SQmag

THE VOICE OF SOFTWARE QUALITY

SCALING DEVOPS IN A DIGITAL ENTERPRISE

Principal Consultant and DevOps Architect Hari Gurumoorthi on how to survive the velocity of change

ORGANISATIONAL AGILITY
Managing Director Debra Paul takes a holistic approach

HIGH-PERFORMANCE QUALITY ENGINEERING
Rik Marselis and Wouter Ruigrok answer the questions Why? Who? How?

MOVING WITH AGILITY
A glimpse in time by Luis Francisco Contreras González

AGILE METHODS SUPPORT GREATER DIVERSITY
"Being a woman in testing, it's important to lift other women up with you" Adriana Benitez, Computer Systems Engineer

09
JUN 2021

A4Q
Design Thinking

I AM A
DESIGN THINKER!
CERTIFY YOUR CREATIVE POWER
ISQI.ORG

iSQI
SUCCESS IS YOURS

Stephan Goericke
CEO, International Software Quality Institute

Following over a year of profound socio-economic disruption around the world due to COVID-19, organizations continue not only to adapt to the challenges that have tested even the most robust of businesses but to seek ways to accelerate change.

Business agility has never been more important and embracing agile ways of working has, and will, give significant advantages as an enabler to digital transformation. Agile working practices are now increasingly being incorporated into the culture and daily activities of teams, not least where they have been compelled to find ways to maintain productivity whilst working remotely. Agile methods are playing an important role in boosting innovation and speed to market with products and services that deliver customer value, and quality, in a time of fast-changing conditions.

Agile is a multifaceted topic and in this new issue of SQ mag, our authors explore agile from multiple perspectives including organizational agility, high-performance quality engineering, agile testing, DevOps, and more.

We hope the articles are of interest to you and thought provoking – enjoy the read!

Yours sincerely,

CONTENT

ORGANISATIONAL AGILITY

The last 12 months have seen dramatic change in all aspects of our lives. Organisations everywhere have had to respond to survive and many haven't succeeded. Everyone who runs an organisation (a grouping of people who are offering a form of service or product) has heard the term organisational agility and are aware of the expectations the term raises. But what does organisational agility actually mean and how is it achieved and demonstrated?

This article explores three dimensions that offer a basis for organisational agility. These dimensions are:

1. Rationale: understand why change is needed.

2. Context: evaluate the organisation and what it offers.

3. Approach: decide on how to act.

The article also introduces four key elements that underpin organisational agility and discusses an approach applied by a case study organisation in response to the global pandemic.

Rationale

There are many factors that cause an organisation to change how it is organised and operates. These drivers for change may be apparent and readily identified, or may need to be sought through business research. The impact caused by change drivers ranges from localised change to extensive organisational transformation. It is possible to consider three types of change driver, as follows:

- Extreme drivers such as a global pandemic where change is inevitable.
- Identifiable drivers such as regulatory change or technology development where change is manageable
- Subtle drivers such as emergent consumer preferences where change is optional (although often advisable).

Identifying and evaluating the drivers are critical first steps. While most organisations are able to do this, they often miss – or elect to miss – some of the more subtle drivers. Sometimes, they may choose to ignore change drivers, even those that are extreme. The ways in which an organisation seeks out and responds to the change drivers, says a great deal about an organisation and its ability to respond with agility.

Context

Analysing the context focuses initially on the different types of organisational leadership. It is also possible to consider three types of leadership styles, as follows:

- Static, head in the sands: trying to ignore what is happening even where the drivers are extreme, or sometimes particularly where the drivers are extreme. Often comfortable with identifiable drivers.
- Aware, waiting for the data to make the decision: deferring action until a business case tells decision makers what to do; always insisting on a business case showing a return on the basis of predictable, financial benefits before approval is given.
- Ambitious, focusing on the outcome: assessing each situation at the required pace; forward thinking; acknowledging ambiguity; making decisions using the available data and accepting uncertainty.

The leadership style has an impact on those working within the organisation and is manifest in two key elements for organisational agility: motivation and mindset.

Motivation is concerned with whether or not there is a drive and acceptance of the need for change. Are the outcomes in view? Is the motivation focused on achieving outcomes? **Mindset** is concerned with whether there is a problem-solving mindset in place. Is there a willingness to uncover the root causes of the problem and move towards the outcomes that will resolve them?

Organisational agility requires positive motivation and a problem-solving mindset. These characteristics must be demonstrated and communicated by the leaders of the organisation and must also be supported at all levels.

Approach

The approach is concerned with deciding how the work to change the organisation will be done. Dave Snowden (BA Conference Europe, 2020) said 'we need chefs not recipe books' and this is a critical element for organisational agility. Where organisational agility is the aim, toolkits, approaches and frameworks are needed, defined methods are less helpful so should be avoided or applied advisedly.

Approach covers two further elements required for organisational agility: customer centricity and capability leverage. **Customer centricity** is at the heart of organisational agility and is concerned with ensuring the focus is on meeting customers' needs rather than achieving internal objectives. It must be based on a clear understanding of the different levels and types of customer, and their value expectations.

