing for a specific test project or test subprocess

Test basis: Bodies of knowledge used to drive test planning and design, including techniques and test cases leading to detailed test plans

Project test strategies are the starting point to define the test plans. Test strategies and plans can be governed by organizational test strategies and policies, including regulations and standards. Test strategies are usually defined at a high level (the ten-thousand-foot view) and lead to test plans that are much more detailed (the thousand-foot level). ISO29119-1 mandates that strategies include the test practices used, the test subprocesses to be implemented, the retesting and regression testing to be employed, the test design techniques and corresponding test completion criteria to be used, test data, test environment and testing tool requirements, and expectations for test deliverables. [1]

Test strategies are typically documented as part of the overall test plan.

Defining the Test Strategy

I find test groups use two common strategies, and both may be used together. Testers use them and the supporting test basis and concepts on a consistent basis as if they were the only viable approaches to perform testing. This results in limitations

of test thinking, including missed errors, expensive testing, and testing that takes longer than stakeholders wish it to take.

The first common strategy is verification checking of code to the written requirements.

While this is necessary for many projects from a legal-contractual perspective, by itself it is not sufficient as a complete strategy for testing. In the safety-critical world, for example, verification of the product being developed to written requirements is mandated.

The second most common strategy is done in combination with verification checking by trying to break the software. This second strategy drives plans that define approaches such as ad hoc testing, exploratory testing, or even error guessing. This combination forms a reasonable test strategy for many organizations and can be successful.

Is That All There Is to a Test Strategy?

From my experience, testers don’t consider academic discussions of test concepts, such as test schools or knowledge certifications, relevant to their work. [2] Instead, they want to simply test better. Better can be defined as testing that takes less time, uses less money, or provides more information that stakeholders will value. Testers are looking for something different, yet they keep trying the same two strategies over and over. Einstein’s definition of insanity was doing the same thing over and over and expecting a different outcome.

Are Testers Insane?

Figure 1 shows how test strategy and test basis form the supporting spokes for improving test plans that lead to improved test designs.

Each of these approaches are some different examples of test strategies and test basis that you may wish to consider to improve your testing.

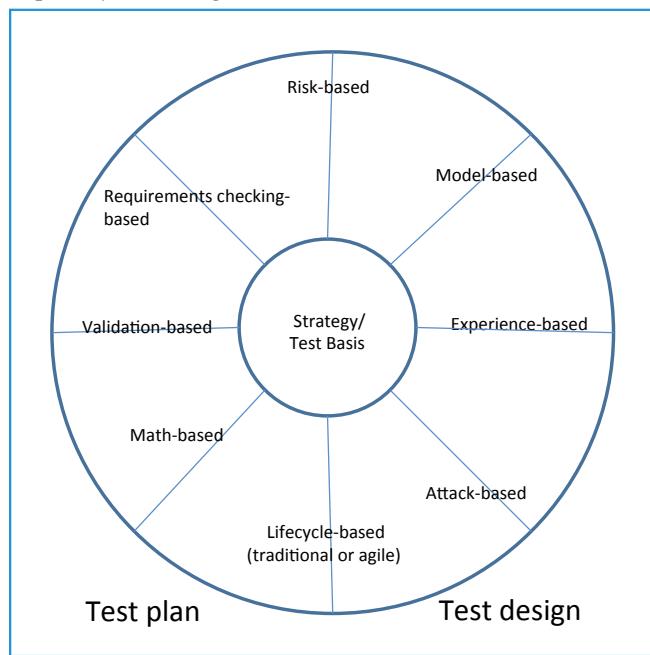


Figure 1: Test strategies and test basis support test plans

These approaches are described below:

- **Risk-based testing:** Testing in which the management, selection, prioritization, and use of testing activities and resources are consciously based on corresponding types and levels of analyzed risk. [2]
- **Model-based testing:** Formal models (not mental models) are used to drive test activities, such as to manage, design, implement, execute, or report testing
- **Agile lifecycle:** These are software development methods in which requirements and implementations evolve, performed by cross-disciplined, self-organized, and collaborative teams. In agile, however, testing is typically practiced by the whole team throughout the project lifecycle.
- **Traditional lifecycle:** This is typically a sequential development process in which activities progress steadily through the phases of proposal, initiation, requirements analysis, design, implementation, testing, production, operations, and maintenance. In traditional lifecycles, testing is often performed by a specialized, independent group near the end of a project lifecycle, usually after integration takes place.
- **Experience-based:** Testing in which the tester's knowledge, skill, and historic practice are used to plan, design, implement, learn, and report testing status. Many testers employ some aspect of experience-based testing, although some use it more extensively. Supporting concepts include error guessing, ad hoc testing, and exploratory testing.
- **Requirements checking:** Testing in which the requirements or other software artifacts are used to verify that the code satisfies these requirements and artifacts. There is often a legal reason to show that contractual requirements are met and to demonstrate compliance with standards or regulatory specifications. Finally, the checking tests are performed and documented in written, scripted test procedures and reports.
- **Attack-based:** The tester attempts to find errors by using patterns (also known as attacks) to break the software. This strategy is closely related to experience-based testing and often includes aspects of risk-based testing. Next to verification checking and experience-based testing, it is likely the most practiced strategy.
- **Mathematics-based:** Testing in which tests are planned, designed, implemented, and analyzed based on mathematical concepts and techniques. These tests include statistical, design of experiments, formal proofs, combinatorial, random, fuzzing, and domain (set theory such as equivalence classes and boundary value analysis).
- **Validation:** Testing in which testers try to show that the development efforts have created the right product that meets user and customer needs. Products being validated can include operational concepts, requirements, design, models, and implementations, as any of these can have errors. Assessing the validity of a product or artifact of development can be very difficult and subjective. Many projects avoid this strategy.

Comprehensive project test strategies will combine several of these strategies. You may incorporate other strategies not in this list that are specific to your situation.

Selecting the Best Strategy

There is no single best starting point for developing a test strategy except to consider a project at a point in time. A tester must be agile even when performing traditional testing to select a good mix of strategies to enter into a plan. Further, once a strategy and test plan have been defined, testers should expect them to change—although detailed plans will likely change more than test strategies.

