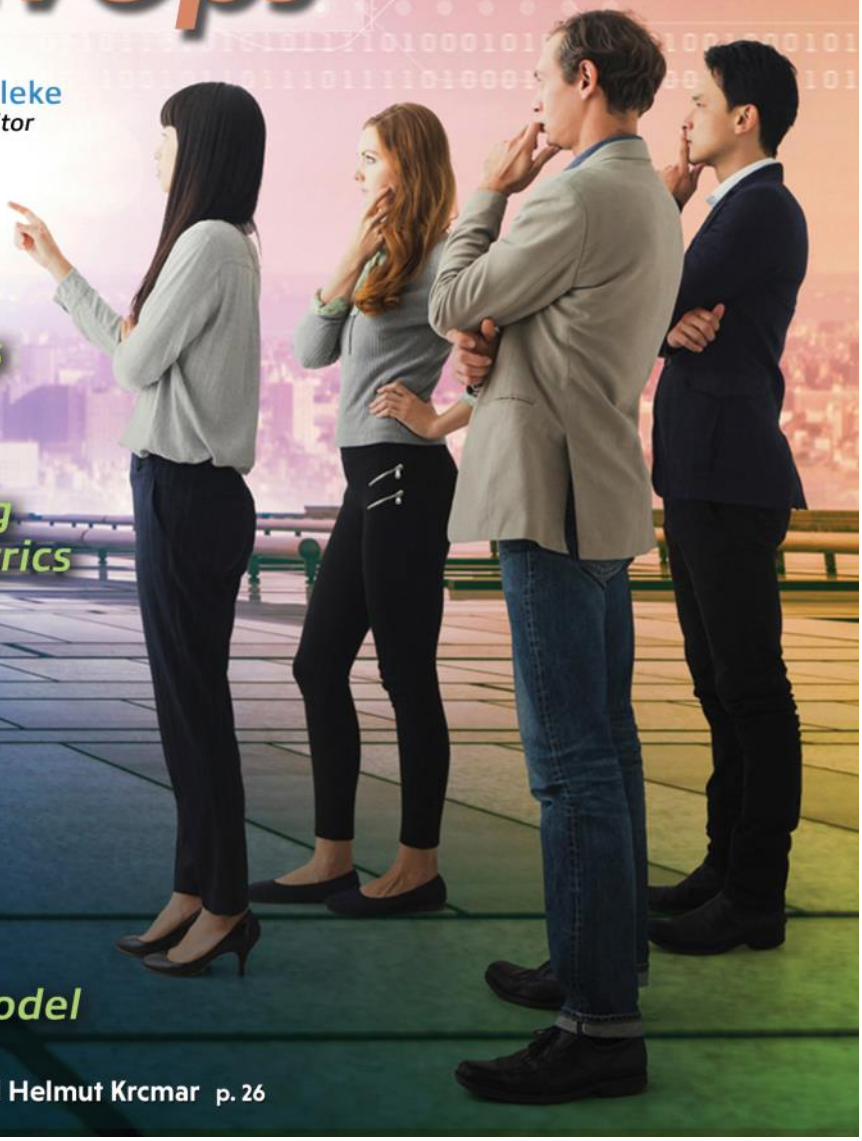
CUTTER Business Technology Journal

Management, Innovation, Transformation

Vol. 34, No. 2, 2021

Exploring the Intersections of Digital, DevOps & Agile

Eric Willeke
Guest Editor



CUTTER Business Technology Journal

As business models for creating value continue to shift, new business strategies are constantly emerging and digital innovation has become an ongoing imperative. *Cutter Business Technology Journal* delivers a comprehensive treatment of these strategies to help your organization address and capitalize on the opportunities of this digital age.

Cutter Business Technology Journal is unlike academic journals. Each monthly issue, led by an expert Guest Editor, includes five to seven substantial articles, case studies, research findings, and/or experience-based opinion pieces that provide innovative ideas and solutions to the challenges business technology professionals face right now — and prepares them for those they might face tomorrow. *Cutter Business Technology Journal* doesn't water down or delay its content with lengthy peer reviews. Written by internationally known thought leaders, academics, and practitioners — you can be certain you're getting the uncensored perspectives of global experts.

You'll benefit from strategic insight on how the latest movements in digital innovation and transformation, artificial intelligence/machine learning, Internet of Things, blockchain, analytics, and cloud, to name a few, are changing the business landscape for both new and established organizations and how cutting-edge approaches in technology leadership, enterprise agility, software engineering, and business architecture can help your organization optimize its performance and transition to these new business models.

As a subscriber, you'll also receive the *Cutter Business Technology Advisor* — a weekly bulletin featuring industry updates delivered straight to your inbox. Armed with expert insight, data, and advice, you'll be able to leverage the latest business management thinking to achieve your organization's goals.

No other journal brings together so many thought leaders or lets them speak so bluntly — bringing you frank, honest accounts of what works, what doesn't, and why. Subscribers have even referred to *Cutter Business Technology Journal* as a consultancy in print and likened each month's issue to the impassioned discussions they participate in at the end of a day at a conference!

Get the best in thought leadership and keep pace with the technologies and business models that will give you a competitive edge — subscribe to *Cutter Business Technology Journal* today!

Founding Editor: Ed Yourdon
Publisher: Karen Fine Coburn
Group Publisher: Christine Generali
Production Manager: Linda Dias
Editors: Jennifer Flaxman, Tara K. Meads
Client Services: service@cutter.com

***Cutter Business Technology Journal®** is published monthly by Cutter Consortium, an Arthur D. Little company, 37 Broadway, Suite 1, Arlington, MA 02474-5552, USA • Tel: +1 781 648 8700 • Fax: +1 781 648 8707 • Email: cbtjeditorial@cutter.com • Website: www.cutter.com • Twitter: @cuttertweets • Facebook: Cutter Consortium. ISSN: 2475-3718 (print); 2475-3742 (online).*

Subscription rates are US $485 a year in North America, US $585 elsewhere, payable to Cutter Consortium. Reprints, bulk purchases, past issues, and multiple subscription and site license rates are available on request.

NOT FOR DISTRIBUTION
For authorized use, contact Cutter Consortium +1 781 648 8700 or service@cutter.com.

☐ Start my print subscription to *Cutter Business Technology Journal* ($485/year; US $585 outside North America).

Name _____ Title _____

Company Address _____

City _____ State/Province _____ ZIP/Postal Code _____

Email (Be sure to include for weekly *Cutter Business Technology Advisor*) _____

Fax to +1 781 648 8707, call +1 781 648 8700, or send email to service@cutter.com.
Mail to Cutter Consortium, 37 Broadway, Suite 1, Arlington, MA 02474-5552, USA.

Request Online License Subscription Rates

For subscription rates for online licenses, email or call: **sales@cutter.com** or **+1 781 648 8700.**

CUTTER CONSORTIUM
Access to the Experts

Opening Statement

by Eric Willeke, Guest Editor

I've worked in the technology field for more than 20 years across a wide variety of development, architecture, leadership, and change management roles. In my career, I've seen significant change in every expectation around technology creation. End customers demand more refined experiences, workflows perfectly suited to their individual needs, and learning curves measured in seconds rather than days. Technology buyers expect solutions nearly out of the box, eschewing long configuration and customization periods in favor of easily integrated point solutions. Technology builders have access to thousands of tools and components that enable in moments what used to take months of custom coding to provide.

Operations involve account management of a cloud provider hosting hundreds of immutable instances rather than fixing individual servers. Deployment means monitoring a pipeline operation rather than an all-weekend manual process of moving files and updating configuration. Funders expect results in weeks rather than quarters or years. None of these stakeholders has any appetite or tolerance for defects or other poor quality, and security incidents can damage brands and careers in moments.

In theory, all these demands can be routinely met through the increased capability provided by modern technology methods. Businesses rely completely on digital capabilities in every aspect of their operations. Technology creators have several well-proven frameworks for identifying, funding, creating, and validating technology. Operations groups, where they are still separate from development groups, have access to automation capabilities for every aspect of their responsibilities, and capabilities for managing cloud infrastructure improve daily.

However, a great number of companies are incapable of taking advantage of these capabilities. Somehow, technology projects are taking longer, demonstrating lower quality, and increasing stress on workers, leaders, and customers. Why? Fundamentally, our existing management and operational habits prevent us from benefiting from the intersection of these innovations.

Every new expectation and innovation-driven opportunity requires companies to work together across historical silos. These collaborations can't be addressed with traditional organizational processes; they require rapidly changing, highly dynamic collaborations that cannot be easily represented on an org chart. They need to bypass existing approval and communication pathways to operate at the required speed; this places incredible stress on the company's control and monitoring processes.

This difficult transition is not unexpected. Socio-economic scholar Carlota Perez identified five historic technological ages that drove massive change at every level of society.[1] The means of production changed, changing dominant business models, sources and application of capital, types of required infrastructure, skills across the workforce, organizational shapes required to deliver against the promise of the new technology, and management models required to operate those businesses. The current transition into the age of software and digital technologies is well into its turning period, with the associated disruption and stress placed on any company with its roots in the previous age.

Quite simply, those companies are trying to exist and thrive in the new age while using operating models from the previous age. This is the real context of digital transformation. It isn't about using digital technology to implement existing models or creating digital-enabled models that complement existing businesses. Companies must completely change their shape, mindset, and engagement model to meet the expectations of the new age.

Unfortunately, proponents of various technologies, methodologies, and products often use the new expectations as a scare tactic to drive change. It is better viewed as a huge opportunity for companies to serve their customers in a fundamentally more satisfying

way, improve the lives of their employees and vendors, and capture greater market share as the digital age transitions from the turning point to the deployment period.[2] Fear-driven tactics for change often result in successive layers of duct tape and Band-Aids, resulting in an ever-more convoluted company. The combination of accepting reality and choosing to aspire to greatness enables alignment around a simpler, cleaner organizational model, and it's one of the most important tools of creating a future-state organization capable of performing in the digital age.

Fear tactics aren't required. In the words of Tasktop CEO Mik Kersten, "The problem is not with our organizations realizing that they need to transform; the problem is that organizations are using managerial frameworks and infrastructure models from past revolutions to manage their businesses in this one."[3] The focus on aspiration and openness to significant change creates the potential to evolve and practice new frameworks and models. This practice brings the people along, engaging employees, leaders at every level, and even vendors in activities that will shape the future. With nurturing, this practice creates a culture that is purpose-aligned, customer-centric, decision-empowered, and actively transparent. In short, it mobilizes the entire company to behave differently. It is transformative.

