

FALL 2017

[BETTERTM SOFTWARE]

A TECHWELLTM PUBLICATION

THE SECRETS OF HIGH- PERFORMANCE SOFTWARE TEAMS



**BRIDGE THE DIVIDE BETWEEN
WATERFALL AND AGILE**

Help your traditional practices
and agile methods coexist

**THE IMPACT OF QUALITY
INTELLIGENCE**

Make smarter quality
decisions in DevOps and IoT

**WE LOOK FORWARD TO
SEEING YOU IN ORLANDO!**

**Special Offer for
Better Software
Subscribers:**

**Register
with promo
code BSME
to save up
to \$200 off***

2017
Your Name

Delegate

Agile Dev
Better Software
DevOps **EAST**

A TECHWELL EVENT



Nov. 5-10, 2017
Orlando, FL
Hilton Orlando
Lake Buena Vista

BSCEAST.TECHWELL.COM

Benefit from a custom week of learning and discovery through all aspects of the development lifecycle with:

- Comprehensive tutorials
- Exceptional concurrent sessions
- Inspiring keynotes
- Pre-conference training and certification classes
- Networking activities
- The Expo
- And more

*Discount valid on packages over \$600



Perfecting Software

GET THE COMPETITIVE ADVANTAGE WITH A COMPLETE IOT SOFTWARE TESTING SOLUTION



“

The importance of software testing and error prevention has risen dramatically, paralleling the continued escalation of software complexity. Parasoft provides developers with the tools and infrastructure necessary to test early and regularly, ensuring quality throughout the software development lifecycle.

Theresa Lanowitz, *voke inc.*

”

STATIC &
DYNAMIC
ANALYSIS

UNIT
TESTING

API, LOAD,
PERFORMANCE, &
SECURITY TESTING

SERVICE
VIRTUALIZATION

Learn more at parasoft.com/iot

LEARN MORE

ZAPTEST

Any Software Cross-Platform

Automate testing of ANY software user interface AS-IS!

ZAPTEST can automate any type of GUI based software application with no limitation Mobile, Web, Desktop

Browsers, Mobile native apps, SAP, PDF, Oracle, POS, Liquid UI, Windows, iOS, Android, Windows, WPF, WinForms, Chrome, Safari, Firefox, IE, Chromium, Opera, Flash, Java, SVG, XML, Mac OS X, BlackBerry, Terminal, Shell, Qt, MFC, GTK+, Cocoa, .NET, OpenGL, Flex, jQuery, AIR...

1 Test Script = 1 Business Process

Reduce number of test scripts for test development and maintenance

Auto-generate Test Documentation with one click from automated script

No need for development manual test cases

Execute 1Script across ALL testing environments Cross-Platform at once in PARALLEL!

Same test script runs on all testing environments: Mobile and Workstations

Expedite time to Regression testing

Support test process for AGILE / DevOps!

Start Shift-Left testing at the SDLC Design phase

Use application mock-ups for test development

Start test cycle with test automation

Integrate with test management technologies: HP ALM & CA Rally!

Keep test management within your corporate standard

Convert automated scripts into manual test steps

Execute test automation and integrate test results for reporting



GET STARTED!

INSIDE

Volume 19, Issue 4
FALL 2017



On the Cover

The Secrets of High-Performance Software Teams

Of all issues that impact getting quality products out on time, the team should never focus on simply managing costs. To minimize the risk of perpetual product delivery delays, define what “done” really means.

by Steve Mezak

14

Features



18

Bridging the Bimodal Divide between Waterfall and Agile

Most software developers are in either the agile or the waterfall camp. Agile is required to be competitive, but many enterprise processes still rely on waterfall practices for stability. They can coexist. *by Steve Elliott*



24

Scaling Agile Thinking through Empowered Teams

Just because a software team adopts agility doesn't mean they'll see results. Being flexible has its benefits, but ensuring that the team is given total responsibility to make decisions may be more important. *by Bob Costello*



28

The Impact of Quality Intelligence on DevOps and the IoT

QI is gaining momentum for providing fast and accurate test data to make better decisions. Just running your app through test suites isn't enough for DevOps and the IoT. *by Steven Winter*

Columns

09 TECHNICALLY SPEAKING

The New Normal for Software Development and Testing

Customers expect fast-paced, quality product releases. To meet these demands, you must consider test automation, testing in production, and the extensive use of data and analytics. *by Mike Sowers*

36 THE LAST WORD

How Agile Has Shrunk Documentation

Agile teams enjoy focusing their time on product features while keeping documentation to a minimum. But every team needs to consider what documentation is really needed. How much is enough? *by Sandeep Maher*

Departments

06 Mark Your Calendar

07 Editor's Note

08 Contributors

12 Interview with an Expert

34 TechWell Insights

37 Ad Index



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.

MARK YOUR CALENDAR



Helping organizations worldwide improve their skills, practices, and knowledge in software development and testing.

Software Tester Certification— Foundation Level

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

Oct. 15–17, 2017
Toronto, ON

Oct. 17–19, 2017
Nashville, TN

Oct. 24–26, 2017
Washington, DC

Oct. 30–Nov. 3, 2017
Virtual Classroom

Nov. 5–7, 2017
Orlando, FL

Nov. 7–9, 2017
Chicago, IL

Nov. 14–16, 2017
Santa Clara, CA

Dec. 5–7, 2017
Austin, TX

Dec. 5–7, 2017
Phoenix, AZ

Agile Testing Training Week

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

Nov. 14–17, 2017
Santa Clara, CA

Dec. 5–8, 2017
Raleigh, NC

Mobile Testing Training Week

<http://www.sqetraining.com/mobile-week>

Nov. 14–16, 2017
San Jose, CA



Conferences

Cutting-edge concepts, practical solutions, and today's most relevant topics. TechWell brings you face to face with the best speakers, networking, and ideas.

STAR CANADA

A TECHWELL EVENT

October 15–20, 2017
Toronto, ON

[LEARN MORE](#)

Agile Dev
Better Software
DevOps **EAST**

A TECHWELL EVENT

November 5–10, 2017
Orlando, FL

[LEARN MORE](#)

STAR EAST

A TECHWELL EVENT

April 29–May 4, 2018
Orlando, FL

[LEARN MORE](#)

Agile Dev
Better Software
DevOps **WEST**

A TECHWELL EVENT

June 3–8, 2018
Las Vegas, NV

[LEARN MORE](#)

STAR WEST

A TECHWELL EVENT

**September 30–
October 5, 2018**
Anaheim, CA

[LEARN MORE](#)

Planning for Productivity Improvements in 2018

It is rare to have two authors with the same first name in a single issue of *Better Software* magazine. Surprisingly, our fall 2017 issue has three authors with the first name Steve! Our featured cover article is by Steve Mezak, and I can't think of a more important topic than "The Secrets of High-Performance Software Teams." One of Steve's secrets is critical for anyone leading a development team. According to Steve, the team must always prioritize the product backlog. And work is only completed if the entire team agrees to a unified understanding of what "done" means.

It is never easy transitioning from waterfall to agile. Steve Elliott explores the benefits of adopting both approaches in "Bridging the Bimodal Divide between Waterfall and Agile." Software development leaders believe that teams will outperform expectations if you place full trust in them. Bob Costello shows how in "Scaling Agile Thinking through Empowered Teams."

In Steven Winter's "The Impact of Quality Intelligence on DevOps and the IoT," you'll learn why providing fast and accurate test data is required to make better decisions. It isn't enough to just run test suites on software anymore.

Mike Sowers shows how continuous approaches have become the mainstream in "The New Normal for Software Development and Testing." For those of us who struggle to determine just how much technical documentation is really needed, you'll enjoy Sandeep Maher's "How Agile Has Shrunk Documentation."

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 reading this issue as much as we enjoy working with these wonderful authors. And let me know what you think of our new magazine layout.

Most of the TechWell team is located in Jacksonville, Florida, and despite Hurricane Irma, everyone pitched in to get this issue out on time. Thanks, everyone! And, readers, please don't forget to spread the word about TechWell and *Better Software* magazine.



Ken Whitaker

kwhitaker@techwell.com

Twitter: @Software_Maniac



FOLLOW US



PUBLISHER
TechWell Corporation

PRESIDENT/CEO
Wayne Middleton

DIRECTOR OF PUBLISHING
Heather Shanholtzer

Editorial

BETTER SOFTWARE EDITOR
Ken Whitaker

ONLINE EDITORS
Josiah Renaudin
Beth Romanik

PRODUCTION COORDINATOR
Donna Handforth

Design

CREATIVE DIRECTOR
Jamie Borders
jborders.com

Advertising

SALES CONSULTANTS
Daryll Paiva
Kim Trott

PRODUCTION COORDINATOR
Alex Dinney

Marketing

MARKETING MANAGER
Cristy Bird

MARKETING ASSISTANT
Allison Scholz

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
TechWell Corporation
350 Corporate Way, Ste. 400
Orange Park, FL 32073



CONTRIBUTORS



A senior agile coach at the Eliassen Group, **Bob Costello** has more than twenty years of experience in software product management, IT engineering, team leadership, and organizational leadership. For the past eight years, Bob has been using agile principles to help both teams and executives think differently about the process of software development. He places special attention to help entire organizations support that process. Reach Bob at rcostel@comcast.net.



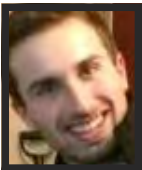
Steve Elliott founded AgileCraft in 2013 to help leaders excel in their management duties through the use of agile. He is passionate about finding ways to help organizations scale all aspects of business operations. Formerly a senior executive in multiple startups, Steve has led several agile transformations including a large-scale fifty-team project with a major technology corporation. Contact Steve at steve@AgileCraft.com.



Playing multiple roles has always come naturally to **Sandeep Maher**, so it is no surprise to find his thirty-year career covers areas of finance, project management, tools development, quality assurance, business analysis, and testing. His professional mantra is to “make a difference.” When not working, Sandeep is interested in photography, cricket, and reading. He can be reached at sandeepmaher@gmail.com.



Steve Mezak is the founder and CEO of Accelerance, a US-based outsourcing advisory firm providing outsourced software development and augmenting staffing for small- and medium-sized enterprises. He is the coauthor of *Outsource or Else!: How a VP of Software Saved His Company* and the author of *Software without Borders*. Reach Steve at steve@accelerance.com.



A longtime freelancer in the tech industry, **Josiah Renaudin** is now a web-content producer and writer for TechWell Insights, StickyMinds.com, and *Better Software* magazine. He wrote for popular video game journalism websites like GameSpot, IGN, and Paste Magazine and now acts as an editor for an indie project published by Sony Santa Monica. Josiah has been immersed in games since he was young, but more than anything, he enjoys covering the tech industry at large. Contact Josiah at jrenaudin@techwell.com.



Mike Sowers has more than twenty-five years of practical experience as a quality and test leader of internationally distributed test teams across multiple industries. He is a senior consultant who works with large and small organizations to improve their software development, testing, and delivery approaches. Mike has worked with companies including Fidelity Investments, PepsiCo, FedEx, Southwest Airlines, Wells Fargo, and Lockheed to improve software quality, reduce time to market, and decrease costs. Reach Mike at msowers@techwell.com.