Consider a checklist I use for test strategy selection:

1. Focus on context. Context will include things such as cost, schedule, product nature, organizational policy, regulations and standard, test team skill set, and customer and stakeholder expectations.
2. Consider how strategies can be mixed and matched for an optimal mix given the context you are focused on.
3. Select those strategies that offer the most effective and optimized approach for your testing. Some strategies that result in effective testing may be difficult and time-consuming to set up and implement.
4. Finally, how does the project development lifecycle of the product impact the strategy? For example, a new product will employ a different strategy from one used for an existing product that requires only maintenance testing.

Final Recommendations

I approach a test effort by first thinking about what my strategy and test basis will be. I do not want to be trapped by thinking there is a single best or most familiar strategy to use. The strategy and basis will flow into the test plans and then into the test design and implementation. Incorrectly limiting strategy and basis can limit my test results, and product success may suffer. Good testers should balance and optimize the test strategy and basis going into planning.

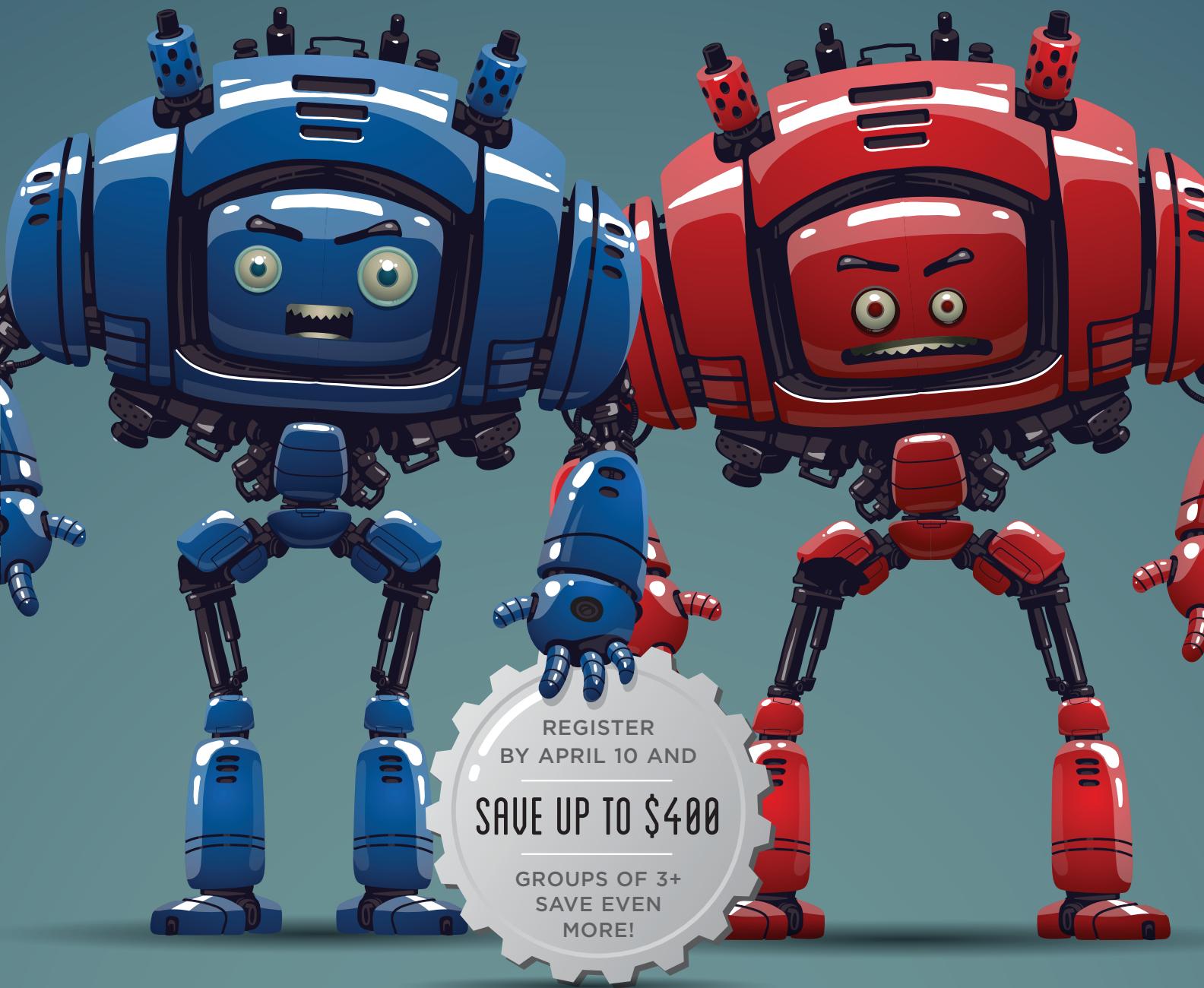
Testers should practice more strategic thinking as well as the bigger picture about testing approaches before jumping into detailed test plans and test design. When appropriate, incorporate strategies for checking and experience-based software testing as starting points. Testers should work on mastering when to apply other strategic approaches and bases to both large and small test efforts. Testers who master the skills of applying different strategies and bases will find new career doors opening because they will be able to provide a broader variety of useful information and quality results to stakeholders. **{end}**

embedded@ecentral.com



Click here to read more at StickyMinds.com.

■ References



REGISTER
BY APRIL 10 AND

SAVE UP TO \$400

GROUPS OF 3+
SAVE EVEN
MORE!