Digital Age Company Characteristics

There are a handful of behaviors and characteristics that can dramatically accelerate change if they are embedded in leadership's culture and mindset. The first three are internally focused (how the company is led); the next four are externally focused (what customers experience):

1. **Align to customer purpose; enable action.** Leadership focuses on providing clarity of purpose and ensures that purpose is completely centered around customers and their needs. Leaders then encourage and permit significant latitude in how to achieve that purpose at every level of the company.
2. **Invest in outcomes, not efforts.** Investment is aligned to those purposes rather than the projects and work required to achieve goals. This creates significant flexibility and permission to explore alternative approaches at the point where that flexibility has the greatest impact.
3. **Release imposed control to gain steering control.** Leaders practice letting go of old habits like directly controlling collaborations, backlogs, organizational structures, and fine-grained investment decisions. Instead, they learn to steer by funding to purpose, evolving purpose, and establishing lightweight guardrails.
4. **Be responsive and connected to customers.** Customers expect an ever-higher level of responsiveness and easy answers, and digital companies orient to provide that service however necessary. Cost of service is still a concern but is not permitted to damage the level of service.
5. **Be honest and authentic with customers**. The nature of the conversation changes, and the way a company shows up with its customers becomes a more dominant component of brand perception. This authenticity expands through the supply chain, requiring companies to care about what their customers care about.
6. **Personalize to individuals**. The age of one-size-fits-all software is over, and customers expect the capabilities they receive to be a great fit for their needs. They also expect to feel individually valued and respected by the companies from whom they choose to buy.
7. **Prioritize buyer needs over seller products.** The ease of new entry into markets requires companies to be constantly attentive to the problems their customers are actually buying solutions for, rather than focusing on what the company sells; otherwise, other companies will quickly discover that gap and exploit it.

Upcoming Topics

Low Code/No Code: Empowering the Citizen Developer & Unleashing Innovation
Michael Papadopoulos

Quantum Computing
San Murugesan

The Change Journey

The transformation to a digital age company cannot be accomplished using methods from the old age. This is one of the hardest aspects of a successful digital transformation effort: it cannot be a project or initiative.

It must be a focused journey of creating and sustaining new capabilities. The transition must be led and exemplified by the senior management, not delegated into any existing silo or role.

Companies attempt to decompose the change into a set of individual programs, leading to multiple parallel transformations. It isn't uncommon to see a digital transformation, an Agile transformation, a DevOps transformation, and a customer experience transformation all running in parallel. This attempt to decompose change is reminiscent of the previous-age mindset of fixing different pieces of a machine with different specialists. Trying to change a company this way results in a tangled ball of yarn because of how interdependent all the efforts are. A digital age company is more like a living organism, requiring a holistic approach to training new behaviors, creating organizational health, and promoting structural flexibility. Be agile to be digital: set goals, experiment for impact, decentralize action, measure impact, and celebrate success.

In This Issue

In our first article, Matt Ganis introduces the core elements of digital, Agile, and DevOps individually. Regardless of individual experience with the concepts, this provides an important opportunity to reflect on the elements addressed throughout this issue: What does "digital" really mean to a company? What makes Agile different from what came before? What are the implications of DevOps and the speed it provides? The article highlights the degree of change required for various heavily impacted functions within the company and the impact of new behaviors that go against decades of habit.

Continuing the theme of understanding the impact of digital, Cristina Popescu and Danish Aziz introduce objectives and key results (OKRs), one of the core tools for creating alignment in times of intense change and anchoring the types of steering approaches capable of keeping up with the decision speed required to be effective in the digital age.

As a 20-year-old movement, Agile has picked up a number of misconceptions, offshoots, and antipatterns as the world has figured out how to make it effective in different environments. In our third piece, Jacek Chmiel examines potential biases and the impacts they have on how Agile and DevOps show up in our organizations, helping us reflect on how we might reimagine various aspects and break out of our old ways of thinking.

Overcoming existing biases and creating effective digital agility require putting energy into an incredibly broad range of change efforts across the company, impacting nearly every role. As leaders, we naturally focus on the exciting, high-impact areas, which unfortunately leads to blind spots around the boring, "un-fun" areas. In her article, Cheryl Crupi shares six hygiene factors that can be crippling if not addressed. The factors highlight the need for a holistic approach to agility that integrates elements of digital, Agile, DevOps, and many other modern practices for companies that want to be successful in the digital age.

Successful transformation requires organizations to structure and operate in new ways, which requires creating many new patterns for designing teams and their collaborations with other groups. Anna Wiedemann et al. close out the issue by introducing one example of an organizational structure tactic for enabling successful evolutionary change with a small group of development teams supported by a vendor-managed operations environment.

These articles were selected to challenge mindsets and prompt readers to inspect their existing beliefs from a variety of directions. As you read this issue of *Cutter Business Technology Journal*, let go of existing biases and assumptions about the concepts and listen for new insights. Reflect on your own behaviors and those of your organization, and seek opportunities to engage differently.

References

[1]Perez, Carlota. *Technological Revolutions and Financial Capital: The Dynamics of Bubbles and Golden Ages*. Edward Elgar Publishing, 2002.

[2]Kersten, Mik. *Project to Product: How to Survive and Thrive in the Age of Digital Disruption and with the Flow Framework*. IT Revolution Press, 2018.

[3]Kersten (see 2).

Eric Willeke is cofounder and Principal of Elevate.to. He partners with leaders at all levels in applying Lean-Agile techniques to get better results through improved change management, better steering and alignment, and a deeper focus on the outcomes that matter. Mr. Willeke engages across many industries in some of the world's largest and most complex companies, as well as the smallest, high-growth startups. He helps by instilling Agile business management techniques, accelerating the deployment of critical initiatives, and mentoring executives leading large-scale Agile transformation efforts. Mr. Willeke is a principal contributor to the Scaled Agile Framework (SAFe), a SAFe Fellow, and a SAFe Program Consultant Trainer. He can be reached at eric@elevate.to.

COMPETITION IS STEEP

Exploring the Full Implications of Digital Transformation

by Matt Ganis

As organizations transform by moving themselves deeper into the digital world, care must be taken to adjust deep-seated behaviors and processes. A digital transformation likely involves adopting software development methodologies such as Agile and DevOps to enable this digital footprint. In this article, we'll look at the types of business processes and mindsets that likely need adjusting as this shift occurs.

For example, the simple task of adding a resource to an established Agile team could cause unintended interruption. For an Agile team to operate properly, members need to build and maintain a level of trust with each other, believing that each team member will meet their commitments in any given sprint or iteration. That trust could be disrupted by the addition of an "outsider," especially if there are personality differences or a lack of understanding of how an Agile team operates.

Integrating digital technologies makes it possible to significantly improve an organization's processes, both operations and, most importantly, those affecting the customer experience.

Moreover, hiring for proficiency at coding may have been the top attribute in the past; today, however, HR's role may be more about focusing on finding someone to help complete a team's "personality." In some companies, this means shifting from centralized hiring to team-level hiring. Regardless of the hiring structure, the process of interviewing and onboarding new members of a team will require considerably more care and attention.

Implications of a Transformation

A digital transformation is no less than a change in an organization's activities, business processes, competencies, and models that allows it to fully leverage the opportunities of current and future emerging digital technologies. The effort, expense, and pain involved with this type of change may lead some to question the necessity. Perhaps we'd be better off listening to US President Jimmy Carter's Director of the Office of Management and Budget (Bert Lance) who famously said, "If it ain't broke, don't fix it!"[1]

Of course, burying our collective heads in the sand is not the answer. In a 2020 Dell Technologies survey of more than 4,000 business leaders globally, nearly 90% said that recent world conditions highlighted the need for a more agile and scalable IT environment. They called out five reasons why companies should look toward digital transformation as a fundamental business strategy:[2]

1. Creation of on-demand services and functions
2. Increased employee effectiveness
3. Improved security
4. Stronger business partnerships
5. The ability to make better, faster decisions

Indeed, businesses with high digital maturity are 62% more likely than their counterparts to experience strong sales growth.[3] Clearly, organizations that move to the digital world first will have a significant competitive advantage over their competitors. Integrating digital technologies makes it possible to significantly improve an organization's processes, both operations and, most importantly, those affecting the customer experience. But how does one effect this change across an organization?

Since we're discussing a digital transformation, let's begin with the organization's approach to software creation and delivery. Consider Figure 1, where the "inside" of an organization contains several business processes, often acting as silos, that need to communicate with one another. This communication can happen verbally, via pen and paper, or through some other

non-digital means. Introducing digital technologies significantly increases the efficiency between those silos. Indeed, the more seamless we make interactions between processes, the greater the value to each business or process owner and the more efficient we make the organization.

But what of the public face of the organization? Consider external Web applications or e-commerce storefronts as the digital facade on an organization. In today's always-connected world, every business needs a Web presence and, increasingly, an online storefront to compete.

Whether we're looking at internal interactions or external ones, the organization's digital infrastructure is right in the middle of it all. Therefore, any discussion of a digital transformation must center around technology creation and the delivery of digital artifacts.

Agile

In the early 1990s, as the personal computer was finding its way into businesses around the world, the need for new and varied software to drive these new devices grew at an accelerated rate. The problem was that the time between identifying the need for a software package or application and the delivery of that software was measured in years rather than months. By the time the software was delivered, the need had often diminished. Consequently, a number of elite software practitioners met in Snowbird, Utah, USA, to discuss various best practices related to several exciting new methods for crafting software. From that meeting emerged the now-famous Agile Manifesto.[4]

Early Agile practitioners were looking for ways to quickly build working software and get it into end users' hands as soon as possible. This fast-delivery approach brought several important advantages. Agile's incremental approach to delivery enabled customers to realize some business benefits of the new software sooner, rather than waiting for a fully completed product. Equally important, the development team gained critical insights about progress and direction, allowing customers to make requirements changes as they saw the software unfolding.

Workings of an Agile Methodology

"Agile" is an umbrella term describing a variety of different methods and techniques. Some methodologies, like Scrum, focus on the project management aspect of development. Others, like XP, center on engineering aspects (i.e., how teams should behave and perform work), emphasizing things like pair programming, test-driven development, and frequent code refactoring.[5]

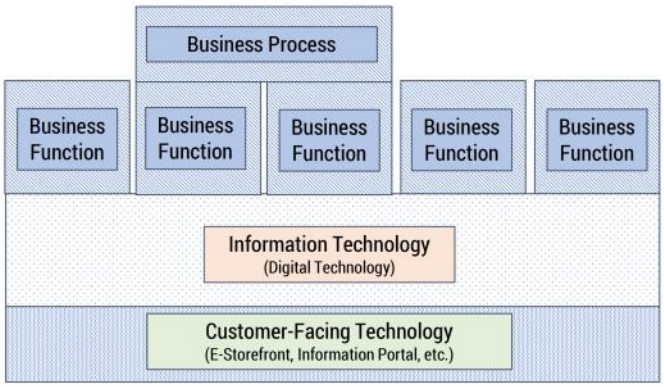


Figure 1 — Siloed business processes.

However, all Agile techniques conform to the four fundamental values laid out in the original manifesto:

1. Individuals and interactions over processes and tools
2. Working software over comprehensive documentation
3. Customer collaboration over contract negotiation
4. Respond to change over following a plan

These values are accompanied by a set of 12 principles that define how a team should go about its work.

Instead of deploying software in large, scheduled releases, teams break down work into small tasks, working on them in customer-provided priority order and delivering updates over frequent iterations (usually two weeks long). By quickly getting products to a deployable state, users can provide feedback on what worked and what didn't.