President and chief strategist at GuerrillaQA, **Steven Winter's** has twenty-two years of experience in quality assurance and team management. With a focus on mobile, test automation, and startups, Steven has managed quality for America's leading financial technology services providers, including the innovative power behind the Starbucks Card mobile app, the nation's first successful large-scale mobile payment solution. Contact Steven at steven@guerrillaqa.com.



The New Normal for Software Development and Testing

AGILE, DEVOPS, AND CONTINUOUS DELIVERY HAVE RESHAPED QUALITY PROCESSES AND PROCEDURES USED BY MODERN SOFTWARE DEVELOPMENT TEAMS.

by Mike Sowers | msowers@techwell.com

There's an endless stream of research, commentary, and conjecture forecasting how technology will eliminate or transform traditional jobs. This is not new, of course. This conversation has been going on for centuries. Technology and the software that supports it continue to move forward at ever-increasing speed. We are now experiencing another step in the transformation of work in software development and testing.

Software development and testing are more challenging due to security and vulnerability risks, diversity of user interfaces, big data, analytics and machine learning, the pursuit of autonomous transports of every kind, connected everything, and constantly evolving technology stacks.

At the same time, agile, DevOps, and other adaptive approaches allow us to innovate faster and deliver improvements almost continuously—with the added benefit of real-time user feedback in our live environments. To complicate matters, rapid deployments keep IT management awake at night worrying about the lingering risk of major or catastrophic failures. For development and testing, the “new normal” means more change, less certainty, and shorter development and testing windows. Our industry must adapt because there is a constant flow of new functionality to deploy.

To keep pace with customer demands, we've always challenged ourselves to improve and enhance the way we do development and testing. This fast-paced industry requires that every technology company stay ahead of the competition and ultimately drive revenue growth. However, the changes we see now in our approaches to development and testing are more significant than those we've seen in the past.

Here are some characteristics of the new normal for development and testing in this digital age of agility and continuous integration and deployment.

Development and testing are a team sport. To successfully complete a project, we must leverage every team member's skills and capabilities and provide everyone visibility to all the work being done. Development is not just for developers any longer, and testing is not just for testers any longer. Development and test professionals are now collaborating much more closely during the entire product lifecycle while business users and analysts join the quest for much higher quality. As a result, testers also need to know more about business and development methods.

Data and analytics are now playing an increasing role. Development is about creating solutions to solve business problems. Testing has always been about providing information to stakeholders about the developed solutions' readiness for the user. Simultaneously looking at data concerning both future and current use of a software solution should drive just enough development and testing to get new features to market with the right balance of functionality and quality. Analytics are used to influence software design, development, integration, testing, deployment, and business decisions.

TestDev thinking is pervasive. Traditional roles are becoming more collaborative. Those with testing expertise are advising the team early and throughout the project lifecycle. Developers are advising those in testing roles. Traditional testing has become more test-driven, often incorporating model-driven and behavior-driven approaches. This expertise is embedded into the agile and DevOps teams, and the entire team is collaborating on testing

To successfully complete a project, we must leverage every team member's skills and capabilities and provide everyone visibility to all the work being done.

TECHNICALLY SPEAKING

and developing everything—from the architecture to the product features, the nonfunctional attributes, and the automation infrastructure that supports development, testing, and delivery.

Continuous everything means we automate everything.

Those with development, testing, and operations knowledge are contributing to the build, automation, and deployment processes as well as to the automation of development, testing, and release for the entire project.

Testing in production is a frequent practice. As systems scale and the number of connected nodes increases (think of the Internet of Things), it is impossible to create test environments that emulate the real world. There is no choice but to test in production. Systems and applications are being architected and packaged using microservices and containers. As a result, new features and functionality can be deployed into production but not enabled using feature switches. This allows the team to reduce their risks and control the exposure to new or modified code, as well as to manage which users see which features or to compare two or more approaches using A/B testing.

Deeper skill sets are a requirement. Those in analysis, development, testing, and operations roles require analytical knowledge, such as data engineering and data science, as well as deeper technical skills in architecture, development, and automation. The lines between all these roles have become blurred.

Automation is pervasive. Just like the transformation of manufacturing, automation is the focus of every part of the software lifecycle. Extensive automation at every step yields consistency and repeatability. This enables the team to focus on more technically challenging areas while receiving timely feedback on things

that need attention. This is especially true for testing. While there will always be a need for some manual testing, automated testing dominates the teams and environments that are deploying frequent updates and changes.

The whole team is accountable. This has been the norm for agile for a while. With DevOps, however, total team involvement should include a wider range of stakeholders, such as operations staff and end-users. This effectively removes any barriers between the customer and the team.

Near real-time measures and metrics are expected. After the initial setup, most of today's project lifecycle tools produce data and information easily and quickly. Using business intelligent tools, teams are establishing near real-time information dashboards that report on what's being planned in the pipeline, the status of user stories, and monitoring end-user experiences.

The tolerance for risks may change. Throughout software engineering history we have strived to build in quality. Some experts call this *quality engineering*. Our emerging continuous development, deployment, and testing processes, tools, practices, and expertise are allowing teams to think and act differently relative to defects. There are now better ways to avoid defects in the first place. Small tested changes can be deployed into production, exposing them to just a few users. An update that does not perform as expected can be immediately withdrawn or corrected.

The new normal in software development and testing looks like many of the things we've dreamed about. Of course, there will be future new normals, but this current evolution is yielding better software, delivered more quickly, and improved means of managing risks. **[BSM]**

WANTED! A FEW GREAT WRITERS



TechWell is always looking for authors interested in getting their thoughts published in *Better Software*, a leading online magazine focused on 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
- DevOps
- Project and people management
- Configuration management

I'm looking forward to hearing from you!

Ken Whitaker

Editor, *Better Software* magazine | kwhitaker@techwell.com

Be a Software Genius.

CAST Analytics take you from
Software to Superware™



www.castsoftware.com



INTERVIEW WITH AN EXPERT

“There are lots of tools that you can gain from very unexpected places. You can go to things like improv, learn very simple exercises, and it help you practice using that muscle that makes an agile mindset really well-tuned.”

“The first thing teams need to have in order to build is to be able to listen—to be able to hear what other people are saying enough to understand it and be inspired by it and move forward. I think that is the most basic need.”

“It’s hearing what people have to say and being nonjudgmental and just playing around with what those ideas are and seeing where they go... they may be part of what got you to the golden solution.”

“By setting it up to have small mistakes, or small wins along the way, I don’t think failure is much of a scar or damage on the team.”

Jessie Shternshus

Years in Industry: 8

Email: jessie@improveffect.com

Interviewed by: Josiah Renaudin

Email: jrenaudin@techwell.com

“The nice thing about agile is you’re not pushing something forward before you’ve ever tested it. So you’re doing things in small chunks and constantly testing things so that when you make mistakes, it’s a lot more low-risk, if you will.”

“You should have that back-and-forth dialogue with your customer so that they’re able to tell you in real time what they like, what they need, how things should change, and how to move forward.”

“You have each other, you have a goal in mind, but there’s a lot of improvising to get to the goal. So, agile and improv are very, very similar in that they set each other up for team success.”

“The more that people see there’s a new perspective or a new idea that comes out of the mistakes, the more they’re willing to put themselves out there and not just do the status quo.”

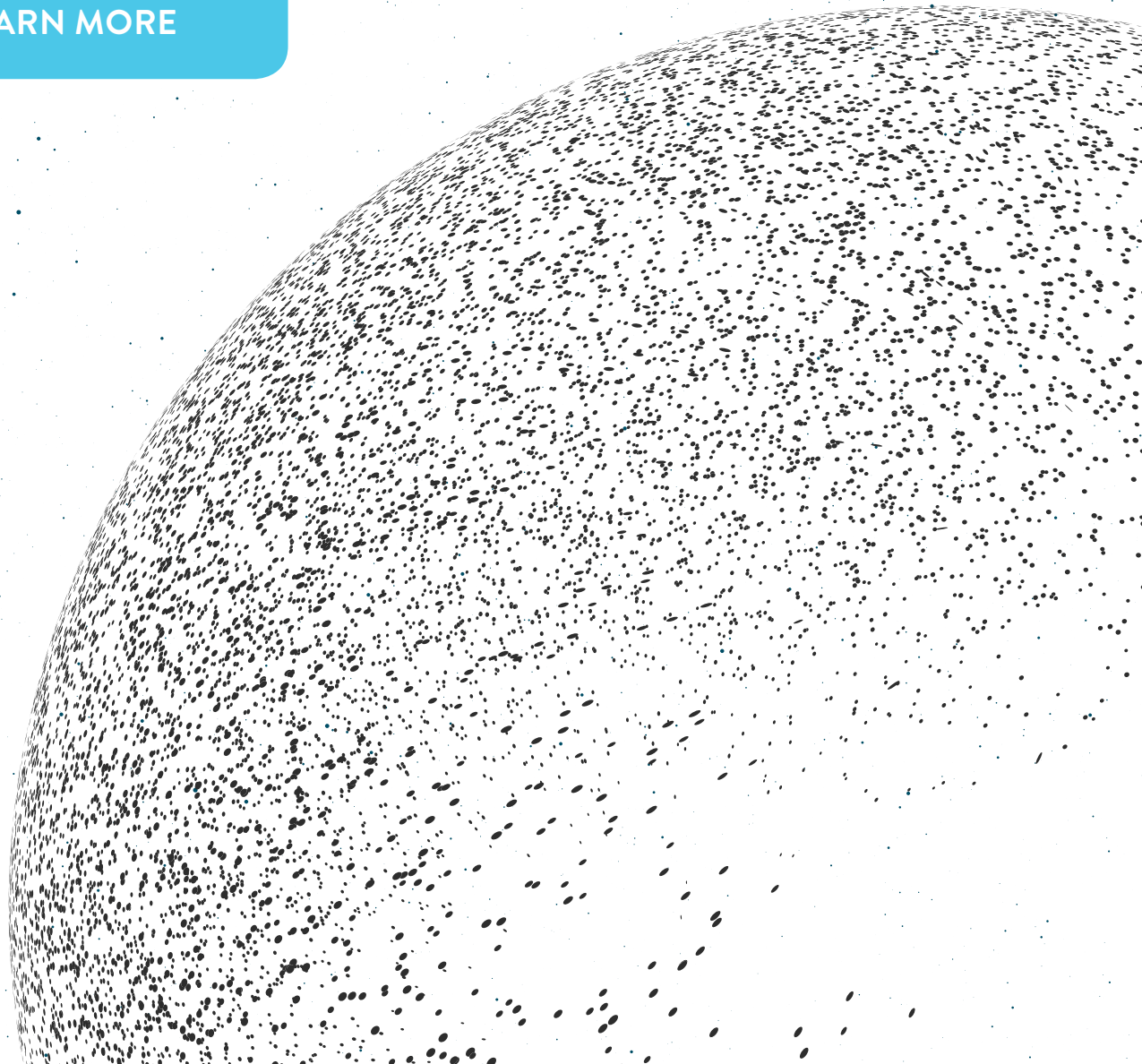
[CLICK HERE FOR THE FULL INTERVIEW >>](#)

DELPHIX

THE DELPHIX DYNAMIC DATA PLATFORM

Fast. Secure. Everywhere.