Capability leverage requires the existing and required capabilities for organisational change to be known, developed (if necessary) and leveraged. Applying capability in different contexts or to deliver new or enhanced services enables organisational responsiveness and adaptability. The elements required for organisational agility are summarised in Figure
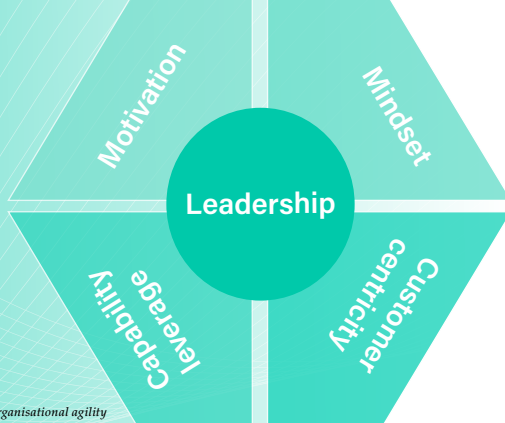
Figure 1: Elements required for organisational agility

Case study: AssistKD

Like many other companies offering training services, AssistKD was facing a complex and problematic situation in March 2020. The global pandemic was imposing constraints on physical interaction that were likely to prevent the delivery of many of our services. With venues closing and most travel prohibited, my colleagues and I faced a potentially bleak future. At an early stage, the leadership team were in agreement that this was an extreme situation and an immediate response was necessary. Motivation to take action was very high. Fortunately, our team included specialists in business analysis, digital technology and business change, expert problem-solvers who used every analytical tool in our collective toolbox to deal with the situation. Perhaps even more fortunately we had a highly motivated, skilled and dedicated team, a robust and adaptable technical infrastructure, and many customers who willed us to succeed and helped where they could. There was a palpable sense of goodwill and collaboration. We ensured we understood the rationale and the issues we were facing. We thought about our organisational context, our desired outcomes and our willingness to take action. Finally, we decided on how we were going to approach the situation. There were six key steps we carried out to respond to the situation. These steps are summarised as follows:

1. Ensure leadership and team commitment.

2. Set up the team to do the work. Empower everyone as much as possible. With transformational change, organisational agility is key and control over every element is not possible. Trust must be explicit and real.

3. Apply service thinking. Analyse the service offering to be delivered, the value proposition, the activities and the intended beneficiaries. Consider the value expectations of the beneficiaries, including financial, functional and experiential needs. Ask how it would be possible to collaborate with customers and co-create value. Understand the outcomes all parties want.

4. Apply design thinking. Use the Design Council's double diamond and encourage divergent and convergent thinking to explore and define the problem before exploring and formulating the solution. Apply iterative processes to generate ideas, build prototypes, obtain evaluative feedback and learn. Explore customer experience requirements. Keep in mind that people are people not machines.

5. Understand and leverage capabilities. True enterprise architects are vital and should be supported by business, data and infrastructure architects. Use their skills but direct where they are needed. Evaluate the capabilities to identify where new skills, tools, techniques and equipment are needed.

6. Deliver incrementally where possible. Strive for continuous improvement and value co-creation. Request feedback from customers and everyone in the team. Look for new opportunities for further collaboration and value realisation.

One year on, we look back on the work we did as a team and are amazed at what we have achieved. The three tenets of organisational agility – rationale, context, approach – provided a strong foundation. The four elements for organisational agility – motivation, mindset, customer centricity and capability leverage - were applied holistically across every part of our service delivery. Everyone learned a lot and continue to do so. Organisational agility doesn't fit a time frame, it is embedded in continuous effort.

Conclusion

While the phrase 'change is constant' is often uttered, some organisations fail to act. Many people talk about 'organisational agility' and fail to appreciate that this requires leadership, trust and holistic action. Along with so many other concerns, last year highlighted where organisational agility really exists and where it is just a convenient tag line. In a world where global change can emerge so quickly, organisations need leaders to understand the essence of organisational agility and the responsibilities this confers upon them. This has never been more important and relevant.

↗

Debra Paul

is the Managing Director of Assist Knowledge Development Ltd, a training and consultancy company specialising in business analysis and business change. Debra co-authored the publications, Business Analysis, Agile and Business Analysis, Business Analysis Techniques and Delivering Business Analysis. Debra is a keen advocate of service science and design thinking, and developed the Business Analysis Service Framework during her doctoral research. Debra is a regular speaker at business seminars and IS industry events. She has delivered keynote presentations at conferences, is a founder member of the BA Manager Forum and was the chief architect of the Advanced International Diploma in Business Analysis.

WHAT'S

SOFTWARE TESTING COMMUNITY IN NIGERIA

NGSTQB has been reaching out to testing stakeholders in Nigeria. NGSTQB has also partnered with The International Software Quality Institute (iSQI) has been appointed to be its exam provider. iSQI has thus been handed the responsibility of conducting all ISTQB exams either online or location-based in Nigeria.

More information:
https://blog.isqi.org/what-are-the-first-steps-of-the-nstqb-in-order-to-further-develop-the-software-testing-community-in-nigeria

NEW

ADVANCED PRODUCT OWNER TRAINING AND CERTIFICATION

The Advanced Product Owner curriculum by nextnormal.academy is a framework-neutral open source training scheme with a corresponding optional certification. The aim is to convey the relationships and interplay between different methods and tools of agile product management.

Training and certification are currently only available in German. However, they will be offered in English later this year!

More information:
https://blog.isqi.org/new-advanced-product-owner-training-and-certification

A4Q WORLD CONGRESS AUTUMN 2021

This April 2021 the Alliance for Qualification was proud to host one of the world's largest virtual international conferences. The mission was to bring together and connect the software quality industry. Speakers and experts presented current findings, comments and discussed future trends in the IT field.
Coming Soon in Autumn 2021

The A4Q World Congress team was happy to announce the next virtual conference in autumn 2021!

More information:
https://a4qworldcongress.com/

A4Q WORLD CONGRESS

NEWS FROM CANADA

The International Software Quality Institute (iSQI) is happy to announce the new cooperation with the Canadian Software Testing Board (CSTB) to provide exams in the Canadian territory. iSQI will be the exclusive exam provider for all ISTQB exams, supporting the Canadian Board with remote and in person exams.

All participants wishing to take an exam in Canada will have the chance to choose between English and French language, among others.

Schedule your exam:
www.isqi.org

CSTB ● CCTL

HAPPENING

HIGH-PERFORMANCE QUALITY ENGINEERING… WHY? WHO? HOW?

