

# SOmag

# 06  
MAY 2019

THE VOICE OF SOFTWARE QUALITY

## **AUTOMATIZATION IN A LABORATORY ENVIRONMENT**

*Christopher P. Rüger  
on developments  
in research*

ARTICLE

## **ABOUT CLOUDS AND SERVERLESS TECHNOLOGIES**

*Abby Kearns on function  
as a service*

INTERVIEW

# **THE WORLD OF SOFTWARE QUALITY**

## **DESIGN THINKING IS A "TEAM SPORT."**

*Mark Tannian  
on what it means to be  
a Design Thinker*

INTERVIEW

## **THE REGRESSION MANAGEMENT QUADRANTS**

*Pieter Withaar and  
Johan van Berkel  
on code and test quality*

ARTICLE

## **TESTING APPLICATIONS FOR CHILDREN**

*Nadia Soledad on the  
difference of testing apps  
for adults or children*

ARTICLE



# ***PRACTITIONER IN AGILE QUALITY.***

***GET AGILE. ACCELERATE QUALITY.***



[www.isqi.org/lp/practitioner-agile-quality](http://www.isqi.org/lp/practitioner-agile-quality)



**Stephan Goericke**

CEO, International Software Quality Institute

## LET'S GET AHEAD TOGETHER!

So much has happened since our last update. The International Software Quality Institute (iSQI) is proud to present a very special new product: the Practitioner in Agile Quality (PAQ). PAQ is designed for anyone who wants to learn how to accelerate quality without compromising the speed of delivery.

For this certification, participants complete a training course where they master agile working in an experiential and collaborative learning environment with an emphasis on practical application. PAQ helps participants to experiment with new practices and techniques taking their expertise to a practitioner level; this expertise is validated by a practical assessment against the PAQ competency framework.

As you can see, iSQI is committed to developing its portfolio driven by the changing demands of organizations. Another great new certification is A4Q Design Thinking. A critical element to digital transformation is creating an excellent experience for the customer – Design Thinking supports this through customer empathy and a collaborative, innovation-seeking approach to problem-solving, ideation and prototyping. For sure, this is important in software development but it is also essential in other fields.

We are committed to developing new, relevant and exciting certifications during the next few months and we will keep you in the loop.

While waiting for more exciting news, I recommend reading our new SQ mag. We have a lot of interesting articles and interviews exploring and analysing developments in software quality.

Enjoy the read!

Yours sincerely,

# CONTENT

## **AUTOMATIZATION IN A LABORATORY ENVIRONMENT**

*Christopher P. Rüger*

4

## **ABOUT CLOUDS AND SERVERLESS TECHNOLOGIES**

*Interview with  
Abby Kearns*

10

## **THE REGRESSION MANAGEMENT QUADRANTS**

*Pieter Withaar  
Johan van Berkel*

14

## **DESIGN THINKING IS A "TEAM SPORT."**

*Interview with  
Mark Tannian*

20

## **TESTING APPLICATIONS FOR CHILDREN**

*Nadia Soledad Cavalleri*

26






# **AUTO- MATIZATION**

**IN A LABORATORY ENVIRONMENT**

*FROM HARDWARE SOLUTIONS  
TO SOFTWARE TESTING*





“ THE TURNOVER TIME OF SOFTWARE SOLUTIONS HAS DECREASED RAPIDLY, MAKING SOFTWARE TESTING A KEY FOR SUCCESSFUL VENDORS

A

Analytical chemistry, in terms of the qualitative and quantitative description of sample mixtures, has a key role in current technologies. Around the world, companies and research institutions from various fields, such as biotechnology and the pharmaceutical industry, rely heavily on rapid and highly qualitative analytical results in their production and development chains. As an omnipresent process in our daily lives, digitalization has not stopped for these fields; indeed, the opposite is true: research laboratories are a significant booster for this transformation. Novel concepts for the automation of the actual measurement – in terms of hardware components but also the automation of data processing and report generation – have changed the laboratory environment drastically over the last decade. In this respect, the turnover time of software solutions has decreased rapidly, making software testing a key for successful vendors. The largest firm for scientific equipment, Thermo Fisher Scientific Inc., made a multi-billion revenue in 2018. Hundreds of millions of Euros were spent on research and development, with an increasing proportion being diverted to the field of software development.

More than 60 years ago (decades before modern computer technology, the internet, and digitalization), the prestigious Journal of Analytical Chemistry reported on automation in analytical chemistry as “one of the most spectacular current achievements in the field” [1]. At the time, they were discussing developments in the hardware components, which simplified and accelerated work schemes. However, this “current achievement” is, from another perspective, still an actual topic. With the emerging progress in robotics,

software, and inter/intranet-of-things (IoT), we are facing a radical change in the way how laboratories work and deliver their results. Some crucial aspects of hardware communication in the internet-era, as well as data reporting, are shortly discussed here.

### A modern analytical laboratory

In a modern analytical laboratory with state-of-the-art instrumentation, measurements are most often already carried out by various types of so-called autosamplers. Those devices ensure an automated introduction of the sample into the analytical equipment, e.g., spectrometer or chromatograph. Additionally, for the sample preparation, a variety of automated solutions exists. As we have seen above, this concept is not new. The novel aspects are the intelligent connection and communication of those devices. Complete workflows can be accelerated, and human errors/workload can be minimized, starting from recording the sample and providing a bar code ID, deploying the correct sample preparation based on this ID, utilizing the right equipment settings for this sample type, and generating fully automated reports. This workflow is, in particular, useful for high-throughput laboratories, such as in a clinical environment or for food safety testing.

Laboratories are still adapting and implementing these new concepts for their specific needs and requirements. A new direction one step further for this workflow is the intelligent communication of laboratory equipment. It is anticipated that such devices and sensors communicate (i.e., sharing readbacks or events) to the employees, but also between each other.