LEARN MORE





THE SECRETS OF HIGH- PERFORMANCE SOFTWARE TEAMS



by Agent
Steve Mezak

In my thirty years in software development, I've seen many teams struggle to meet company expectations. Most of the perceived lack of performance is due to missed deadlines and cost overruns. Most software executives believe that the important thing a team must achieve is an expected level of profitability.

What do successful software teams have that the laggards don't? What are their development mechanics? What practices do high-performing software teams embrace?

A lot depends on the mindset of the software product manager or project manager who leads and directs the team. Ultimately, this individual is responsible for keeping the initiative on its visionary rails and in touch with users and customers. Because managers are closest to the company's objectives for software development, much of the team's energy comes from the manager's thinking and behavior.

A right-minded software manager contributes to development success based on three critical factors defined in Actuation Consulting's "The Study of Product Team Performance": [1]

1. Don't overemphasize development costs when prioritizing requirements.
2. Strive for a clear definition of "done" within the team.
3. Always prioritize the backlog by value.

The study looks at ways to improve a software product team's effectiveness and increase profitability. All three of these key performance indicators (KPIs) closely correlate with successful, high-performance teams. And in my experience, a high-performance software development team that adheres to these KPIs will usually meet company expectations and deliver profitable products and solutions.

So, what makes a high-performance team?

Focus on Quality, Not Costs

Overemphasizing development costs when prioritizing requirements is an all-too-common leadership mistake. Teams that prioritize work based on limiting development costs are more likely to underperform, according to the study. Underperformance can mean a late or delayed launch, missing requirements, or buggy features. Failing to deliver a product that end-users expect is a horrible outcome.

Why invest at all if you're not committed to quality as an end result? Certainly, complex software features require more time and money to develop, but to push out cheap, unsatisfactory software is a big waste of time and resources.

For example, some companies are motivated to outsource development for cost savings only. Although outsourcing your software development can give you an entire up-and-running development team quickly, it's impossible for your team to get top-quality

results unless you're willing to think about the cost appropriately. Use outsourcing as a flexible strategy to get expert development resources you need—not simply as a way to cut costs.

A good manager sets budget expectations with realistic assumptions up front, which is different from just ongoing monitoring of the budget. However, even a minimum, nominal, and maximum budget may not anticipate changes that must be made to the application under development.

For example, a major change based on user feedback can require a shift in direction that impacts the schedule and cost, but ultimately it may be the right thing to do. This occurred in a web application I recently had developed that contained a list of software technologies (such as PHP

and JavaScript) from which users could select. Most users of the initial alpha release made clear that they wanted to define their own technologies (Symfony, AngularJS, Node.js, and so on) in addition to selecting from our list. There was no way we could anticipate every software technology they would want to use.

Although it took two additional unplanned sprints to solve this problem, the switch to a tag or folksonomy feature really had to be done to ensure the growth of the number of users. Ultimately it positively impacted the profitability of the app. Isn't that what being flexible and agile is all about?

Define "Done" and Prioritize the Backlog

I can't emphasize enough the importance of working with the team for a clear definition of "done" and to consistently prioritize the backlog by value. High-performance teams that have a definition of "done" always outperform their counterparts. But make sure your entire team collectively comes up with the definition.

Early in my career I worked at a startup in the computer-aided engineering market. The company leadership could not decide whether we wanted to create an app for circuit board design or for integrated circuit design. Each had a large market segment but each also had different sets of feature requirements.

After a management change that brought in a new CEO and VP of engineering, the decision was made to focus on delivering an application to design circuit boards, and we finally put a release plan into place.

My advice is not to wait for a regime change. Be clear about what your software will achieve for your company—at both a high level and for every release.

Actuation Consulting's performance study shows a strong correlation between an effectively prioritized backlog and a high-performance team. Not surprisingly, teams that clearly and efficiently move from task to task are more likely to deliver high-performance results. Conversely, a jumbled backlog has the opposite effect on both performance and financial results.

**HIGH-PERFORMANCE
TEAMS THAT HAVE A
DEFINITION OF "DONE"
ALWAYS OUTPERFORM
THEIR COUNTERPARTS.**

Encourage Vision and Innovation

Instead of focusing on costs at the beginning of a project, use that time for market research to sculpt the vision for your software. Talk with potential customers and users. Go exploring. It is unlikely that you'll have time for this once the project is underway.

Look at current trends and brainstorm potential ones so your plans aren't disrupted by a better way of doing something. This will promote more innovative thinking with the team and help you set a more realistic budget. To find your vision, think and plan at a much higher level. This approach simplifies setting the overall project goals.

Sit down with your team to talk about the vision. Instead of announcing, "This is exactly *what* I want," say, "This is what *I think* I want." This gives permission to let fresh ideas and solutions emerge. You want to encourage your developers to give their opinions.

Allow room for your team to explore and draw upon previous experiences. This frees them to look beyond what you originally thought you wanted and deliver better software than ever imagined. Empower your developers—on internal, distributed, and out-sourced teams—to innovate.

After market research and team conversations wind down, create your product roadmap. Properly executed roadmaps direct where you want the software to go in the future. Start by planning the steps needed to fulfill the vision with every release and update. Think product sequencing, pricing iterations, and new features and functions that you want to add down the road. Consider future releases as experiments in order to ensure what you are releasing is what users want and need. Spend some time considering the budget—as well as resources, time, staff, and outsourcing—as you plan the timing of your releases.

I've observed many managers who felt they didn't have the team bandwidth or budget to do everything. Because they can't imagine how the software's complex features could actually get done, they mistakenly limit future scope. They end up shortchanging some of the most important aspects and features.

But just because you limit the software based on its budget doesn't change the fact that the product still needs sophisticated features and functions to be successful. If you don't develop the software to do what it's supposed to do, then why even bother developing it?

Cultivate a High-Performance Team

In the end, a high-performance team comes down to the manager having the right mindset—one that doesn't overemphasize costs, drives market research at the start, encourages innovation, and stays on track by prioritizing tasks. All of this leads to better software profitability for your organization. [\[BSM\] steve@accelerance.com](mailto:steve@accelerance.com)

CLICK FOR THIS STORY'S **REFERENCES**

NEWSLETTERS FOR EVERY NEED!

Want the latest and greatest content delivered to your inbox? We have a newsletter for you!

AGILE CONNECTION™ *To Go*
A TECHWELL COMMUNITY

AgileConnection To Go covers all things agile.

DEVOPS *To Go*
BROUGHT TO YOU BY CMCROSSROADS

DevOps To Go delivers new and relevant DevOps content from CMCrossroads every month.

STICKYMINDS™ *To Go*
A TECHWELL COMMUNITY

StickyMinds To Go sends you a weekly listing of all the new testing articles added to StickyMinds.

TECHWELL™
INSIGHTS

And, last but not least, **TechWell Insights** features the latest stories from conference speakers, SQE Training partners, and other industry voices.

Visit AgileConnection.com, CMCrossroads.com, StickyMinds.com, or TechWell.com to sign up for our newsletters.

An aerial, black and white photograph of a rowing team in a scull on the water. The team consists of five rowers, each with their own oar, moving in a synchronized fashion. The water is dark with visible ripples and the wake of the boat. The text 'SERVING AS A LEADER' is overlaid in large, white, bold, sans-serif capital letters across the center of the image.

SERVING AS A LEADER



iSQI Scrum Master Pro

realscrummaster.org

BRIDGING

the

BIMODAL

DIVIDE

between

WATERFALL

and

AGILE

STEVE

ELLIOTT

Moore's law predicted in 1965 that computing power would dramatically increase while costs would decrease exponentially. [1] This theory has historically held true—we've witnessed explosive growth in almost all areas of technology. This phenomenon includes processors, microchips, batteries, cellphones, vehicles, and many other products. In short, technology innovations will continue to disrupt nearly all segments of commerce and production.

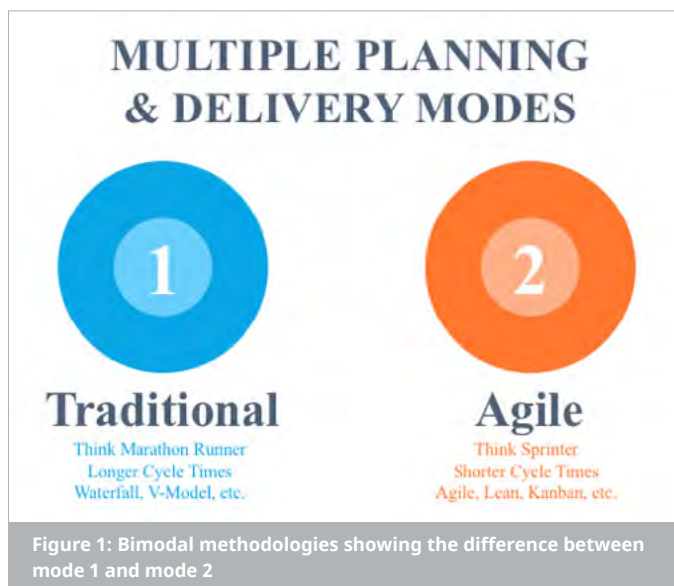
While the consensus among chip manufacturers is that Moore's law is a fact (until recently), the same theories of increasing demand for smarter, faster, and leaner software may not be possible. To keep pace in a more sophisticated software-rich world, enterprises must either look to become an innovator or risk being left in the dust by the competition.

Choosing the Best Methodologies

Software drives much of the intelligence for these fast-paced technology product offerings. And as we all know, there are good and bad approaches (or methods) to deliver this software.

Many of these enterprises are following an approach that mixes traditional IT methodologies with those built to deal with the dynamic nature of today's technology. Gartner calls this approach bimodal, and warns their clients and the marketplace to either prepare to incorporate digital innovations into the way they operate or risk being disrupted—or worse. Gartner recommends that “marrying a more predictable evolution of products and technologies (mode 1) with the new and innovative (mode 2) is the essence of an enterprise bimodal capability.” [2] This is depicted in figure 1.

While many development teams have lived in the world of agile for quite some time, most IT organizations have not taken that same approach to the broader aspects of their PMO, portfolio, and enterprise levels, nor have they incorporated their business partners into the world of agility. In my experience, many enterprises have agile teams under waterfall management.



There are some fundamental approaches to bimodal concepts and capabilities that organizations can follow in their agile adoption and transformation to reduce the risk of large-scale agile change management initiatives.

You want to enable organizations to embrace a culture of innovation while allowing select projects to use traditional models.

Let's look at a few key principles organizations can adopt to successfully drive the alignment, coexistence, and collaboration between waterfall and agile mindsets.

YOU WANT to ENABLE ORGANIZATIONS to EMBRACE a CULTURE of INNOVATION WHILE ALLOWING SELECT PROJECTS to USE TRADITIONAL MODELS.