DevOps

DevOps is a software development approach aimed at enabling collaboration between development teams and IT operations. This collaboration of people, process, and working product enables continuous integration (CI) and continuous delivery (CD) of value to end users. Development and operations teams are siloed in many organizations; this separation leads to

miscommunication and conflicts that slow down production. DevOps combines these teams (along with QA and security in some cases) and fosters a collaborative environment that puts the needs of the people and clients in sharper focus.

> *Just as an Agile team becomes more efficient and productive when it has the resources it needs, so do all disciplines as they move toward digital transformation.*

DevOps is a natural extension of Agile, especially as we look at how Agile's main tenant — frequent delivery — can be accomplished. Once created, the operations team must deploy the software onto servers and other IT infrastructure, and the DevOps team aims to remove the barriers to this.

Agilists call this "The Whole Team Approach." It's about striving to involve everyone with the knowledge and skills necessary to ensure project success. Instead of having to gather various skills from around the organization to work on a project in bits and pieces, all workers become one team — a team that's 100% dedicated to the project and can therefore deliver much more quickly. The Whole Team concept is critical to an Agile development team, and coupling DevOps skills with Agile teams to generate a larger Whole Team creates a tremendous boost in customer value (see Figure 2). The curve in the figure represents the measure of value as teams move from a non-Agile environment to an organization driven by Agile techniques coupled with DevOps.

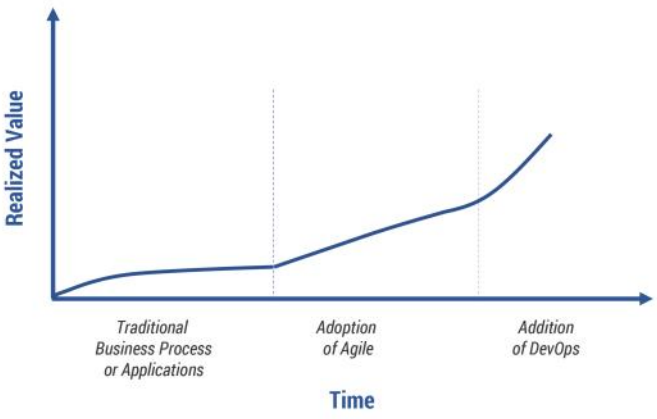


Figure 2 — Realized value over time as organizations adopt Agile and DevOps.

Agile methods bring an increase in value because what's ultimately delivered to a customer has been constantly reviewed and modified to meet to an ever-changing set of requirements. Frequent deployment comes at a cost, though: development teams are not as familiar with IT concepts as traditional operations teams. Bringing the operations team closer to the Agile team results in continuous development, continuous automated testing, and CI/CD. The value curve rises sharply as the organization introduces a set of DevOps practices to enhance its Agile development.

Transformation & Implications to the Organization

The move to digital platforms has ramifications far beyond the development and operations teams. Just as an Agile team becomes more efficient and productive when it has the resources it needs, so do all disciplines as they move toward digital transformation.

If the development team is operating with an Agile mindset, its actions may appear counterintuitive to other functions if team members don't understand the basic concepts of Agile. Certainly, the entire organization needs a basic understanding of how application creation (i.e., development and deployment) is now being done. In many cases, a fuller understanding and acceptance of an iterative approach is needed. Let's look at a few examples.

The Development Team

The development process on an Agile team is quite different from the one a traditional software developer follows. On Agile teams, the culture is very open, creating an environment where teams plan and size work efforts collaboratively.

Team members often argue (in a good way) about the size of the work effort to complete specific requirements (what an Agile team would call a "story"). A traditional developer might look at a given requirement and, based on his or her knowledge and experience, size the effort accordingly. On an Agile team, all members get to weigh in on what they believe the effort is, generating

a team velocity (i.e., the amount of work a team can complete in a time interval). Imagine the friction caused by a junior developer challenging the estimates of a very senior team member.

The coding process can also be problematic. Agile teams often practice pair programming in which two developers sit side by side, one with hands on the keyboard and the other commenting or perhaps driving the development. Some developers love this work method, but traditional developers, used to working on their own, can find this a difficult way to work.

Management/Business Process Owners

Line managers often have a difficult time when development teams adopt Agile methods. Traditional project managers report milestone achievements and status on a regular basis. Management tends to get involved in solving scheduling problems to help keep a project on time and budget.

In an Agile environment, teams are constantly accepting changes to projects as each new iteration or release is deployed. Recall that one of the four fundamental values of an Agile project is customer collaboration over contract negotiation. This is a fancy way of saying an Agile team would prefer to work with changing requirements than a fixed set of features/requirements. The implication, however, is that there can be no defined end date, or milestones, if the customer changes direction. Thus, the project manager role changes from keeping a project on track by managing plan milestones to removing roadblocks in an effort to keep the Agile team moving forward.

Human Resources

Although not an obvious place for change during a digital transformation, HR can play a crucial role in the success of a digital transformation. Recruiters look for the best and brightest when interviewing potential new hires — and having top developers is always a plus. But in the Agile world, we need developers with a wide range of "soft" skills. Knowing how to best communicate ideas (especially to long-tenure teammates) is crucial, as is a diverse background that helps the developer better understand the needs of the customer.

On the surface, this may not appear to be a large differentiator, but one "wild duck" or "superhero" developer (someone who loves to jump in to save the day) can be hugely damaging to the team's productivity. In an environment where the goal is quickly turning around changes and making course corrections to a system's inner workings or architecture, a non-team player can derail a project.

Conclusion

A digital transformation aims to fully leverage the possibilities and opportunities of new technologies and realize their impact faster, more efficiently, and in more innovative ways. For most, this transformation requires taking a staged approach that involves various stakeholders, breaking down silos between organizations, and modifying business processes.

Once an organization's digital applications are deployed, users will expect (and demand) frequent upgrades, additional features, and a fast turnaround on errors/defects. This means the supporting infrastructure inside the organization must be ready to respond quickly and efficiently. This entails changes not just to the applications themselves, but the way they're created and maintained. It's a tall order but the price we must pay to stay competitive in today's digital world.

References

[1]Martin, Gary. "The Meaning and Origin of the Expression: If It Ain't Broke, Don't Fix It." The Phrase Finder, accessed March 2021.

[2]"Digital Transformation Index 2020." Dell Inc., 2020.

[3]Cayer, Marie. "Digital Transformation, a Competitive Advantage in all Markets." Raymond Chabot Grant Thornton, accessed March 2021.

[4]Eby, Kate. "Comprehensive Guide to the Agile Manifesto." Smartsheet, 29 July 2016.

[5]"ScrumXP." SAFe Scaled Agile, accessed March 2021.

Matthew Ganis is an IBM Senior Technical Staff Member in Armonk, NY. He is also an Adjunct Professor of Information Technology and Astronomy at Pace University, where he focuses on the implementation of the Internet of Things. Dr. Ganis is the coauthor of Social Media Analytics: Techniques and Insights for Extracting Business Value Out of Social Media. *He can be reached at mganis@pace.edu or via Twitter @mattganis.*

A RALLYING CRY

Improving Agility & Facilitating Digital Transformation via Metrics

by Cristina Popescu and Danish Aziz

The business landscape we find ourselves in today is incredibly dynamic, with significant levels of change and variability that can make it hard to understand where or how to focus. Technology has added even higher levels of uncertainty into this mix, with low barriers to entry for most small businesses to experiment and discover opportunities that scale unexpectedly quickly.

Incumbents, however, find it increasingly challenging to lead innovation in what would be considered their core competency. Customer expectation, once a key trend in the B2C sector, has become prevalent in the B2B sector as well.

Metrics are common throughout the business landscape, but the complexity of organizations and the markets they operate in rarely allows the linear comfort of using a single measure to define success.

Putting the user at the heart of everything we do is one of the most common pieces of advice when it comes to leadership in the digital age. Yet many organizations are still not approaching this highly uncertain environment using a scientific approach to experimentation. They are willing to invest in new technologies and ways of working, but more often than not, they don't understand the impact of their investments well.

Metrics are common throughout the business landscape, but the complexity of organizations and the markets they operate in rarely allows the linear comfort of using a single measure to define success. For example, Web traffic on an e-commerce store might increase, which is encouraging, but traffic alone doesn't necessarily lead to a purchase or a satisfied customer.

The growing availability of product, service, and vendor choices across the board means that organizations need a balance of metrics to provide accurate, timely insight for decision making and continuous improvement. This would typically include value metrics that indicate whether or not expected benefits are being delivered to customers. It would also include delivery metrics, to understand if the organization's ways of working allow for rapid discovery in a dynamic landscape. Finally, it would include portfolio metrics for a view on benchmarks and trends that can help shape the system of work according to the context.

Balance is also key from the perspective of leading and lagging indicators, where using only one or the other can become a dangerous trap in dynamic markets and complex organizations. Lagging indicators such as sales or revenue tell us whether a result has been achieved, but they may not be timely enough and explanatory enough to inform necessary changes during delivery. Similarly, leading indicators such as product quality or number of marketing campaigns are predictions of future success that may or may not materialize.

Metrics can also be misused in ways that harm the organization's ability to learn and make progress. This is especially true in fast-paced environments where the definition of success can change rapidly; metrics and assumptions can easily become outdated. And holding people accountable for metrics that no longer make sense encourages them to pursue an outdated plan at all costs, without taking into account new information and lessons learned. The flip side, where an organization overreacts to metrics, is harmful as well, creating an unhealthy fear-based culture. This leads to people caring more about the metrics than the well-being of the organization and the outcomes the organization seeks.

Similarly, comparing metrics across teams without understanding the hugely different contexts or challenges faced by two seemingly identical groups can be dangerous. This, of course, does not only affect an

organization internally, it also translates to how partner organizations operate though contractual terms, buyer-vendor relationships, multi-vendor teams, joint ventures, and so on.

Using metric trends over time rather than absolute numbers can be an effective way to address some challenges mentioned, simultaneously increasing the focus on continuous improvement. Choosing what trends to follow, how to interpret them, and when to use them for timely decisions is nevertheless subject to many of the same pitfalls. Overall, a balanced approach to metrics is typically best because it provides a holistic perspective on the organization while reducing the risk of focusing solely on metrics that are not driving business value.

The concept of a balanced scorecard that weighs various types of metrics from several aspects of the organization is well known. However, the complexities mentioned above can make defining such a scorecard difficult for even very experienced professionals. Using a standard set of metrics that may have worked for other organizations is not very useful either.

As explained by the Cynefin framework,[1] best practices can work well in obvious or complicated environments. Complex environments like the ones in which a large majority of our activity now takes place require a much more contextual and flexible approach, as our understanding and insights emerge through ongoing experimentation.

No metrics are perfect, and choosing the ones to focus on has become as much an art as a science, but making business decisions solely on anecdotal evidence, gut feeling, or the highest-paid person's opinion is risky at best. Metrics are ultimately meant to improve our objective awareness and understanding of reality, so we benefit most when they are relevant, unambiguous, timely, and transparent.

Using Objectives & Key Results to Measure What Matters

Now that we have established how metrics can be useful, we need to recognize there is also an effort associated with collecting and using them. Having too many metrics complicates decision making as much as having insufficient metrics, especially if a large portion of these metrics distract from what is really valuable. So how do we identify the metrics that matter most in our context? A great approach is the goal-setting system known as objectives and key results (OKRs).

