

SOmag

THE VOICE OF SOFTWARE QUALITY

PERSPECTIVES

TESTING FROM THE PERSPECTIVE OF QUALITY ENGINEERING

Some people seem to think it's only about testing, but that's just one aspect of the activities in the IT delivery process.

Rik Marselis

TEAMWORK AND NOT COMPETITION

Teamwork and effective communication are skills that are often taken for granted.

Luis Francisco Contreras González & Martin Contreras Romo

THREE PILLARS FOR SUCESS:

Sometimes it helps to look outside of software development.

Erik van Veenedaal

SOFTWARE TESTING PERSPECTIVES

Agile frameworks are everywhere, but just because you are working on these frameworks does not mean you are doing or being agile.

Sebastián Viquez López

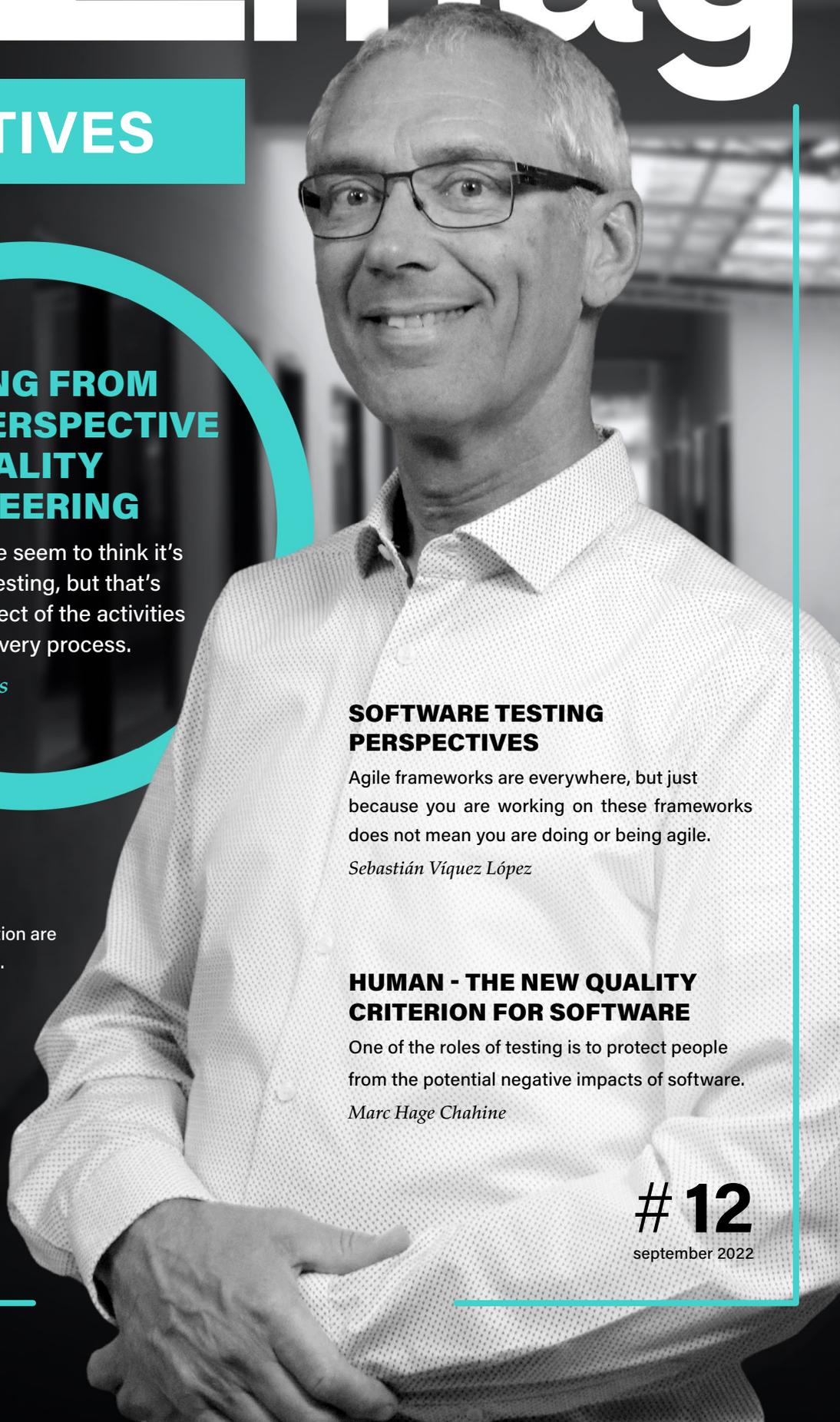
HUMAN - THE NEW QUALITY CRITERION FOR SOFTWARE

One of the roles of testing is to protect people from the potential negative impacts of software.

