

SUMMER 2018

BETTERTM SOFTWARE

A TECHWELLTM PUBLICATION

GREAT BIG AGILE AN OS FOR AGILE LEADERS

All Customers Are Not Created Equal

*Learn how to
improve your
skills to improve
customer
satisfaction*

Supporting Continuous Testing with Automation

*DevOps won't be
the same after you
master the art of
test automation*



INSIDE

Volume 20, Issue 3
SUMMER 2018



14

On the Cover

Great Big Agile: An OS for Agile Leaders

Following agile ceremonies may make an organization feel good, but that's only a start. "Great big agile" requires leadership at all levels to focus on self-organization and empowerment as a universal framework.
by Jeff Dalton

Features



22

All Customers Are Not Created Equal

Software developers may not think they have much to do with customers, but it is wise to consider the customer in all you do, from collecting requirements to design and implementation.

by Regina Evans



26

Supporting Continuous Testing with Automation

Create tests that can be used in a continuous testing environment, build the right number of tests, and don't fall victim to the mistaken belief that everything can be automated.

by Bas Dijkstra



30

QA Is More Than Being a Tester

QA testers often take on more of a role than just testing software code. When the team needs help, QA should lend a hand in assisting with business analysis, customer communication, user experience, and user advocacy.

by Amanda Perkins

Columns

09 TECHNICALLY SPEAKING

Using Agile and DevOps to Achieve Quality by Design

When software nears completion, it is the wrong time to focus on quality. Product delivery improves if you invest in a plan, validate in small increments, and focus on continuous testing.

by Mike Sowers

35 CAREER DEVELOPMENT

Be Indispensable: Cross-Train like a Testing Athlete

Stretching your skills and preparing for organizational and work-related change is vital for any QA tester. Your best approach is to work on complementing your skills and competencies.

by Bonnie Bailey

Departments

06 Mark Your Calendar

07 Editor's Note

08 Contributors

12 Interview with an Expert

33 TechWell Insights

36 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.

Stepping Back into History

Sometimes it can be fun to look back at history. *Better Software* magazine has provided innovative and pragmatic information for the software development and testing community for more than twenty years. TechWell has focused on finding the best articles that can benefit its readers over the long term. By avoiding deep dives into specific technologies that could easily become outdated with new innovation, we've concentrated on principles that will serve you well beyond the latest fad.

Better Software articles will help you create—no surprise—*better* software! The magazine started out as *Software QA* magazine in the mid-1990s with a focus on testing. The magazine was renamed to *Testing & Quality*, then to *Software Test & Quality Engineering* before finally evolving into *Better Software*. The design of the magazine has definitely changed along the way:



Whether you are a tester, developer, project manager, or another seasoned software professional, *Better Software* strives to help you improve the delivery of quality software products and services.

In this issue's cover article, "[Great Big Agile: An OS for Agile Leaders](#)," Jeff Dalton gives us something to think about in terms of how an agile approach shapes a company's culture. Regina Evans has an urgent call to arms on how to engage with customers in "[All Customers Are Not Created Equal](#)." For those of you who test for a living, you'll enjoy Amanda Perkins's article, "[QA Is More Than Being a Tester](#)." Over the past few years, we've focused on automation and continuous processes. But just because you use automated tests, it doesn't mean that product quality improves. To round out our feature articles, you'll learn three ways to avoid major test failures in Bas Dijkstra's "[Supporting Continuous Testing with Automation](#)."

In our regular "Technically Speaking" column, Mike Sowers offers some creative ways to bake quality into your product in "[Using Agile and DevOps to Achieve Quality by Design](#)." If you are looking for some career advice, Bonnie Bailey states the importance of expanding your knowledge in "[Be Indispensable: Cross-Train like a Testing Athlete](#)."

Thanks for helping us celebrate over twenty years of *Better Software* magazine. Enjoy this issue, folks.



Ken Whitaker

kwhitaker@techwell.com

Twitter: @Software_Maniac



FOLLOW US



PUBLISHER
TechWell Corporation

PRESIDENT
Alison Wade

DIRECTOR OF PUBLISHING
Heather Shanholtzer

Editorial

BETTER SOFTWARE EDITOR
Ken Whitaker

ONLINE EDITOR
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



Bonnie Bailey Bonnie Bailey is a software developer for an emergency services information technology company. Bonnie is an avid reader of science fiction and books on investing, the future of humanity, and software engineering. Bonnie can be reached at bonnie.bailey@motorolasolutions.com.