OKRs are a lightweight, structured approach to defining success criteria and aligning the team to deliver specific results. Agile teams are meant to continuously deliver (some sort of) value, and without clarity on what value really means, many teams are only guided by output metrics (e.g., velocity, lead time, number of features delivered, and code quality).

Consistently delivering output with speed does not necessarily translate to business results. That's where OKRs come in, clarifying the meaning of value and shifting the team's focus from output to business outcomes. An OKR can be thought of as an immediate goal or commitment structured on two components: an objective and a set of key results.

The objective states what we want to achieve, describing where we want to go and setting a clear direction. You can think of it as a destination on a map, such as Amsterdam. Objectives are meant to be tangible and unambiguous, so it is obvious to any observer whether the objective has been achieved. They are also meant to be ambitious and inspiring, motivating the team to stretch beyond its comfort zone.

Key results are metrics that show how we are progressing toward our objective. You can think of them as the GPS used on a drive from Paris to Amsterdam. Key results are often numerical and can be based on growth, performance, revenue, engagement, and so on. Importantly, they are a focused set of value-based metrics tied specifically to the desired improvement.

Key results are not a list of activities to be completed, and they are not health metrics that monitor the ongoing state of the business. Rather, key results indicate if we indeed get the value that we intended from the initiatives we're investing in.

OKRs Complement Business Agility Efforts

Three areas where OKRs can bring significant advantages are alignment, focus, and flexibility. Consider the following example.

Imagine that as part of a digital transformation journey, the HR team wants to streamline the employee engagement experience to develop and retain the best talent, as measured by the following key results:

- Increase the average time employees dedicate to learning activities from one hour per week to five hours per week.
- Increase the amount of feedback shared between employees from an average of three feedback records per month to an average of nine feedback records per month.
- Increase the employee satisfaction with HR processes from 55 to 88 on a 100-point scale.

> *OKRs are not meant to be static or all-encompassing. They are meant to clarify what success means and rally the organization in the same direction.*

Say we also have two teams in IT responsible for the development and maintenance of HR applications, and they want to simplify the HR application landscape to enable a modern HR experience, as measured by the following key results:

- Reduce the number of HR applications from 30 to 10.
- Reduce the total HR application maintenance costs from US $3.3 million to $2.2 million per month.
- Reduce the number of support tickets for HR applications from 90 to 30 per month.
- Increase the employee satisfaction with HR tools from 47 to 75 on a 100-point scale.

The first level of alignment these two OKRs serve is within each team, as the team members know with clarity what results they are working to achieve. Each objective could exist on its own, but there's more value in aligning OKRs across teams.

For example, when the IT teams are evaluating cloud-based software-as-a-service (SaaS) apps that could allow them to sunset 10 custom-built legacy systems, they look for options that include good-quality tools for employees to request and give feedback.

Once the teams are aligned and driving in the same direction, OKRs facilitate focus though clearly defining what success means and continuously using this definition to guide activities. When the HR team notices that its activity in the past three weeks has not led to any change in employee turnover, the team realizes most of its time was spent organizing for an upcoming university hiring fair. Although hiring is important, it is not an activity that contributes directly to the current OKR, so HR decides that instead of participating in the hiring fair this quarter, it will focus its efforts on organizing an employee festival to boost happiness and reduce turnover.

Since OKRs describe results and not activities, coupled with a continuous awareness of progress, they also facilitate more flexible decision making. This enables teams to redirect efforts as early as possible when new information becomes available, minimizing investments with limited or no return.

For example, after a month and a half of analysis and pilots, the IT teams find that any of the feasible SaaS options would cost at least $3 million per month to support all current HR functionality. Instead of selecting and rolling out an application they know from the start will not meet their success criteria, they decide to pivot their objective from simplifying the entire HR application landscape to eliminating low-value functionality and processes.

What is important to note is that OKRs are not meant to be static or all-encompassing. They are meant to clarify what success means and rally the organization in the same direction. The relationship between the objective and the key results is hypothetical until realized. So the challenge is to adjust hypotheses as new information becomes available.

In addition, key results are typically lagging indicators, so it's a good idea to use additional output metrics that provide leading indicators to balance any decisions. Learning when and how to adjust an OKR to the organization's advantage can be difficult, so a culture of transparency, communication, and trust is key. Next, we highlight two cases to support the need for an optimal OKR process.

Avoiding the Sunk Costs Fallacy

An organization we worked with was running a multiyear digital transformation based on replacing a complex, homegrown legacy system with a new, homegrown application. Six teams were building the new system, loosely using Agile practices in a mostly waterfall environment. After more than three years'

efforts and more than $47 million spent gathering requirements and designing and building the new system, limited functionality had been made available to the users.

With budgets stretched and little value realized, the leadership team was making decisions largely on the budget numbers. Delivery was repeatedly delayed due to unforeseen complexities, quality problems, missing skills, and missing development capacity. So much had been invested in the initiative that one way or another, it had to get done. Was the original business case still relevant? Did it really make sense to continue investing in building this system? Perhaps yes, but without a clear definition of success and timely objective metrics to continuously inform decision making, there's an enormous risk of unknowingly allocating extensive effort and resources to activities that do not lead to the results we hope to achieve.

Learning from Digital Natives

Digital native firms such as Google, Spotify, Twitter, LinkedIn, and Airbnb have been using OKRs effectively for years as they navigate their dynamic environments. Their experience has inspired a wide range of organizations across industries, including ING, Walmart, and Target, to consider how they can benefit from OKRs as they transition their operating model and working culture to compete in the digital age. Such transformations take significant time before benefits are realized at scale, but the growing use of OKRs has provided clarity and flexibility for teams to deliver value in an aligned and autonomous way.

Getting Started

OKRs can be a powerful strategy execution tool for teams and organizations operating in complex and uncertain environments. Use the following guide to get started with OKRs in your environment:

1. **Choose one objective that is most valuable for your team to achieve within the upcoming three months.**
    - Start with one objective at a time. There is much more value to gain from focused and aligned action than from multiple efforts running in parallel. Even as you gain experience with OKRs, maintain your focus on valuable results by selecting at most three objectives to work on in parallel.
    - Start with difficult but achievable goals. As you develop a more mature OKR culture and consistently achieve your objectives, introduce stretch goals to develop further.
2. **Choose up to five key results that measure your progress toward achieving the objective.**
    - Key results are not a task list to complete. They are value-based, not activity-based, and to the extent possible, every key result should be measurable.
    - Aim for a set of key results that represent all that is needed to fully achieve your objective. This minimizes the risk for an unexpected delay or miss on the objective.
3. **Align your OKR with the other teams and with the overall objectives of the organization.**
    - Don't cascade OKRs top-down through the organization. Instead, align them in a simultaneous top-down and bottom-up process, empowering the teams to decide how they can best contribute.
    - Use a shared OKR for multiple teams when they are working toward the same objective rather than artificially trying to split the OKR into a separate result for each team.
4. **Baseline your key results before you start working toward your objective.**
    - Discussing these baselines within the team is a great way to build shared understanding.
    - Expressing key results as a target compared to a baseline conveys information more clearly than using percentual changes.
5. **Review your key results at least once per week with the entire team.**
    - Dedicate ample time to reflect on your progress, identify impediments, and implement improvement initiatives.
    - Measuring how confident the team feels about achieving each key result is a useful technique to maintain alignment and surface issues early.

6. **Reflect on your objective at the end of the three months (earlier if necessary).**

    - If you achieve your objective before the three-month deadline, that's a perfect reason to celebrate! And then you're ready to choose another objective.
    - When your key results indicate that you may be unable to achieve your objective on time, recognize this as early as possible and be willing to pivot.

For the best results, remember to keep your entire process transparent and collaborative. We invite you to experiment with this approach and would love to hear about your experience!

Reference

[1]See Wikipedia's "Cynefin framework."

Cristina Popescu brings a holistic approach to her work in the areas of business transformation, strategy, agility, and coaching. Ms. Popescu is looking for ways to connect the dots and engage the world from a systems thinking perspective in her efforts to build high-performing teams. She can be reached at cristina.popescu@ibm.com or via LinkedIn at linkedin.com/in/icpopescu.

Danish Aziz has been working with delivery teams and senior executives in shaping their systems of work in a number of digital transformations in the banking, insurance, consumer goods, and information services industries. The experiences of day-to-day team delivery, balanced with the challenges of leading large organizations, help him bring forward multiple options for organizations to discover what works best for them. Mr. Aziz firmly believes that the coming decade will see a big change in the default mindset of how we work to a more experimental and adaptive approach. He can be reached at danish@ibm.com or via LinkedIn at linkedin.com/in/danishaziz1980/.

MOVING ON ...

Escape the Agile Bubble & Bring Back Productivity

by Jacek Chmiel

It's been 20 years since the Agile Manifesto was created and published. I remember the excitement and the feeling that all known methodologies at the time (except XP) had suddenly become obsolete.

Agile won global acceptance and became the de facto standard for software projects. It then spread across organizations, influencing other departments, digitalization initiatives, and projects. As a result of its enormous popularity growth, we now have Agile project management, Agile processes, Agile enterprises, Agile marketing, Agile … everything.

Let's stop for a moment and reflect on the relevance of the original Agile ideas and methodologies.

In the context of what has happened during the digitalization of all aspects of our businesses and private lives, 20 years is the length of an era. We lived in a very different world in 2001, with only a fraction of businesses online. The majority of people were not using digital tools for work or in their private lives.

Agile was promised as a cure for all the problems inherent in previous methodologies, completely changing every assumption about how people should work with each other.

Twenty years later, there's substantial criticism about the relevance of so-called Agile methodologies — specifically, how much they've become disconnected from the business reality of enterprises and the Agile spirit itself.

Before we go further, it's important to remember that many of Agile's practices and base assumptions were and are unrealistic, and some have become obsolete in our present digital world. Indeed, "Doing Agile wrong" has been a popular mantra for the last 10-15 years.

Nevertheless, it's high time we moved past that and verified Agile's real relevance. This is the inevitable cycle of progress: we're in need of a major step forward into a new era of ever-more-digital businesses and lives.

Agile Problems and Solutions

We need new solutions — software developers would call it a major refactoring. It's likely you will recognize many of these suggestions as changes you've already made to address the flaws of the original Agile methodologies. For those who have just become Agile believers, this may come as more of a surprise. Let's go through the most common problems with Agile versus reality and discuss possible solutions.

Manifesto as the Source of the Problems

Many practitioners, myself included, believe the Agile Manifesto was too ambiguous, too generic, and too disconnected from the realities of modern enterprises and even human nature. It was so open to misinterpretation that the true spirit of Agile was lost in a set of ceremonies, processes, and procedures within a family of Agile-derived methodologies.

So do we need a new manifesto? Sure, they're being created, but the truth is we must all find our own way, and it's usually not necessary to call it a manifesto.

You are free to borrow from multiple sources, old and new, to adopt something fresh, something that will work for you. Actually, new Agile (or Agile replacement) is more about personalization than using generic manifestos and playbooks.