Marc Hage Chahine

12

september 2022





skillsclub

SkillsClub is the biggest club on
earth in the field of Software Quality

skillsclub.com



Stephan Goericke
CEO, International Software Quality Institute

DEAR READERS,

In order to develop new ideas and see new opportunities, we sometimes need to adopt new perspectives and look at things from a different angle. In the field of software testing, the ability to change perspectives is in high demand: in order to ensure high-quality digital products, software testers need to put themselves in the user's position and develop an understanding for their needs and wishes. With the advancing digitization and the increasing complexity of IT products, the professional field of software testing has become an important industry in recent years - and a real future perspective for IT professionals all over the world. Perspectives: This is also the topic of this new issue of the SQ mag that you are holding in your hands. In this edition we would like to look at software quality from different angles. Our authors Luis Francisco Contreras González and Martin Contreras Romo have taken a closer look at the relationship between testers and developers and have found that - contrary to what many people think - this is much less a competition than an understanding of teamwork. Sebastián Viquez López takes a new perspective on agile frameworks and states: just because you work with these frameworks does not mean that you actually work or are agile. Marc Hage Chahine looks at the human factor as a quality criterion for software. And Rik Marselis considers testing from the perspective of technology and explains that for the new development of quality engineering, new technologies will also play an important role, in particular artificial intelligence (AI). You may have noticed it already: Software testing and software quality are a sophisticated interplay between different positions, people and technology. To be flexible and to take new perspectives is especially necessary. We would like to inspire you to do the same and hope you enjoy reading this issue.

Yours

#12

CONTENT

TESTING FROM THE PERSPECTIVE OF QUALITY ENGINEERING

Rik Marselis

4

THE PERSPECTIVE OF THE DEVELOPER AND THE TESTER: TEAMWORK AND NOT COMPETITION

*Luis Francisco Contreras González
& Martin Contreras Romo*

8

THREE PILLARS FOR SUCESS: THE PPT FRAMEWORK

Erik van Veenedaal

11

HUMAN - THE NEW QUALITY CRITERION FOR SOFTWARE

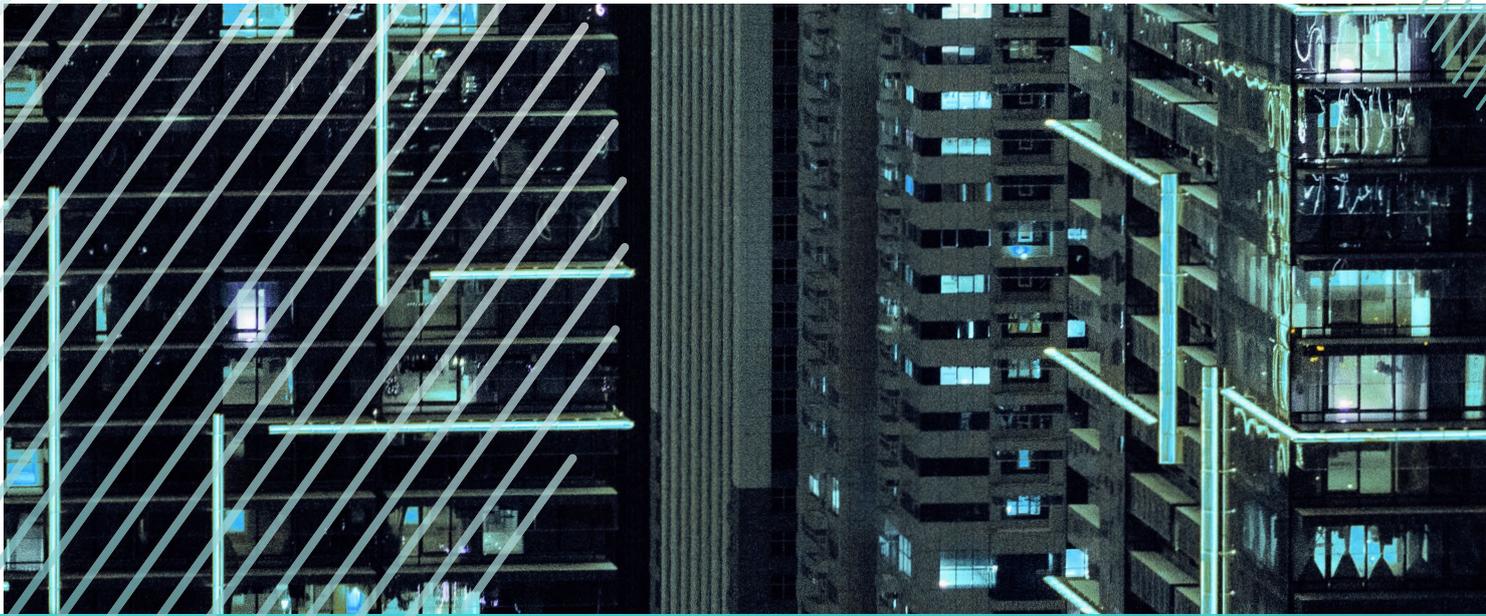
Marc Hage Chahine

16

SOFTWARE TESTING PERSPECTIVES

Sebastián Viquez López

19



TESTING FROM THE PERSPECTIVE OF QUALITY ENGINEERING





Quality engineering consists of many activities. Some people seem to think it's only about testing, but that's just one aspect of the activities in the IT delivery process. In the 1980's testing was seen as an activity at the end of the IT delivery activities to find the faults that were introduced in previous phases, and fix them. In four decades, I witnessed the evolution towards quality engineering where an IT delivery team takes joint responsibility for building-in quality from the start, in order to deliver business value with quality at speed. With this attitude, the main objective of testing is to demonstrate that the quality indeed is at the required level.