#### Sources

[1] - Automation in Analytical Chemistry, Analytical Chemistry 1958, 30, 12, 25A-34A, DOI: 10.1021/ac60144a717



## AUTOMATIZATION IN A LABORATORY ENVIRONMENT

## “STANDARDIZATION OF REPORTING WITH A SUFFICIENT AMOUNT OF INFORMATION IS NEEDED



➤

### **Christopher P. Rüger**

*Dr. rer. nat. Christopher P. Rüger studied Chemistry at the University of Rostock, Germany. In 2018, he completed his doctorate with distinction. During his studies, he published thirteen peer-reviewed articles and over twenty oral presentations at national and international scientific conferences. At the moment, he is a PostDoc at the University of Rouen, France, working in the field of data processing and method development for the mass spectrometric analysis of ultra complex mixtures.*

For example, the oven or shaker will save a report at the local server if finished and send a notification to the laboratory manager. This event triggers the spectrometer, scheduled for those samples, to be put in the right conditions for the samples, and samples are loaded by the autosampler reading the sample metadata from the bar code ID. After analysis, the report is created and forwarded to the personnel as well as to the cloud-based server platform. Apparently, this topic is strongly related to the discussion about Industry 4.0 and leads to new questions about safety (i.e., can a sensor or device be manipulated or unauthorized?) and data security/privacy. Hence, along the whole workflow, stable, safe, and reliable software solutions are needed.

### **Challenges and Requirements**

In this context, the challenges and requirements for successful software distribution are mostly the same as in other fields, which is also reflected in the utilization of similar software testing utilities. Nonetheless, the reporting of analytical results bears some specialties. Results, such as the quantity of a specific drug or pharmaceutical compound in an environmental or process stream sample, are crucial for the further handling of the material or legal reporting. Hence, based on those obstacles, delivering the right results is even more crucial and requires multi-level software testing and development. The potential malfunction of the instrumentation needs to be considered, and meta-data needs to be written entirely hard-linked to the data set. In this regard, data integrity is currently a big issue. Software solutions should ensure that the user is not able to modify the raw data and all changes to the report need to be tracked, saved, and visualized.

These approaches are not only valuable for routine laboratories or core facilities (analysis

as service for research institutions or companies) but also for general research laboratories. Most research laboratories are faced with a drastic increase in the number of investigated samples. Thus, to cope with the requirements of large sample cohorts, a substantial degree of automatization has to be put in place. As we are also facing an era in which an increasing proportion of published results cannot be recreated from other laboratories [2], standardization of reporting with a sufficient amount of information is needed. Adapted software solutions, such as web-based platforms allowing to upload the raw data linked to the complete set of meta-data for general use, or the development of uniform data formats, are needed here. On first glance, this might be inconvenient and related to a higher workload for the researchers. Nonetheless, it will surely accelerate multidisciplinary research in individual teams and also between the institutions, allowing to tackle more complex research questions than ever before.

### **The transition from hand-written to automatized documentation**

We have seen a great transition from the traditional laboratory work, mostly reporting in hand-written “lab-books”, to an automatization of the hardware and even more relevant an automatization in how results are delivered, with solutions, such as e-mail reporting and error-notification, tracking data integrity and history, as well as cloud-based data storage concepts. This transition is far from complete and still in full progress. Software solutions will undoubtedly be the key for institutions and companies to stay competitive in this area. Thus, software testing will become one of the most important aspects for many companies involved in distributing laboratory equipment, ultimately leading to a need for qualified personnel capable of dealing with these new challenges. ■

[2] – Challenges in irreproducible research, Nature special (ISSN: 0028-0836), <https://www.nature.com/collections/wjsrmdnsm> (accessed 26.03.2019)



# PAQ – PRACTITIONER IN AGILE QUALITY

The Practitioner in Agile Quality (PAQ) is an accelerated learning agile certification, designed to support people with current agile knowledge to take their skills to a practitioner level in an intensive learning environment.

PAQ immerses participants in agile practices with an emphasis on the key competencies essential for high performing agile teams working together collaboratively to increase value to the business and the customer. A Practitioner in Agile Quality can use agile as an effective pathway to accelerate the delivery of quality software.

## → Benefits of PAQ certification

### For individuals

- Hands-on, collaborative learning
- Accelerated path to practitioner
- Prove your agile quality competency
- Develop your agile mindset (soft skills)
- Stand out from the crowd with a certification that shows you can do!

### For organizations

- Accelerate quality in the organization with Agile Quality Practitioners
- Develop agile competency within your organization to leverage more benefits from agile
- Demonstrate agile competence to internal and external customers
- Get ahead of the competition with qualified PAQs who can do!

### For the Team

- Bring the agile team together with a focus on quality
- Develop team competence, collaboration and confidence
- Motivate the team with a career path to practitioner
- Challenge the team to validate they can do with a practical certification!

### For Training Providers

- Boost your training portfolio with a highly practical course in agile quality
- A course for anyone involved in testing but also the whole team
- Practical assessment exam – a market differentiator
- Soft skills focus plus strong agile quality content
- Licensed materials available
- Support your customers to develop their organization's agile competency

## → Who should take PAQ certification?

Agile quality is the collective responsibility of the whole team. PAQ certification is not just for testers but anyone, and everyone, in the team who needs to continuously deliver better products, services and user experiences using agile.

The course builds on existing testing, software quality management and agile theory inspiring and building confidence through hands-on exercises and experiential learning aligned to the PAQ Competency Framework. The final practical assessment ensures that PAQ holders have demonstrated that they can achieve the expected PAQ competency outcomes.



For more information visit:

<https://isqi.org/lp/practitioner-agile-quality/>

## iSQI RESPOND TO GLOBAL DEMAND FOR DESIGN THINKING CERTIFICATION!

The Alliance for Qualification (A4Q) has released a global certification in Design Thinking. The certification is available at Foundation Level and validates the holder's knowledge and ability to use this innovation-seeking approach to problem solving and solutions development. The leading global exam provider iSQI included the Design Thinking Foundation Level certification in its respective portfolio..

<https://blog.isqi.org/en/2018/10/30/isqi-respond-to-global-demand-for-design-thinking-certification/>



*With over 30,000 certifications per year, iSQI paves the way for successful career development. Join an exclusive insight into iSQI's world.*

# WHAT'S

## EXPANSION IN EASTERN EUROPE

The neighboring countries see Germany as an economically interesting market. For programmers in Poland, IT specialists in Ukraine, developers in Romania, Belarus etc., Germany offers great business potential. Especially in the IT market, the interest is reciprocal. German companies are increasingly looking for skilled specialists in the eastern European market.