Sticking to a single manifesto wasn't a very agile thing to do in the first place. It tended to become an object of a blind cult with fanatics unable to listen to any feedback, instead blaming the organization when it didn't work in its pure, fundamentalistic form.

The reality is more complicated, and one-size-fits-all solutions quickly fail.

It's a 1940s (but never wrong) idea of a *constant improvement loop* based on results and practical experiences. The current Agile (1.0) seemed to forget about the need to improve itself, especially in Scrum and related methodologies with their artificial ceremonies and false assumptions about people and organizations.

> *The realistic solution is to involve as many people as possible in ideation and the creative process.*

Financial Incompatibility

Let's talk about money and cost first, as these things are not addressed by Agile at all. Agile methodologies did not survive contact with CFOs and the economy in general. Money is not iterative in its nature, nor are budgets, and there's a striking imbalance between financial aspects, reporting, and Agile principles. As a result, Agile projects nowadays are often waterfall projects disguised in Agile clothing.

This is too often a combination of flaws from both approaches: waterfall's hope of not being wrong (i.e., fixed scope, fixed date, fixed feature set) with tons of Agile ceremonies on top of it (i.e., to appear Agile, not to miss out on Agile).

One of the solutions that works is to let go of some of the false impressions of control. Letting go is hard at first, but you'll get more business value with less frustration in the end.

It might be best to create yearly budgets for constant digital product developments without a fixed scope. Avoid infamous big-bang releases and adopt continuous features delivery, technology improvements, and delivery models (e.g., continuous improvement, continuous delivery).

This is all to enable real progress without guessing what will happen half a year later. No one knows what will happen next, and it's better to stop lying to ourselves and embrace the dynamics of the organization and business context.

Product Owner Case

One of the key flaws of Scrum (for many, it is almost synonymous with Agile) is the definition of the role of the product owner, which in reality is hard to find. In essence, we look for someone, a single physical person, who knows all the business goals, processes, and technologies and has the authority to make final decisions on the spot.

Requirements and their prioritization have been a challenge throughout every project, so here is a "brilliant" solution — simply appoint one single person … a magician of sorts … an exclusive business knowledge holder … a technology-savvy genius … who is also equipped with decision-making power.

That would be great and has even happened from time to time, especially in smaller startups. But more often, that role was played by someone with a blurry product vision and not enough power to make clear decisions fast. It was the team that was defining and delivering new features with near-silent acceptance from a product owner present in name only. Everything depended on a single superhero who was too often a utopian idea, especially in large, multidisciplinary projects.

The realistic solution is to involve as many people as possible in ideation and the creative process and not to assume things like "developers cannot deliver great functional ideas" or "developers lack imagination or creativity." Anyone who can and is willing to contribute to the product design should not be afraid to participate.

A collaborative spirit should also be applied to decision making about features and priorities.

Communication

Agile emphasized direct verbal communication, creating another set of problems. A verbal format is not optimal for team members who want time to analyze the question, research different options, and prepare for a discussion. Often, it's better to write up/draw up something ahead of time and give people a chance to analyze it, or even sleep on it.

Additionally, verbal communication is not effective and motivating for shy people or introverts, characteristics of many IT professionals. **Conversations are often dominated by extroverted personalities rather than those with excellent ideas that add value.**

Eventually, this turned into meetings for everything, along with calls, teleconferences, and gatherings to address items that could have been achieved by a simple message on Slack/Teams or a shared document with a request for comments. This obsession with meetings is a main reason for diminished productivity, reduced attention spans, and increased anxiety.

Another common misconception (not related to the manifesto but often attributed to Agile adoption) is waiting until the next meeting to discuss important communication activities or make decisions. In the modern era, it's better to embrace a continuous time spectrum. When something is important and needed, it should be done without waiting for a ceremonial gathering the next morning. That applies not only to sharing information, but also key project decisions.

In this new era, what is going to work is the combination of tools that take into account human nature and new technologies while being open to remote work. Let people do their jobs. Synchronizations are needed, but they also are distractions; focus is what we really need to move projects forward.

Architecture and Product Design

Architecture is a key success factor for software projects. It was, unfortunately, deliberately omitted in the Agile Manifesto. It's time to bring the good practices back. **Evolutionary architectures and composable architectures are better responses than avoiding architecture design completely and hoping things will somehow happen.**

Data Aspect

Data has become the fuel of modern enterprises, and vast amounts of data growing exponentially require new approaches and tools. Machine learning is fed with data, and real-time analytics is a standard now, not in the future. **Data is a missing part in the original manifesto that must be addressed in all data-driven organizations.**

Velocity vs. Business Value

Pretending there's progress because every two weeks (default Scrum sprint length) something visibly changes is another problem that can lead to perfectly (or imperfectly) executing the wrong idea. Burning money is irreversible, and the encouragement to iterate is often misunderstood and brings with it a lack of focus on the goals.

There will always be another sprint, so let's move it to the next one. There's also: "Let's overestimate the sprint items to make sure the managers won't accuse us of not delivering the expected results on time." This attitude is safe for developers but extremely counterproductive.

The business value, meaning the assessment of what really was achieved, is much harder to evaluate than points in the Agile productivity measurement tool. It becomes more rewarding for developers to pick simple tasks and become velocity leaders.

This has to stop immediately, as we cannot lie to ourselves by confusing velocity with productivity or business value. Yes, it will be harder to work with a business value perspective (assessing the business benefits of given feature sets), but far more productive and satisfactory in the end. Most project team members want to make an impact, see the results of their work, and feel they are making a difference for business users.

Micromanagement in Disguise

The Agile-esque methodologies of enterprise projects are the bureaucracy that younger generations want to avoid. They hate daily Scrums and feature/task management with yellow sticky notes posted all over whiteboards or their digital equivalents. Things that were fresh and cool for the early adopters of Agile seem strange and unnecessary to newcomers. Again, 20 years have passed.

Agile is a type of discipline, but today it's too often used as a micromanagement technique, which goes against the spirit of Agile. Getting rid of daily sprints is becoming a default response to these frustrations, but it's just the first step.

Disrespect for Documentation

A lack of documentation is often attributed to the Agile Manifesto; this is then used as an excuse for the lack of discipline within the project teams, resulting in system maintenance and further development becoming a nightmare. This is one of the major misinterpretations of the Agile Manifesto; probably the biggest one.

For instance, passing a product through the various phases and managing it without adequate documentation requires hours of one-on-one sessions with the previous developers. I witnessed a famous consulting company once propose the methodology of taking over the system development from one vendor by another.

> *Everyone deserves equal respect and attention, but we're all different, so the unrealistic utopia of equal team members should be abandoned once and for all.*

To me, losing the ability to document things, despite the big advancement in dynamic documents for collaboration (live documentation), is nothing but a sad degradation of organizational maturity.

Today, I see a movement back to documentation. A new kind of documentation that is useful and readable, but documentation nevertheless. Verbal communication is not the only solution for transferring information and knowledge, nor is it the most efficient.

In the API-driven world, developers need good documentation more than ever, so let's bring it back and appreciate it in a new form, eliminating this harmful misinterpretation of the Agile Manifesto and the resulting consequences.

Constant Team Assumptions

Project managers tend to like this one because they can use it as a justification for not sharing developers with other projects. People are confined to the space of a single project — some like it, but others consider it a huge artificial limitation.

Not changing team members for a long time is a recipe for burnout. Everyone needs a break or change eventually.

The solution is to embrace more flexible models, letting developers contribute to a variety of projects while staying open to new team members with different degrees of seniority and skill sets.

Seniority and Experts

Agile promoted a so-called T-shaped model (a broad range of skills possessed by each team member) to allow team members to fluently change tasks and reduce wait times. It's a nice concept that is now being reinforced with new trends like full-cycle development and full-cycle developers.

However, specialized expert team members with highly focused skills are also of great value and should be appreciated, not avoided. They are underappreciated in Agile, but again, real-life projects demand an effective combination of experts and T-shaped developers in the right proportions at the right time.

Seniority and experience were completely neglected by Agile through the false assumption that everyone is the same. Even worse, many people misinterpreted Agile as a process for twentysomethings. **Young, dynamic teams can greatly benefit from more experienced people, especially when it comes to architecture, code design, and product design: essentially, any area that benefits from experience.**

Technologies change, but not as often as some initially thought (Java is more than 20 years old; HTML is the same), so the practical experience of senior team members can be of tremendous worth. The same applies to business domain knowledge, as it's invaluable to the success of the project.

The focus should be on creating a balanced team so that experienced individuals work alongside less costly, less experienced workers. Everyone deserves equal respect and attention, but we're all different, so the unrealistic utopia of equal team members should be abandoned once and for all.

Collaboration vs. Distraction

Agile teams often showed off their colorful offices full of nice gadgets, with everyone working in a large room, close to each other to enable collaboration. For many developers, this was a nightmare, stealing their ability to focus and killing productivity and creativity.

People wearing noise-cancelling headphones all day are too common in open spaces; they are trying to focus on their work by minimizing distractions. Offices built around this concept need to change. There's time for brainstorming, discussions, and even quarrels or going

to lunch together, but most of our time should be spent on the deliverables of the projects, which require focus. That means a return to smaller, preferably individual, rooms (these could be remote) with physical offices that serve as collaboration spaces rather than working spaces.

Servant Leaders?

Many organizations renamed managers, calling them leaders and telling them to earn their Agile certificates so they could become "Agile leaders."

True leaders don't just do the project reporting, they actively participate in the project. Converting generic project managers with no technological and/or business background into true leaders is seldom possible and requires a great deal of training.

By the way, in the original Agile Manifesto, there's no one responsible for the project, but of course, by definition, everyone is responsible for the project as a team of peers with shared responsibility.

Does this happen often in the enterprise? No.

In reality, there's always a person appointed as the project manager. So, this isn't an Agile thing at all; let's call this person an Agile project manager. Ok, now it sounds better, but let's not forget that we've just rolled over the idealistic Agile assumption that no project leader role is needed.

A common Agile misinterpretation is that servant leaders should simply make sure all the requests of the Agile team are met and not interfere with the work. In return, the team will create the best products. Hierarchies of any kind are so out of fashion.

But what about hiring people, promoting people, and managing their careers and salaries? Yes, some interpret these in the Agile way, but it's up to the team to decide all that. This usually results in an explosion of salaries and cost, plus constant infighting, resulting in chaos and a loss of focus on the project itself.

Another false assumption is that people are the same and have no other ambition other than to deliver the best solution.

Of course, we are humans, so there are always leaders in the group. It's better to name them and give them additional responsibilities than to pretend there are no leaders just because it's Agile. This idea goes against human nature and can be easily fixed by recognizing leaders and outlining what is expected of them.

Where Is DevOps?

DevOps is often believed to be the cure for Agile problems. Many practitioners, and I concur, think it was born out of desperation to address the efficiency problems of the inevitable friction between development and production. Indeed, DevOps came about as a way to introduce transparency and promote effective collaboration between operations and developers.

This can happen in practice, but more often, the idea becomes another situation in which the DevOps teams are unable to communicate efficiently with both developers and operations, making the problem it was supposed to solve even worse.