What is quality and why would we want it?

What is quality?

Quality is defined as: "the totality of features and characteristics of a product or service that bear on its ability to satisfy stated or implied needs."

How much quality is enough?

In today's world many people and organizations rely on IT systems, many things would not be possible without IT systems. So, we need to be able to trust that these IT systems are working good enough to support business processes. This means that the quality must be at the level that fits the purpose and delivers business value.

If we want to know if our IT system indeed satisfies the needs, we must measure the quality. We need to define quality indicators and measure these indicators. Most of this measuring is a

testing task. Since in Scrum or DevOps, quality is the responsibility of the whole team, this measuring of indicators can be done by any team member that has the required quality engineering skills.

Quality engineering is about taking joint responsibility

Quality engineering is defined as:

“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 businesspeople and their customers. It is a principle of software engineering concerned with applying quality measures to assure the quality of IT systems.” (source: the TMAP book “Quality for DevOps teams”).

When applying an Agile mindset with or without a Scrum framework and/or when working in a DevOps culture, TMAP talks about High-performance IT delivery. This is defined as:

“High-performance IT delivery is an approach that enables cross-functional teams to continuously improve the products, processes and people that are required to deliver value to the end users.”

To measure the quality of the IT system we use indicators. Measuring indicators is done for example by testing. Testing is defined by TMAP as: “Testing consists of verification, validation and exploration activities that provide information about the quality and the related risks, to establish the level of confidence that a test object will be able to deliver the pursued business value.”

Quality is the responsibility of the team and high-performance IT delivery is today’s way of implementing IT delivery. What does that mean for the organizations and people involved?

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

“
HIGH-PERFORMANCE IT
DELIVERY IS ABOUT CON-
TINUOUS IMPROVEMENT
OF PRODUCTS, PROCESSES,
AND PEOPLE.”

06

the territory of technical people only. Since the 1990’s businesspeople 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.

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 improving 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 TMAP introduces the concept of “quality engineering”.

Quality Engineering is about team members 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. I am 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



Figure 1 the Organizing and Performing topics of TMAP (source: Quality for DevOps teams)

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.)

The future of quality

In the four decades that I am personally experiencing the development of quality in information technology, I have seen that IT systems have become much more connected and complex. Therefore, quality assurance needed to evolve from “finding errors and fixing them” towards quality engineering where teams take joint responsibility for “built-in quality”. And now that the focus is on business value, you may wonder, what will be next? My expectation is that the IT-world will see a further evolution toward “purpose”. Business value is rather materialistic, purpose addresses doing good for the company, society and our planet as a whole. In the near future we will see how this will further develop the quality engineering practice. In this new development of quality engineering new technology will also play a major part, especially Artificial Intelligence (AI). These intelligent machines will go through three stages of supporting quality engineering. Currently AI is mainly good in descriptive

analytics; machine learning algorithms analyze massive data sets and derive information from that. Next there will be predictive analytics, that (based on the analysis done in descriptive analytics) will predict the future level of quality. Finally, we will see prescriptive analytics where AI supports quality engineering by prescribing what actions need to be taken to ensure that the quality stays at the pursued level. And even AI may trigger actions automatically if the quality is at risk!! My conclusion is that the quality of IT systems remains as important as ever, but other IT systems will automatically guard that quality and people in IT only need to use their imagination to invent new applications of IT to pursue purpose!

Happy quality engineering!!



Rik Marselis

is a principal quality consultant at Sogeti in the Netherlands. He is a highly regarded consultant, coach, presenter, trainer, and author, who supported many organizations, teams, and people in improving their quality engineering & testing practice by providing useful knowledge, tools & checklists, practical support, and having in-depth discussions. Rik is the chairman of the TMAP special interest group, and he is a fellow of Sogeti's R&D network SogetiLabs. He has contributed to over 20 books in the period from 1998 through today. His latest book is “Quality for DevOps teams” as part of the www.TMAP.net body of knowledge for quality engineering & testing. In 2022 Rik received the ISTQB International Software Testing Excellence Award and the EuroSTAR Best Tutorial Award.



TEAMWORK

AND NOT

COMPETITION

In the digital world in which we find ourselves, where there are endless systems and applications that help us make our lives simpler and more organized, it is vital that these systems and applications have good quality, not only for proper operation and user experience but also because many of these control sensitive information such as banking, school and financial systems as well as those that help and support lives in the field of health care, etc. Systems development is the work of a group of people in which the sum of different skills, knowledge, capacities and efforts lead to achieve the objectives and quality and safety standards to reach the desired goal. To achieve these objectives, people with different profiles, responsibilities and tasks, complementary to each other, are grouped into teams.