TWO CONFERENCES
IN ONE LOCATION

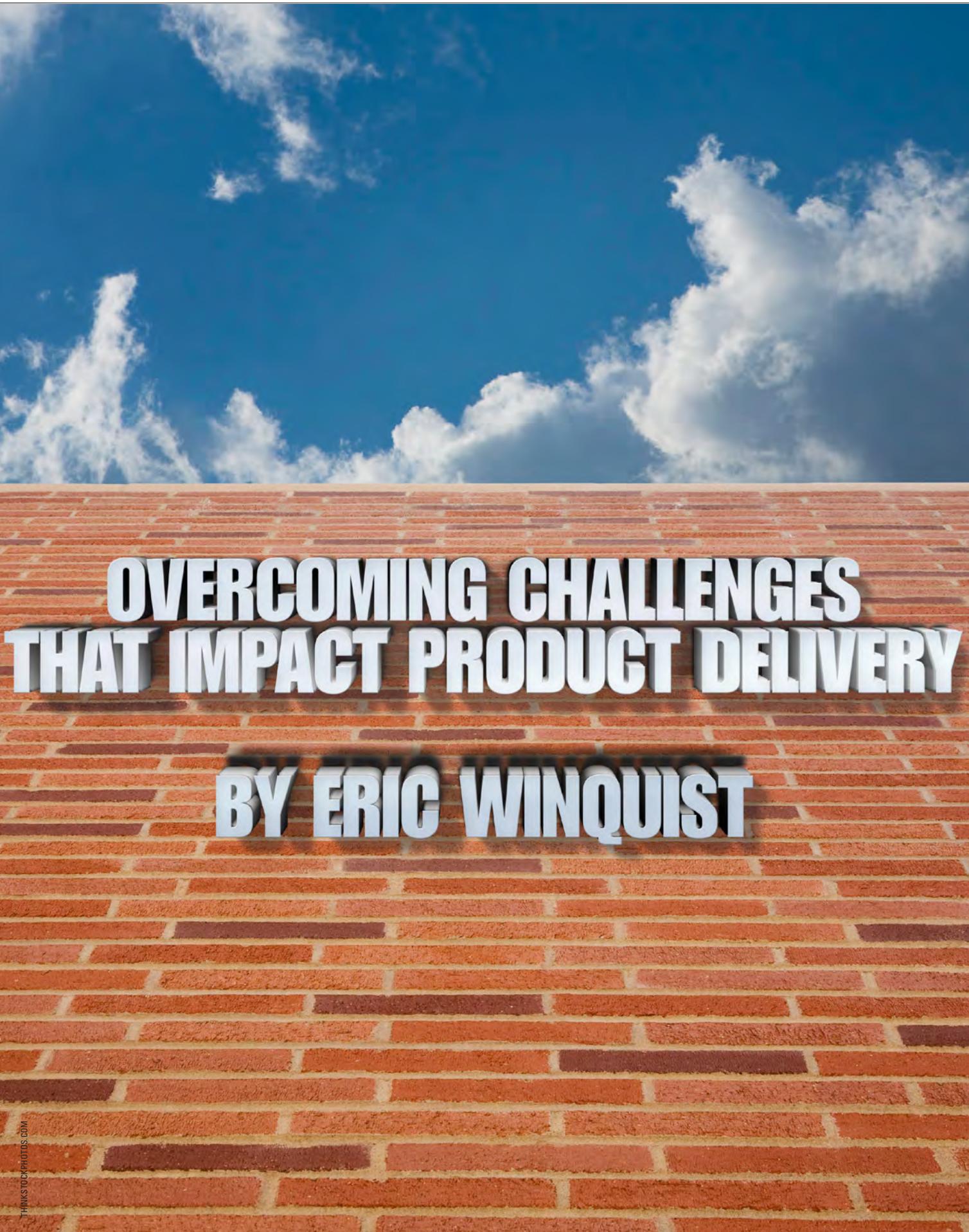
REGISTER AND
ATTEND SESSIONS
FROM BOTH EVENTS!

June 7-12, 2015
LAS VEGAS, NV | CAESARS PALACE

Explore the Full Program at
ADC-BSC-WEST.TECHWELL.COM

 Project
Management
Institute
SM R.E.P.

PMI® members can earn
PDUs at this event



OVERCOMING CHALLENGES THAT IMPACT PRODUCT DELIVERY

BY ERIC WINQUIST

Although this may sound like a cliché, it's a fact: Change is the only constant in today's technology-driven economy. The rapid development of various hardware devices used in the home, in the workplace, and for personal engagement requires equally rapid and robust software development to support these devices. Based on a changing market and customer expectations, the software needs to be updated in real time. According to a May 2013 report on disruptive technologies from the McKinsey Global Institute, the number of connected machine-to-machine devices has increased 300 percent since 2008. [1] As software becomes more and more embedded into technology, the rate of innovation accelerates.

Our industry is now firmly in what has been called the age of the customer, when the availability of information and ease of communications have shifted much of the power from the seller to the customer. For technology organizations to stay competitive, they must constantly innovate. Product delivery is critical to success, and effective execution provides a significant competitive advantage to become a market leader.

Executives are increasingly elevating the strategic importance of products as a key foundation to providing business value. Product delivery is cross-functional, spanning executive management, sales, marketing, service, support, and operations in addition to the traditional product management, development, quality assurance, and release management functions.

The margins of operation in many areas of business are tight, with little to no room for error. Maintaining quality and maximizing return on investment are key business drivers, yet shorter product development cycles require new manufacturing technologies and techniques to balance the demands for functionality and robustness. The aggressive timelines for product delivery bring a set of challenges throughout the supply chain as product developers face relentless demands for innovation.

This new economic reality means that most companies across all industries are becoming technology companies. However, many organizations are having a difficult time keeping up with the rapidly accelerating pace of change, especially when their teams are working in silos using outmoded systems.

Obstacles to Efficient Product Delivery

Product delivery is plagued by preventable delays. In a research project conducted by Forrester Consulting on behalf of Jama, [2] senior business and IT professionals at enterprise organizations with annual revenues of more than \$1 billion said that "unclear or changing requirements" was the most common reason for product delays. Delays also result from misalignment when multiple teams, across functional groups and spread around the globe, exacerbate communication challenges. Forrester identified five factors as the major contributors to product delays:

1. Product teams often lack a clear understanding of customer needs. Unclear or changing requirements impact product delivery. Not being able to get timely feedback on possible solutions results in delays and wasted time, effort, and money.