<https://blog.isqi.org/en/2018/12/12/expansion-in-eastern-europe/>





---

## THERE ARE 3 NEW ISTQB® CERTIFICATES IN THE iSQI PORTFOLIO



The ISTQB® Certified Tester Foundation Level – Automotive Software Tester can demonstrate knowledge of the most important test procedures and standards used as part of testing E/E systems in an automotive environment.

<https://blog.isqi.org/en/2019/01/30/new-istqb-ctfl-automotive-tester-exam/>

The ISTQB® Certified Tester Foundation Level – Usability Testing offers candidates an opportunity to prove and improve their usability testing skills. It will support anyone responsible for usability testing to test the ease with which the user interfaces can be used and whether the application, or the product, is user-centric or not.

<https://blog.isqi.org/en/2019/02/04/new-istqb-ctfl-usability-tester-exam/>

The ISTQB® Certified Tester Foundation Level – Performance Testing plays a critical role in establishing acceptable quality levels for the end user and is often closely integrated with other disciplines such as usability engineering and performance engineering.

<https://blog.isqi.org/en/2019/02/22/performance-matters/>

# HAPPENING



---

## iSQI CERTDAYS 2019

Very recently iSQI hosted the 2019 Certification Days (CertDays) which is starting to become a tradition. This year we had a five-day line-up with two webinars every day including topics such as Test Automation, Usability and Design Thinking. The full program appealed to the professional audience as the number of people tuned into every session was overwhelming.

If you want to see a trailer of the webinar first, just use this link to find the trailers and decide which one you would like to watch in full:

[https://www.youtube.com/results?search\\_query=isqi+certdays+2019](https://www.youtube.com/results?search_query=isqi+certdays+2019)

<https://blog.isqi.org/en/2019/04/16/isqi-certdays-2019-04/>



# ABBY KEARNS ABOUT CLOUDS AND SERVERLESS TECHNOLOGIES

IaaS, PaaS, BaaS, SaaS – the evolution of cloud history is going on. The newest step is FaaS – Function as a Service. That technology does not include any server and enables the developers to concentrate on the application they develop instead of taking care of the infrastructure and needed disk space. Cloud Foundry offers interoperable cloud technologies, some of which could be described as “serverless” as they do not require developers to build infrastructure, so they can instead concentrate on app development using the PaaS Cloud Foundry Application Runtime. The SQ mag spoke with Executive Director Abby Kearns about it:





# “CREATE APPS, NOT THE PLATFORM.”

## A

**Abby, on your website, we find “Create Apps, Not the Platform.”  
What does this mean?**

Developers have enough to worry about without the unnecessary challenges of running an environment in-house. Building infrastructure to support and maintain enterprise applications is a complicated and expensive endeavor. Our mission at Cloud Foundry is to empower developers to focus on building awesome applications with an automated, open source platform that relies on the latest in cloud native technologies. By embracing interoperability and supporting a platform that equips developers with a framework for constant iteration, Cloud Foundry provides a reliable and secure platform for innovation at scale. In short, Cloud Foundry allows developers to focus on their applications.

**Cloud has become increasingly popular in the last few years.  
Why did developers start to program in the cloud and leave servers behind?**

Cloud native technologies are a crucial component of modern infrastructure and a necessary step in digital transformation. The speed, agility and reliability of the cloud is well proven and evident in essentially every major technology product or service in use today. On prem infrastructure, and company managed data centers, are increasingly being left behind so that companies can minimize the amount of infrastructure overhead they need to manage, and want to develop and run more apps at scale. Enterprises are moving more and more workloads to the cloud as part of their shift to fundamentally change how they run their business.

**Now, a new technology called “serverless” wants to replace technologies like PaaS. What is the difference between these two?**

Serverless infrastructure is not a replacement for PaaS, but rather a promising advancement in cloud computing that is quickly becoming an important part of cloud native solutions. Speed is a key characteristic of serverless computing as it empowers enterprises to initiate and deliver automated tasks within milliseconds. Servers are still involved at some level, but it’s run in stateless containers that are triggered by events and managed by a separate entity or platform.





## ABBY KEARNS ABOUT CLOUDS AND SERVERLESS TECHNOLOGIES

### What are the advantages of serverless technologies?

Serverless is quickly becoming a viable solution for developers that want to create and iterate on smaller, focused apps. Serverless apps can be easy to create, run, and then shut down when the job is over. From a business perspective, the most significant benefits of serverless computing is dynamic pricing and allocation of resources based on real-time demands. In short, serverless can provide a great deal of flexibility for some use cases. There is a lot of innovation currently happening in the serverless area, so I expect to see a lot more this year in this space.

### Are there any risks to using serverless? How can you minimize them?

Every technology has risks and benefits. You should use serverless for those workloads where it makes sense. It is not going to be an ideal solution for every workload. Serverless is a powerful approach for a great many cloud native workloads, but it can create a significant reliance on a single cloud provider. There will be more options in the near future, many of which will be based on open source technologies which gives users optionality on which technologies to choose for serverless workloads. Over the next 18 to 24 months, I expect to see significant alignment in the cloud native industry around serverless standards.

### How many companies are already working in that way?

As with any new technology, the integration cycle can be spiky, with organizations rushing to evaluate the tech only to pull back if they believe they are not ready. Serverless is no different. After a sharp increase in evaluation of serverless in our research from September 2018, which showed an increase to 42 percent of respondents evaluating serverless from only 25 percent evaluating in November 2017, our research from April 2019 shows a little pull back to 36 percent today. However, while those trying out serverless might have retreated a bit, for those who are using and evaluating the technology, they are going deeper. We see broad use of serverless up to 18 percent, doubling year over year since April 2018, and early use continuing to decline to only 38 percent.

### What does a change to cloud-based working/development mean for a company?

For many organizations, shifting to a cloud-based solution is a great forcing function. Developing and running cloud native applications will push an organization to change the way that it works. You will find yourself running CI/CD pipelines and pulling together cross-functional (agile) teams, you are now looking to iterate on apps quickly, and get customer feedback. All of these changes are critical to working in a cloud native way, and are part of the digital transformation journey.