THE SOFTWARE FROM THE PERSPECTIVE OF THE DEVELOPER AND THE TESTER

Some of these in a general and brief way can be the following:

- **Business:** Provides the project requirements and gives us progress feedback.
- **Management:** It is responsible for the estimation, administration and fulfillment of objectives and goals, manages the start-up and evolution of projects, manages and resolves problems and escalations as well as project completion and approval tasks.
- **Infrastructure:** Create environments and services. These elements include hardware, software, cloud computing, network elements, an operating system (OS), and data storage.
- **Technical Lead:** Translates business requirements into technical solutions.

- **Database administration and developers:** create, administer and manage the database in different environments.
- **Software development:** Take the requirements and create the product or system based on them.
- **Quality Assurance and testers:** they are in charge of designing and executing tests in order to find bugs in that system and report them so that they can be fixed before reaching production, they also provide analysis and suggestions to improve and ensure quality.
- Among many others depending on the project, company and methodology.

In this way all the teams create parts of the puzzle and the testers make sure that they fit correctly and that they fully meet the requirements.

The good relationship of the teams is very important to achieve the objectives and among all these relationships a very significant one is the one between developers and testers since a sense of competition can be created given on the one hand by the different perspective with which you can see the product of the work by both teams, the developer often has a closer but therefore less complete view of the product, the developer can thus overlook conditions that he does not know and that can generate errors, lack of uniformity in the system, incorrect presentation or even misspellings of the screens for capturing and/or presenting the information. On the other hand, the tester has a vision with less detail of the development of the system but broader in terms of what the system is expected to do, how it should be done and how to present the information to the end user, the old story of seeing the results. trees and not see the forest or on the contrary see the forest and not know in detail what trees form it.

On the other hand, among others, the objectives of testing are:

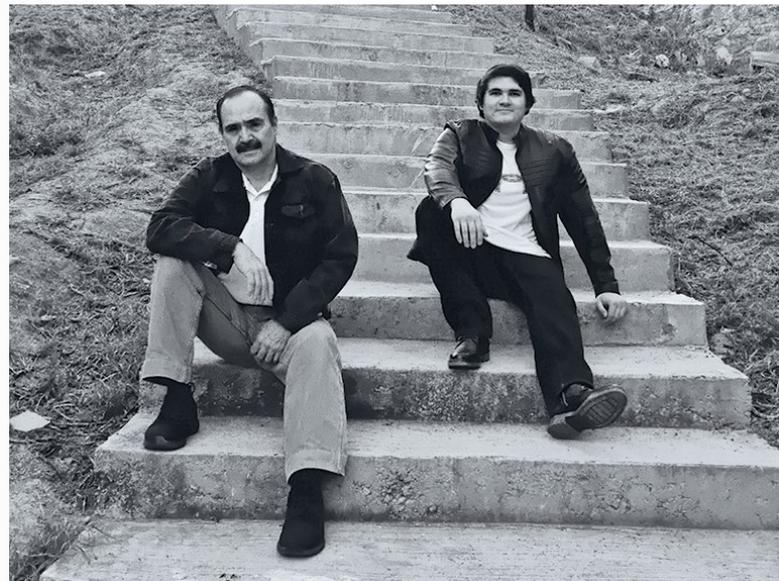
- Prevent defects by evaluating products such as requirements, user stories, design, and code.
- Check if all specified requirements have been met.
- Check if the test object is complete and validate if it works as expected by users and other stakeholders.
- Find defects and flaws to reduce the level of risk and inadequate software quality.
-

Evaluating, verifying, checking, finding defects are tasks that are not always pleasant for those who are creators of the products under observation, that is why many people place the developer and testing as opposite poles and that one is in charge of proving the other or they have the idea that it is testing vs development. The reality is that both complement each other and at the end of the day what is sought is the same goal, to deliver a quality product and service to the end user, for which it is also crucial to have good communication skills since you will be collaborating with other people and teams whether you are working with an agile or traditional methodology. Teamwork and effective communication are skills that are often taken for granted when in fact they are one of the

most important since they serve to keep the different teams and stakeholders aligned.

It may even be somewhat beneficial to take a bit of the position or optics of the other so that in this way the developer will look for a broader spectrum (see the forest) to, for example, carry out different tests and think about how the system can fail and how to ensure a greater quality from programming, and as a tester (get closer to the tree so as not to ask for pears from the elm) to become familiar with some technical programming issues and how systems are created in order to be able to analyze in more depth and do more and better tests as well as allow both teams a more pleasant and simple communication and collaboration. It cannot be claimed that the systems are developed free of bugs and/or that the testers resolve them, but it is possible that the collaboration of both teams achieve systems with the least number of errors possible, which leads us to one of the principles of testing: absence of errors is a fallacy so looking for 100% in this case will only lead to frustration and anger.

Finally: It is said that when the water rises at the dock all the boats tied to it rise together as well.