Business people don't ask for IT systems anymore. They want business value. And the supporting IT as such is not relevant for them. Also, they expect to get the right quality at the right moment. IT delivery teams may struggle to deliver this "quality at speed". In this article, you will learn how a focus on quality engineering by cross-functional teams in a DevOps culture (or other organization with an Agile mindset) will enable organizations to achieve the business value they are longing for.

High-performance quality engineering: Why?

Organizations today cannot exist without information technology (IT). In 2020 we learned that silicon chips (in smartphones, tablets, laptops, etc.) keep the world turning when people can't travel to get together. The first computer was created almost 80 years ago. And for about half a century IT was the territory of technical people only. Since the 1990's business people started to get used to what IT could do to achieve business value. Today they don't want to be bothered by technical talk, they want business value and they want it fast. So, IT delivery teams need to adjust. It is exactly 20 years ago that the Agile manifesto was written by a group of visionary IT people. "We are uncovering better ways of developing software by doing it and helping others do it." they said. Their vision has inspired many new approaches, but organizations still often struggle with implementing such new ways of high-performance IT delivery.

> **QUALITY ENGINEERING IS ABOUT TEAM MEMBERS AND THEIR STAKEHOLDERS TAKING JOINT RESPONSIBILITY**

Today's IT culture: DevOps

One of the ways to achieve a higher speed of IT delivery is to integrate development and operations efforts. This movement, using the simple name "DevOps" started about a decade ago and currently is being implemented by many organizations. We define DevOps as a cross-functional systems engineering culture that aims at unifying systems development (Dev) and systems operations (Ops) with the ability to create and deliver fast, cheap, flexible and with adequate quality, whereby the team as a whole is responsible for the quality. Usually, other expertise's like Business Analysis and Quality Assurance (including testing) are integrated in the team. A DevOps culture has an Agile mindset that can be supported/implemented by, for example, the Scrum framework.

High-performance quality engineering: Who?

To properly implement a DevOps culture, the people need to organize themselves in cross-functional teams. The main goal of this type of organization is that teams are, to a certain degree, autonomous. Which means the team members together have all skills, knowledge, and facilities to perform their tasks. This makes that they can execute almost any of their tasks without support from outside the team. To be able to work as a truly cross-functional team all team members are allowed to pick up any task, so they regularly switch roles. A team member for example can pick up a development task at one moment in time and perform a testing task at another moment. Also, the team members apply various quality measures, such as "pairing" in which two team members pick up one task together. Applying such quality measures enables them to quickly deliver the right quality, and at the same time improve their skills by learning from each other. And this brings us to one of the core concepts of high-performance cross-functional IT delivery teams: they constantly strive to improve the product (IT-system), process (for IT delivery) and people (both individual and team skills).

High-performance quality engineering: How?

To implement this continuous improvement focus in your Agile, Scrum or DevOps organization we introduce the concept of "quality engineering". Quality Engineering is about team members and their stakeholders taking joint responsibility to continuously deliver IT systems with the right quality at the right moment to the business people and their customers. It is a principle of software engineering concerned with applying quality measures to assure built-in quality. Quality engineering is very broad, it encompasses quality assurance and testing, but also other engineering and IT delivery activities that relate to creating built-in quality. With today's wide variety of IT delivery models in mind, in our book "Quality for DevOps teams" we have described a common set of topics that are always relevant for quality engineering, regardless of the IT development, operations and maintenance approach that is followed by the organization. The way these topics are addressed in your situation depends on many factors, not in the least by the IT delivery model you use. We are convinced, however, that for effective and efficient QA & testing, all of these topics need to be addressed in one way or another. While describing the topics we noticed a distinction should be made about the kind of activities a topic relates to. This resulted in two overarching groups: Organizing topics and Performing topics. The organizing topics are aimed at orchestrating, arranging, planning, preparing, and controlling the quality engineering activities. The performing topics are aimed at the actual operational quality engineering activities. (This division is not purely black-and-white, some topics are mostly organizing and somewhat performing or the other way around.)

Figure 1: the Organizing and Performing topics of TMAP

The TMAP certification scheme for quality engineering in high-performance IT delivery

Three different groups of people are involved in quality engineering.

1. All members of a cross-functional team need to collaborate to perform their tasks in delivering business value with quality at speed. They therefore constantly focus on delivering business value with the right quality at speed.

2. Some team members specialize in performing dedicated quality engineering activities (such as automating tasks in a CI/CD pipeline, specifying Infrastructure as Code, reviewing as part of pull requests, applying unit testing principles and mutation testing, etcetera) and support the other team members in their tasks. These engineers we call performing people.