2. Conflicting priorities caused by stakeholder disagreements put product delivery teams in an unfortunate bind. Lack of clarity about objectives, assumptions, and possible solutions leads to a lack of focus. In particular, priorities between "must-have" and "nice-to-have" activities can make it appear that there is no priority at all.
3. Effective collaboration spans roles, teams, and geographies. Modern products are complex, requiring a wide variety of expertise to deliver successfully. The reality of the global marketplace means that collocated development is rare; globally-distributed teams are increasingly commonplace.
4. Unnecessary handoffs and delayed decisions reduce speed and impair quality. Rapid delivery is an increasingly competitive differentiator. Any delay in making decisions or obtaining feedback needlessly hampers product delivery.
5. Delivering winning products requires unprecedented collaboration across diverse roles, spanning the organization from executives to operations and from marketing to quality assurance.

The Advent of Modern Product Delivery

We're looking at a paradigm shift in the approach to the product development cycle that is fully collaborative, with live data accessible to all teams at all times. The software used here is the key differentiator and an enabler. There's an old way and a new way. We call this transformation modern product delivery.

Successful product delivery requires you to implement a product-centric mentality, focused on the outcome and delivery of customer value. We also frame this as a victim/victor perspective. If you see your company as a victim to disruption and change, you probably will not succeed. Our customers in traditional industries saw disruption not as a threat, but as an opportunity. They embrace change rather than manage it.

Modern workforces collaborate and communicate across departments and teams, they understand customer needs, and they work together throughout the entire process.

The old way of sharing documents via email attachments and hosting meetings to discuss decisions doesn't work when you need to move fast. Decision making needs to happen in real time with everyone impacted by these decisions informed immediately, even if they are mobile or work in remote offices.

Table 1 summarizes the features of the old way versus the new way of approaching product development.

Old Way	New Way
Focus on work	Focus on outcomes
Control change	Embrace change
Victim mentality	Victor mentality
Siloed communication	Collaboration in context
Share only in meetings	Share early and often
Duplicate efforts	Reuse base IP

Table 1: Comparison of the old way versus the new way of approaching product development

We need to redefine product delivery to stay relevant, stay innovative, and remain profitable. We are going to need to change the way we work, find ways to be faster, and manage complexity to innovate and better understand customers. The good news is that there are things product design teams can focus on and start doing to evolve their processes.

1. **Define the why.** In addition to defining all the features and functions, take time to understand, define, and share the why within your organization. Share why you prioritized this project and put the other one on hold. Why do your customers need this feature—what problem does it solve for them? Most importantly, define the business outcomes you are hoping to achieve.

Team members want to understand and feel connected to the outcomes and bigger picture. When they understand, they get excited and are more likely to put in the extra effort. When people are invested and committed to the project, they're less likely to give up. Give your team the context to make better decisions with less churn. Sure, you will always have missing information and numerous questions, but if your product team understands the why and the outcomes you've identified shooting for, they will make much better decisions when they need to fill in the gaps. When you define the why, it makes you faster.

2. **Focus on core business value.** For every initiative, find and hold true to the core business value. By core business value, I mean the absolute, essential set of deliverables that is necessary to fully realize the idea, and no more. This may sound like minimal viable product because that's associated with the first iteration of a product. But the problem is that minimal viable product is often associated with just the features, not the business value. What is the core business value the release offers customers? We want something more meaningful for our customer than "It was functional." We want to provide a better business outcome.

Embrace agile and iterative processes, but approach development with a strong viewpoint of what success looks like based on feedback loops—customer interviews, value testing, and design reviews. Build with a strong view of what outcomes the product should deliver to your customers.

Product development is all about negotiations and trade-offs. You start with this beautiful idea of what you want to build, but the reality of what ultimately comes out at the end will require decisions and compromise. If you and your team have a clear understanding of the core business value of the product, you can cut through discussions, trust that team members will make the right decisions, and navigate through the technical complexities to obtain the business outcomes.

3. **Reuse.** When you are ready to start building, look around and see what you can reuse rather than reinvent. It is not uncommon for teams within large companies to recreate the wheel over and over and ultimately restart projects from scratch. If you can figure out ways to start sharing and reusing information, you will be faster and can deal with product complexity.

Different versions of a product often share 80 percent to 95 percent of the same intellectual property. Industry-leading companies are finding ways to reuse their intellectual property at many levels. Code reuse happened first, but now we see customers reusing design artifacts, specifications, test cases, content for data sheets, and process information.

A customer of ours works with satellites and found that 70 percent of the planning doesn't change from launch to launch. They now reuse all the launch manifest information, update with any new information they learn, and improve every time. This frees them up to focus their energy on the features and services that differentiate them: the 30 percent that's customized for each launch.

Similarly, a semiconductor customer decreased cycle times by creating a library of specifications that different teams developing multivariant product lines reuse, resulting in eighteen times the efficiency gains.

4. **Review early and often.** Reviewing early and often with a broad set of stakeholders can easily become the fastest path to dramatic improvements in quality and speed. Move away from daylong meetings to go over requirements line by line. You need to be nimble enough to incorporate feedback, get approval, and communicate the impact of change in real time—switch to lightweight, iterative reviews to engage your team and stakeholders.

Draw stakeholders in the project lifecycle early, then transition to value testing, then to design reviews, then to more formal requirements reviews. The key is to iterate and split work into manageable chunks.

5. **Rethink change.** Change was a completely different beast twenty-five years ago when software products were built using the same traditional methodology we used to build buildings. New ideas, information, and change requests came in the form of memos and faxes. We read about the market and competitors in monthly magazines and newspapers. Information was slow and infrequent.

Today, there is a constant, never-ending flow of new information feeding into your product team. The question is: How can you and your team take advantage of this flow? How can you disrupt your market by harnessing feedback loops?

You need to adopt modern tools that are inherently social so your team can share and discuss new information in the context of their work and in real time. The user experience matters—today's workers need easy-to-use tools that mirror what they use in their personal lives.

Empower team members to make decisions. Take time to define clear decision-making responsibilities, but push decision making as far and wide as possible. Find ways to make faster decisions than waiting for a bi-weekly status meeting or change control board.

To ensure that the best decisions are being made, link people to the work. Good decisions need to consider context or situational awareness, an understanding of possible impacts, and a way to receive input from others.