Luis Francisco Contreras González

Software Test Lead QA Specialist/Test Developer, with multiple 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 tester amongst other professional certifications from several entities such as Microsoft, Google, Harvard and Scrum.

Martin Contreras Romo

Electronic and software Engineer specialized in programming with more than 30 years of experience in design and development of systems for private companies and government agencies, master of programming and robotics during the same time.



ISTQB, TMMi OR TEST AUTOMATION

**THREE PILLARS FOR SUCCESS:
THE PPT FRAMEWORK**

In trying to improve software testing and thereby achieve higher levels of product quality, the testing community doesn't seem to agree. There are those that claim that in Agile, it's all about people following the manifesto statement "people over process". To them processes are no longer needed. They more or less coincide with those that state that models like TMMi, despite tangible results having been reported [1], are obsolete and processes are not needed anymore. Of course there are also those to whom test automation is the answer to everything. Who needs skilled testers, a mature process, let's just start with test automation and every software quality problem will be solved. Who is right, or at least partly right? Who is wrong? This has been an endless, and perhaps even senseless, debate over the last so many years. Sometimes it helps to look outside of testing, or even outside of software development. In business management, a highly popular framework is the People, Process, Technology framework (also known as the PPT framework). It refers to and exhibits how the balance of people, processes, and technologies drive successful organizational change, improvements and re-engineering.

The PPT Framework

The PPT framework has already been around since the early 1960s [2]. The original model featured four elements:

1. *People: those who perform the tasks*
2. *Structure: how the people are organized*
3. *Tasks: what the people are doing*
4. *Technology: the tools that are being used.*



Figure 1: The PPT Triangle

Many have since combined structure and tasks into processes creating a sort of triangle shaped framework (figure 1). "People, process, technology" has become a mantra in business management. Since the 1990's there has been a shift away from individuating people, processes, and technology from one another. Instead, the focus when using the PPT framework is on looking at how these elements work together and influence each another. This has been a major shift in thinking. As separate components, people, process, and technology are essential for organizational growth, transformation, and management. To achieve organizational efficiency and effectiveness, all three elements must balance and sustain good relationships and interactions among themselves. As such the PPT framework is widely applied in for example the security domain and throughout the digital transformation process that organization are undergoing. When it started security was often largely considered to be technology-only issue, using the PPT framework the notion that people and process needed to be incorporated into an overall security system was established. Understanding each element of the PPT Framework individually is a pre-requisite, before trying to understand the relationships between the elements. Learning how to regulate each aspect, implies

better being able to control them. Although the balance is critical, let's first briefly explain and discuss each element of the framework.

People

The PPT framework considers people to be the most crucial part of the triangle. People refers to the employees within the organization. They are the ones who complete the process tasks, sometimes supported or leveraged by technology. Employing and hiring the right people is essential. An organization needs to identify which skills, experience, attitude, and values are required for their employees. People also require clearly defined roles, so everybody knows their responsibilities. Ensuring that a team consists of the right (mix of) people, with the right communication between everyone who's a part of a change is also critical. Finally, businesses need to get a buy-in from their employees. They need to understand what they have to do, why they're doing it, and how changes affect them. The more they understand and believe in the changes that are being made, the more effort they will put into implementing them.

Without discussing the people aspect of the PPT framework in detail, trying to relate this to testing, we can very easily state that the International Software Testing Qualifications Board (ISTQB) scheme and portfolio plays an essential part for complying with the people aspect for testers with regards to the PPT framework.