3. Especially in organizations with several teams, organizing IT delivery requires quality engineering at scale. Organizing this requires managing dependencies, thus enabling good communication and collaboration between teams. These product owners, release train engineers, end-to-end quality orchestrators, scrum masters, agile coaches, and others we call organizing people.
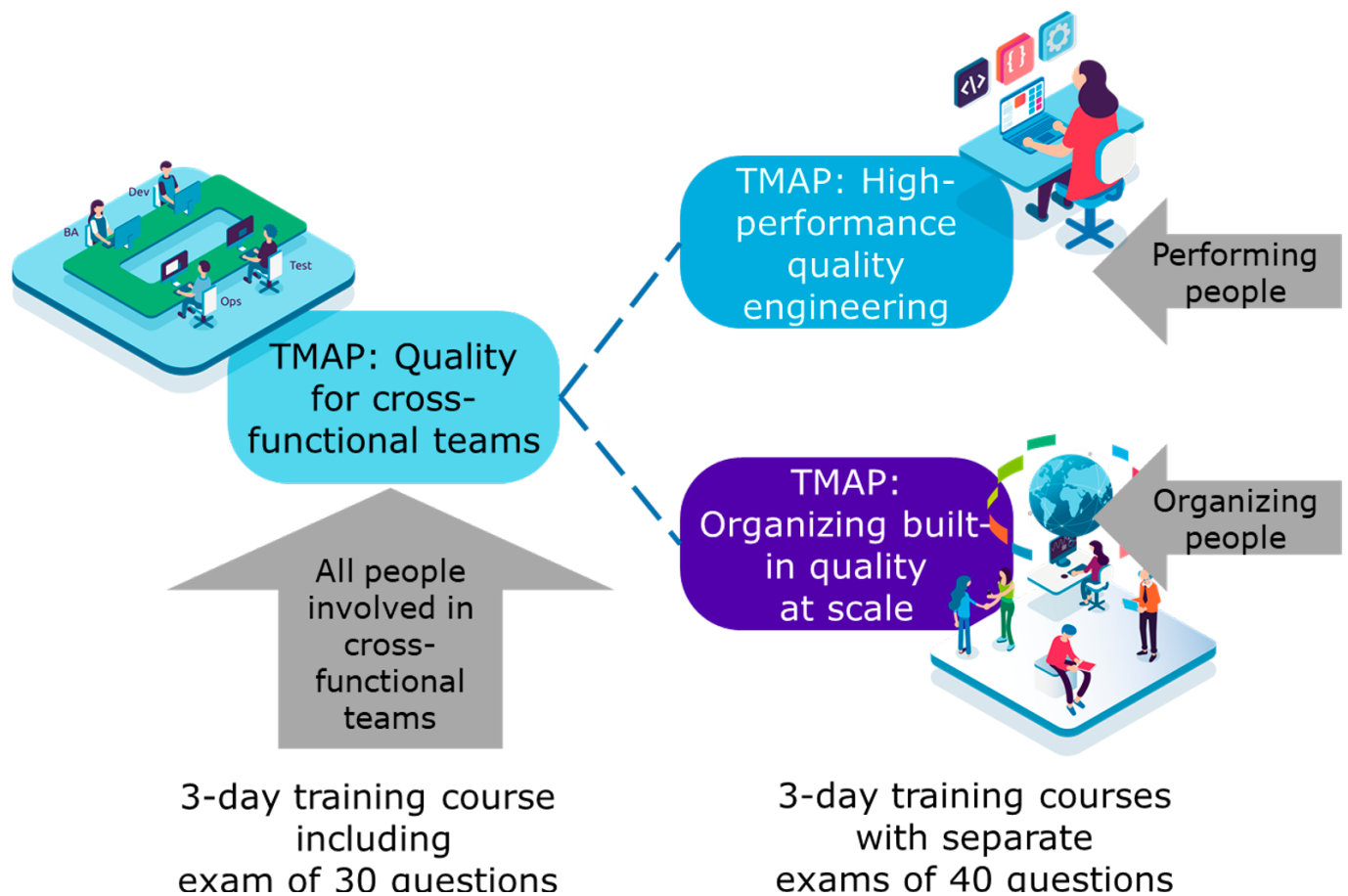
10

The renewed TMAP certification scheme (that was created by Sogeti together with exam provider iSQI) supports these three groups of people with the following certifications:

- TMAP: Quality for cross-functional teams
- TMAP: High-performance quality engineering
- TMAP: Organizing built-in quality at scale.

TMAP started in 1995 with a focus on testing. In the 25 years of its existence gradually the scope broadened to the current body of knowledge for quality engineering in IT delivery.

Conclusion

To continuously deliver IT systems at speed with a focus on business value, high-performance IT delivery teams (such as in a DevOps culture) integrate quality engineering in their way of working. These cross-functional teams continuously improve the products, processes and people that are required to deliver value to the end users. TMAP supports people in and around teams in implementing quality in their IT delivery culture, with practical examples, useful knowledge and some theoretical background. Modern teams benefit from this: instead of relying on the strength of departments or individuals, they aim for being a high-performing team. By following "you build it, you run it" they put quality dead center because they would feel the pain of issues themselves.

↗
About the authors and more info
Rik Marselis is Principal Quality Consultant. Wouter Ruigrok is Agile Quality Coach. They both are co-authors of the book "Quality for DevOps teams" in the TMAP body of knowledge, and they work at Sogeti in the Netherlands.

For more information on the TMAP body of knowledge for quality engineering, visit www.TMAP.net.

For more information about TMAP certifications visit https://pages.isqi.org/tmap-2020/ TMAP is a registered trademark of Sogeti.

A4Q SCHEME "A4Q TESTING FOUNDATIONS FOR DEVELOPERS (TF4D)" RELEASED

The A4Q Testing Foundations for Developers (TF4D) has been developed by A4Q in cooperation with the German Testing Board e.V. This certification makes the content of the ISTQB® scheme even more accessible and relevant to developers. The TF4D curriculum will help participants to gain a basic understanding of testing in the software development lifecycle. All main test techniques are explained in the syllabus and static test, review processes and static analysis are given special attention. White Box & Black Box testing approaches are also covered.

More information:

https://isqi.org/en/82-a4q-testing-foundations-for-developers-tf4d.html

iSQI®
SUCCESS IS YOURS

IS EVERYWHERE!

iSQI has several Social Media Channels. Follow us to keep up with new trends in software testing, development, agile working methods and much more.

- **iSQI Group**
www.facebook.com/iSQIgroup
www.twitter.com/iSQICert
www.linkedin.com/company/isqi-gmbh/
www.instagram.com/isqicert/
- **Serbia**
www.facebook.com/isqiserbiaeasterneurope
www.instagram.com/isqi_serbia/
- **DACH**
www.facebook.com/isqideutschland
www.linkedin.com/company/isqi-dach
- **iSQI en Español**
www.facebook.com/isqies
www.facebook.com/groups/isqicertificaciones
www.facebook.com/isqies
www.linkedin.com/company/isqi-gbmh/
www.instagram.com/isqi_es/
t.me/isqi_espanol

HAVE YOU

12

SEEN ::

RWANDA UPDATE

The first ISTQB CTFL training in Rwanda was concluded virtually in Kigali. Provided under the Beaming Knowledge in Africa initiative with the African Institute for Mathematical Sciences (AIMS) and the iSQI Group. Following a 3-day training and sitting for the certification exam, 12 students passed successfully and were awarded their globally recognized certificates!