Jennifer Bonine is the VP of global delivery and solutions for tap|QA Inc., a global company that specializes in strategic solutions for businesses. Jennifer began her career in consulting and implementing large ERP solutions. She has held executive-level positions leading development, quality assurance and testing, organizational development, and process improvement for Fortune 500 organizations. Contact her at jbonine@tapqa.com.



Jeff Dalton is chief evangelist at AgileCxO, a research and development organization that studies organizational agile performance. He is an author, conference speaker, blogger, coach, and technology leader with more than thirty years in the software development industry. Jeff is author of "A Guide to Scrum and CMMI: Improving Agile Performance with CMMI." In his spare time he plays jazz bass in live bands and on recordings. Jeff can be reached at jdalton@agilecxo.org.



Bas Dijkstra is an independent professional helping teams and organizations improve their testing efforts through smart application of tools. He is also a trainer on various subjects related to testing and automation. Living in the Netherlands, Bas likes running, reading a good book, and listening to good (mostly classical) music. Contact Bas at bas@ontestautomation.com.



As an engineering program manager at NetApp, **Regina Evans** has been working in IT for more than fifteen years. Most of her experience has been working with customers in software development and test engineering. In addition to being a customer evangelist, Regina enjoys the beautiful state of North Carolina with her husband and two children. You can reach her at Regina.Evans@netapp.com.



Amanda Perkins has six years of experience as a quality analyst. With experience in customer service, business analysis, product ownership, and automated testing, she brings a unique mindset to test approaches. A constant student, Amanda is learning all there is to know about cyber security testing approaches. Amanda can be reached at agiese79@gmail.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. He 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.

GREAT BIG AGILE

AN OS FOR AGILE LEADERS



BY JEFF DALTON

As the prolific and popular SPaMCAST podcaster Tom Cagley proclaimed during his keynote at the 2017 Agile Leadership Camp, “Values aren’t really what matters; behavior matters.” Cagley, who has interviewed more than five hundred technology leaders on his podcast series, hit the nail squarely on the head. Culture is usually derived from organizational values.

The behavior of team members, business stakeholders, partners, and leadership is all that matters, as it demonstrates real, as opposed to stated, culture. Too many companies say the words while demonstrating antipatterns that proliferate throughout the organization. Instead, companies should project and promote behaviors that build scalable and sustainable self-organization at all levels.

While many leaders are asking about scaling agility these days, they’re asking the wrong question. They should be asking how to scale self-organization using a healthy dose of agile values, frameworks, and techniques.

From time to time, people have made attempts to transition from the “rules of men” to the “rules of nature,” a system that more closely mimics the natural world. This is where the scales are inverted, roles and accountabilities are dispersed throughout the organization, and people go about the sometimes messy process of organizing themselves without having to ask permission from any business leader.

In more than one hundred organizations we’ve assessed, tech leaders tell us that they want to push decision-making down and give their teams greater autonomy, but their behaviors are in conflict with agile values, creating an organizational type mismatch. In other words, the API is broken, and the architecture needs to change.

Introducing Great Big Agile

The concept of “great big agile” requires leadership at all levels, just not the kind we are used to. Simply working with an agile coach to implement well-known ceremonies is not enough. Metaphorically, the operating system needs an upgrade.

In today’s corporate hierarchies where command-and-control structures, low trust, long-term planning, and risk management reign supreme, the skills required to thrive and survive are anything but agile. This leaves agile teams to push the culture uphill, leading to unpredictable results once business operations expand beyond the boundaries of the core agile team. This creates chaos because information technology, operations, marketing, infrastructure, business development, sales, and end-users are not on the same page.

Agile without self-organization isn’t agile at all. There is nothing wrong with adopting ceremonies and techniques that are most commonly identified as being agile, and many companies have found some success with that, but the power of agile values and their associated frameworks grows exponentially once self-organization is perfected.

How Agile Is Your Organization?

I have witnessed only a few examples of large organizations that have been successful with true agility. Far more insist that they are agile by merely adopting a couple of techniques or ceremonies within an otherwise command-and-control, low-trust, and traditional operating model.

Even with impediments to self-organization and agility, companies and government agencies are increasingly turning to agile frameworks because they sense, correctly, that by improving their methods and tools, they may increase customer satisfaction, speed delivery of value, and raise the quality of software, systems and services. The problem is, they often think that it’s *only* about changing their methods and tools, and they give short shrift to the power of culture.

Once the domain of mid-size software companies, “agile-like,” a term that describes an organization that adopts some agile ceremonies without the accompanying organizational change, has become mainstream in the IT shops of Fortune 100 companies and government agencies.

The concept of “great big agile” requires leadership at all levels, just not the kind we are used to.