### In what ways do cloud and serverless technologies support the quality of applications?

Cloud native technologies exist to help organizations develop and run cloud native apps, at scale. These technologies work to automate as much as possible. Running cloud native apps mean that you are building, and iterating, on apps that are resilient, scalable, and can be quickly created and updated. If you are developing and iterating on new apps in production every few weeks, then by that very process your apps will improve in quality. You are constantly iterating on apps, based on feedback from your teams and customers.

### Regarding qualification: How hard is it for companies in Europe compared to other regions in the world to find skilled employees in software development?

Skilled developers are in short supply and high demand around the world. When we conducted research on this topic almost a year ago, 51 percent of IT decision makers said there is now or will soon be a shortage of skilled developers. This concern is particularly acute with fast growing technologies like artificial intelligence (AI), machine learning (ML), as well as any cloud native technology. You also have the demands of IT development growing at a rapid pace. More than three quarters of IT decision makers we surveyed reported that they need more skills now than they did five years ago. In short, the need for skilled technologists is increasing, as the skills necessary are evolving. And this is happening globally.

### What role does certification in software skills play?

Certification can be a great tool to acquire the skills needed to keep up and prepare for the future of cloud-based development. Certification can offer a level-set of the skills in a given area (in this case cloud and cloud native), as well as showcase that even though you may not have five years of experience in this space (which, who does?) but that you know enough of the fundamentals to be certified, and you can build on those skills in the job.

Training programs can help organizations unleash opportunities to innovate and grow their business. Enterprises of all sizes and types should be in constant pursuit of reskilling and upskilling the workforce. Giving employees the space and resources to learn new things is also an incredibly cost effective way to increase an organization's competition for talent.

Thank you, Abby. ■



#### **Abby Kearns**

*With nearly twenty years in the tech world, Abby Kearns is a true veteran of the industry. Her lengthy career has spanned product marketing, product management and consulting across Fortune 500 companies and startups alike. As Executive Director of Cloud Foundry Foundation, Abby helps the ecosystem of developers, users and applications running on Cloud Foundry, and works closely with the Board to drive the Foundation's vision and grow the open source project. Prior to Cloud Foundry Foundation, Abby focused on Pivotal Cloud Foundry as part of the Product Management team at Pivotal. She spent eight years at Verizon where she led Product Management and Product Marketing teams dedicated to the early days of cloud services.*

“THE NEED FOR SKILLED TECHNOLOGISTS IS INCREASING, AS THE SKILLS NECESSARY ARE EVOLVING.

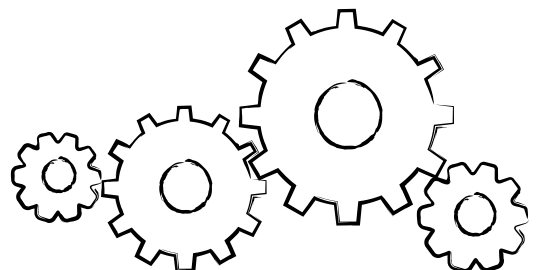
# THE REGRESSION MANAGEMENT QUADRANTS

# C

*Capable of detecting regression at low execution costs as well as saving engineers from death by boredom, automated checks are often considered to be the silver bullet to regression. But when applied in the real world, the promise seems to be more of a fairy tale: after all the investments are done to automate the regression tests, the amount of regression doesn't seem to decrease. And even though the execution is cheap, maintenance sure isn't.*

“

ANY CHANGES MADE TO THE CODEBASE FROM THE VANTAGE POINT OF ONE USER NEED CAN YIELD UNEXPECTED CHANGES IN ANY NUMBER OF OTHER USER NEEDS.





“KEEPING THE IMPACT OF REGRESSION  
IN CHECK IS A NEVER-ENDING STORY  
THAT REQUIRES CONSTANT EFFORT  
TO MAINTAIN.

### The burden of regression

When a product is changed for whatever purpose, developers will primarily focus on changing the codebase to support the new user need. Assuming that the user need is implemented in an existing product, this means building new code as well as refactoring existing code to integrate the user need in the product. The resulting cycle of development, test, and fix will ensure that –eventually – the new user need is met.

Unfortunately, user needs are never implemented in perfect isolation. Instead, the code is often reused to fulfill multiple user needs. As a result, any changes made to the codebase from the vantage point of one user need can yield unexpected changes in any number of other user needs. Whenever such an unexpected change occurs, we speak of regression. Although regression, technically, doesn't include a specific impact on the business value within its definition, positive regression rarely ever occurs. More likely, the impact is either negligible or detrimental (in the form of increased maintenance costs and/or decreased product performance).

With the risk that regression poses towards the business value of the product, the need to prevent regression from reaching the customer is obvious. Roughly speaking, the level of detrimental regression that reaches the customer can be lowered by:

- Preventing the occurrence of regression during development
- Detecting any (residual) detrimental regression before reaching the customer

Keeping the impact of regression in check is a never-ending story that requires constant effort to maintain. But since it prevents potential issues from occurring, it is virtually impossible to quantify the economic benefits that come from the effort spent on managing regression. As a result, organizations are inclined to spend as little effort on the topic as possible.

Following this line of reasoning, extensive automated regression testing is often seen as the silver bullet to regression. By writing tests that focus on verifying the business value of all user needs, only detrimental regression is caught and fixed. Only code that causes detrimental regression is reworked, and since automated tests are cheap and fast to execute, the cost-benefit ratio of this approach is supreme. Or so it seems... Because take a step back and you will see that this approach only results in endless drudging through the swamps of maintenance hell while chasing a unicorn that doesn't exist.

If that last bit doesn't make a lick of sense at the moment, then you probably haven't been introduced to the Regression Management Quadrants. Luckily, that's exactly what this article is about.

### Introducing: The Regression Management Quadrants

The Regression Management Quadrants take the two methods of managing (detrimental) regression (i.e. prevention and detection) and plot the relationship between them into four quadrants that can help organizations determine on how they should (and shouldn't) manage regression. But before we start talking about the quadrants, let's start by explaining what we feel are the most impactful factors when it comes to the prevention and detection of regression:

#### *Preventing regression by increasing code quality*

The first method to limit detrimental regression is to prevent regression from occurring in the first place. Its success rate is mostly determined by the cognitive effort it costs to correctly assess the impact of any change in code on the whole codebase. Although experience plays a role, code quality is vastly more impactful. When talking about code quality, we mean any characteristic that impacts the effort it takes to translate the code into a relevant and correct mental model. This can be anything ranging from readable naming conventions to cyclomatic complexity, code cohesion, and code coupling.