Taking the Bimodal Journey

Most software enterprises consider software and their company's ability to deliver software as a strategic asset. Because speed to market and providing innovative solutions represent competitive advantage, many companies are involved in some form of agile transformation. That usually results in a mixed, or bimodal, environment. This journey typically moves through four stages of evolution: team, program, portfolio, and enterprise (figure 2 on the next page). As an organization progresses through each stage, concepts usually become easier to understand and implement.

Stage 1: Team. This stage, usually involving both agile and waterfall teams, reflects where most organizations are today. The teams are learning how to work together faster, more efficiently, and with greater quality. This may not be apparent to everybody. Management is still using traditional budgeting, finance, and planning methods, and the middle and top management layers have very little insight into the work being done at the team layer.

Stage 2: Program. At this level, the transition has expanded from the team to the program level. Teams have transitioned to agile and are realizing its value, but the product portfolio is still very much using a traditional waterfall approach. The organization's primary challenge becomes how to translate planning and progress reports between those charged with driving strategy (portfolio) using traditional methods and those charged with execution (program and team) using agile methods.

The next two stages—portfolio and enterprise—are where organizations aspire to reach on their agile journey.

Stage 3: Portfolio. The organization is managing innovation-driven projects with an unknown set of requirements using

FOUR LAYERS OF SCALE

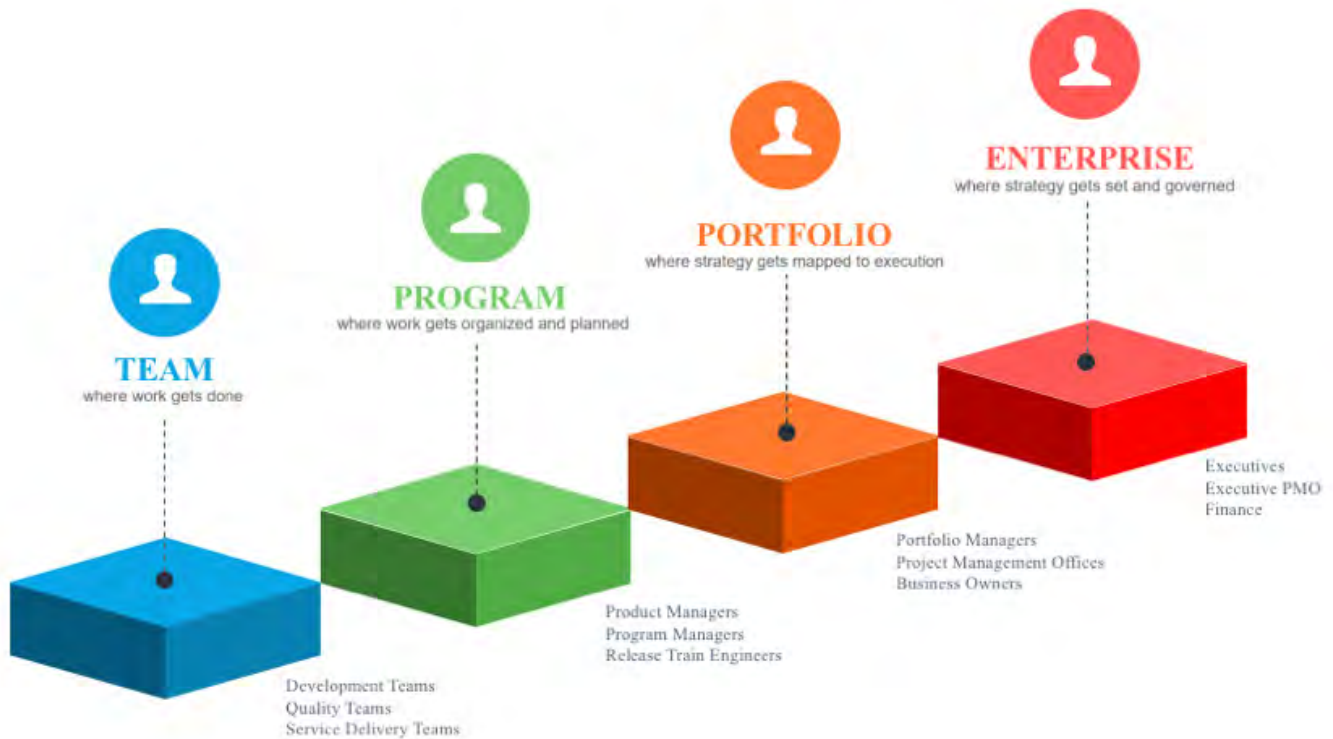


Figure 2: The four stages of agile transformation

an agile process. Highly predictable projects with well-defined scope, on the other hand, are managed in a more traditional way. A thin layer at the very top of the organization ties the two together through reporting, budgeting, and strategic alignment. Many in the analyst community would say that attaining this level is the core of the bimodal enterprise.

Stage 4: Enterprise. The enterprise level is the ultimate destination, where the executives know where to place their bets in real time. Executives can determine if there is enough value in one budget or if funds should be re-allocated to another budget. The ability to make those value decisions in real time and have those decisions drive all the way down to the teams doing the work bridges the gap between strategy and execution.

Staging Traditional Versus Agile

Figure 3 illustrates the four stages of a bimodal journey. The symbol “T” represents traditional methods and the symbol “A” represents agile methods applied at each stage. In the first stage

we see mostly traditional methods, and as we progress to the later stages, an organization transitions to a fully agile enterprise.

Making a Bimodal Approach Work

In a perfect example of taking a bimodal approach, development teams are engaged in long-term projects to build and maintain legacy systems, and other teams are charged with innovating to stay ahead of the pack. Management must provide the tools and techniques so that both can peacefully coexist.

Both modes offer significant advantages. By allowing some parts of the organization to keep operating using their existing model, organizations can maintain their current methodology while identifying other opportunities where agile thinking fits best. This approach

allows the organization to begin implementing the transformation to agile without disrupting current traditional processes.

In another example, a CIO of an insurance company or bank could manage an enterprise resource planning (ERP) modernization project using traditional methods. Process, tools, and outcomes

**MANAGEMENT MUST
PROVIDE the TOOLS
and TECHNIQUES
SO THAT BOTH CAN
PEACEFULLY COEXIST.**

are well defined, and the abundance of legacy data provides a competitive advantage over younger companies. Meanwhile, short-term objectives around online banking and bill pay or mobile applications can follow an agile approach to support changing customer, market, or regulatory needs.

Key Principles for Bridging the Bimodal Divide

For a bimodal approach to succeed, alignment, collaboration, and transparency between agile and waterfall must be enabled to bridge the divide between the two distinct approaches. It implies a deep understanding of culture and balance.

objectives when needed and agree to deliverables for longer timeboxed periods. This cadence also applies to those who prefer to plan continuously, implement a steady rhythm to plan releases, or synchronize teams and business units.

4. Define a single process to manage dependencies and risks. Create a unified way to manage risk and align handoffs among all teams. Map dependencies between work items and the teams that need to deliver them, and account for any differences in delivery time that could lead to risk. Tie dependencies to the risk value if the dependency is not fulfilled. Investigate recurring dependencies to reduce complexity in your projects and weaknesses in your architecture or organizational structures.



There are five essential ground rules that organizations must follow to successfully drive alignment to a bimodal approach:

1. Institute a common language. Align terminology and lingo early on, allowing agile and waterfall teams to effectively communicate. Do the development teams write requirements (traditional) or user stories (agile) to explain desired functionality? Is the complexity of a project measured in task hours (traditional) or story points (agile)? At the very least, try to create a Rosetta stone consisting of a comprehensive list of core terminology understood by all teams.

2. Construct a unified flow. Define a centralized intake process, preferably one that is business case-driven. This means projects are funded through a gated approach, where value is the key driver. When ranking the value of projects, be sure to examine all facets of a project—such as revenue potential, cost, time to complete, and required resources.

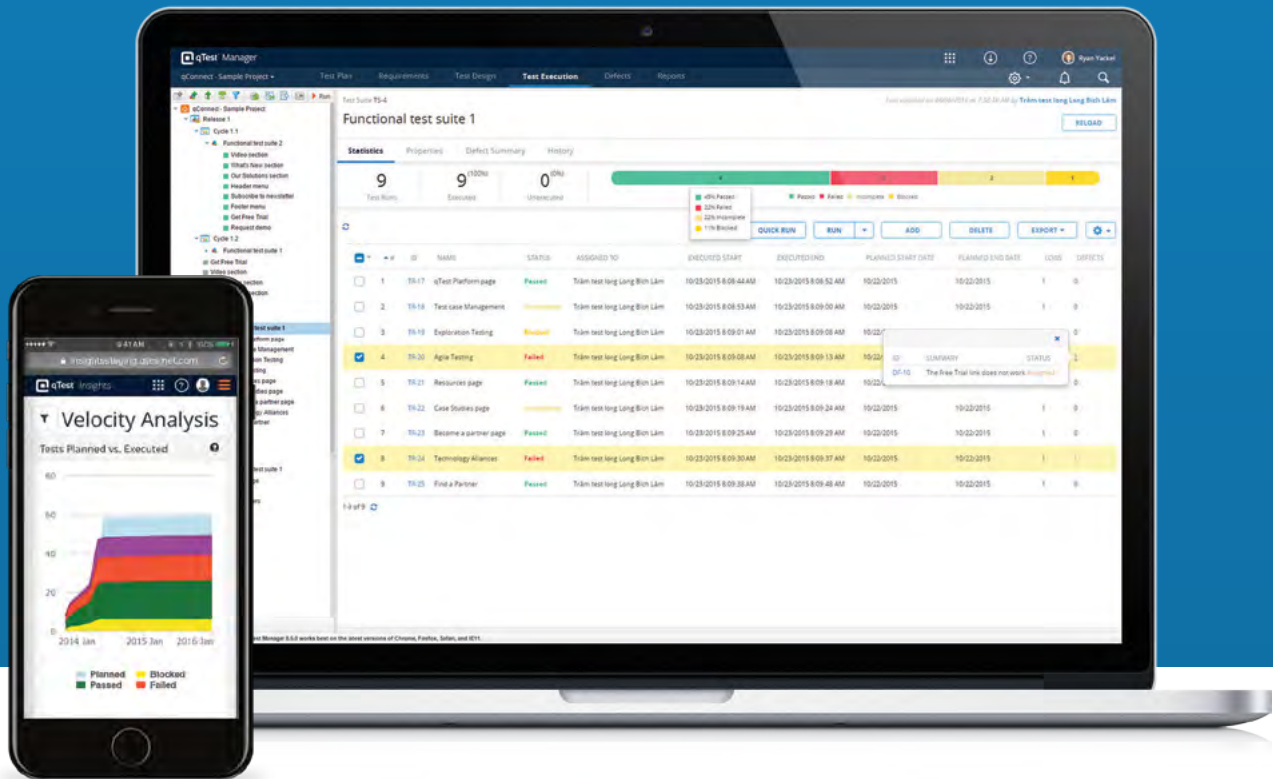
3. Implement a planning cadence. To align the way the business operates, have the organization conduct planning on a set cadence. Whether you use sprints, planning increments, monthly updates, or quarterly updates within an annual plan, recalibrate

5. Enable multilevel reporting. Provide staff at every level of the organization—from executives to portfolio managers to project managers to developers—with one source of truth. The same data that is used to report on project progress at the team level should also translate into value and funding at the program and portfolio levels, respectively, and business strategy at the enterprise level. Reporting data should flow as easily from the bottom up (spend, results, predictability) as it does from the top down (strategy, goals, priorities).