If you use traceability on your projects, rethink how you can use that information. Traceability is much more than making sure you have a test case for every requirement. Rather, it's a guide that identifies who is connected to whom at any point in the project, who is impacted, and who should be brought together to make key decisions. It also connects you to the why. It's a map that shows you who the people are who best understand or define the core business value. By providing context, strong relationships, and understanding of the why, your teams will be able to react to the new information more effectively to produce better outcomes.

Going Forward

People are wary of change. Many organizations are still stuck in the old way of doing things by focusing on a project's tasks while looking for ways to stop or manage change. Progress can be slow. Most companies still email documents as attachments, keeping information locked up and hard to find, and in most organizational cultures, meetings are the forum for decision making.

To survive, technology companies need to transform business by shifting to the new way of working. Embrace change instead of thinking about how to manage or contain change. Empower people to make decisions based on the new information they receive every day. To make sure your product isn't straying from your core business value, bring in stakeholders and provide context so that everyone understands what they're building and why.

Modern product delivery can help you fulfill customer satisfaction and think differently about how you approach the issue. The bottom line is that product delivery is a business issue—it cannot live just in the dark caves of the development world but instead needs to be treated as a strategic priority.

Change is taking place in the enterprise, with customers transforming their businesses to innovate and out-compete. You can transform your business by rethinking how you ap-

proach product delivery. If you recognize that product delivery is a business process, products will not get built faster or better by throwing more technology resources at the development and product teams. You need to keep the customer at the forefront of every decision, and elevate product delivery as a top five strategic initiative. At the end of the day, your business builds value from what your team delivers to market. There is nothing more important than doing this right. [\(end\)](#)

[linkedin.com/in/winquist](https://www.linkedin.com/in/winquist)



[Click here](#) to read more at StickyMinds.com.

■ References

NEWSLETTERS FOR EVERY NEED!

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

- **AgileConnection To Go** covers all things agile.
- **CMCrossroads To Go** is a weekly look at featured configuration management content.
- **StickyMinds To Go** sends you a weekly listing of all the new testing articles added to StickyMinds.
- And, last but not least, **TechWell To Go** features updates on the curated software development stories that appear each weekday at TechWell.com.

Visit StickyMinds.com, AgileConnection.com,

CMCrossroads.com, or TechWell.com to sign up for our weekly newsletters.

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

How Can I Create Good Definitions?

Conrad Fujimoto

There is a necessity for a common understanding when it comes to software development. It is vital that everyone—including stakeholders, business analysts, developers, and quality assurance—communicates properly if we are to efficiently build the software applications required to meet the needs of our organizations.

Nowhere is this truer than in the area of requirements. We know good requirements have certain characteristics; two of these are that they be clear and unambiguous.

<https://well.tc/N72>

Internet Explorer Updates Help Attract Developers and Consumers

Josiah Renaudin

For the longest time, it was almost assumed that you were viewing your favorite websites on Internet Explorer. Clicking on that big blue “e” sitting on your taskbar was just a part of the daily routine. But over time, bigger and better alternatives arrived.

Firefox and Safari are enjoyed by millions of consumers, and Google Chrome has just recently overtaken Internet Explorer as the most-used browser in the United States. Competitors have continued to show initiative to innovate while Microsoft seemingly rested on its laurels. But a fresh update for Internet Explorer 11 has patched twenty-five of the browser’s key vulnerabilities and even added new tools to entice additional developers.

<https://well.tc/NdQ>

Applying Emotional Intelligence to Your Testing

Thomas McCoy

Is software testing an emotional activity? I’m not talking about encountering code that makes us want to cry, but the range of emotions we feel when undertaking our role. What “emotional intelligence” tries to teach us is how to identify and respond to those emotions in the most productive way.

The concept was popularized by Harvard psychologist and author Daniel Goleman through his 1995 best-seller, *Emotional Intelligence: Why It Can Matter More Than IQ*, which has much to offer the testing profession. He argues that the first thing we need to do is listen to our emotions without letting the “amygdala hijack” take control.

<https://well.tc/NhN>

Successful Performance Testing Begins at Requirements

Dale Perry

Traditionally, performance testing occurs at or near the end of a project. Attempting to do all that is required in a performance test at the end of the development and functional testing process rarely, if ever, succeeds.

This approach is based on the idea that a complete system or application must exist in order to execute a performance test. While stability of the software is an essential aspect of performance testing, “stability of the software” does not necessarily refer to the entire system.

<https://well.tc/N6r>

How to Work with Defensive People

Naomi Karten

It’s irritating to work with people who are always defensive. These are the people who, when things don’t work out as planned, are quick to blame circumstances, bad luck, or the people they work with. They take comments seriously, create excuses for things that have gone wrong, and are skillful at repackaging their goofs and glitches to avoid being found at fault. In a sense, they react as if preparing to fend off an attack.

So if a coworker asks a chronically defensive programmer how he’s doing and the programmer knows he’s way behind, he might lash out, find fault with the coworker, or point to problems elsewhere in the project. And he’s likely to stop listening to anything else the coworker says.

<https://well.tc/Nhe>

Turning to the Turing Test for Agile

Cameron Philipp-Edmonds

What if there were some kind of a Turing Test for agile teams? A test that could separate those who are just mechanically following the process and those who are emoting the agile essence?

Just as the Turing Test asks for more than 30 percent of unbiased participants to unknowingly mistake a computer interaction for one with a human, shouldn’t at least 30 percent of outsiders be able to look in on an agile team and think they are profoundly agile?

<https://well.tc/N4r>

Glogster iPad App Gives Learners a Key to Worldwide Knowledge

The Glogster iPad application gives users a library of top-quality resources for inspiration, an editing platform for expression, and the opportunity to collaborate on an international scale.

Available worldwide, Glogster's iPad app answers learners' questions. Using the app's sleek interface, learners can be inspired by over 8,000 hand-picked glogs and resources covering eighty topics across nine disciplines. Glogpedia allows users to instantly interact with any topic through a combination of text, images, video, audio, and more. A simple button lets users save glogs to their dashboard ready to be viewed, shared, or edited using the integrated creation tool.

The app's enhanced editing function enables users to mix multimedia from their own files, Glogster's library of pre-designed graphics, or anywhere around the web courtesy of the powerful Web picker browser.