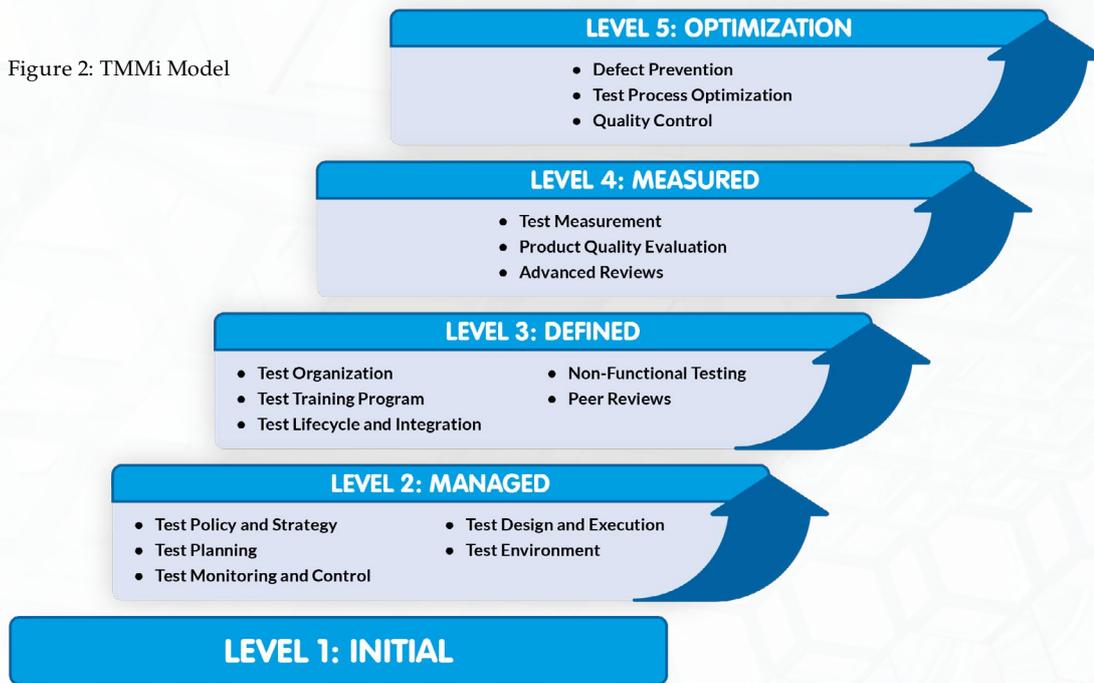
Process

With the framework process refers to the steps or actions to produce a particular result. A process in the PPT framework mostly focuses on the "how to do" aspect. How will we reach our results? How do we utilize the people and technology to reach this result? Without processes in place, people don't have a clear idea of what and how to do. Without people in place, processes don't get done. Processes are repeatable actions that theoretically produce the same result independent of who performs them. Implementing processes in an organization is most often not as easy as it initially looks. It's typically a difficult and extremely challenging task. The PPT framework provides guidelines for designing and implementing processes. It also states that once the people and processes are in place, organizations should consider the technology to support them thereby prioritizing the three aspects.

If we think about processes in the context of testing, the leading model or process framework for test process improvement today is the Testing Maturity Model integration (TMMi). The TMMi is a five level staged framework with process area at each level (figure 2). Testers perform the testing processes, and of course skilled and experienced testers typically produce better results using the same processes. The TMMi sur-



Figure 2: TMMi Model



vey [1] revealed that no less than 87% of the TMMi worldwide users have also embraced ISTQB to train their testers (and sometimes developers). This is good example where in practice both the people and process aspect are used in conjunction to improve testing and thereby achieve higher levels of product quality.

Technology

The technology provides the tools that the people can use to implement and perform the process. It also helps automate some parts of the process. The industry is coming up with new, helpful technologies and tools almost every single day. Ideally, the latest technology creates the most impact. It's very tempting to get attracted to "shiny" new tools. However, technology alone cannot solve all of your problems. Given the PPT framework, technology needs people and processes to work correctly and achieve the expected benefits. Too often companies make huge investments into technology to gain strategic advantages with people and processes being a second thought. Then they try to fit the people and process into this new technology. But this typically won't bring out the best outcome. Technology is only as good as the processes that are implemented around it, and processes are only as good as the people who execute them. If the people have not been trained how to use it or the process doesn't utilize it well, then the technology will not bring the best return on investment. Therefore, according to the PPT framework technology cannot be the solution to any problem by itself. Businesses need to articulate the objectives ("their needs"), define the process, and train the people to leverage technology to its fullest.

Translating the technology aspect to testing, points in the direction of test tools and test environments. There are many type of test tools but of course the most popular is test automation also referred to as test execution tooling supported by

various methods, techniques and frameworks, e.g., Selenium being a highly popular test automation framework. Especially with the uptake of Agile, more focus on unit testing, these tools including their framework, have become highly popular. Indeed the quality of this technology has improvement tremendously over the last decade. At the same time, I still see many organization struggling with test automation. Quoting Rex Black "less than half of the major test automation efforts I've seen with my clients are still achieving a positive ROI after five years" [3]. Putting test automation in the context of the PPT framework, technology needs people and processes to be really successful. Remember, people (testers) know what to test, they assess the risks and design the most interesting test cases that make the difference. All of this seems like an obvious statement, but apparently it is less obvious looking at every practice.



Balancing

The PPT framework is all about how the three elements interact. The three elements must balance one another. The three elements exist independently, but they do affect each another. This means that the actions of one component will affect another. If you change technology, you'll see changes in people and processes. The same relationships exist with each intersection. Some people refer to the relationship between these three elements as a 'triple constraint.' If one element shifts, the other two must do so as well. Without compensation, the

three elements will fall out of balance. Balancing the PPT Framework isn't easy. It takes constant management and restructuring. Many businesses throw new technology at their problems. However, technology is only as effective as the processes utilizing it and the people who handle it. In this way they will also most probably not take full advantage of the value delivered by technology. If an organization is too much process focused, they'll end up with a good plan on paper but without the right people or the technology to support it. Without mature processes, the actions of the people will be highly ineffective. Thus, businesses need to find the right balance between the three critical components. The PPT framework encourages organizations to think multi-dimensionally. The framework helps to map the entire value streams of people, processes, and technology. Understanding the balance between the elements can be difficult, with the balance being different for each context.