Enterprise agility is required to effectively compete in every industry, especially the technology arena. At the same time, traditional approaches can provide stability and proven experience that edge out the competition. Therefore, a bimodal approach is the reality of today’s digital transformations, and it may be here to stay. Driving alignment and coexistence between waterfall and agile will help your organization survive and thrive in this rapidly changing world. [BSM] steve@AgileCraft.com

CLICK FOR THIS STORY'S **REFERENCES**

WATERFALL, AGILE, OR DEVOPS? YES.

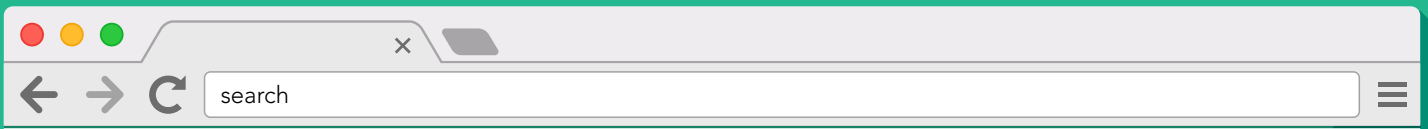


QASymphony is the creator of qTest —
A more efficient software testing platform
that any team can use with any methodology.

Until now you've been forced to choose between either the same old legacy tool from a QA dinosaur or a promising new tool from some startup that can't support enterprise-scale needs.

In the real world, you don't have the luxury of choosing between innovation and scale. That's why leading enterprises like Cisco, Salesforce, Barclays and Amazon rely on qTest by QASymphony. qTest helps hundreds of companies across the globe test smarter, test seamlessly and test at scale. Finally, a testing solution built for the real world.

Start a **Free Trial** Today at
www.qasymphony.com



Search

where to learn about the hottest topics in testing



TOP RESULTS:

STAR EAST

A TECHWELL EVENT

STAREAST Testing Conference - April 29-May 4, 2018 - Orlando, FL ...

<https://stareast.techwell.com/>

Come to STAREAST to get answers to all your toughest software testing questions:

- Testing in DevOps
- Test Transformation
- Test & Release Automation
- Big Data, Analytics, A/I Machine Learning for Testing
- Agile Testing
- Testing for Developers
- Security Testing
- Test Strategy, Planning, and Metrics
- Test Leadership
- Performance Testing & Monitoring

Register early for the best pricing plus a \$50 bonus gift card!

[SEE PRICING & PACKAGES](#)

SCALING AGILE THINKING THROUGH EMPOWERED TEAMS

A photograph of three hikers climbing a steep, rocky mountain slope. They are silhouetted against a bright, overcast sky. The hiker at the top is reaching out to help the middle hiker, who is in turn helping the bottom hiker. They are all wearing backpacks and climbing gear. The background shows a vast, rugged mountain range with patches of snow or light-colored rock.

**BOB
COSTELLO**

Since its publication sixteen years ago, the Agile Manifesto and the application of its principles and values have become a successful framework for creating productive software development teams. Agile thinking—emphasizing team communication, cooperation, and cohesion—has grown increasingly pervasive as a way to promote a dynamic oriented toward team empowerment. The Agile Manifesto gives us the language to talk about that empowerment and provides a framework for implementing autonomy.

Why is autonomy so important to team success?

This compelling question is driving a growing interest in the science behind motivation and its relevance in an increasingly

WE DO OUR BEST WORK WHEN WE FEEL THAT IT HAS VALUE OR MEANING.

complex and global economy. At the heart of this research lies a fundamental relationship between meaning and motivation.

The Importance of Empowerment and Autonomy

In her book *Why Motivating People Doesn't Work ... and What Does* [1], Susan Fowler identifies two important breakthroughs in motivation science:

- **The desire to thrive:** The need to find meaning in our work is innate in all of us. When our contributions feel effective, valued, and purposeful, we will experience greater satisfaction, engagement, and productivity
- **Meeting psychological needs:** We also require our psychological needs to be met based on competence, relatedness, and autonomy

Our natural tendency toward positive motivation is optimized when both our desire to thrive and our psychological needs are met. Conversely, when our social context fails to satisfy these needs, this inherent tendency is thwarted. This research is highly suggestive of an underlying connection between meaning and motivation. We do our best work when we feel that it has value or meaning.

Agile thinking is built on a foundation of leadership, supported by the pillars of mindset, self-organization, and practice. Interestingly, these three agile pillars provide a supportive framework for meeting the psychological needs of the individual.

Based on this research, it should come as no surprise to discover that autonomous agile teams are more productive and more highly motivated than other, more traditional team models. When we are given the freedom to self-govern at work, we feel more invested in that work because we have the power to affect both the process and the outcome. We are motivated to do good work when we feel connected to, challenged by, and proud of that work.

Scaling Agile Thinking

It is the success of autonomous software teams that has executives attempting to scale the Agile Manifesto and agile thinking to nonsoftware development teams, with the end goal of transforming into an agile organization.

These attempts have been met with minimal success. The “11th State of Agile Report” reflects a rather grim trend for agile scaling. [2] When respondents were asked what the top three challenges to adopting and scaling agile were, 63 percent stated organizational culture at odds with agile values, 47 percent stated lack of skills or experience with agile methods, and 45 percent stated lack of management support.

The report supports the idea that successfully scaling agile depends on leadership that is empowering and supportive. This is true at both the organizational and team levels. In my experience, true autonomous teams cannot thrive within a hierarchical organization. Support for autonomy must come from the top down.

How Leadership Influences Team Success

Teams depend on leadership for empowerment. A team cannot effectively self-regulate and self-govern if the prevailing leadership does not empower them to do so. The success of autonomous teams is based almost entirely on the leader's ability to establish an empowering team dynamic. The presence or absence of certain leadership behaviors can even predict whether or not a team will succeed.

Researchers have identified five leadership behaviors that are essential to effectively empower teams: leading by example, coaching, participative decision-making, informing, and showing concern while interacting with the team. [3] These behaviors lead to the success of autonomous teams because each emphasizes the role of the individual team member rather than the team leader.

Making Leadership Intentional

Intentional leadership refers to the ability to effectively employ these behaviors to promote team growth, empowerment, and productivity based on the specific needs of the team. Understanding that team empowerment boils down to leadership behaviors rather than characteristics is encouraging for leaders and teams alike because it establishes the learned, rather than innate, nature of leadership.

These behaviors reinforce a dynamic that empowers team members to feel meaningful within the team. Here are some examples of intentional leadership:

- Reinforce the understanding that each team member plays an important role and has value within the team dynamic

- Establish your own ability as a leader to put the group before yourself
- Encourage the active participation of each team member and provide support when facing unexpected challenges
- Advocate the importance of learning through experience by handling mistakes as opportunities for long-term growth

BEFORE YOU CAN SUCCESSFULLY SCALE AGILE THINKING, THERE MUST BE MEANINGFUL EXECUTIVE ENGAGEMENT, INVESTMENT, AND COMMITMENT TO THE TRANSITION.

When you tailor these behaviors to meet the needs of your team, you are leading with intention. Intentional leaders act in accordance with the needs of the team—not the other way around.

When you choose to lead with intention, you acknowledge that you play a fundamentally important role in the empowerment and long-term success of your team. Ultimately, intentional leadership nurtures a dynamic that prioritizes long-term development over short-term gain. This commitment to potential makes team members feel valued and engages their psychological desire to find their work meaningful.

According to a study published in the Ivey Business Journal, “Just as successful autonomous teams praised support from top managers for their accomplishments, participants on failed autonomous teams blamed weak support from management for their failure.” [4]

How Leadership Influences Scaling Agile

When transitioning from a more traditional, hierarchical structure toward one that is autonomous and self-governing, organizations face an interesting and challenging paradox: The ability to self-organize is contingent on the strength and support of the prevailing leadership. In the absence of skilled leadership, autonomous teams will fail to thrive. This apparent paradox is clarified when we examine the relationship between empowerment and autonomy.

For companies replacing more traditional hierarchical management structures with agile teams, commitment to the transition

must come from the top down and must penetrate the entire organization. The same Ivey Business Journal study found that an organization-wide commitment to autonomous teams is one of the most powerful factors behind team success.

This makes sense when we consider the role of the team leader within an organization. Team leaders report up the chain of command in any organization and are therefore subject to the same empowering leadership behaviors they are expected to instill. Meaningful engagement at the executive level encourages C-level investment and commitment to an agile transition. Empowered team leaders will, in turn, empower.

Scaling, or implementing agile beyond the software development team, cannot be blindly implemented. To successfully scale agile, the organization will most likely need to reframe and redefine agile concepts in a language that speaks to the culture of the organization itself.

Spotify is a perfect example of a company that uses a scalable agile framework specifically for its culture and business needs while staying aligned with the agile mindset and principles. Spotify developed its own vocabulary, calling autonomous teams “squads.” Each squad is accountable for a discrete aspect of the product, which it owns cradle to grave. Squads have the authority to decide what to build, how to build it, and with whom to work in order to make the product interoperable. The key to the success of this model is the ongoing executive commitment to nurturing and supporting agile thinking.

Agile Frameworks to the Rescue

Agile frameworks such as the Scaled Agile Framework and Large Scaled Scrum must address executive goals related to coordination across groups and organizational goals within teams, as well as the impact on the organizational culture and structure.

While it is tempting for executives to seek out a one-size-fits-all model, thinking it will ease the challenge of applying agile thinking concepts to the organization, the complexities of leadership, business, and culture demand an adaptive approach. Scaling frameworks should not be prescriptive. They can be a useful starting point for conversation, but at a certain point companies must choose best practices based on their own needs and goals.