The solution is to embrace the original meaning of DevOps: tasking developers with ops as well, adding one or two DevOps tool experts as internal consultants and coaches, but not as separate teams, thus avoiding the creation of DevOps silos.

Goodbye Agile Manifesto, Long Live Fusion

Budgets and time frames of digital transformation programs are exceeded by huge percentages, and it's getting harder to deliver on promises made to investors and business owners. We can't continue to pretend that current Agile methodologies are still the best way to go.

Let's embrace the foundations of productivity and the feedback loops for continuous improvement, along with valuable criticism and skepticism. Get rid of Agile ceremonies that don't work in a particular context, inventing new ones if the team needs them but keeping them to a minimum.

Let people focus on what they *do*, don't force them to be physically cramped, and above all, don't mix up velocity with adding business value.

Recognize that people are different, and don't force them to behave as if they are the same — embrace seniority and embrace the experts. Do not fall into the trap of servant leader failures. Bring back the discipline of documentation using modern tools.

The time has come to set free the Agile spirit, abandoning its original manifesto and replacing generic, first-generation Agile with a personalized one of our own, a fusion of better methodologies for the 2020s and beyond.

Horses were replaced by cars, zeppelins were replaced by modern airliners, paper libraries replaced by the Internet, and waterfall by Agile. In the digital age it's only natural that the pace of change will keep on accelerating. We need revolutions, not evolutions, and right now that means the dawn of the post-Agile era.

It's so exciting to be an active part of this transition. Yes, there will be mistakes, but they will be fixed eventually, until … a completely new idea arrives and steals all the thunder.

Jacek Chmiel is head of Avenga Labs, a division of Avenga focused on technology research and trend analysis as well as innovative, technology-oriented proof-of-concept projects with business partners. He is a technology evangelist and a well-known keynote speaker and presenter at some of the most prominent tech conferences all over Europe. Over the course of his 26 years' experience, Mr. Chmiel has been an insider into strategy and business development, working both on the business side and with technology partners. He has played every role in the production process, has overseen complex projects, and has successfully created concepts for highly sophisticated software and consulting services. As owner of the software development lifecycle process, Mr. Chmiel has profound experience in managing teams of more than 200 specialists, setting career paths, directing the professional development system, and handling recruitment. In addition, he supported the successful implementation of an innovative platform for one of Europe's biggest fintech companies. As CTO, Mr. Chmiel managed a distributed team from several companies comprising the Capital Group Platform and actively supported projects, concentrating on the health and pharma, finance, and automotive industries. He holds a degree in computer science from the Warsaw University of Technology. He can be reached at jacek.chmiel@avenga.com.

STOP! AND PAY ATTENTION!

Removing Agile Roadblocks Brings Exponential Benefits

by Cheryl Crupi

Speed, adaptability, and technical excellence are just a few reasons large enterprises are heavily investing in becoming Agile. Unfortunately, many forget the impact of deeply rooted historical processes and culture around enterprise controls, systems-of-record data, goal setting, funding, and change management. These roadblocks will not change without attention, and ignoring them will drain the momentum from your transformation, tarnishing Agile and DevOps with leaders and workforce alike.

This article examines these critical hidden hygiene factors as well as measures to remedy them. Removing these roadblocks is painstaking, tactical work. It doesn't have the inherent fun and adrenaline of "going Agile." However, investing time in making Agile easy will produce enormous benefits. In my experience … exponentially so.

Simplify Controls and Process

We know that controls and process ensure safety and quality. They evolved to support efficient operations at scale, coordinating the work of hundreds or thousands of people. But look closely at the processes of your organization: procurement, legal, regulatory, audit/risk, architecture, and even new product development. You may be surprised to find they pose a series of waterfall phase gates. Sequential phase gates are a relic of an earlier time. Gates drive up batch size, reduce frequency of delivering to your customer, and impose costly delays and handoffs. A new product might pass through five or six of these waterfall processes before delivery to market. The aggregate wait time is staggering.

Sponsors, practitioners, and teams alike pale at the prospect of repeating enterprise processes on a sprint or quarterly cadence. Literally, it might be impossible to iteratively complete gates designed to be satisfied once in the lifetime of a project. Under the demands of a continuous product delivery model, these processes break down. Teams look to complete a gate once and not look back. Adjustments based on learning are sacrificed by lengthy change-control loops. In the worst case, a major change lands an initiative back at square one.

You must retool core bureaucratic processes and controls to achieve business agility. Incremental improvement is not fast enough. The authors of *The DevOps Handbook* tell us that "bureaucracies are incredibly resilient and are designed to survive adverse conditions — one can remove half the bureaucrats and the process will survive."[1] These processes accreted over years, even decades. Often, no one remembers the conditions or risks they were intended to guard against in the first place. The processes themselves are a source of division and blame between functions, undermining the collective ownership necessary for agility.

You must retool core bureaucratic processes and controls to achieve business agility.

Instead of incrementally fixing these processes, stand up an alternative path for Agile. At first, this path will involve some countermeasures, a "skip the line" fast pass for Agile, and expert support. Make new processes simple and low friction. Minimize handoffs. Work carefully with enterprise control owners to ensure safeguards are intact. Employ Lean tools like the "Five whys" to figure out what you need and what you can throw away.

Build adaptability into your new process and controls. Eliminate, minimize, and automate repeatable activities, in that order. If you automate before eliminating and minimizing, you will rebuild the flaws of the old in the new. Guide your efforts with flow metrics. Validate and fix actual constraints instead of taking aim at commonly

held scapegoats. Everyone may complain about funding approvals, but flow metrics may show server provisioning is the real choke. Good telemetry, and a baseline to start, focuses everyone on the right problem and boosts morale.

As you retool controls and process, consider all the people being impacted. Remember that people are accustomed to using those things. Don't assume they will share your relief at eliminating old process. Change, even good change, is stressful. Provide a clear vision, a roadmap, and a change plan. Help people understand the why, how, and when of change, as well as how they fit into the future. This transparency will dial down resistance and reduce distracting requests to improve legacy processes.

Be clear on your vision to steer confidently with actionable data.

Ensure Actionable Data Fit to Steer

To truly be Agile, we must capture the right data and use it well. Much like enterprise process, data collection habits can perpetuate embedded waterfall thinking. I have witnessed organizations fall short of agility goals for years due to lack of actionable data fit to steer. Fitness relates both to the data you collect and the quality of that data.

Data tells a story about an organization. The data you collect reveals beliefs and frames the questions you can answer. Agile organizations optimize for predictability and outcomes, delivering on cadence and measuring business value. Waterfall organizations measure scope, schedule, and cost (and, sometimes, ROI). The iron triangle — scope, schedule, cost — is rooted in waterfall thinking. Inspect the executive dashboards in your organization. If top performance measures relate to scope, schedule, and cost, your organization is optimized for the wrong variables. Speed and adaptability are served by different measures. Create a product-oriented scorecard that includes business outcomes, predictability, and end-to-end flow metrics in addition to financial and operational metrics.

To adapt is to steer, and to steer well requires quality source system data. Is your data trustworthy? How you, your colleagues, and company leaders talk about your data offers clues. Is the proficiency of people who create the data disparaged? How much time is spent validating and cross-checking? In meetings, do debates center on data reliability rather than what the data tells you? Do you make decisions to proceed or cancel based on data or despite it?

If you spend a great deal of time cross-checking, disparaging, or debating reliability, you have some work to do on data quality. Consider dedicating a team to it. Give this team a clear charter and latitude to work proactively across your organization. Often, process complexity contributes to data quality issues. Process simplification can accelerate data improvement efforts. Also, invest in connecting data end-to-end along product lines. This will involve tracing work across enterprise systems, such as product development, funding, work management, and release.

Be clear on your vision to steer confidently with actionable data. As with legacy processes, apply the Five whys to legacy metrics. Eliminate those that reinforce waterfall thinking. Do not make the mistake of hanging onto legacy metrics too long; this sends the wrong message. Your workforce will march to what is measured instead of what is aspired. As part of your change roadmap, consider ceremonially retiring old metrics that do not align to new ways of working.

Empower Directional Change Without Guilt

Adaptability requires objective evaluation *and* the ability to pivot without guilt or blame. Fixed goals promote people (or teams) who predict and deliver over those that experiment and learn. Organizations can look Agile (hold stand-ups, retros, demos), but fail to achieve value because they cannot or will not pivot. Although not the topic of this article, psychological safety and culture change figure heavily here. Starting with practical hygiene factors, ensure HR goal setting, performance reviews, and rewards make explicit the permission to experiment without personal repercussions.

First, look for indicators of whether your company walks the walk of adaptability. When goals change or an initiative gets cancelled, does it negatively harm people? Does it impact compensation, limit growth potential, or result in people being exited from the company? Is there a culture of minimizing bad news

or of watermelon initiatives (look green on the outside but are red on the inside)? Is the workforce resistant to new practices? If the pace of adopting Agile is slow, despite training and enablement, the workforce may not feel safe trying something new.

The inability to pivot has ramifications at every level. It prevents personal and collective agility. It fatigues rapid learners, experimenters, and innovators. Dissonant empowerment talk will eventually cost retention of top talent you dearly want to keep. Agile practices such as design thinking and customer centricity help evolve a "pivot without mercy" mindset and culture. While this thinking is taking hold, consider specific measures to reinforce adaptable goals and experimentation.

Experimentation, like anything else, requires time and practice. Leaders must model the basic tactics — the stance they take regarding their own experimentation really matters. They must be willing to learn and experiment in front of everyone else. Last year's *Harvard Business Review* cover feature "Building a Culture of Experimentation" notes that "leaders have to live by the same rules as everyone else and subject their own ideas to tests."[2] When leaders use language that emphasizes "the right experiments" or praises "successful" experiments, they create an environment that is actually hostile to experimentation.

Good experiments are well framed and fact focused, and they produce learning regardless of whether they produce the desired outcome. Leaders sharing their experiments — and the results — establishes credibility for changing direction safely. Similarly, changing from annual performance rewards for hitting a goal set a year ago to adaptable objectives and key results (OKRs) makes it explicit that informed changes in direction are the new norm.

Town halls and internal social media are excellent forums to empower directional change without fear or guilt. Airtime in these forums is commonly reserved for success stories. Consider the impact of giving equal airtime to termination of product concepts found unlikely to produce profitable outcomes. This endorsement will promote experimentation. Excellence looks like running thousands of experiments, not tens or hundreds. When a team finds a line of experimentation unproductive, cancel the effort and assign exciting next challenges to that team. Also, frequently communicate strategies and OKRs, and provide guidance for new Agile performance and goals practices.

Accelerate Business Responsiveness Through Lean Strategy and Investment Funding

There is no question that iterative funding is a must-have for Agile. Lean Portfolio Management (LPM) dramatically speeds time to test top-priority ideas. Work is easy to cancel and thus easier to start. LPM is orders-of-magnitude better than protracted annual planning that locks in big bets for next year and beyond. Conventional annual planning kills speed and responsiveness by adding months, quarters, or years of lead time, and by locking in inflexible and unrealistic commitments with tens of approvals. In contrast, LPM offers speed, adaptability, and collective ownership.

The inability to pivot has ramifications at every level. It prevents personal and collective agility. It fatigues rapid learners, experimenters, and innovators.