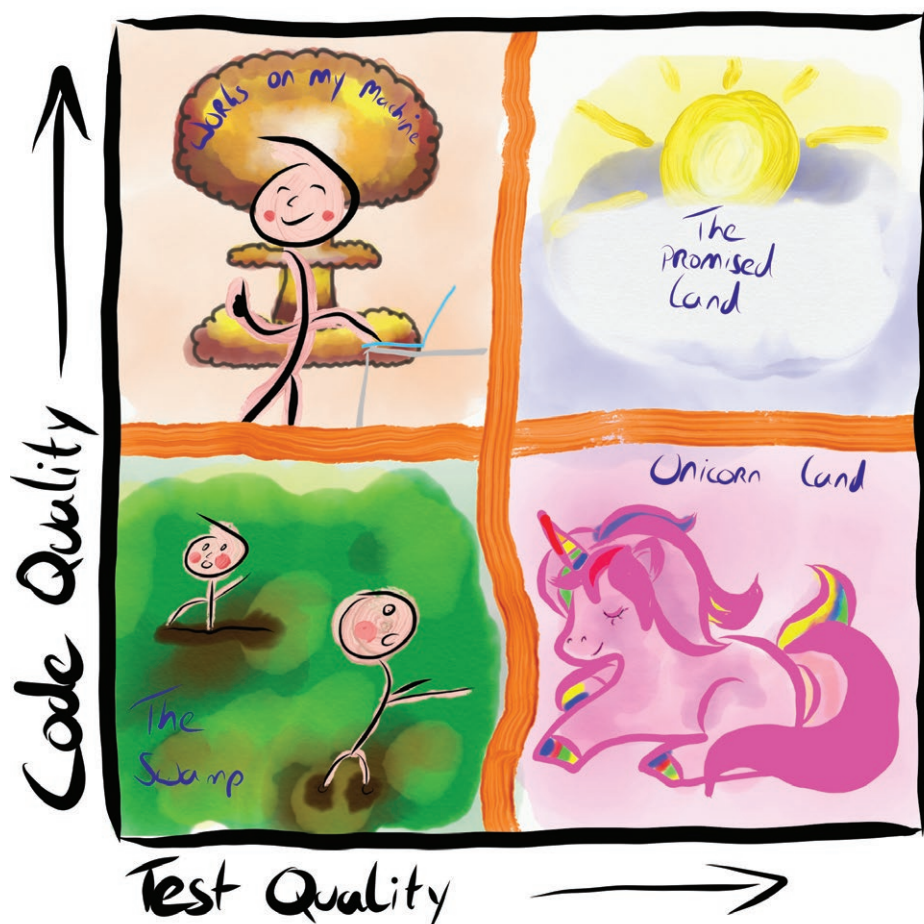
# THE REGRESSION MANAGEMENT QUADRANTS

## Detecting regression by high-quality tests

The second method to limit detrimental regression is to detect regression before it reaches the customer. Detection is done by testing; therefore, the most impactful factor is the quality of testing. When talking about test quality, we mean any characteristic that impacts either the effectiveness or efficiency at which tests can detect detrimental regression. Effectiveness is determined by the amount of detrimental regression still reaching the customer (the less regression goes through, the more effective the tests are). Efficiency is determined by the time and effort it takes to sustain the regression tests, which can be characterized by, for example, test redundancy and test maintainability.

## Characterizations as result

Now that we have defined the most impactful factors for successfully preventing or detecting regression, we can plot both of them, resulting in the four quadrants mentioned earlier:



Each quadrant in the RMQ represents a potential strategy for managing regression and comes with its own reasons for why organizations end up using the strategy, and why – as is the case with any self-respecting Four Quadrant Matrix – the top right quadrant is the only correct answer to managing regression. Let's dive into the quadrants, shall we?



**Low code quality, low test quality: Welcome to the swamps of maintenance hell**

Most product developments are extensions or improvements to an existing product. Whether it is the result of prioritizing new features over maintenance activities in the earlier days of the product life cycle or legacy code inherited from days long gone that no one dares touch, most organizations are stuck with a product that contains (a lot of) technical debt. Since assessing the impact of a change in a product with low code quality is nearly impossible, the product will yield high levels of regression. Initially, this might result in a storm of complaints from customers. This quickly backfires into a reflex response by the organization: development needs to stop regression bleeding through yesterday. Since the fastest way to stop bleeding is to apply a Band-Aid, the effort to detect regression is intensifying, but with low code quality, regression doesn't decrease; it merely shifts to other areas of the product. With each failure found, the automated checks are expanded, gradually growing to unmanageable proportions. Welcome to the swamp of maintenance hell.

**Low code quality, high test quality: unicorns are still mythical**

The most common approach to dealing with maintenance is to prioritize testing, which is the reflex response to issues in the field for any organization. However, sustaining regression testing results in high costs, and eventually, people will repeat the logic of this article's introduction: we need good tests, and since we can't predict where regression occurs, we have to test everything all the time. From there, test automation is only a step away: "If all regression testing is done extensively and automated", they reason, "we can limit the code intervention to fixing parts that cause actual issues, and automated tests are cheap to execute, so it's a no-brainer!".

In practice, this reasoning neglects a critical component of the cost of testing, in general, and test automation, specifically: maintenance. Even when the test suite is effective at detecting regression, its application is inherently inefficient. Remember: regression is not always detrimental. In some cases, its impact is trivial. But a trivial change in behavior is a change nonetheless, and automated tests are binary in their result. This means that with any trivial regression in the product, the effectiveness of the tests will decrease as some tests will fail while not detecting detrimental regression.



↗  
**Pieter Withaar**  
*Test automation enthusiast, consultant and trainer at Improve Quality Services.*



↗  
**Johan van Berkel**  
*Test consultant and technical writer at Improve Quality Services.*

“

REGRESSION IS NOT ALWAYS DETRIMENTAL. IN SOME CASES, ITS IMPACT IS TRIVIAL. BUT A TRIVIAL CHANGE IN BEHAVIOR IS A CHANGE NONETHELESS...