Why Agile Matters

Without exception, all of the organizations I work with have expressed an interest in “going agile,” if they have not already done so. This is a strategic decision that has deep-rooted cultural implications and should not be taken lightly. Many leaders do not realize the extent to which they have to change the way *they* behave.

There are several reasons why an organization should transition to a model that is agile and self-organizing:

Agile frameworks reduce the cost of failure. It is conventional wisdom in the technology industry that failure is inevitable, with many companies seeing failure rates as high as 70 percent. [1] Research conducted by organizations such as the Project Management Institute and the Software Engineering Institute has consistently confirmed high failure rates, so it makes sense to seek solutions that assume failure, not success, and to simply reduce its cost.

Failure is not just an option; it should be expected. A foundational premise of agile is to acknowledge that failure is normal, and we should plan to fail fast and learn as much as we can.

This reduces a project's cost while allowing teams to redirect efforts toward a more successful approach through the use of experimentation, retrospectives, and short, timeboxed iterations. Quality professionals will recognize this as an application of W. Edwards Deming's "plan-do-check-act" framework of continuous improvement applied in short iterations. [2]

Agile methods deliver business value to end-users more quickly. Value is delivered more quickly with an iterative and incremental delivery approach due to low-value features being de-prioritized or discarded, freeing up valuable resources to focus on the high-priority needs of the customer.

Self-organization pushes decision-making downward, freeing leaders to focus on strategy. For decades, the technology industry has explored ways to push decisions downward. Agile frameworks finally provide a model that can make that a reality, if only leaders are willing to accept their role as enablers rather than task managers. A successful agile team requires minimal oversight, makes day-to-day operational decisions, collaborates with business customers, and delivers business value without the need for continuous management intervention.

Agile complements important IT industry models. If CMMI, ISO 9001, and the *PMBOK® Guide* are models we use, agile is something we are. For example, CMMI has a perspective of defining what needs to occur for a product or service to be successfully delivered, while agile values describe why we take those actions. If adopted in this way, CMMI makes agile stronger. [3]

Many leaders do not realize the extent to which they have to change the way they behave.

All Is Not Well with Agile

While the popularity of agile frameworks like Scrum, Extreme Programming, and Scaled Agile Framework cannot be understated, in some ways, they have been a victim of their own success.

Large companies eager to replicate small company successes; satisfy younger, more self-organizing employees; and to just simply "go agile" have jumped on the agile bandwagon. Unfortunately, they often give inadequate attention to the changes in governance, infrastructure, measurement, and training required to succeed. The results have been chaotic, with large organizations adopting some elements of Scrum (e.g., daily standups and sprints) and force-fitting them with more traditional roles and techniques that are in conflict with agile values. This conflict dilutes the value of

the very agile ceremonies they use and leaves the organization without the benefits they were hoping to achieve. Out of more than two hundred companies assessed by AgileCxO and its partners:

- More than 90 percent assigned project managers for task management, oversight, and control of agile teams
- More than half did not conduct regular retrospectives
- Almost half conflated story points with hours yet still considered velocity to be a reliable metric
- Most made no changes to governance, infrastructure, or training to support agile adoption

These obvious conflicts with agile values result in a scenario where leaders may desire agility but continue to apply low-trust defined process control models to run the business, when a high-trust, empirical process control model is required for successful agility. This friction, often manifesting itself as "Scrummerfall" or "ScrumBut" ("we're agile, but..."), corrupts and degrades the very performance that agile leaders are seeking to achieve.

Jim Bouchard, author of *The Sensei Leader*, sums it up: "Don't even attempt to transform your organization until you can transform yourself." [4]

The Missing Layer in the Operating System

While the Agile Manifesto excels in describing *why* we do what we do, and industry frameworks and models describe *what* we need to accomplish, there is little guidance for leaders or teams on *how* to experience consistent success with self-organization and agility.

This layer isn't a process, but a set of guide rails that help leaders and team members recognize what large-scale agile looks like and provide the ability to recognize, evaluate, and improve agile performance. As I often tell conference audiences during my talks, "It's not magic. You just need to be able to recognize it."

To succeed with "great big agile," technology leaders and teams can start by categorizing capability into three interdependent layers: *why*, *what*, and *how*.

"Why" models: The set of values and guiding principles that are traced directly to the goals and methods of the organization. With its guiding principles, the Agile Manifesto is perhaps the best example.

"What" models: The set of frameworks, methods, roles, and artifacts derived from industry-standard models or internal methodologies. These models define what needs to be done and often provide examples that help us understand what we need to do while executing the software product development process.