Deciding what to fund is hard. The list of great ideas always outstrips the capacity to deliver. More effective funding mechanisms enable flow of work. However, it is not the only important factor. A better funding process will benefit your business only if you successfully pick top-value work. Look upstream from mechanics and ensure your strategy and alignment practices are working. Rather than focusing on funding alone, have *alignment, scope, then fund* be your goal.

Fixing the mechanics of funding is much easier than getting alignment on outcomes across top leadership. Value-stream alignment and clear commercial outcomes will help. If you haven't done so, build product-aligned teams. In addition to simplifying the landscape, this provides an opportunity to reduce dependencies and boost predictably. It will also bring all stakeholders to the funding conversation — a tremendous game changer. In joint-funding discussions, I have seen years-long conflicts resolved with mutual understanding as a marketing leader and a technology leader realize building an automation capability is the top priority for both.

Keep an eye on cultural considerations. Going back to experimentation, consider tactics that promote curiosity

instead of advocacy. In an advocacy environment, sponsors strive to convince others that their idea is best. Success looks like winning the debate. A context of curiosity supports keen conversation of alternatives. Success is arriving together at the best answer. Getting this right requires trust, a healthy forum for prioritization, and frank discussion about what your customer really needs. Practices like participatory budgeting offer a different way to facilitate these discussions.[3]

It is normal to prefer habits (even bad ones) over change. Provide clarity as to why new habits are needed.

Partner with Sponsors

As you tackle Lean funding and investment strategy, you will surely encounter sticky questions about already-funded initiatives. Do you play through all committed work? Start fresh? The answer is neither. Rather, provide an outcome-oriented "on-ramp" that is empathetic to teams and sponsors.

Partnering early is essential. Sponsors may be excited about Agile, but the prospect of needing to rewin funding for a big multiyear investment could stop them in their tracks. Likely they have spent months, even years, securing funding. Any progress you have made on making Agile easy will help reduce concerns about onboarding in-flight work. Conversely, high-friction funding will garner resistance.

As with process and controls, build trust and confidence with countermeasures, "skip the line" fast pass for Agile, and expert support. Articulate the case for change in terms that matter to sponsors: maximizing business outcomes. Most organizations have too much work in progress to deliver with speed. Highlight the benefit of strategy alignment to make it easier to "set down some work" so the organization can focus on top-priority work.

Risk management is also compelling. Most sponsors have experienced big commitments gone bad in the form of slipped dates, jettisoned scope, and too many corners cut. No doubt, small, iterative deliverables are less risky. However, it can be hard to imagine big initiatives delivered in increments. It isn't as simple as drawing two-week iterations on a Gannt chart. Offer expert facilitation to break big-bet initiatives into successive minimum viable deliveries (aka minimum viable products, or MVPs) that can be tested with customers.

Use these MVP discussions as field research for simplification work. Look carefully for impediments to breaking down work. The generally accepted "long poles" in the product pipeline might be processes you can work to simplify. Feedback from teams and sponsors offer terrific insights for streamlining. If these top constraints are external, talk with sponsors about ways to exploit these constraints to your advantage.

Bring the Workforce Along

In *Dare to Lead*, Brené Brown says, "We should all be held to be accountable for being both optimistic *and* realistic."[4] As change leaders, we must show a clear pathway to the future. This involves inspiring and empathizing. It requires an eye to practical considerations as well as aspirations. Keep in mind that at every level of your organization, there are people who advanced their careers through waterfall and conventional ways of working. Most have had no opportunity to work in an Agile team.

Agile change leaders are asking their organizations to take a leap of faith. Yes, we all want to be fast, high performing, and dynamic. But for those who have not worked in a thriving Agile environment, the path is not clear. Taking a page out of *Switch: How to Change Things When Change is Hard*, "What looks like resistance is often a lack of clarity."[5]

It is normal to prefer habits (even bad ones) over change. Provide clarity as to why new habits are needed. Offer how-to guidance and job aids to make getting started easy for first-time Agilists. Get feedback from new adopters, and fix the things they don't like. Most people prefer to learn from others. Provide a safety net through learning networks, job shadowing, and communities of practice. As noted, leaders must serve as exemplars. Set them up for success through one-on-one coaching and help in preparing for events like town halls and first demos.

In the course of "going Agile," you will eliminate old processes, practices, and mindsets. Take care not to marginalize the people who are expert in the previous way of working. Success depends on everyone in your organization understanding the vision, the why, and the road to get there.

Conclusion

Achieving true business agility requires a solid foundation to go fast, with safety and focus. To do so, your organization must have simple processes, actionable data, the ability to pivot, alignment, a sponsor on-ramp, and a plan to bring your organization on the journey. Achieving these things involves grueling tactical work. This often-overlooked investment in basics is necessary for both Agile and DevOps to take root in your organization. By fixing the hidden friction points addressed here, you'll inject oxygen into your transformation — breaking down hidden resistance, activating your workforce, and increasing speed. You will make Agile easy, and in doing so you'll accelerate the switch to Agile and greatly amplify the benefits to be gained.

References

[1]Kim, Gene, et al. *The DevOps Handbook*. IT Revolution Press, 2016.

[2]Thomke, Stefan. "Building a Culture of Experimentation." *Harvard Business Review*, March/April 2020.

[3]"Participatory Budgeting." Scaled Agile Framework (SAFe), Scaled Agile, accessed March 2021.

[4]Brown, Brené. *Dare to Lead*. Random House, 2018.

[5]Heath, Chip, and Dan Heath. *Switch: How to Change Things When Change is Hard*. Broadway Books, 2010.

Cheryl Crupi is founder and leader of the MetLife Global Lean-Agile Center of Excellence, where she strives to establish and activate a corps of confident, adaptive, and innovative MetLife Agilists in 40+ countries. For the past 15 years, she has been an Agile change champion promoting Lean-Agile practices in Fortune *500 companies. Ms. Crupi has over 25 years' experience in the delivery of technology solutions in a broad range of fast-paced, results-oriented businesses, including small startups, large enterprises, nonprofits, joint ventures, and university settings, with particular expertise in financial services and insurance. Passionate about delivering the benefits of agility, starting with the building blocks of team agility to the challenges of full business agility, she has worked with the Scaled Agile Framework (SAFe) since its inception in 2011. Ms. Crupi is a SAFe Fellow and a SAFe Program Consultant Trainer. She can be reached at cheryl.crupi@metlife.com or via LinkedIn at linkedin.com/in/cherylcrupi.*

WE'RE ALL IN THIS TOGETHER

Introducing a Client-Vendor Model for DevOps Implementation

by Anna Wiedemann, Dirk Heiss, Nick Bartlett, and Helmut Krcmar

There are two main challenges when it comes to organizations trying to build DevOps teams in collaboration with vendors. First, established firms usually outsource complete projects to vendors with predefined start and end dates linked to concrete results that must be achieved. Strong client-vendor control mechanisms and measurements are implemented, such as result-driven milestones.[1] Second, classic client-vendor project management often focuses on software development work rather than operations. Trying to close the gap between classic approaches to development and operations leads to serious constraints.[2]

A client-vendor DevOps model can solve these challenges. In this article, we provide a detailed description of a digital operations service in which an IT consultancy (the vendor) supports several digital product teams at a large, global financial services company (the client). We describe how a DevOps concept can be integrated beyond company borders such that client and vendor achieve a common goal: providing a stable, high-quality software delivery lifecycle to the end user. By following the DevOps mantra, "you build it, you run it," coined by Amazon CTO and VP Werner Vogel, the digital product teams and the digital operations team built and ran the applications hand in hand.

The Organizational Setup of Client-Vendor DevOps Teams

The client's insurance platform serves as the basis for several digital products created to provide end-user solutions. Overall responsibility for development and operations of the applications lies with the digital product teams (build and run) on the client side. When the number of end users and workload of the digital products increased, further support was needed to guarantee stable provision of the applications (run).

The overall aim is to support the digital products that run on the digital platform. Every digital product has individual development and operations activities and processes, but many operations tasks repeat. The client saw the potential for savings by implementing a shared operations and support service for the digital products. Through scaling effects and continuous improvement, the digital products would benefit from individual service with a high level of reliability.

Digital operations operates in a 24/5 "Follow the Sun"[3] delivery model. The members are in four countries to support all digital product teams in various time zones. Currently, digital operations consists of nine people divided between seven digital product teams located around the world. Digital operations is not only a support service for the digital products. As the name implies, the vendor focuses on operations and support tasks such as case management and proactive monitoring. However, the digital operations team has been fully integrated into the routines and tasks of the digital product teams in order to achieve continuous learning and improvement.

The digital operations team defines a subject matter expert (SME) for every digital product. The SME attends the product team's daily stand-ups, review meetings, and retrospective meetings. The information from the meetings is then shared with the digital operations team in knowledge management sessions. Every SME has a representative in the digital operations team (see Figure 1). By bringing these two perspectives of development and operations together, the overall aim is to achieve a DevOps process, forgoing the highly standardized processes and strict responsibilities of traditional infrastructures.

Tasks in the Client-Vendor DevOps Team

The client-vendor DevOps team facilitates self-management. The team delivers the complete software delivery lifecycle to the user.[4] The client and the

vendor work together on plan, build, test, deploy, and operation steps (see Table 1).

Planning Activities

Digital operations SMEs are part of the planning meetings. Digital operations and the digital product teams elaborate the backlog for the next sprints. Digital operations brings in experience from user support and case management and explains how to improve user satisfaction through new features or changes in the applications. In addition, digital operations can discuss which new features are being planned with the developers. This makes it much easier for digital operations to answer product-related user questions.

Building Activities

The majority of software development lies with the digital product teams. However, digital operations automates manual processes by using scripting languages. This means digital operations analyzes user requests and incidents on a regular basis. With the help of IT service management tools such as ServiceNow,[5] the digital operations team can automate repeating service requests such as user access requests or application registration processes. In addition, due to measures such as shadowing development sessions, the vendor's staff is integrated into the build activities of the digital product teams.

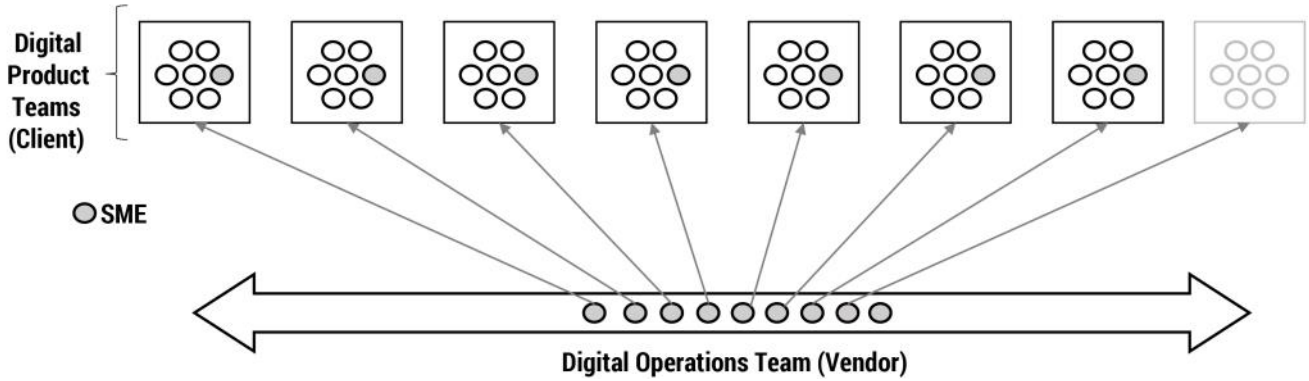


Figure 1 — Client-vendor DevOps model.