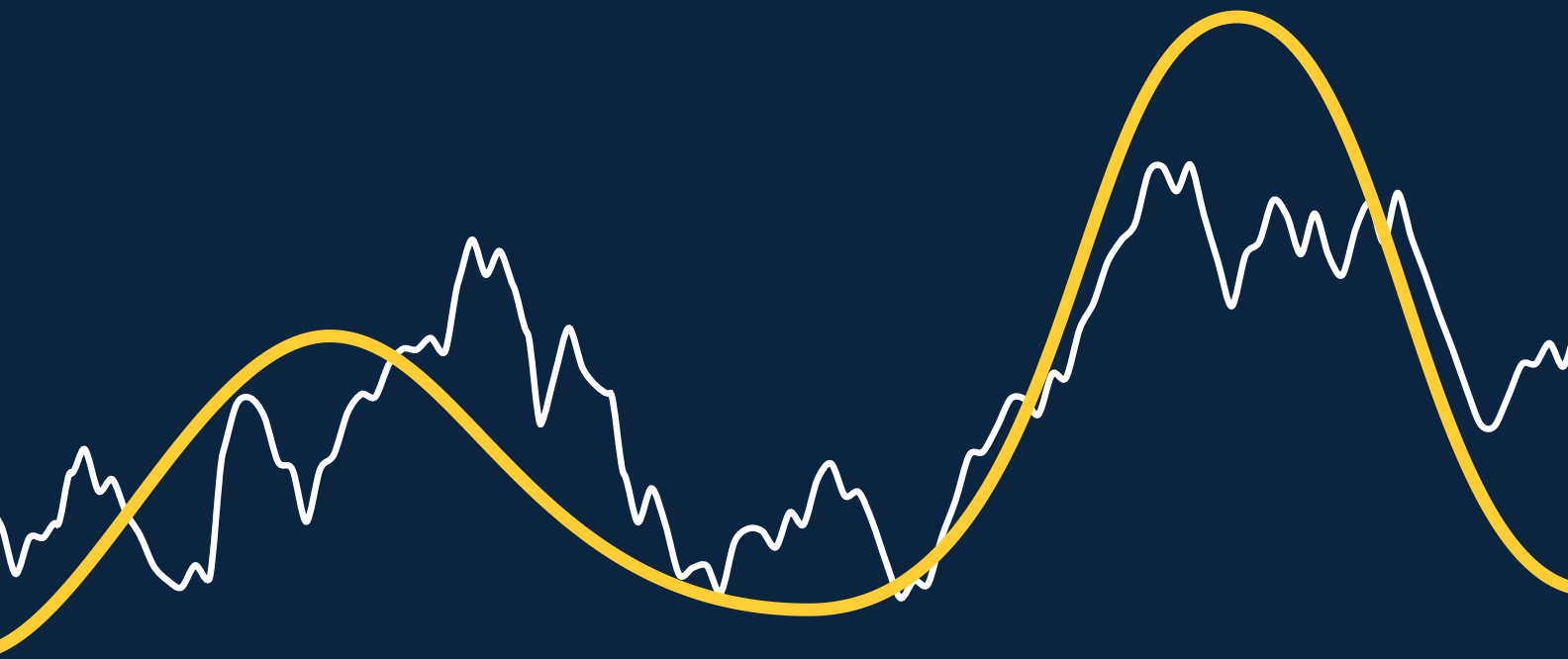
Like the Spotify model, the scaled framework must embody the needs of the entire organization to ensure the long-term success of agile thinking—and, by its very nature, it must be agile enough to evolve and adapt with the culture.

Before you can successfully scale agile thinking, there must be meaningful executive engagement, investment, and commitment to the transition. In order to empower the teams and promote their autonomy, intentional leadership must be practiced at all levels of the organization. [\[BSM\] rcostel@comcast.net](mailto:rcostel@comcast.net)

CLICK FOR THIS STORY'S **REFERENCES**



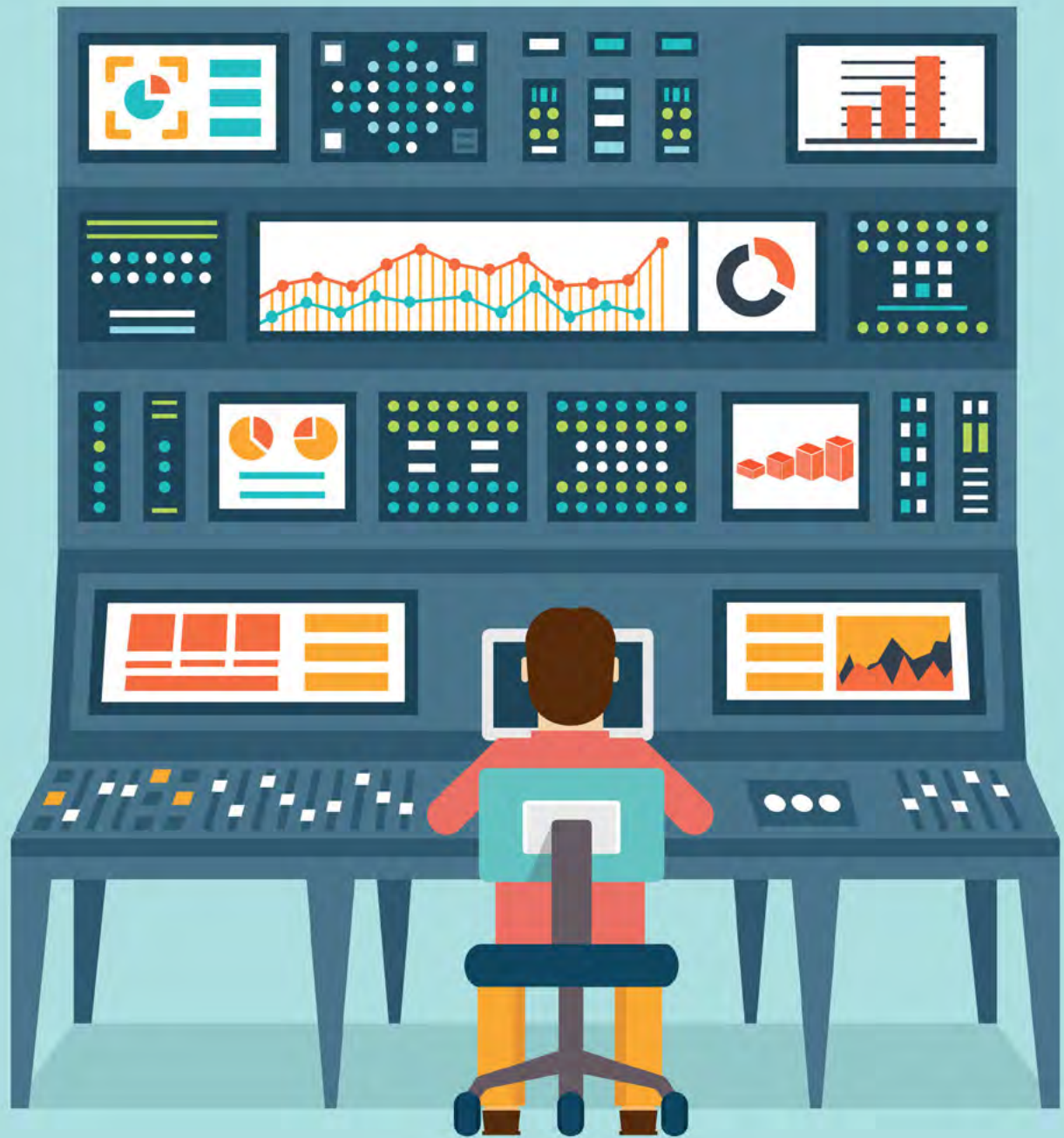
Simplified, massively-scalable
load testing on demand



start your free trial now

flood.io

THE IMPACT OF QUALITY INTELLIGENCE ON DEVOPS AND THE IOT



BY STEVEN WINTER

The role of quality intelligence has dramatically increased in popularity in the past couple of years in terms of its pivotal role in our “continuous everything” industry. No longer is waiting for days, weeks, or even months to get test results acceptable for any company that cares about keeping pace in their market. Immediate and trustworthy quality metrics need to be available at every gate of the software delivery chain—and anything less is a losing proposition.

On top of DevOps rapidly becoming the normal way of a developer’s life, the Internet of Things (IoT) explosion is bringing a staggering amount of new data. And if that’s not enough, mass automation means that every day more and more tasks are being delegated to software automation or automated machines. From fast food cashiers being replaced by touch screens to auto assembly lines staffed by highly efficient robotics, our industry is automating all kinds of testing and delivery tasks. With universal income becoming a strong consideration to combat robots from taking over, automation has become the key differentiator in the growing age of continuous delivery and deployment.



What new key performance indicators (KPIs) or service-level agreements must we measure now? How do we manage aggregating this new onslaught of data? How do we move faster in our analysis and reporting so we can make faster and better decisions for our business?

Quality Intelligence Moves Mountains

Before we dig into the impacts and solutions, let’s define quality intelligence (QI). QI is all about providing fast and accurate test data to make better decisions. It is the core output of our testing services. It’s not about the tests, the test environments, or even the skills of the testing team. It’s about test results. And these results must be believable without question.

Though QI has been at the heart of software delivery through the ages, never has its criticality and necessity been as important as it is today. Every major online retailer leverages some form of continuous deployment that is enabled by continuous testing.

Testing at code commit via continuous integration, mass and continuous regression testing, daily performance and security tests, A/B testing against live production, and advanced rollback

procedures are all based on testing and the scrupulous review of the results. This enables immediate business decisions that directly and instantly affect the company’s bottom line.

A successful QI approach takes into account all critical levels of business. This includes unit and regression testing, performance and stress testing, security, real-time performance profiling, critical service uptime, mean time to defect detection and resolution, and customer satisfaction scores. The power of testing results is clearly the enabler for success.

The Relationship between Quality Intelligence and DevOps

It should be no surprise that we’ve continued to see a steady increase in DevOps adoption. According to RightScale’s latest cloud survey, DevOps adoption increased among respondents from 66 percent in 2015 saying they were embracing DevOps to 74 percent in 2016. [1]

The reason DevOps works is because teams who are doing it right have made the commitment and investment in creating continuous testing throughout the development and delivery chain.

In my experience, you cannot have continuous anything if you do not have continuous testing supporting it. And remember, it’s actually not about the testing—it’s about the test results. It’s about the QI necessary to move the software down the line to production and keep it there.

With this continuing advancement of DevOps adoption, many teams are running into big challenges when making the transition. Issues like lack of staff experience, immature technologies, and lack of skills working with containerization and configuration tools can sidetrack a team from staying focused on their QI.

A list of challenges would not be complete without mentioning security. The mere fact of speed being one of the key results of a successful DevOps solution can wreak havoc on a team that doesn’t invest in improving their security development and testing practices. Because things move faster, you need to test faster and keep pushing security best practices left. And don’t forget about the increased skills needed to enable that faster security practice.

All these challenges put together pale in comparison to what I see as a deeper and more fundamental challenge for these teams: to establish clear measures of value and success before any efforts are made to adopt DevOps practices.

During the past decade of my attending, presenting, and speaking at software quality conferences, I have heard this mistake as the root of evil for so many. DevOps is not the answer but the enabler, and many organizations have made the adoption decision in the name of good faith. Taking on a shiny new methodology toy won’t solve delivery woes if you’re unclear on what success looks like on the other side.

I recently interviewed Sofia Palamarchuk, CEO of Abstracta, on this very subject. Her testing and automation service company works with clients ranging from waterfall organizations all the way up to advanced and highly efficient DevOps teams.

“We have seen clients jump into DevOps and not be clear with what they need to know. Didn’t take the time to identify their specific KPIs to enable their business,” Palamarchuk said. “It is not the goal to be DevOps, it is to be successful in your business. We make sure to identify those measures and help build the testing functions to provide usable test results to achieve those goals.”

The Importance of a Quality Intelligence Dashboard

In my last development project, we had the opportunity to start from the ground up and not be burdened with a legacy system that needed to be transitioned. Our solution was to create a quality intelligence dashboard as our first step. This should take place before coding or writing any tests. We worked backward from the point of a single piece of software being deployed to production, identifying all the necessary gates allowing for that deployment to take place automatically and with trust. We kept working backward until we identified the tests that needed to be in place to support a developer checking in code to the master repository.