Nominating a glog to Glogpedia puts users at the heart of the library's content, allowing them to share ideas and expertise with 17 million learners and 1.9 million educators worldwide as part of the ongoing initiative to expand Glogpedia into a complete interactive resource.

The app integrates into English, Spanish, and Czech system languages with more to be added, allowing worldwide users to consume content in a truly unforgettable way within an ongoing cycle of inspiration, expression, and collaboration.

<https://itunes.apple.com/us/app/glogster/id907433564>

CodinGame Turns Programming into a Video Game

CodinGame is launching the first-ever coding platform that turns programming into a visually stunning online multiplayer video game.

With CodinGame, all the in-game actions are dictated by code commands developers write in real-time, allowing coders to improve their programming skills at each level, while having lots of fun.

The CodinGame community brings together more than 20,000 developers from 120 different countries, and the puzzles can be solved in twenty different programming languages (Java, Javascript, C/C++, C#, Python...).

Code source sharing and features such as live chat and forums allow developers to give advice, help other participants, and comment on their achievements, creating a sense of community among beginners and professionals.

Players can help Indiana Jones escape a maze filled with rolling boulders, land the Opportunity rover on Mars, or help Thor win the battle of Ragnarok. Gaming events such as "Game of Drones," or "Skynet Revolution," plunge developers into a unique immersive world of fun and excitement, combining the atmosphere of a hackathon with a video game, and bringing traditional coding to a whole new level.

The CodinGame platform supports a single-player mode where participants can keep track of their results, as well as a

multiplayer option. Just like in MMO games (massively multi-player online games), a large number of gamers can play artificial intelligence games simultaneously.

An in-game Integrated Development Environment (IDE) with its useful features such as code completion, coloring, etc., allows developers to write, test, and compile their code directly in the browser.

CodinGame organizes online coding contests and multi-player battles once a month where over 2,000 students as well as seasoned professionals from around the world come together. Some of the contests' sponsors include major companies like eBay, Nintendo, Sage, Adobe, and Ubisoft.

<http://www.codingame.com/cg>

Desktop Version of Anti-Malware Program Gate Scanner Released

A new desktop version of Sasa Software's enterprise security software, Gate Scanner, has been released.

The Gate Scanner blocks, scans, and cleans all files and ensures that data is exchanged within the network in a controlled and secure way. It uses a combination of six anti-virus programmes, mathematical processes which clean the data, file conversion, the removal of macros and other embedded elements, and deep scans of archived material. Encryption and verification systems allow managers to monitor and verify all activity.

The three different versions of Gate scanner are tailored to large or small sized networked or unconnected, stand-alone, systems. It can also be used on remote laptops or other devices. Regular updates are provided from a central server and, despite the intensity of the scanning and blocking, system users are not markedly inconvenienced.

<http://www.sasa-software.com>

Electric Cloud Launches ElectricFlow to Accelerate and Automate Software Build, Test, and Deployment

Electric Cloud has announced the release of ElectricFlow, a family of Continuous Delivery applications. ElectricFlow, built on the highly scalable ElectricCommander DevOps process automation platform, automates build, test, and deployment activities, eliminating software delivery bottlenecks and providing real-time visibility. For organizations developing mobile, embedded systems and enterprise web/IT applications, it makes software delivery repeatable at scale, and as routine as pushing a button.

ElectricFlow features a new deployment automation application called Deploy that automates and standardizes software deployments throughout the delivery process. It features a model-based approach that allows teams to centrally define and manage application and environment details to reduce delivery cost, increase quality, and accelerate time-to-market. Built from the ground up to be mobile ready, Deploy provides an intuitive, easy-to-use way to manage application deployments.

<http://www.electric-cloud.com>

Electric Cloud Launches ElectricAccelerator Huddle

Electric Cloud announced ElectricAccelerator Huddle, a free product that dramatically accelerates software builds and tests for development teams. Available through public beta, it empowers teams to be more agile and productive by tapping unused desktop CPU capacity within a peer-to-peer group of developers to distribute builds and tests, and reduce cycle times. The result is a highly efficient way of supercharging the “front-end” of the continuous delivery process.

With Huddle, teams can accelerate builds and tests, for free, forever, on up to eight local cores—or parallelize and distribute them across multiple machines up to a limited number of hours, ensure correct builds through build dependency management and file system isolation, add new members to the huddle and increase the available pool of virtual build and test resources, plug Huddle into an existing environment with no recoding or reconfiguration, and opt-in or out of a huddle at any time to control the amount of CPU cycles being shared

<http://electric-cloud.com/huddle>

VSDC Free Video Editor 2.2.0 Offers New Possibilities for Animation Experts

Flash-Integro releases VSDC Free Video Editor 2.2.0, capable video editing software that now allows for creating an animated object from a set of static images forming a video sequence. A series of optimization techniques applied to the frame processing resulted in a notable performance increase in the new version.

VSDC Free Video Editor already gained a good amount of popularity because it houses more than a user would expect from a free tool. Along with special non-linear video editing and subpixel manipulation with video fragments and pictures, VSDC also offers decent performance, an extremely intuitive user interface, and a range of exporting options including one-click DVD/Blu-ray burning.

Previously, the program allowed for adding video, images, and audio fragments to the final footage. However, many professionals who are experienced in the arts often need custom-drawn animated objects on the video—rotating logotypes, animated indicators, or labels and so on.

It is this gap that the new version of VSDC Free Video Editor fills. Now, a user can prepare a set of static images in any preferred format, including the native format of the Photoshop editor, and import the files to the program that compiles them into an animated object.

Just as in the other objects that the tool supports, animations are easily configurable. A user can specify the frame duration, choose key frame settings, and select the position of the individual image in a frame thereby enabling an animator artist to save a lot of time.

VSDC Video Editor is absolutely free and is available for Windows 2000/2003/XP/Vista/7/8.

<http://www.videosoftdev.com/free-video-editor>

Innotas Announces Partnership with cPrime to Deliver SAFe Professional Services

Innotas has announced a partnership with cPrime that for the first time provides users with a complete solution of agile portfolio management software and professional services. The cPrime/Innotas partnership enables customers to solve their enterprise agile scaling challenges with portfolio solutions and services, including Scaled Agile Framework (SAFe).