"How" models: A set of behaviors, actions, and outcomes that helps define and evaluate organizational success and supports the culture, goals, and objectives of the organization. "How" models trace directly to established values, guiding principles, and frameworks to ensure that the behaviors exhibited by teams reflect the values of the organization.

An Operating System for Scalable Agility

AgileCxO's Agile Performance Hierarchy (APH) is an organizational operating system that encapsulates all three layers, providing leaders with an integrated view of organizational agile performance.

APH provides agile leaders and teams with a model to build, evaluate, and sustain great agile behaviors and habits. It is not an agile maturity model or a process, but an operating system for sustainable agility. Figure 1 shows how APH defines performance circles and holons.

Introduced in the 1967 book *The Ghost in the Machine* by Arthur Koestler, holons are described as self-reliant entities that “possess a degree of independence and can handle contingencies without asking higher authorities for instructions.” [5]

Koestler defines a holarchy as a “hierarchy of self-regulating

holons that function first as autonomous wholes in supraordination to their parts, secondly as dependent parts in a sub-ordination to controls on higher levels, and thirdly in coordination with their local environment.”

A holarchy works well for describing and evaluating agile performance, where behaviors are self-organizing and empirical and the sequence of actions and outcomes is unpredictable, iterative, and recursive, rather than procedural.

The APH is composed of interdependent actions and outcomes that provide guidance for the behaviors, ceremonies, and techniques that might be performed to meet the outcomes by team members, functional groups, and leaders throughout the organization. Sequence, rigor, and intensity are determined by functional and project teams, not by management. There are several key APH components.



Figure 1: The agile performance holarchy

PERFORMANCE CIRCLES

Performance circles encapsulate a discrete set of behaviors with a set of actions and outcomes that are essential to successfully step through the process of adopting, transforming, and mastering large-scale agility.

Organizations wishing to benchmark performance against the

APH may evaluate performance circles to determine how they are adopting, transforming, or mastering the behaviors of that circle.

There are six performance circles, each with a specific objective for leadership, depicted in the classic user story format of role, mission, and business value. These are described in table 1 in terms of goals and benefit.

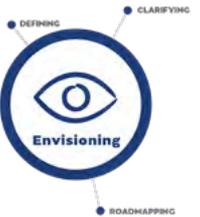
| Performance Circle | Objective User Story |
|---|--|
|  | <p>As an agile leader,</p> <p>I want to project agile values, provide the environment, and establish a vision</p> <p>so that my teams can be agile and successful in everything they do.</p> |
|  | <p>As an agile leader,</p> <p>I want agile team members engaged in the planning and building of high quality products</p> <p>so that we deliver the solution as expected.</p> |
|  | <p>As a product owner,</p> <p>I want to establish a roadmap, release plan, and backlog</p> <p>so that the overall vision of the product or service can be realized.</p> |
|  | <p>As an agile leader,</p> <p>I want teams and functional areas to learn and master self-organization and agile ceremonies and techniques</p> <p>so that the entire organization can benefit fully from agile adoption.</p> |
|  | <p>As an agile leader,</p> <p>I want to confirm that teams are demonstrating agile values, methods, and techniques as expected</p> <p>so that I can understand what is working well and what needs improvement.</p> |
|  | <p>As an agile leader,</p> <p>I want to foster a continuous improvement environment and engage with agile partners</p> <p>so that agile teams can grow their capabilities.</p> |

Table 1: Performance circles and objectives

HOLONS

Several holons are encapsulated within each performance circle, and they represent a set of actions and outcomes that can effectively stand alone but are also an integral part of a greater whole. All the actions and outcomes should be implemented in order to realize the value of each holon.

There are eighteen independent holons within the APH, as shown in table 2.

As an example, the engaging holon encourages the use of *gemba walks*, a process where leaders walk around to observe their teams and “understand by seeing” in order to improve engagement and enable quality.

ACTIONS

An action is the specific behavior that is applied to meet a holon’s objective. All behaviors in the holon should be demonstrated in order to meet the intent of the objective, and each must always be aligned with agile values. The APH recommends agile ceremonies and techniques that, when executed successfully, will meet the intent of the actions. Not being a process, the APH does not require any ceremony or technique, although it does provide a list of potential options for those who wish to benefit from their guidance.

CEREMONIES AND TECHNIQUES

Each action provides a recommended set of ceremonies and

| Leading | Crafting | Envisioning | Teaming | Affirming | Providing |
|-------------|------------|--------------|------------|---------------|--------------|
| Valuing | Planning | Road-mapping | Organizing | Confirming | Contributing |
| Enabling | Solving | Defining | Growing | Understanding | Partnering |
| Envisioning | Delivering | Clarifying | Governing | | Equipping |
| Engaging | | | | | |

Table 2: Holons within each performance circle define expected behaviors

OBJECTIVES AND OUTCOMES

Each holon describes objectives in a user story format that should be met in order to instantiate the value of the holon. An objective can be met by taking the defined actions in a manner and behavior that is consistent with agile values.