As a result, we created a focused and specific dashboard consisting of gates and measures that completely supported our continuous delivery goal. And it worked. We didn’t know how to fill out the dashboard at first, but that was just a coding logistics problem. Given that we started with the end in mind, we had a small list of what needed to be developed to make that QI dashboard a reality.

I recently had the opportunity to interview Fred Beringer, VP of Engineering, Photos, and Digital Customer Experience at Shutterfly. Fred described the type of metrics they have in place to support their enterprise-level continuous deployment strategies for the millions of photo greeting cards they produce each day during their peak holiday season. The types of KPIs they track are highly usable at every level of the company—from the CEO to the entry-level developer.

Fred breaks down Shutterfly’s metrics into two main categories: quality and efficiency. At the engineering quality level, they track code coverage, functional testing results, backward compatibility testing results, performance profiling for both web and mobile, daily load and performance test results, and a testability score that ensures every build is worth testing. From the efficiency view, they track number of deployments per team, number of patches per team, mean time to detect a defect, and mean time for defect resolution.

From the management and monitoring view, covering both quality and efficiency, they constantly monitor services uptime and availability, page speeds, mobile image upload speeds, success rate of mobile image uploads, stability of apps, and the customer satisfaction scores generated directly from their customers. And because all developers are tied directly to the code they deploy, the immediate QI feedback they get allows for faster and more accurate responses when issues arise.

While these examples are only a partial list of their metrics, this real-time QI data allows them to not only ensure the highest level

of quality for their customers but also move from being generally reactive to aggressively proactive in managing their bottom line.

Shutterfly’s commitment to a quality intelligence strategy is a glowing example of how to do it right. Given their DevOps implementation, their organization has invested in a mass amount of test automation to successfully support an ecosystem of mobile and web platforms generating millions of orders every day.

The Impact of Quality Intelligence on the IoT

Many of you have heard the hype around the IoT in terms of its staggering number of connected devices and projected market revenue. Depending on whom you ask, the projected number of connected devices could exceed thirty billion by 2020, with revenues exceeding \$470 billion by that year and overall infrastructure investments topping \$60 trillion in the next fifteen years. Whereas 2016 was the year of media attention and growth, 2017 is the year of rollout and monetization. [2]

While IoT growth projections are extraordinary, the ability to deliver quality solutions rests on the actual adoption, implementation, and support—and more importantly, on having a tight quality intelligence strategy. IoT technology generates huge amounts of user data that needs to be aggregated, analyzed, and leveraged.

Beyond the noise of mega data, there are very real challenges around the IoT that can distract us from getting the quality intelligence results we need. The sheer volume of connected devices creates highly dynamic environments, which mean extremely complex use cases. Accuracy and scalability of the necessary test environments create additional challenges. And the security and privacy of IoT systems have great room for improvement, which was clearly illustrated by the major attack from IoT devices that caused widespread web outages in October 2016. [3]

It Is Time to Take Quality Intelligence Seriously

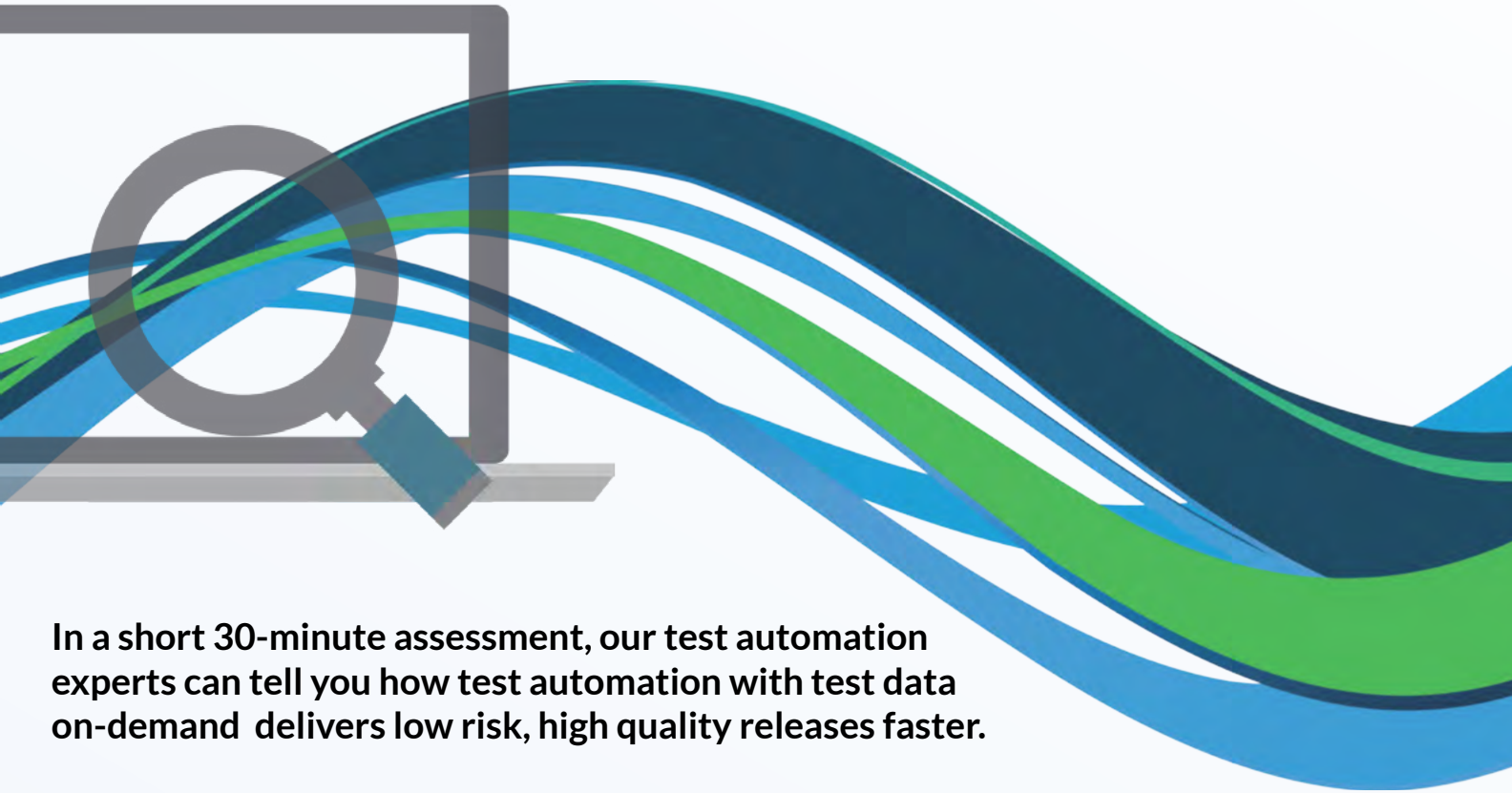
There are plenty of daily testing issues to distract you from staying focused on what’s most important to make better decisions faster. It’s essential to align your quality intelligence strategy to the data points that move your business forward. From the success of checking in a single line of code to monitoring the uptime of your key services, these data points are sometimes fueled by hundreds of deployments a day.

Make sure you have the right skills on your team to best analyze QI data, and be bold enough to deprioritize or even cut tests if they don’t matter anymore. Managing the increased amount of new data from DevOps, the IoT, and mass automation should be propelling your delivery and helping you focus on the right measures. [BSM] steven@guerrillaqa.com

CLICK FOR THIS STORY'S **REFERENCES**



Learn how **ACCELERATE** test cycles,
MITIGATE risk, and **DRIVE DOWN** costs.



In a short 30-minute assessment, our test automation experts can tell you how test automation with test data on-demand delivers low risk, high quality releases faster.

Plus, use our simple ROI Calculator to give you an indication of how much you can save with TurnKey test automation.

Get a 30 Minute FREE, HONEST Assessment
turnkeysolutions.com/free-assessment/ | 844.428.4678

Test Earlier and Faster | Streamline Test Creation | Slash Maintenance Costs Up to 80%

A brief history of web and mobile app testing.



BEFORE SAUCE LABS

Devices. Delays. Despair.

AFTER SAUCE LABS

Automated. Accelerated. Awesome.

Find out how Sauce Labs
can accelerate your testing
to the speed of awesome.

For a demo, please visit saucelabs.com/demo
Email sales@saucelabs.com or call (855) 677-0011 to learn more.



Testing at the speed of awesome.



ATTEND LIVE, INSTRUCTOR-LED CLASSES VIA YOUR COMPUTER.

Live Virtual Courses:

- » Agile Tester Certification
- » Software Tester Certification—Foundation Level
- » Fundamentals of Agile Certification—ICAgile
- » Testing Under Pressure
- » Performance, Load, and Stress Testing
- » Get Requirements Right the First Time
- » Essential Test Management and Planning
- » Finding Ambiguities in Requirements
- » Mastering Test Automation
- » Agile Test Automation—ICAgile
- » Generating Great Testing Ideas
- » Configuration Management Best Practices
- » Mobile Application Testing
- » and More

Convenient, Cost Effective Training by Industry Experts

Live Virtual Package Includes:

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

Featuring fresh news and insightful stories about topics important to you, TechWell Insights 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. The following is a sample of some of the great content you'll find. Visit [TechWell.com](https://www.techwell.com) for the full stories and more!

Is Agile versus DevOps a Real Debate or a Misconception?

By Josiah Renaudin

Some organizations are seeing all the buzz surrounding DevOps, not quite reaping the benefits from agile that they had hoped, and putting all their eggs in this new basket. However, if you ask Jeff Payne, the CEO and founder of Coveros, you don't often have DevOps without incorporating some aspects of agile.

[Read More](#)

Getting Started with API Testing

By Justin Rohrman

For testers interested in moving into test automation, Justin Rohrman usually suggests that they get started by learning to write a little code—more specifically, working with their software's API. You don't have to jump into tooling just yet. You can begin testing the data and workflow of the API first. Here's how.

[Read More](#)

The Digital Transformation: Turning Lead into Gold

By Lee Copeland

For years, IT has been seen as support for organizations' primary internal functions. But IT can enable revolutionary processes—processes that generate revenue, rather than just consume it, and create innovation, helping organizations remain relevant and competitive. How can you transform this idea of lead into gold?

[Read More](#)

Ensure That Your Current Cloud Solution Will Stand the Test of Time

By Pete Johnson

It's still early in the lifecycle of cloud adoption. This means certain cloud vendors and technologies will fall by the wayside as adoption takes on critical mass. How, then, do you future-proof your cloud solution to make sure you don't make a decision that you'll regret later? Here are three ideas to consider.

[Read More](#)

Catch Small Failures Early with Agile Practices

By Ryan Ripley

Agile is designed to keep failures small and manageable. It's essential to be able to talk about small failures and ways to improve during the retrospective so that the teams can advance their agile practices. If your teams can't talk about their small failures openly, there is a great risk of bigger troubles soon.