Under the partnership, Innotas will provide customers with Portfolio Management software, dashboards and reporting, with an Integration Platform to interface with agile work management software, such as Atlassian's JIRA. cPrime will offer process consulting, training, coaching, configuration, adoption, and staff augmentation.

Innotas' Cloud Agile Portfolio Management solution delivers portfolio management across agile projects, providing executives, PMOs, and program managers with improved visibility, cost tracking, and resource management.

With the complete Innotas/cPrime agile solution, customers will benefit from increased visibility and alignment of agile projects, with a top down portfolio view. In addition, the joint solution helps improve predictability and planning for releases, and increases buy-in at all levels of the organization.

<http://www.innotas.com>

Quadrant 4 Announces the Launch of Its New Cloud-Based Media Platform qBLITZ

Quadrant 4 System Corporation announces the launch of its qBLITZ cloud platform for the media and publishing industries.

qBLITZ is a first-of-its kind scalable cloud-based platform targeting high-traffic digital properties with footprints on web, mobile, and social. The unique proprietary platform tackles three key aspects of digital media, including core services around content authoring, storage, rights, and publishing; audience engagement through social modules; and monetization tools.

The platform is offered in the form of a PaaS (Platform-As-A-Service) recurring revenue model coupled with a retainer model for Professional Services. The platform will be dynamic and scalable to meet the changing needs of media and publishing companies so they can achieve higher efficiencies while improving audience engagement.

<http://www.qfor.com>

F A Q

expert answers to
frequently asked
questions

by Arlen Bankston

arlen.bankston@lithespeed.com

Scrum or Kanban—Which Is Better?

There is no reason for a fight; the Scrum and kanban agile methodologies can be complementary. Scrum provides simple rules imposing consistent inspect and adapt iterations and is the most widely adopted of all agile methodologies. Contrast this with kanban that has few rules: visualize your workflow, limit your work in process, and continually find ways to optimize cycle time. Should you choose between them or find ways to combine them?

Role Definitions

Scrum prescribes a product owner to manage demand, a team to manage end-to-end delivery, and a ScrumMaster to guide the process. This works as long as roles are well understood and appropriately staffed. However, stark differences from existing corporate role definitions can make adoption difficult.

Scrum emphasizes cross-functional behavior. Making activities parallel and highly collaborative can improve development speed and quality. For instance, developers often pair with testers and do testing on their own.

Kanban has no rules about roles at all. Team composition can be dynamic in kanban, but cross-functional team structures like those in Scrum are often practiced.

Managing Demand and Delivery

Scrum achieves focus through timeboxed sprints, which help teams concentrate and limit multitasking. Unfortunately, this can result in quality tradeoffs and last-minute heroics. Scrum requires a sprint's worth of work to be planned in advance, while kanban is made to deal with unplanned work. It is easy enough to leave buffers in Scrum while establishing simple rules to deal with unexpected demand. The timeboxed nature of sprints can help control unwanted tendencies like gold plating and scope creep.

Kanban, on the other hand, uses explicit limits for how many items are allowed to be in process at once. These approaches can effectively be used together. For instance, an imbalance in development and testing capacity in a Scrum team might be addressed through different work-in-progress limits for each activity.

Work that flows to different individuals rather than to a team is ideal for kanban, as is work involving repeatable processes commonly used in maintenance and operations projects. However, kanban can handle planned work just as well by requiring process and product review sessions.

Kanban uses classes of service to define rules around different types of work, such as defect resolution versus new development. Measuring these separately can provide more granular delivery speed information than batch velocity measures in Scrum.

Planning and Requirements Management

The product owner manages scope and requirements in Scrum; kanban leaves the job of prioritizing and managing the flow of work to the team as an open-ended proposition. Scrum uses release and sprint planning meetings to identify scope and goals by bringing teams and stakeholders together at regularly scheduled intervals. While Scrum's sprints set a development cadence, releases are made at the product owner's request; kanban simply includes them as a final stage in the workflow.

Grooming is the process of getting stories ready for the team to accept as work. In Scrum, preparing a few stories beyond what is likely to fit in a sprint is recommended and can effectively allow for continuous flow of work. In kanban, grooming is necessary but undefined, as wide variation in input size can generate unpredictable delivery times.

In Summary

Scrum and kanban are ideal for different situations, but they can be blended to good effect. Let teams start where they're most comfortable; then allow flexibility in technique and tool adoption while maintaining reasonable standards around the capture and sharing of information. **{end}**

Are You Done or Done-Done?

Don't fall victim to the "almost done" optimism prevalent in software development. Use these techniques to ensure tasks are completed.

by Masha Nehme | masha.nehme@abremod.com

You are in an elevator with your boss and he asks, "Where are you with your project?" You, scratching your head, guess, "Oh, around 40 percent complete." The boss walks away satisfied, but you are probably left still scratching your head. While using a percentage completion metric is widely popular among project manager practitioners, project management literature doesn't usually support this policy for status tracking, and it's best avoided. There has to be something better!

Why We Gravitate to Metrics

Everyone likes a number. There's no denying that a numeric value is a nice reporting metric. It can sound simple, clear, and accurate—even when it's not. Bosses prefer numbers and don't have time for details and prefer not to hear about other ambiguous metrics.

Numbers sell. Percent complete is a good sell because it creates an illusion of quantitative measurement. Customers buy into a quick number much easier than into a prospect of working together on a product that best fits their need. Consequently, we tell them what they like to hear and worry later.

That's what we've got time for. Guessing that your project is, for example, 40 percent complete takes no time, while any useful project performance analysis will require more time than that. When project managers are putting out fires all day long, there's no time to properly plan and track project progress.

Everyone does it. Providing a percent complete is right up there with other "We've always done it this way" excuses. If everyone reports percent complete at your company, you might feel comfortable doing the same.

We're not ready for anything more sophisticated. You don't know what your team is capable of unless you give them an opportunity. Until then, your developer will estimate the duration of each activity at exactly the same amount of time you've allocated for it in your project plan—because that's what you want to hear.