The era of new technologies, digital transformation and technology-focused businesses

When the age of digital transformation began, some people started assuming that the PPT framework would no longer be relevant. However in practice, successful businesses were leaning on the model more than ever while implementing new technologies in their organization workplace. Without the PPT framework, organizations would fall out of balance with all of the new, innovative technologies being implemented.

Traditionally, the order for structuring the framework was defined as: 1. People, 2. Processes, 3. Technologies. Technology-focused organizations may reverse the steps. Since they focus on technology use, they will typically be more successful with an altered approach. Instead of completely balancing technologies with people and processes, they preference technologies. In information security, people and processes can be unreliable. Therefore, a focus on technology can lead to more impactful change faster. The PPT framework may be slightly altered, but the main idea is still the same. There is a favor towards technology, but this does not change the fact that everything must be balanced. Favoring technology does not mean that technology should overpower people or processes. One must make sure that technology is being used in a way that it complements the people and processes within an organization.

Bringing it all together, changing the mindset

Going back to the initial problem stated, with the testing community debating what has more impact: ISTQB, TMMi or Test Automation. It should not be 'or' but rather 'and'. The focus of the discussion should be how the three can work together successfully. We need people, processes and technology to make the change and achieve higher levels of software quality being delivered to our customers. Many things from the business oriented PPT framework can easily be translated to testing and/or software development. Already in the 1960's the framework stated that people should be the primary fo-

cus, we now finally have Agile stating "people over process". The balance will be different depending on the context. In Agile people will core, but in regulated environment processes tend to be very important as well. There are certainly many technology-focused organization in the testing industry; they will favor the technology aspect to make the most impact. Of course the people aspect is much more than just ISTQB, but ISTQB with over 1.000.000 exams is certainly part of it. It's great to see that the ISTQB organization and the TMMi Foundation have signed an alliance to work together. This is what is needed.

Let's not spent our negative energy anymore on debating the differences and who is the winner. Let's change our mindset towards a positive attitude by focusing on how people (ISTQB), process (TMMi) and Technology (test automation) can successfully work together and make the impact that is so much needed in today's industry.



Erik van Veenendaal

Erik van Veenendaal (www.erikvanveenendaal.nl) is a leading international consultant and trainer from Improve IT Services BV (Bonaire), and a recognized expert in the area of software testing, quality and requirement engineering. He is the author of a number of books and papers within the profession, one of the core developers of the TMap testing methodology and the TMMi test improvement model, and currently the CEO of the TMMi Foundation. Erik is a frequent keynote and tutorial speaker at international testing and quality conferences. For his major contribution to the field of testing, Erik received the European Testing Excellence Award and the ISTQB International Testing Excellence Award.

A4Q TESTING SUMMIT AND ISTQB®

It's official. A4Q Testing Summit is an ISTQB® recognized conference. It is the 20th anniversary of ISTQB®, and we are very proud to be part of this history. Let's celebrate live from Marrakech on the 19th of October. We will broadcast directly from the ISTQB® General Assembly through our social media channels.



ISTQB®

More than 1.1 Million ISTQB® - International Software Testing Qualifications Board® exams delivered! By the end of 2021, 1.1 million ISTQB® exams were delivered, and 806k individuals were certified! Congratulations to all the individuals, organizations, Accredited Training Providers, academic institutions, Member Boards, and Exams Providers who have been a part of this success! The ISTQB® Certified Tester Scheme is the leading certification for the testing profession globally!



THE FIRST A4Q TESTING SUMMIT - DUBLIN

The A4Q event was created to connect the world of Information Technology, and we are thrilled that the First A4Q Global Testing Summit directly from Dublin was such a success.

The next summit is about to start, and we are sure that the following special Jubilee edition will be even bigger with your support.



AWARD WINNER

Rik Marselis has been honored by the ISTQB® with the 2022 ISTQB® Software Testing Excellence Award! The ISTQB® selected Rik in recognition of his significant contribution to the fields of testing and quality. His skills and efforts align extensively with the Award criteria: Increasing public awareness of the importance of software quality and testing. Contributing to the improvement of software development and testing processes. Advancing the publication of research findings in the field of software quality and testing Promoting further education and lifelong learning in software quality and testing. Supporting the creation of standards and best practices in the area of software quality and testing.

Congratulations to Rik!

Take the opportunity and read Rik's articles in this special SQ Mag issue #12 – Perspectives.



When I took (and passed) my ISTQB® Certified Tester Foundation Level certification in 2014 there was one point that “amused” me in the introductory chapter. One of the reasons we test software is because “Software that does not work correctly can lead to many problems, including loss of money, time, or business reputation, and even injury or death”. (ISTQB® Certified Tester Foundation Level syllabus). At the time, this seemed “obvious” to me with the simplistic example of the software used to fly an aircraft. Nevertheless, for me, this remained in the realms of the “anecdotal”, the “marginal” case but which I would never be confronted with... However, I accepted that the testers are supposed to measure quality and that they must take the human element into account which means the tester should evaluate the potential impact on users... This point of software impact has now become paramount. In fact, even if I have never worked on software whose failure could endanger the lives of its users (or beneficiaries), this postulate has recently become more and more important in my tester life. Indeed, I can no longer hide the huge impact on people’s lives of the software I test!