[Read More](#)

Don't Let Too Little Planning Tank Your Agile Adoption

By Alan Crouch

Many organizations turning to agile believe it means you don't have to do any planning. This couldn't be further from the truth. A healthy agile team does just as much (if not more) planning than a team using a waterfall methodology. Preparing and setting goals sets up the team for a more successful agile adoption.

[Read More](#)

Something to Remember When Managing Change at Work

By Naomi Karten

Grieving is usually associated with the loss of a loved one, but it's equally relevant to loss triggered by changes in the workplace. If you are introducing or overseeing a change, it's a mistake to belittle people's reaction to the loss or excessively tout the benefits of the new way. Be sensitive to those affected.

[Read More](#)

Avoiding Continuous Bugs: Speed and Quality in DevOps

By Jeffery Payne

Lots of DevOps initiatives focus on speed and frequency of deployment without an emphasis on quality. Bad testing practices in DevOps only deploys buggy software faster. Here are some tips to move toward a more effective testing process that supports a continuous delivery approach—without sacrificing quality.

[Read More](#)

A Beginner's Guide to Mobile Application Testing

By *Max Zheleznyy*

Mobile is a great area to focus on for testers who are looking to expand their skills. So, what do you need to know before you can start testing mobile applications? Here's some information about tools and techniques that can help you get a good start in the expanding arena of mobile application testing.

[Read More](#)

Data-Driven Testing Skills in an Agile and DevOps World

By *Michael Sowers*

For agile and DevOps, an understanding of the role of data analysis in the test strategy is helping teams accelerate development, testing, and deployments. As we continue to enhance our testing effectiveness, data analytics skills are an important dimension in managing risks in a “continuous everything” world.

[Read More](#)

To Improve Agile Teamwork, Think about the Individuals

By *Steve Berczuk*

Given the emphasis on teams, it can be easy to forget that agile has the value of individuals and interactions as a central principle. As much as an effective team dynamic is what makes Scrum work, teams are comprised of individual people, and it's important to acknowledge each person's role and to express appreciation.

[Read More](#)

Testing in the Pipeline

By *Hans Buwalda*

With DevOps becoming the norm, we're entering a world of pipelines. With frequent or continuous deployments, streamlining and automating the process of building, configuring, testing, and releasing developed software components becomes a high priority—with testing posing its own unique set of challenges.

[Read More](#)

If You're Working Too Much, Is It a Challenging Project or Bad Management?

By *Payson Hall*

Projects sometimes encounter challenges that require team members to put in extra work. But if this is happening repeatedly, it's worth figuring out where the pressure is coming from. You may need to ask, “Is this project simply challenging, or is it being badly managed?”

[Read More](#)



Gearing Up to Test in the Augmented and Virtual Reality World

By *Mukesh Sharma*

With close to \$9 billion being funneled into virtual reality investments in the past five years and the number of users buying these devices continuing to rise, the augmented reality (AR) and virtual reality (VR) sectors are important areas in which testers should focus their efforts.

[Read More](#)

Naive Innovation in Software Testing

By *Rajini Padmanaban*

The idea behind naive innovation is that a person with the right mindset and passion for a certain area is a great candidate for innovation, even without having the same subject matter expertise as a domain expert. Rajini Padmanaban looks at naive innovation in software testing.

[Read More](#)

Can the IoT Help Save the Bees?

By *Pamela Rentz*

More than one-third of all crops in the US require insect pollination, and managed honey bee colonies are our main resource. According to the US Department of Agriculture, beekeepers have been steadily losing colonies, but the IoT may be able to help with some interesting projects.

[Read More](#)



How Agile Has Shrunk Documentation

THERE IS ALWAYS A BALANCE BETWEEN TOO MUCH AND TOO LITTLE DOCUMENTATION. SOME GUIDELINES ARE NEEDED TO IDENTIFY HOW MUCH IS NECESSARY.

by Sandeep Maher | sandeephmaher@gmail.com

One of the lines in the Agile Manifesto proclaims preference for “working software over comprehensive documentation.” Agile adopters often go ballistic about this line, particularly the part about documentation. It has been quoted, misquoted, and abused more often than perhaps the entire manifesto itself. Just how much documentation is really enough?

The house is divided on the role of documentation in the software development world, and despite the rapid acceptance of agile, this subject remains as contentious as ever.

Traditional Documentation: A Beacon and a Burden

Historically, software code has relied on a range of documentation. It has taken various forms, such as specifications and design documents, to define an application in a way that supports its existence, implementation, and evolution. Documentation has supported customer requirements, functional specifications, and information that supports design, testing, installation, and operation.

Both code and documents are associated with contrasting motivations. Coding gels with the geek, hip, and creativity while documentation can be viewed as the equivalent of a burdensome donkey, needed but never respected. Software documentation is always expected to be complete and comprehensive, with rich content, to serve as a beacon so that the software is developed and tested as expected.

The business analyst (BA) traditionally has been responsible for creating these detailed specifications, supported by flow diagrams, wireframes, field descriptions, validations, and quality attributes related to performance, security, etc. The developers and the testers engaged themselves with the documentation details, sometimes interpreting them correctly—and sometimes not. When they missed or messed up with the requirements, bugs and change requests resulted.

The BA was pressured to get the details right and was hounded if the document missed a point or was prone to ambiguity. On the other hand, the BA's expectations rightfully were that the team should think more, ask questions, and interpret implicit as well as explicit requirements.

Software documentation, was both a refuge and an excuse, resulting in much debate and conflict. The documentation usually never stayed current since the impetus or inclination to update it reduces with the passage of time while the software being built gets more attention. There were soldiers of quality who would collaborate and question the document contents and its interpretation, helping it evolve to become more correct and precise. In hindsight, the collaboration and the documentation's evolution were harbingers of change waiting to happen.

It appeared that agile had sounded the death knell for documentation as we knew it.

Agile Cometh

The advent of agile created a new paradigm consisting of rapid, iterative development, quick feedback cycles, customer collaboration, and automation. Roles changed, too. The BA has become the product owner, the project manager has been replaced by a Scrum Master, and the developers and testers have united as cohorts on the team. Discussions grew in scale and the spoken word took priority over the written one. It appeared that agile had sounded the death knell for documentation as we knew it.

Enter the *user story*.

User stories list requirements, and the conversation about the story description and acceptance criteria needs to fit on an index card. The granular nature of stories encourages plenty of discussion and refinement by the team and the customer.

Compare this approach with the non-agile era, when the BA would write the software documentation largely in isolation,

attempting to remember the interviews and conversations with the customer. Product documentation, due to its large size, was never prone to as much focus or discussion as user stories.

Collaboration Replaces the Document

In the agile world, the reliance on collaborative thought processes builds and enables application solutioning, thus reducing the need for detailed documentation. Agile helps the stakeholders to imagine the application-to-be before it is developed. The virtual forest view of the application landscape precedes the incremental pieces developed over time. The increments are driven by intense discussions among the team and with the customer representatives. These discussions help not only in writing the stories but also in building the shared application knowledge within the team from a very nascent stage. The discussions usually bring out risks and assumptions, and clarify the understanding for the team. Ultimately, the customer benefits from these building blocks that lead to successful iterations and application releases.

The smart product owner provides the team with a holistic view of the solution using tools such as story maps, user journey maps, as-is and to-be views, and clickable wireframes. These visual artifacts facilitate intuitive interaction, better insight, and higher knowledge retention by the team. Provided that the right people participate in these discussions, better-shared understanding and faster decision-making leads to accelerated application development.

Traditional waterfall approaches require upfront, detailed documentation that struggles to stay current with burgeoning application functionality. With agile, design documents or test plans have given way to visual diagrams and mind maps. Detailed test cases

have morphed into a behavior-driven development style of test definition. The use of code analyzers and automated testing tools has led to traditional coding standards and system test cases not being as prescriptive and voluminous as they used to be.

Long Live Documentation!

Despite this evolution, software documentation shouldn't be made out to be a villain. After all, user stories are still a form of documentation, albeit a sleeker, more easily adaptable version of it. The value of usable documentation to inform, educate, and serve historical need must never be forgotten. Its content must survive and stay relevant to the needs of the team and to the context of working software.

There is still a benefit that documentation can serve by taking on a variety of written and visual forms. If a team thinks it necessary to create documents that will assist their end-users to understand or use the application better, then they should do so. Any investment in documentation must provide value. For example, if a team thinks a mind map could be more evocatively expressive of an application's many components and interactions than a bloated, unreadable document, then so be it.

A long-standing misconception is that agile advocates for no documentation. In reality, an agile team does as much (if not more) discussion around requirements and expectations than a team using a waterfall methodology and traditional documentation. The team simply defines application criteria using more communication and collaboration assisted by visual aids, which sets them up for fewer misunderstandings, fewer change requests, and ultimately, more successful releases—which means happier customers. **[BSM]**

LINK TO OUR ADVERTISERS

Agile Dev, Better Software & DevOps East	2
CAST Software	11
Delphix	13
iSQI	17
Parasoft	3
QASymphony	22
Sauce Labs	32
SQE Training—Live Virtual	33
SQE Training—On-Site Training	38
STAREAST	23
Tricentis	27
TurnKey Solutions	31
ZapTest	4

DISPLAY ADVERTISING

advertisingsales@techwell.com

ALL OTHER INQUIRIES

info@bettersoftware.com

Better Software (ISSN: 1553-1929) is published four times per year: January, April, July, and October. Entire contents © 2017 by TechWell Corporation, 350 Corporate Way, Suite 400, Orange Park, FL 32073 USA unless otherwise noted on specific articles. The opinions expressed within the articles and contents herein do not necessarily express those of the publisher (TechWell Corporation). 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.

TRAIN YOUR TEAM ON YOUR TURF



BRING THE TRAINING TO YOU

Software Tester Certification—Foundation Level
Mastering Test Design
Agile Tester Certification
Agile Test Automation—ICAgile
Integrating Test with a DevOps Approach
Mobile Application Testing
And More!

60+ ON-SITE COURSES

 **40**
TESTING
COURSES

 **7**
MANAGEMENT
COURSES

 **9**
REQUIREMENTS
COURSES

 **4**
DEVELOPMENT
AND TESTING
TOOLS COURSES

 **17**
AGILE
COURSES

 **2**
SECURITY
COURSES

For more than twenty-five years, TechWell has helped thousands of organizations reach their goal of producing high-value and high-quality software. As part of TechWell's top-ranked lineup of expert resources for software professionals, SQE Training's On-Site training offers your team the kind of change that can only come from working one-on-one with a seasoned expert. We are the industry's best resource to help organizations meet their software testing, development, management, and requirements training needs.

With On-Site training, we handle it all—bringing the instructor and the course to you. Delivering one of our 60+ courses at your location allows you to tailor the experience to the specific needs of your organization and expand the number of people that can be trained. You and your team can focus on the most relevant material, discuss proprietary issues with complete confidentiality, and ensure everyone is on the same page when implementing new practices and processes.

IF YOU HAVE 6 OR MORE TO TRAIN, CONSIDER ON-SITE TRAINING

[SQETRaining.COM/ON-SITE](https://sqetraining.com/on-site)