Congratulations!

More information:

https://www.beamingknowledge.com/first-group-of-trainees-obtain-the-istqb-ctfl-certificate-in-rwanda/

SCALING DEVOPS IN A DIGITAL ENTERPRISE

From the lens of Agile and Lean

DevOps has revolutionized the way organizations manage software delivery. As every business is becoming an IT business, Software Delivery and Operational Performance plays a vital role in deploying changes, defect fixes, or configuration faster by removing waste in the delivery process. The Outcomes of software delivery and operational performance are measured not just by the speed of delivery but also by the software's stability.

There are reported isolated islands of success, but the undeniable fact is that automation tools or cloud adoption to try to achieve enterprise transformation have not worked effectively. The challenge emerges when scaling the success of small teams to an enterprise-level adoption, like how Scrum, an Agile framework, struggled. Using the yardstick of a small team's DevOps practices for enterprise fails because each business unit's dynamics differ in their culture, structure, technology stack, and the competition it faces. This article analyzes the challenges in managing high-velocity changes in an enterprise from the lens of Lean and Agile principles.

Surviving High Velocity of Change

As digital technology is becoming the core of every industry, organizations need to thrive in adopting technology faster, and infuse it into the mainstream to gain a competitive edge. But what is more important at the enterprise level is using these technologies to observe, measure, and quickly respond to customer needs. As software becomes ever more integrated into every product and service, making a big shift in rethinking and reorganizing quickly becomes the key operational challenge for businesses of all kinds. We need a management model that does not merely account for, but embraces, continuous change (Seiden & Gothelf, 2017).

Adoption of Agile at scale for enterprise

Agile paved the way for development teams to embrace change but did not do enough for business and operations. Agile at scale was challenging for larger organizations with a legacy mindset. They picked only the rituals like the daily scrum but implemented the "Water-Scrum-Fall" model (West, 2011). Large-scale Agile frameworks like SAFe, LeSS, Spotify, Nexus, and Scrum-at-Scale evolved to streamline the adoption in larger organizations. Some of the common challenges (Carrol, 2019) associated with large-scale Agile frameworks are:

- Inconsistency in defining the concepts and terms of scaling Agile.
- Readiness and appetite for adopting large-scale Agile frameworks.
- Balancing the organizational structure and large-scale Agile frameworks.
- Top-down versus bottom-up implementation of large-scale Agile frameworks.
- Lack of evidence-based use of large-scale Agile frameworks.
- Maintaining developer autonomy.

However, there is very little empirical research examining these large-scale Agile frameworks and their impact on the organization's transformation process in response to change.

Managing cross-functional requirement changes

Cross- or non-functional requirements like security and infrastructure are traditionally handled by Operations and Infrastructure teams. These teams are seen as support services in an enterprise. Research shows that most of these teams

> "
> ## WHATEVER GOT YOU HERE MAY WELL PREVENT YOU FROM GETTING THERE.
>
> —Marshall Goldsmith

are "outsourced as a function" and that their agility is mostly governed by the processes (State of DevOps Report, 2018). Outsourcing is viewed as a quick way to expand capabilities and bandwidth. Driving change takes a long time as there are huge hand-offs involved. Generally, Operations teams are built around stability and not for speed.

Responding through DevOps

DevOps emerged a decade back, laying a foundation layer on top of Lean principles, and resolving the weaknesses of Agile frameworks. Most organizations started their DevOps journey by building audacious goals to automate their current painful processes, like build and deployment, using CI/CD tools. These tools provided a quick win in terms of seeing features and defects move faster, mostly for product owner review in staging and sometimes to production. Beyond these advantages, it became difficult to showcase the value of a DevOps transformation to the business. Organizations recruited talent in the tool stack and claimed to be unicorns of DevOps. That DevOps team of unicorns helped automate the processes but did not have the opportunity to understand the end-to-end value stream. Ultimately, organizations suffered from having created a siloed team between dev and ops. As a result, DevOps is viewed as a methodology for automation and not as a value-driven initiative for its customers.

DevOps Research on High-Performing Organizations

At the beginning of 2014, DevOps Research and Assessment (DORA), which is now part of Google, built an empirical model to identify capabilities adopted by high-performing organizations and patterns that impede the others. The "State of DevOps" research annual report shared the relationship between organizational performance, IT performance, and DevOps practices.

DevOps research has consistently shown that high-performing organizations are building the intelligence needed to know their customers better, experiment with ideas that add value, and preserve the ideas that add customer value in a faster and more efficient way. They are also able to embed these capabilities across teams and business units.

Scaling from the Lens of Lean

Understanding the challenges an organization faces and best practices of high performing organizations in addressing them helps define the capabilities an enterprise should consider during their DevOps journey. Below are two models used in Lean manufacturing and currently in Lean IT.