Holons contain a set of outcomes, like the performance circle they are surrounded by, that can be used to evaluate, improve, and sustain organizational agile performance within that context. The outcomes at the holon level are categorized into three levels, adopting, transforming, and mastering, and are used to help leaders evaluate and improve their organization’s agile maturity.

For example, the objective and outcomes of the delivering holon, part of the crafting performance circle, are depicted in table 3.

techniques derived from agile and lean frameworks that can be adopted to demonstrate the desired behavior and meet the intent of the action.

The sixty-eight ceremonies and techniques described in the APH include all the typical visual information indicators, roles, expected behaviors, and actions, allowing leaders and all levels to effectively play their roles as servant leaders and recognize, evaluate, and enable improved organizational performance where needed.

Large-scale agility requires large-scale self-organization, but it isn’t magic. Contrary to the claims of many agile-like practitioners that “agile doesn’t use process,” agile uses a lot of process; however, it may not be the kind you’re used to. The freedom to successfully self-organize results in freedom through mastery.

| Adopting Level Outcomes | Transforming Level Outcomes | Mastering Level Outcomes |
|--|---|--|
| <ul style="list-style-type: none"> * Current state of organizational performance is defined. * Future state is identified and displayed. | <ul style="list-style-type: none"> * SWOT is complete and published. * Backlog for future state exists in visual format. * Culture transformation release plan exists. | <ul style="list-style-type: none"> * Organizational performance sprints are executed. * Progress is visually displayed using VIM. * Impediments to organizational performance are regularly identified and removed. |

Table 3: Leaders can assess each holon’s objectives and determine which outcomes have been met

Self-organizing teams must also learn their craft, practice their forms, and progress through the stages of adopting, transforming, and mastering agility in order to be self-reliant.

Put In the Work and Reap the Rewards

According to VersionOne's "11th Annual State of Agile Report," 98 percent of respondents who believed they were agile reported success with agile projects. [6] That's a stunning statistic, but it isn't a coincidence. When agile works, the results are spectacular.

However, the same survey said that many organizations report a conflict between corporate policies and agile values, that leadership lacked the skills to enable agile teams, and that more than half were still maturing years after adoption.

Successful agile organizations are not successful because they adopt popular ceremonies or frameworks. They are successful because they are committed to open, collaborative, and transparent servant leadership at all levels, and they have cultures where failure and risk are not punished, but celebrated as a way to learn and improve. Strong agile organizations are learning organizations that can demonstrate a mastery of self-organization.

For those companies, the behaviors they exhibit are the natural outcome of organizational culture change, and they produce better results when they align with the rules of nature, which consist of, among other things, iteration, continuous learning, incremental wins (and failures), transparency, and team collaboration. They succeed because they rely on trust, collaboration, and deep respect for team members, and they recognize that while teaming is highly valued, personal commitment to behavioral excellence is the prime directive. In other words, an agile culture aligns with the rules of nature, and successful agile teams are those that have mastered both self-organization and personal self-reliance.

Conversely, if an organization is autocratic, with a high degree of secrecy, distrust, and negativity; blames people for failures; and is generally low in trust, then they will struggle with agile adoption. Without an "operating system" upgrade, they can never realize the benefits of organizational agility. For those companies, adopting Scrum or Extreme Programming, both excellent frameworks, can be misleading. They feel agile, but, in practice, they are anything but. This will come back to bite them in the form of missed deadlines, high turnover, unhappy customers, low quality, and high cost. This eventually leads management to declare, "We tried agile, and it didn't work for us."

To summarize, the agile performance hierarchy provides leaders and teams with objectives, outcomes, and behavioral guide rails to succeed. Along with an assessment method, training, and certifications you can chart a course to a high-performing agile future. [BSM] jdalton@agilecxo.org

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 has everything you need to know about 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™
I N S I G H T S

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.

Agile Leaders Know that Agile Performance Is Critical To Their Success.

Is Critical To Their Success.



Organizations can be certified in three levels to describe their state of agile performance

The Agile Performance Hierarchy is a Model for Building, Evaluating, and Sustaining Great Agile.



Contact us at agileleader@agilecxo.org if you are:

- Interested in benchmarking and improving organizational agile performance
- A company that is interested in expanding your business by providing APH assessments, training, and coaching solutions