## THE REGRESSION MANAGEMENT QUADRANTS

“ADDITIONALLY, THE QUADRANTS CAN HELP TO DETERMINE WHICH QUADRANT YOUR ORGANIZATION IS AT...”

To make the regression tests effective again, all regression tests failing due to trivial changes need to be adjusted to account for the change. This, basically, moves the maintenance burden from code to test, and since test only detects regression and does not prevent it, this maintenance effort is endless and ever-changing. Where the organization thought it found the magical unicorn, they actually end up running circles in the swamps of maintenance hell.

### High code quality, low test quality: well, that escalated quickly...

It may be that the organization has spent considerable effort on code quality in the initial version of the product, understanding that doing so would lower the burden of maintenance. Or it could be that the organization stuck in the swamps of maintenance hell didn't fall for the unicorn and understood the value of preventing regression over detecting it. Regardless of their reason, the organization could explicitly strategize to focus their attention on code quality. Although this is, arguably, the “lesser evil”, it still poses some challenges on its own.

The problem with good code quality is that – on the surface – it devalues regression testing. If regression rarely ever occurs, then why spend all this effort on building extensive automated regression tests that rarely ever detect detrimental regression? Alternatively, there might be a lot of effort spent on test automation, but the resulting tests are actually ineffective in detecting detrimental regression. How can you know whether they are effective or not if nothing ever fails? Regardless of whether the automated tests are deprioritized or ineffective, the result is the same: all tests are “green” and no complaints from customers, so all is well, right?

The answer is “yes, for now”. The primary purpose of regression tests should not be to find practical failures but to help identify patterns that emerge from finding those failures. Consider this: degradation in code quality should cause regressions tests to fail regularly and erratically. But with automated tests being ineffective at detecting regression, this pattern doesn't occur. Instead, the regression caused by the gradual decline in code quality builds until it reaches critical mass and blows up in their faces. Customer complaints start pouring in on a regular basis until the damage is too much to

ignore and the strategy is revisited. Since the issues were not detected, the first inclination is to analyze the automated tests, which will uncover that the quality of the automated tests is, in fact, insufficient. Suddenly, the unicorn in the bottom-right starts to look very real as it winks seductively. And before we know it, we're back in the swamps of maintenance hell.

### High code quality, high test quality: at long last, we reach the Promised Land

With all the wrong ways of going about managing regression explained, we end our story in the quarter where everyone wants to be: the Promised Land. It should no longer come as a surprise that proper regression management requires investing in both high-quality code and regression tests. Hopefully, we helped you realize that the true purpose of regression testing shouldn't be to detect failures but to prevent them from occurring in the first place. So even though the initial costs of setting up your product and tests for proper regression management can be costly, the upkeep of such a strategy is significantly lower than the costs your organization would incur from buying all those silver bullets.

Three of the four quadrants turned out to be dead ends, but they still serve a purpose: providing arguments that prevention and detection both have their own merit. High code quality is needed to achieve low regression rates, whereas high-quality regression tests are required to retain it. Additionally, the quadrants can help to determine which quadrant your organization is at, which – as it turns out – requires a broader perspective than just analyzing the current list of incidents. Instead, focus on patterns that occur over time: do the automated tests fail frequently? Better double-check your code quality. Structurally reworking failed test cases that shouldn't have failed in the first place? You might be chasing that illusive unicorn. Perfect scores on static code analysis with all builds green? No time for complacency, but remain critical about your test coverage to prevent the need to “duck and cover” from the sudden influx of customer complaints. And regardless of the characterization applicable to your organization, we hope we helped in the process of turning the maintenance hell, into a maintenance “swell”...

Ok, we'll show ourselves out now. ■

Code Quality ↑

Test Quality →



# CERTIFY FOR SUCCESS

Testing professionals need to be able to develop and demonstrate their expertise; certification can be a valuable, and often essential, asset to this. The most convincing accomplishments are measurable, and iSQL certification contributes to this with independent validation of skills.

The benefits of certifications are numerous. They divide into three main aspects:

## **Professional achievement**

iSQL certified professionals build exceptional careers. They understand common processes and standards as well as points of references in a specific field, provide high-quality structured service, show commitment to professional development as well as to discipline, take knowledgeable responsibility for a topic and are able to communicate with standard QA language.

iSQL certified professionals use their certifications to validate their professional knowledge and mastery of skills and attitudes to their employees, coworkers, and clients.

## **Professional opportunity**

iSQL certified professionals compete in the global market for the best IT jobs and salaries. They continuously advance and grow in their careers, improve their job performance and are ready to overcome challenging tasks. iSQL certified professionals show with their certifications that they are interested in their professional development and that they meet the requirements and preparation for a certain task.

## **Personal Accomplishment**

iSQL Certified professionals report pride, fulfillment, success, and empowerment as a result of passing the exam and obtaining the certification. It enhances their contribution in their organization and more satisfaction in their roles. They know how to handle challenges, complexity, change, and they build exceptional careers based on invaluable personal and professional confidence.

---

## **iSQL as your certification partner**

iSQL dedicates to the success of partners and clients. The company has three main roles as a certification body: knowledge auditor, knowledge editor, and knowledge mediator.

iSQL cooperates with qualifications boards around the world, and this is why our certifications meet the highest requirements for standardization and quality assurance. Also, because they are based on syllabuses and guidelines created by independent experts.

iSQL certifications create security for professionals and businesses alike. In addition, they provide the guidance businesses need to gain a better understanding of the qualifications of professionals in a non-regulated industry.

Certifying with iSQL means certifying for success!

[www.isqi.org](http://www.isqi.org)





# DESIGN THINKING IS A “TEAM SPORT.”



“ EACH MEMBER OF THE TEAM BRINGS  
THEIR SKILLS, INTERESTS, AND  
PERSONALITY INTO A DESIGN PROJECT.

Everybody talks about Design Thinking. It plays a crucial role in how companies formulate their business models. It is a collaborative, innovation-seeking approach to problem-solving and solutions development.

iSQI introduced the certification A4Q Design Thinking into its portfolio last year. The certification is for anyone who sees the potential in collaborative ideation and evaluation. A certified Design Thinker understands the creative process, knows how to apply the tools of Design Thinking and other methods and can create the right environment and circumstances to enable innovation.