Continuous Improvement Vs. Innovation

From process and operational improvement, Lean practices talk about Kaizen, meaning Change for Better, and Kaikaku, meaning Reform, Innovate. These two models have evolved to address the challenges organization faces and approaches to improve the performance. In Kaizen, most of the impro

vement comes from the team, with small changes gradually over time. This model reaches its limits where you need a Radical breakthrough, Kaikaku, that disrupts the whole model process and practices. In Toyota Production System, Kaikaku is not used. Kaizen is seen for change for better, which could be big or small.
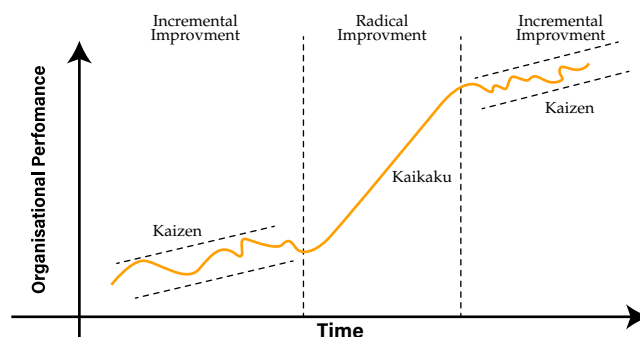
Applying Theory of Constraint

The Theory of Constraint introduced in 1986 by Eliyahu M. Goldratt explains the importance of identifying the most important limiting factor (i.e., constraint) that stands in the way of achieving a goal, and then systematically improving that constraint until it is no longer the limiting factor.

Agile was introduced when organizations saw that the waterfall methodology was a constraint on increasing the time to market while maintaining high quality. Over the years, as the speed of development has improved, the stability of the system became the constraint. DevOps was introduced to build a collaborative model to further improve speed and stabilize the system. Cloud Platforms came as a breakthrough to optimize the constraints in infrastructure management. Now, the constraint is spread across the organization. Unless we consider the value stream of the entire organization, we cannot optimize end-to-end value flow.

Scaling DevOps in an Enterprise

Gene Kim, in his article State of DevOps 2020 and beyond , highlights larger organizations need ability to pivot and adapt quickly when changes hit them hard. Organizations need to have sense of binary thought process to manage changes that are innovative and changes that are improvements to the existing product or services. The DevOps capabilities required for these thought process differ.

Explore Vs. Exploit Domain

Enterprises need to consider the lifecycle of businesses and how organizations balance the exploration of new models for innovation and the exploitation of existing products or services (Jez Humble, 2015). Exploring new opportunities and exploiting existing ones are fundamentally different strategies requiring different structures, competencies, processes, and mindsets. The capabilities that are effective in the exploit domain leads to failure if applied to exploring new opportunities - and vice versa. The differences between these two domains are listed in the below table:

	Explore	Exploit
Strategy	Radical or disruptive innovation, new business model innovation	Incremental innovation, optimizing the existing business model
Structure	Small cross-functional multiskilled team	Multiple teams aligned to the overall mission
Culture	High tolerance for experimentation, risk-taking, acceptance of failure, focus on learning	Incremental improvement and optimization, focus on quality and customer satisfaction
Risk management	The most significant risk is the failure to achieve product/market fit	A more complex set of trade-offs specific to each product/service
Goals	Creating new markets, discovering new opportunities within existing markets	Maximizing yield from the captured market, outperforming competitors
Measure of progress	Achieving product/market fit	Outperforming forecasts, achieving planned milestones and targets

Identifying Capabilities for Explore and Exploit Domains

When identifying capabilities to improve the DevOps practices, most of the organizations uses a static DevOps Maturity Model. This model is used to evaluate all of its business units or teams assuming that "Level 1" and "Level 2" look the same across all teams and organizations. Maturity models are quite often a "lock-step" or linear formula, prescribing a similar set of technologies, tooling, or capabilities for every set of teams and organizations to progress through. In contrast, capability models are multidimensional and dynamic, allowing different parts of the organization to take a customized approach to improvement, and focus on capabilities that will give them the most benefit based on their current context and their short and long-term goals. For example, DevOps Research now part of Google and DevOn Enterprise Capability Model derives the future capability state based on Outcome to drive Software delivery and operational performance than by the static model.

Building Enterprise DevOps Competencies

It is often challenging to nail down the competencies of a resource from DevOps' lens, not because it is one of the most challenging roles in the current dynamic workforce, but because it is still evolving. Currently, it is hybrid with both technical and non-technical skills in equal measure. Professionals in a DevOps Organization must balance soft, process, functional, and relevant technical skills. When transforming to DevOps, collaboration, cooperation, and acceleration are at their core. We need a T-Shaped profile in cross-functional teams focusing on the continuous flow of value and feedback. T-Shape profiles are essential for the Enterprise compared to core technical or domain skill because they can perform tasks within their specialization and help their teammates out.

Summary

Organizations accept DevOps to accelerate its value delivery and build resilience for sustenance in the high-velocity change. Identifying the capabilities to accelerate, continuously training on these capabilities, applying these capabilities across the Enterprise, and measuring the overall outcomes that impact the organizational performance are critical across the value delivery.

↗

Hari Gurumoorthi
a Principal Consultant at DevOn, is an Azure and DevOps Architect. Since 2000, he has played multiple roles in Software Development as well as in Operations. He brings rich experience in creating and executing technology strategies in the BFSI domain, especially in Risk and compliance. He is a certified DevOps Leader,

DevSecOps Engineer, Continuous Delivery Architect, Site Reliability Engineer, and Azure DevOps Expert.

Hari has extensive experience in Building DevOps and SRE Competencies for distributed teams. He has authored the iSQI Certified DevOps Professional Course. He has coached and consulted companies around the world, including E&Y, Bechtel, Olam, Fujitsu, Arrow Electronics, Daimler. He has earned a Bachelor of Computer Engineering, Advanced Financial Risk Management from IIM (B) and M.B.A from Anna University

AGILE METHODS SUPPORT GREATER DIVERSITY

Why do women hold only 24% of computer science jobs, down from 37% in 1995! (Girls Who Code, 2017)

Agile is the most inclusive of the modern tech subgroups. It welcomes people from various roles and specialties to work in close collaboration, extending well beyond the realm of software development.

Although women remain under represented within the tech sector, we can still observe a slight preference for testing. Regarding this, I think 'most women like testing as it's a safe and comfortable area in the IT space', unlike other areas, a tester has to be organized, meticulous, methodical, and pay attention to the details. These are skills very typical in women, hence why they could be more drawn to testing. Once women start in the tech industry, 'they are ready to embrace various roles and opportunities within IT be it designing, Development, Project Managing or Testing.'