	Description
Plan	The team does regular planning, and the team lead is responsible for meeting business department requirements and attending planning sessions. All team members plan the development and operations work. The team estimates time needed for unplanned work.
Build	Software development includes the building and compiling of new software code. New deployments run automatically through the different stages of the environment.
Test	Continuous integration enables automatic testing of new code when a team member provides it to the build environment. This approach helps continuously merge different software components.
Deploy	Deployments include the packaging and deploying of new software features across all environments to production. When the software code is in the production environment, it is visible for the end customer.
Operate	Software operations processes begin with the deployment of new software features in the production environment. Operations processes include the monitoring of running software, solving problems, and automatically logging data through protocols.

Table 1 — Process steps of a DevOps team.

Test Activities

The digital operations team has access to test and development environments, so when end users have how-to questions, the digital operations team can test and investigate. In addition, digital operations manages all software defects reported by the users. After the digital product team has fixed the problem, digital operations tests the solution before informing the user. Digital operations also conducts functional tests with the digital product team.

> *Digital operations supports collaboration with partners in a continuous improvement process, identifying ideas to improve collaboration and the overall service.*

Deployment Activities

Digital product deployment includes packaging and deployment of new software features to the production environment. Production deployment of new features is the responsibility of the digital product team, but some digital product teams are responsible for API management. When that happens, digital operations prepares and implements API deployments in all environments.

Operations Activities

The digital operations team focuses on three main operations activities: monitoring, case management, and knowledge management.

Monitoring

The digital product solutions are cloud-based and run on a public cloud. Digital operations proactively monitors the applications using a variety of tools. Depending on the digital product's setup, monitoring dashboards may be implemented in the public cloud environment or using an open source tool (e.g., Grafana[6]).

The digital operations team is automatically notified of any alert peaks or availability outages and sets up notification lines and response patterns. If an alert is determined to be a product-related critical issue that needs to be solved by developers, the information and analysis results are forwarded to the digital product team. If the alert is assessed as uncritical (e.g., the cloud provider updated the system), digital product teams are simply informed that is the case.

Case Management

The digital product developers usually work in one time zone, but the digital operations team works 24/5 to guarantee service for all geographic locations. The digital operations team serves as single point of contact for the end user, taking over service requests and incident management and acting as a classic first- and second-level support.

In the event the incident cannot be solved by digital operations (e.g., a software defect in the application code), the digital operations team collaborates with the digital product team. Digital operations conducts the problem analysis and creates a so-called software bug with necessary information on the virtual Agile (Kanban) board of the digital product team. The digital product team takes over and informs digital operations about a solution. Digital operations SMEs participate in daily stand-ups to explain current cases and ask for updates. When the incident is resolved and tested, the user is informed by digital operations.

Knowledge Management

Digital operations maintains a knowledge base for users that serves to answer frequently asked questions and helps users before they create a case. In addition, a customer management tool was set up, where users could find the knowledge base, along with information about digital product features and application-related issues. With this self-service approach, users get the necessary information with a minimum amount of user requests.

Client-Vendor DevOps Continuous Improvement

Digital operations supports collaboration with partners in a continuous improvement process, identifying ideas to improve collaboration and the overall service. The goal is to enhance the service quality and reduce costs through improvements such as manual activity automation.

The client and vendor work together to drive and implement continuous improvement ideas. New ideas about how to improve the client-vendor DevOps model come from regular brainstorming sessions, including those aimed at time- and cost-reduction for all digital products. A committee comprising client and vendor personnel discusses and assesses the ideas, and when an idea is approved by the committee, digital operations provides a fact sheet with related information and an ROI calculation. After the digital product team collaborates with digital operations to implement the idea, the fact sheet is updated with the final costs and whether or not ROI was achieved.

For example, the digital operations team was tasked with supporting an authentication tool that involved time-consuming manual addition of new applications to the tool in a cloud environment. Since the process was always the same, digital operations took the opportunity to automate it. The digital operations team had several discussions with the digital product team to completely understand the technical details, then other stakeholders were consulted, including those responsible for the cloud environment and the IT service management tool.

Digital operations wrote the script, tested the automation, and implemented the solution, reducing the application-adding process from almost an hour to a couple of minutes. The client became excited about the potential for similar automation opportunities to reduce cost and manual effort. Digital operations presented the automation approach to a broader audience at the client, and several people expressed interest into the procedure. The script, which can be easily adapted to automate other processes, was shared with the stakeholders.

Control Mechanisms Between Client and Vendor

Although collaboration between vendor and client is based on a high level of trust, some control mechanisms have been implemented as a basis for collaboration. The digital operations team is managed by two service managers, one on the client side and one on the vendor side. The activities of service managers include (but are not limited to) client-vendor reporting, checking the digital operations team's utilization, managing knowledge sharing, and scheduling onboarding for new digital products.

On the client side, every digital product has a product manager as a contact person for software defect management and general collaboration improvement. The product managers receive monthly reports and participate in steering committee meetings involving both vendor and client.

The client and vendor have agreed on control mechanisms such as standard service-level agreements and key performance indicators. These are measured on a monthly basis with the help of digital tools and regular quantitative surveys to understand the satisfaction level of the digital product teams. Additionally, improvement ideas are introduced during steering committee meetings, fostering discussions about possible service improvements.

New ideas about how to improve the client-vendor DevOps model come from regular brainstorming sessions, including those aimed at time- and cost-reduction for all digital products.

Conclusion

The client-vendor DevOps concept provides high amounts of value for both sides. The implementation of a value-driven collaboration between digital product teams and digital operations leads to benefits in several areas:

- The digital operations team is seen as part of the digital product team. The client-vendor DevOps model has a product-oriented setup with some control mechanisms derived from classic project management.
- Room for continuous improvement and learning is facilitated because ideas and knowledge are proactively driven and continuously shared.
- Thanks to integration and improvement measurement, digital products and digital operations both benefit from manual task automation, leading to cost reductions and time savings.

The client-vendor DevOps model introduces a new collaboration arrangement that has the potential

to dramatically improve offshore outsourcing engagements.

References

[1]Gregory, Robert Wayne, Roman Beck, and Mark Keil. "Control Balancing in Information Systems Development Offshoring Projects." *MIS Quarterly*, Vol. 37, No. 4, December 2013.

[2]Humble, Jez, and Joanne Molesky. "Why Enterprises Must Adopt DevOps to Enable Continuous Delivery." *Cutter IT Journal*, Vol. 24, No. 8, 2011.

[3]See Wikipedia's "Follow-the-sun."

[4]Krancher, Oliver, Pascal Luther, and Marc Jost. "Key Affordances of Platform-as-a-Service: Self-Organization and Continuous Feedback." *Journal of Management Information Systems*, Vol. 35, No. 3, 26 October 2018.

[5]ServiceNow.

[6]Grafana Labs.

Anna Wiedemann is a Research Associate at Neu-Ulm University of Applied Sciences (Germany). Her current research comprises Agile IT, DevOps teams, and qualitative research methods. Dr. Wiedemann has been published in European Journal of Information Systems, Communications of the ACM, *and several conference proceedings,* including International Conference on Information Systems, European Conference on Information Systems, *and* Hawaii International Conference on System Sciences. *She earned a master's degree in advanced management from Neu-Ulm University of Applied Sciences and a PhD in information systems from the Technical University of Munich (TUM, Germany). She can be reached at Anna.Wiedemann@hnu.de.*

Dirk Heiss is Head of Digital Platforms at Munich RE. Previously, he held the positions of Head of Global Infrastructure and Head of Risk and Security Management. Dr. Heiss has more than 25 years' experience in the IT industry and has worked for various technology organizations, including Gartner, Horváth & Partners, and the IT Service Management Forum. He is a certified Gartner TCO Expert, holds a Foundation Certificate in IT Service Management, and is a PMI-certified Project Management Professional (PMP). Dr. Heiss has also worked in academic research in computer-aided image processing in radiology. He earned a degree in medicine from Ludwig-Maximilians University (Germany) and Harvard Medical School. He can be reached at dheiss@munichre.com.

Nick Bartlett is Head of Digital Services at Munich RE. Previously, he worked at Maersk as Head of IT Delivery for e-commerce/customer engagement as part of a significant digital transformation driving Agile at scale and moving from a project to product-based DevOps organization. Before this, Mr. Bartlett worked for 10 years as a management consultant at PA Consulting specializing in program delivery across a range of industries. He holds a master's degree in mechanical engineering from the University of Birmingham (UK). He can be reached at nbartlett@munichre.com.

Helmut Krcmar is Professor Emeritus of Information Systems in the Department of Informatics at the Technical University of Munich (TUM, Germany), with joint appointment at the School of Management, where he is currently Founding Dean and Delegate Officer of the President. Dr. Krcmar also served as Postdoctoral Fellow with IBM Los Angeles Scientific Center and Assistant Professor of Information Systems with the Leonard Stern School of Business, NYU, and Baruch College, CUNY. His research interests include information and knowledge management, engineering, piloting, and management of innovative IT-based services, along with computer support for collaboration in distributed and mobile work and learning processes. Interdisciplinary work incorporates areas such as accounting, mechanical engineering, and healthcare. Previously, Dr. Krcmar was Chair of Information Systems, TUM; Dean, TUM; Chair of Information Systems, Hohenheim University (Germany); and Dean of the Faculty of Business, Economics, and Social Sciences, Hohenheim University. He has coauthored several research papers in MIS Quarterly, Journal of Management Information Systems, Journal of Information Technology, Information Systems Journal, Journal of Strategic Information Systems, *and* European Journal of Information Systems. *Dr. Krcmar is a Fellow of the Association for Information Systems (AIS) and a member of Acatech National Academy of Science and Engineering. He can be reached at helmut.krcmar@tum.de.*

CUTTER Business Technology Journal

About Cutter Consortium

Cutter Consortium is a unique, global business technology advisory firm dedicated to helping organizations leverage emerging technologies and the latest business management thinking to achieve competitive advantage and mission success. Through its research, training, executive education, and consulting, Cutter Consortium enables digital transformation.

Cutter Consortium helps clients address the spectrum of challenges technology change brings — from disruption of business models and the sustainable innovation, change management, and leadership a new order demands, to the creation, implementation, and optimization of software and systems that power newly holistic enterprise and business unit strategies.

Cutter Consortium pushes the thinking in the field by fostering debate and collaboration among its global community of thought leaders. Coupled with its famously objective "no ties to vendors" policy, Cutter Consortium's *Access to the Experts* approach delivers cutting-edge, objective information and innovative solutions to its clients worldwide.

For more information, visit www.cutter.com or call us at +1 781 648 8700.