The SQ mag talks with Mark Tannian, Professional Development Provider at RBCS. He supported the development of the A4Q Design Thinking certification and the course.

# E

*Hi Mark, tell us a little bit about you.*

My family and I live in the New York City area. I currently teach, do research and consult on security and engineering projects. My primary areas of interest are in cybersecurity and privacy. I enjoy cooking and like to run in the nearby parks.

*Are you a Design Thinker?*

I would like to think so. The desire for innovation and learning motivates me every day. I keep the focus on the user as I participate in product and service design. Users are essential to the success of products and services. Many security and privacy challenges need innovative approaches to utilizing people, process and technology. My first big user-centered design effort was information-visualization research that worked with professionals to improve decision making during complex information technology incidents.

*What does it mean to do Design Thinking?*

Design Thinking is a "team sport." Each member of the team brings their skills, interests, and personality into a design project. Collectively the team participates in Design Thinking. The term "Design Thinking" has

two popular meanings. The oldest and most scholarly meaning is the study of how designers (ex. architects, industrial designers) think as they create. The second meaning is related; however, it is significantly different. It focuses on using the ways of designers to influence how business teams approach innovation so that their efforts will yield success more reliably. From my point of view, innovation is a pioneering practical solution to a significant challenge or problem. There is no one best or right way to go about Design Thinking. Businesses and teams need to pick the methodologies and tools that work best for their culture and lines of business. It is the second meaning that I am referring to in our discussion today.

*Why is it helpful – or better: necessary – to try new ways of thinking?*

We normally consider thinking to be a personal private internal mental activity. Design Thinking externalizes or draws out what team members and users are thinking and feeling in order to guide the collective thinking of the design team. In a sense, a feedback loop is formed. This loop conveys the ideas triggered and feelings experienced by teammates and users as they explore the team's design efforts. Design Thinking embraces the exploratory dimension of innovation.





## DESIGN THINKING IS A “TEAM SPORT.”

At the beginning of a design effort, the design team makes multiple educated guesses. Design teams are unlikely to develop the “right” design of a truly original and valued product or service in a single pass through a design methodology. Users are needed to help avoid deep commitments to poor guesses while identifying and refining good ones.

### *What is the difference between teams which do Design Thinking and those who don't?*

With so many reasonable approaches to Design Thinking, teams may have adopted aspects of Design Thinking without realizing it. A Design Thinking team is user-focused. Beyond broadly identifying what challenge to focus on, a Design Thinking team allows the user community to shape the problem being solved. Using divergent and convergent thinking practices the team identifies and integrates their understanding of the users and their needs, wants and desires into a resulting design. Teams quickly produce multiple candidate solutions early on; however, with resource and time limitations it is necessary that only one best candidate makes it to the end. Teams that design independently from their users are unlikely Design Thinking teams. There are many products and services deep within supply chains, such as tiny watch screws or credit card processing that are barely recognizable by the end user. In those cases, Design Thinking may not necessarily improve those product or service designs.

### *What's about your team at RBCS? Do you use Design Thinking?*

There is a practical question businesses need to consider. How do we get paid? As a consultancy, our work products become our customers' property. There are a number of contract arrangements, such as Firm Fixed Price, that are restrictive and make Design Thinking's exploratory approach difficult to incorporate. Many of our contracts at RBCS specify fixed deliverables. Having said that, we at RBCS see Design Thinking as a powerful approach to adopt for innovation contracts that allow more operating latitude.



### *Why is it necessary for people to prove they can work in the way of Design Thinking?*

Businesses and business units that have external or internal customer facing products or services need to integrate the voice of the customer early and often within design efforts. As businesses become aware of the power of Design Thinking they will need a workforce capable of functioning on Design Thinking teams. In many professional employment marketplaces, employers desire external verification of employee knowledge, skills, and abilities. A candidate or employee who is able to show that they are ready to participate in Design Thinking projects will be regarded above those who cannot prove their understanding of Design Thinking.

### *Why should they do a certification?*

Taking a test or exam is not commonly considered an enjoyable experience by many. However, the need to succeed in proving understanding helps you focus on the material, retain information and integrate new learning with your prior learning. Design Thinking is as much a set of skills as it is a knowledge area. The richest learning experience available is to participate in an in-person course. A student who takes the certification preparation course will experience Design Thinking by working with fellow students on a prepared design challenge. ■



I AM A

**DESIGN THINKER!**

CERTIFY YOUR CREATIVE POWER

**ISQI.ORG**





---

## INTERVIEW WITH LEAD TEST ENGINEER CRISTIANO MILIA

We met Cristiano Milia, a Lead Test Engineer at HERE Technologies in Berlin, at the Testing Stage 2019 in Kiev and talked with him about the conference and what he thinks about certification.

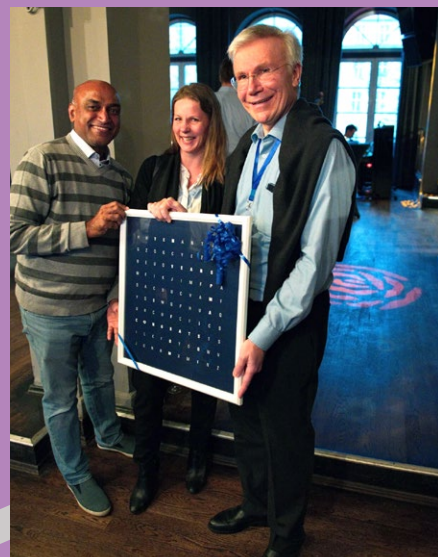
[https://blog.isqi.org/en/2019/04/08/cristiano\\_milia/](https://blog.isqi.org/en/2019/04/08/cristiano_milia/)

# HAVE YOU SEEN ...

---

## ISPMA CELEBRATES ITS BIRTHDAY

The International Software Product Management Association's (ISPMA) 10th birthday was celebrated at this year's Software Product Management Summit in Frankfurt/Main, Germany 9-10 April. Congratulations from iSQI!