It is neither new nor shocking that software development fields underrepresent women. When examining barriers to eliminate the gender gap, it is vital to identify why the lack of diversity is a problem and the reasons it exists. The interesting discussion is not really whether we should have more women on our teams, but rather why don't we have more of them already.

On agile teams, collaboration is paramount. It is included in all the contexts agile touches: team organization, communication, working with clients, etc. The bottom line is that if we consider how effectively teams work and not focus on artificial individual context, we'll end up with a very different assessment of what constitutes the best candidate. Suddenly a lack of great female candidates would not be a problem anymore.

I personally believe that gender should never be a barrier for any career choice. QA as a career requires someone to have an ability to interact with clients to understand their requirements, to empathize, to be methodical in prioritizing those requirements, being responsive, multitask and finally deliver

thoroughness as part of the delivery. Anyone who has these skills and can add some technical understanding on top of these qualities will just excel in QA as their career choice.

Agile Testing is a wonderful area that women are able to master so well and dominate. However, I'd say find something that will always take you off your comfort zone and challenge you, like Test Automation, that way you stay relevant and don't lose the coding skills you have learned. Although women are as capable and hard-working as men, there is still a huge inequality that persists over the years. It is true that many industries, including tech, have hired more women these past few years but the gender gap remains very present…

First of all, there is the fact that girls are usually less encouraged to study and pursue a career in STEM (Science, technology, engineering, and mathematics) than boys. This is quite a recurring issue as, for instance, when you step into an engineering or computer science course, you will often see more men than women. Talking about empowering women, having well-known female leaders is extremely important, as they can encourage young girls and women to follow their lead! Without them, the perception that technology isn't a career for women is reinforced. Shifting the development and quality assurance landscape to be more inclusive will take a ground-up approach, but there is still much that co panies can do to get with the times and embrace diversity.

Agile supports a greater diversity

Agile testing methodologies have been lauded by the most forward-thinking programmers for their ability to reshape QA teams into lean and nimble business units. Agile methods need Agile teams - teams that think differently and work in ways that support responsive delivery. An agile mindset, and a set of shared values, principles and often Agile tools, help Agile teams succeed. Tester. As much Agile work is still carried out in the IT domain, software testing is still a big part of Agile teams. Even in non-software teams, you may want someone who can act as a tester. Being a woman in testing, it's important to lift other women up with you.

↗

Adriana Benitez
*is a Computer Systems Engineer with a
Master in Mobile Communications
at Universidad Politécnica de Catalunya,
has several certifications such as: Scrum
Master and CTFL-ISTQB. She has
almost 8 years of experience in testing
and has worked with international teams
as a tester, test lead and test manager.
Currently, works in a company that
develops software for the banking sector
where testing plays a key role.*

"

BEING A
WOMAN IN
TESTING, IT'S
IMPORTANT
TO LIFT OTHER
WOMEN UP
WITH YOU.

MOVING WITH AGILITY, A GLIMPSE IN TIME

According to the dictionary, a methodology is a set of methods, principles and rules that govern adiscipline in both the arts and the sciences. If we understand systems development as a discipline that has features of both, we will understand the need for methodologies that lead us to the efficient development and quality of computer systems.

1960

- in the 1960's, software was developed through trial and error, first in machine language and then in assembler, with little or no documentation and without a formal method,resulting in low quality and an inaccurate estimate of time and costs.

1970

- in the early 70s, the arrival of structured programming allowed standardized coding. This with improved documentation, facilitated maintenance and quality control. It was also at this time that software engineering emerged as the area of engineering that defines methods and techniques to develop and maintain quality software. Waterfall development, also known as sequential development, emerged as a methodology; stages of development were systematically ordered and a new stage could not begin until the previous stage had been completed - there were no overlapping stages of development.

- In the mid-70s, iterative and incremental methodologies were identified as adding value to the the waterfall model by allowing changes in various aspects of the product during its development stage.

- Shortly after, the V-model emerged as a project management method that describes processes and stages for both management and system development. On one hand, the system specifications were defined (requirements and functional, technical and component design) and on the other hand, the system verification carried out (unit, component, system and acceptance tests).

1980

- At the beginning of the 1980's, database management systems emerged, facilitating the handling of large amounts of information. With the emergence of the relational database paradigm, the iterative development model developed. This model creates a group of repetitive stages that consist of iterations starting with the analysis and ending with the implementation of the system.

- In the mid-80's the prototype model appeared; this was an evolutionary development model in which prototypes representing the basic end user requirements are built in a short time frame using fewer resources. Prototypes are presented to a client for feedback. Developers can then proceed to make improvements and repeat the same cycle until the development process is completed.

- Almost at the same time the spiral model arose; in each iteration, there were a group ofactivities that are not set by stakeholder priorities but based on a risk analysis.

- The Object Oriented Programming paradigm appears; existing programming languages adapt and with this new languages emerged. The client server paradigm also emerged, which helped to optimize communication between computers and launched the internet.

- The Object Oriented methodology emerged according to which systems are conceived as objects that interact with each other to achieve a common goal.

1990

- In the mid-90s Scrum came to light as an agile methodology with incremental development. The development of systems were planned as blocks or sprints of incremental processes with functional deliveries, emphasizing the empowerment of people to collaborate and make team decisions. It had the advantage of being adaptable and reducing the large amount of documentation that were common in more traditional methodologies such as waterfall.

2000

- The Rational Unified Process (RUP) was created as a method that allowed adaptation to the context and needs of each organization. This methodology seeks to balance the requirements of all participants using multiple teams working iteratively taking care of quality in each cycle.