"There is definitely a prevailing culture of planning a project as a progression of tasks as opposed to focusing on results."

What about Reporting Percent Complete?

Define "complete." Gerry Boston, a project manager with a software company in Washington, D.C., recalls crossing off a task as completed after hearing a report that it was done, only to discover the next day that the developer went back to working on the task.

When asked about it, the developer said, "Yes, I am done, but not done-done." Defining completeness of a given task or the entire project is a challenge.

You need to be meticulous in your documentation, and you need to actively involve the team. Otherwise, your reporting "around 100 percent complete" is just as meaningless as reporting 40 percent.

You're almost there (most of the time). As Frederick Brooks mentioned in his classic book, *The Mythical Man-Month*, coding is "90 percent finished for half of the total coding time." [1]

Project managers readily acknowledge that certain project activities eat up most of the project time and resources.

This is especially true about ongoing, iterative activities, such as coding or testing. A percent complete value assigned to such activities is bound to be arbitrary—and, in most cases, useless.

What you really need to know is how much of the desired result has been achieved with all the work, resources, and time that has been put in. This is also referred to as value.

Complete has nothing to do with done. If a project were simply composed of a list of activities, then we could just deliver the promised hundred lines of code to our customer and not worry about whether the code actually does anything for them.

It's amazing how many project managers get caught up in task lists (defining, tracking, and crossing tasks off) and forget about the project's purpose, which is delivering its final product and value to customers.

Yes, There Is an Alternative

One clear alternative to percent complete is earned value management (EVM). This methodology requires simultaneously tracking scope, schedule, and resources. In EVM, a project is always considered within its context; you never look at a schedule without considering costs and the volume of work performed. The focus is on results and solutions as opposed to mere tasks. EVM does provide numeric values you can report and use in managing a project. Most importantly, EVM gives you facts you can use to get the project back on track. For example, you can be within budget but behind in getting the work done due to some risk event. Alternatively, you can be hitting milestones as planned while spending twice as much as you had originally budgeted. Percent complete cannot give you such a holistic glance at the project.

Why Don't Project Managers Use EVM?

There is definitely a prevailing culture of planning a project as a progression of tasks as opposed to focusing on results. Tools we use sometimes further propagate this problem. In his article "Convincing Project Managers to Use EVM (and Making No Excuses)," Jose Barato [2] recommends moving away from a spreadsheet, which many use for project tracking. General tools like these are not designed to compare planned and actual values. Microsoft Project and other project management software tools, such as Clarity PPM or Open PPM, make EVM a lot easier to integrate into your daily practice.

Another obstacle keeping project teams from using methods like EVM is that they appear complex, with too many acronyms and formulas that look intimidating. In reality, there's nothing in EVM that a project manager shouldn't already know. EVM allows you to compare actual value to planned value throughout the project cycle.

In quantifying scope, Williams Chirinos of INEXERTUS, a project management consulting firm, recommends using simply a binary completion theory. This theory states that something is either done (you earned "a one") or not done (you earned "a zero"). [3] When the deliverable is complete, it gets factored into the earned value calculations. Don Cox of VPMSI, a project management consulting agency, also recommends assigning additional weights to work packages to indicate high-priority deliverables, such as those on the critical path. This approach will help portray an accurate picture of your project's performance at any given moment.

How about working with numbers that mean something?
{end}



Click here to read more at StickyMinds.com.

■ References

index to advertisers

Agile Development & Better Software Conference West	http://adc-bsc-west.techwell.com	23
ASTQB	http://www.astqb.org	8
Call for Speakers	http://www.sqe.com/Conferences/Speakers.aspx	Inside Front Cover
Capgemini	http://www.capgemini.com/testing	Back cover
Mobile Dev + Test Conference	http://mobiledevtest.techwell.com	9
Ranorex	http://www.ranorex.com/whyBSM	2
STARCANADA	http://starcanada.techwell.com	10
STAREAST	http://stareast.techwell.com	15
Upcoming Events	http://www.sqe.com/conferences	1

Display Advertising
advertisingsales@sqe.com

All Other Inquiries
info@bettersoftware.com

Better Software (ISSN: 1553-1929) is published six times per year: January/February, March/April, May/June, July/August, September/October, and November/December. Back issues may be purchased for \$15 per issue plus shipping (subject to availability). Entire contents © 2014 by Software Quality Engineering (340 Corporate Way, Suite 300, Orange Park, FL 32073), unless otherwise noted on specific articles. The opinions expressed within the articles and contents herein do not necessarily express those of the publisher (Software Quality Engineering). All rights reserved. No material in this publication may be reproduced in any form without permission. Reprints of individual articles available. Call 904.278.0524 for details.

WORLD QUALITY REPORT

2014-15 | SIXTH EDITION



YOUR ULTIMATE GUIDE TO THE ENTERPRISE APPLICATION QUALITY AND TESTING PRACTICE

- What is the percentage of budget allocated for the testing function?
- Adoption of cloud testing: Hype or reality?
- Does your organization currently test mobile applications and devices?
- How are testing functions adapting to the changing demands of their business?
- How does digital transformation affect organizations' testing practices?

The answers to above queries can be found in the World Quality Report 2014-15!

The *World Quality Report*, jointly published by Capgemini, Sogeti and HP, is established as the largest global research study and expert commentary on enterprise application quality and testing practices. The *World Quality Report 2014-15* (sixth edition) is based on market research carried out via telephone interviews with 1,543 senior IT executives across 25 countries.

To know more about Capgemini Testing visit: www.capgemini.com/testing-services



Top trends highlighted in the report for North American companies are:

- Mobile testing is rapidly gaining momentum – only one in 10 research participants states that they don't engage in mobile testing.
- Only 16% of North American companies report having no interest in creating Testing Centers of Excellence (TCOE) – a steep decline from 26% in 2013 indicating growing Quality Assurance and Testing maturity levels and a strong drive towards consistency of quality process.
- A growing trend among North American companies is to achieve capex reduction by moving application infrastructure to the cloud.



Find out how your organization can adapt to the current trends in Testing. Download the *World Quality Report 2014-15* at www.worldqualityreport.com