## NEW TRAINING AND EXAMINATION CENTER IN CUBA

The CEO of the iSQI Group, Stephan Goericke, welcomes the first students to the new training and examination center at the UCI. Goericke was impressed by the reliability of the Cuban partners: "What was just an idea two years ago has now become reality with the support of many local partners: a training and examination center in Havana".

<https://blog.isqi.org/en/2018/10/19/software-made-in-cuba/>



## NEW iSQI FACEBOOK CHANNEL – IN SPANISH

Friends! Finally, we have created an exclusive channel for our Spanish followers: follow us now at <https://www.facebook.com/isqies/>.

We will be offering webinars and raffles and posting the latest content in software testing and certifications. Follow us and be prepared!





# TESTING APPLICATIONS FOR CHILDREN

Is designing an IT system for adults the same as doing it for children? What differentiates children from adults? How are these differences reflected in a system?

# T

The main difference between children and adults is that the first ones are still developing their cognitive, motor and socio-affective skills. For this reason, the way they interact with the world is different. Moreover, applications are part of that world. In this sense, when designing and testing applications for those under-12-year-old users, we must have some particular considerations.

As always, we must first know our users. And, in this case, our users are digital natives who begin to interact with the systems long before any of our generations. But that interaction is made with mobile devices (cell phones or tablets) and not so much with notebooks or desktop computers. For this reason, when they use our applications, regardless of the device on which they execute them, they will try to make the gestures they are used to do today (scroll with the hand, touch the screen – assuming that all of them are “touch”, turn the screen expecting a resizing, etc.). If they can’t make these movements, they will feel frustrated, particularly because children do not like to be mere spectators – they want to interact with the applications, touch the characters in the games, and live all the experiences as protagonists.

**“ AVOIDING FRUSTRATION IS FUNDAMENTAL**

References:

Nielsen Norman Group. UX Design for Children (Ages 3-12), IV edition, <https://www.nngroup.com/reports/children-on-the-web/>



# “ CHILDREN DO NOT LIKE TO BE MERE SPECTATORS – THEY WANT TO INTERACT WITH THE APPLICATIONS

Avoiding frustration is fundamental. And here, there is a difference between the behavior of children and adults. Children tend to get more frustrated than adults when a system is slow. On the contrary, when something does not work, children take it as a challenge and are more persistent than adults in trying to solve these types of problems. However, if we can prevent mistakes or provide facilities to recover from failures, we must do it because children have no experience in problem-solving. So, any attempt to overcome these issues will require a high cognitive load.

Regarding this point, we must reduce the cognitive weight of the interfaces. This applies to systems designed for both adults and children. But when the users are children, this point becomes more important. Why? Because children have less capacity for attention, concentration, and working memory. If the interfaces are overloaded, they will be unfocused very easily and will not be able to complete any particular task.

Another difference to take into account is that for adults, there are thousands of applications with different purposes. On the other hand, for children, the purpose is usually limited to games or educational programs. In other words, the variety is much smaller.

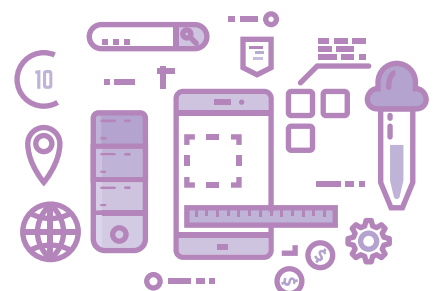
Also, children read much less than adults, some of them due to an evolutionary limitation (they still have not learned to read) and others because they are lazy. One of the difficulties we have when designing software for children is that the reading level of our users is diverse. For this reason, the instructions that we give must be clear in order for them to reach the goals of the game or fulfill the tasks that are required. But be careful! While the instructions must be clear, we must not be prescriptive; we must give them some guidelines to act spontaneously. With this in mind, we can include animations or games as part of the system, in order to explain how to achieve that goal. If this includes explanatory audio, it will be much better.

And talking about audio, let's also think about sounds. Children love everything that attracts attention: animations, sounds, etc.– in short, everything that adults find disturbing. Associated with this, children do not realize when there is advertising included as part of the system. They incorporate it as functionality.

As you will notice, designing and testing software for children are very different from doing it for adults, even though we all have an inner child within us. ■



**Nadia Soledad Cavalleri** lives in Buenos Aires, Argentina. She is an information systems engineer (2008) and psychologist (2012). Nadia has been working in Testing & Quality for more than twelve years. She is the co-founder of BoundLess where she works as Operation and Quality Manager. She is also a founder of Argenteesting, the first Argentinean congress about testing. Nadia has a YouTube channel where she uploads videos about testing and quality topics twice a month. She also has a blog ([www.nadiacavalleri.com.ar](http://www.nadiacavalleri.com.ar)).





---

# PREVIEW

## Any Questions?

---

### PUBLISHER

International Software  
Quality Institute (ISQI)  
Friedrich-Engels-Str. 24  
14473 Potsdam (Germany)  
T: +49 331 231810-18  
[www.isqi.org](http://www.isqi.org)

---

### EDITOR

Responsible for Editorial Content:  
Stephan Goericke  
Editorial team:  
Anja Schreinert, Debbie Archer  
[contact@sq-mag.com](mailto:contact@sq-mag.com)  
Friedrich-Engels-Str. 24  
14473 Potsdam  
(Germany)

---

### PHOTO CREDITS

p. 3 Stephan Goericke\_©Karoline Wolf  
Thanks to:  
p. 4 Andrii Ganzevych on Unsplash  
p. 10 Annie Spratt on Unsplash  
p. 19 John Salzarulo on Unsplash  
p. 26 Patricia Prudente on Unsplash

---

### ADVERTISEMENTS

Detailed metrics for the online  
advertising formats are available.  
Please send your request to:  
[contact@sq-mag.com](mailto:contact@sq-mag.com)



## What's in the next issue of SQ mag?

The next issue is about:  
"Agile". We are looking forward  
to articles about software testing  
and developing.

When you want to share your  
experiences with us, please send  
your suggestion for an article to  
[contact@sq-mag.com](mailto:contact@sq-mag.com)

---

### LAYOUT

Frenkelson Advertising Agency  
[www.frenkelson.de](http://www.frenkelson.de)

---

### 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](http://www.sq-mag.com)

**SQmag**  
**# 07**

COMING  
OCT 2019