Other important developments came to the fore:

- Extreme Programming, an agile methodology based on communication, reuse and feedback recommended for short-term projects where periodic tests are carried out.

- Kanban, a highly effective and efficient production system, originated in the "just in-time" (JIT) production processes devised by Toyota, in which cards were used toidentify material needs in the production chain. Currently, the term Kanban has become part of the so-called agile methodologies, whose objective is to manage in a general way how tasks are completed.

- Test Driven Development (TDD), characterized by reviews, testing, and constant design and redesign that conform to a development cycle that must be followed for successful implementation.

As we can see, the methodologies have evolved from a very rigid, waterfall approach through to the agile methodologies that are recognized as having significant advantages to most organizations and projects and which are now widely adopted.

Among other advantages, agile methodologies create greater collaboration between the different areas thus favoring cross team communication and information sharing. In this way the teams work together more effectively, breaking down silos and , all with the same objective which is to deliver a quality product to the end user. Let's compare this to traditional development approaches where each team carries out its work in different stages in a sequential manner, with disparate objectives and communication is scarce beyond documentation.

Another significant advantage is that it allows better scoping of projects and this helps make changes and adaptations on the fly in terms of processes, scope or even capacity in the teams. This is much more efficient than waiting for later phases to realize these needs, which would entail substantial changes in development costs and time. The agile methodology requires careful planning and strategy since without this, projects fail. It is a mistake to think that agile means no documentation , no plan or organization is required.

↗
Luis Francisco Contreras González
Senior Software Test Engineer and consultant with more than 6 years of experience working in the field of Information Systems and clients support for different companies in their technology departments with strong analytical and problem-solving skills. 4X ISTQB/ ISQI certified and Scrum certified.

Martin Contreras Romo
Electronics Engineer specialized in programming with more than 30 years of experience in systems design and development for private companies and government agencies, professor of programming and robotics, and Scrum certified.

Whether you are transitioning from a traditional to agile methodology or if agile is being implemented as the main work method on a new project there can be challenges in determining the time, personnel, costs and resources necessary to have a successful project. It is very common that unproductive combinations of waterfall and agile are attempted that cause problems in the project.

The Agile Software Development Manifesto (https://agile-manifesto.org/) specifies::
• Individuals and interactions over processes and tools
• Working software over comprehensive documentation
• Customer collaboration over contract negotiation
• Responding to change over following a plan

Following these points and based on the particular needs of each project and organization, an optimal way to work with agile methods can be found. One of the advantages is the different number of agile methodologies that can be used, such as: eXtreme Programming XP, Kanban , Test Driven Development (TDD), Scrum (which is one of the most popular and currently used), among others. Always keeping in mind that what works in one place will not necessarily work in another. Each project and organization has its own requirements that make them unique and require a
different way of adapting their processes. While making sure the guidelines and best practices are correctly followed.

Agile testing is a smarter way to work with an approach where testing is a continuous part of the development life cycle. Rather than leaving testing as a final phase and waiting for all the system developments to be completed, testing is now involved from the start, not only debugging. Testing would now function as a way to prevent bugs and improve the quality of the product.As previously mentioned, it is important to think of the different areas as part of one team, so the testing team works hand in hand with the developers, analysts and other stakeholders involved in the project, where everyone is responsible for quality. Agile testing has many advantages such as saving time and costs, since errors are caught early on and we can make modifications and corrections on the fly. Thanks to this flexibility and adaptability we can reduce documentation. Of course, this would not mean we can exclude all documentation, but we can focus on the most important information.

There are many other agile testing methods, each with its individual rules and specifications, such as Behavior Driven Development (BDD), Exploratory Testing, Acceptance Test Driven Development (ATDD), Test Driven Development (TDD) among others. Working with agile requires good processes and correct implementation to lead to a successful project. In agile testing, similar to other traditional methodologies we also need to have a test plan. The agile test plan is written and updated for each release and generally, in my experience it should include at least the following:

1. Testing Scope and objectives

2. Functionalities

3. Types of testing

4. Priorities

5. Consideration for infrastructure

6. Resourcing (I.e. time, tasks)

7. Risk mitigation plan

8. Milestones and SLAs

9. Along with other specifications, which are related to a specific project

PREVIEW

 # 08
OCT 2020

SQ_mag
THE VOICE OF SOFTWARE QUALITY
ARTIFICIAL INTELLIGENCE

HOW TO TEST A SYSTEM WHEN WE DON'T KNOW THE OUTPUT WE NEED?
by Adam Leon Smith

MOVING TO TEST PREDICTIONS
by Tom van de Ven & Rik Marselis

EVERYONE HAS A PROBLEM, AND EVERYONE KNOWS THE SOLUTION – IT'S AI
by Khimanad Upreti

BETTER JOBS AND HIGHER SALARIES
Interview with Valentina Taseva

PERFORMANCE IN YOUR BUSINESS FOR SUCCESS
by Leandro Melendez

What´s in the next issue of SQ mag?

Creating great experiences in everything you do Guest feature by American Graphics Institute and user experience designer Jennifer Smith.

Please share your experiences with us and send your article to **contact@sq-mag.com**

ADVERTISEMENTS

Detailed metrics for the online advertising formats are available. Please send your request to: *contact@sq-mag.com*

METRICS

21 cm × 29,7 cm
(DIN A4, 4-color (euro scale))
Live area: 170 mm (B) × 242 mm (H)
E-Paper and PDF-Download on
www.sq-mag.com

SQ_mag
#10
COMING OCT 2021

A4Q
Selenium Tester
Foundation

Get Selenium Certified - Now

Perfect for test professionals who desire an understanding of how to design, implement and maintain automated solutions using Selenium WebDriver.

Internationally recognized certification!

iSQI®
SUCCESS IS YOURS

A4Q Alliance for Qualification

isqi.org/de/certified-selenium-tester-foundation

©bejo_shutterstock