HUMAN

THE NEW QUALITY CRITERION FOR SOFTWARE



Software have a huge impact on the lives of its users

This point may seem obvious, but it has never been truer than it is today. It took me a long time to understand it, but now it is obvious to me: software strongly influences our lives, emotions, quality of life and even our relationships.

I can no longer ignore this impact. In my case I started to open my eyes in 2014 by following the media reporting of a Facebook study done without the knowledge of its users via a partial rollout in 2014 (most likely a Dark Launch)

<https://www.theguardian.com/technology/2014/jul/02/facebook-apologises-psychological-experiments-on-users> >.

Facebook’s study was about the users’ mood. To do this, Facebook (without the knowledge of its “guinea pig users”) used its software to display oriented information. For one panel it was only positive information, for the other, only negative. The idea was to see the influence on their behaviour and mood. The result of the study showed the strong impact of this information display on the users’ mood.

The study caused a scandal at the time because even if it was allowed according to the “TOS” (Terms of Use) it is not ethical to make uninformed people participate in a study... especially if part of the goal of this study is to deteriorate the morale of

these people which can lead to unmanaged consequences. I kept this information in the back of my mind and thought, "thank God I don't have Facebook... and anyway it's Facebook, a huge company with far too much power!"

Again, I was wrong and underestimated the phenomenon. It took a few more software-related scandals such as the use of loyalty cards to offer targeted promotional campaigns, legal AIs making it easier to set white people free, or Cambridge Analytica for me to focus on the problem. In the meantime, we have seen the emergence of voice assistants, the multiplication and expansion of social networks, the generalisation of purchases on online platforms, the development of YouTube algorithms, the democratisation of video-on-demand platforms, the multiplication of small "freemium" mobile applications, online betting and so on.

The examples given cover a large number of software packages and that it is quite difficult as an IT worker not to work, at least for a while, on one of these software packages that have a huge impact on people's lives! However, these types of software have all been the subject of scandal:

- **Voice assistants:** with an always-on microphone to retrieve our data and use it for commercial purposes:
<<https://foundation.mozilla.org/en/privacynotincluded/>>.
- **Social networks:** and their algorithms pushing us more and more into an inter-self by only seeing what interests us and posts with a similar opinion to ours.
- **The development of YouTube:** with its algorithm pushing reaction videos which pushes low value content types or worse... videos pushing any type of reaction to attract views. (like Donald Trump's campaign videos)
- **On-demand video platforms:** whose algorithms offer us more and more suggestions corresponding to our tastes and becoming more and more time-consuming, without offering us anything different from our previous viewings
- **Online shopping platforms:** which create artificial desires (same principle as video-on-demand platforms) and, in some cases, exploit their employees and partners through rating systems
- **Freemium or in-app purchases:** which, like scratch cards, are increasingly pushing for purchases by maliciously using behavioural mechanisms
- **Online betting applications:** pushing to bet more and more money and making precarious people (targets of their advertisements) believe that they will become rich... while at the same time they ban people who earn too much from their platforms.

And I'm only talking about examples here, you'll find many others if you search. To have access to a list of concrete products I invite you to go to the "list of shame" of the Mozilla foundation (confidentiality not included)

<<https://foundation.mozilla.org/en/privacynotincluded/>>.

I now propose to take a closer look at one of these mechanisms linked to algorithms.

Example of how software works to impact and shape our lives

Lets take one representative case: Facebook and its algorithms. On 5 October 2021, Frances Haugen gave a vibrant testimony before the US Senate on the use and orientation of Facebook's algorithms

<<https://www.cnn.com/2021/10/05/watch-facebook-whistleblower-frances-haugens-senate-testimony.html>>.

It shows how this algorithm works, with the sole aim of creating more clicks and interaction, to stay on Facebook longer. Even if at first glance this may seem like a good thing (we see what we are interested in) it poses many problems, such as:

- We become totally closed in on ourselves and only have confirmation of our beliefs. Someone who thinks the earth is flat will only be offered articles on it!

- *We no longer know how to debate and exchange ideas because everything becomes an "opinion", which facilitates extremes (it is becoming increasingly difficult to convince someone that the earth is round if he or she stays too much on social networks)*
- *People are being steered towards extremes. People who are careful about their weight can quickly find themselves with articles encouraging anorexia!*

All this happens insidiously and naturally, as the algorithm pushes us more and more towards these subjects without us noticing. This brings us to the issue of freely consented submission raised by Joules-Beauvois and the question "What is our real freedom? It's only one example, but it can be replicated over and over again! However, this observation leads us to notice one thing: tests have so far failed in one of their stated goals: taking into account and measuring the aspect of protection of the life of the users! The idea here is not necessarily to make the tester the guarantor of this aspect of the software (that does not depend on him) but that he is capable of taking this characteristic into account. It is now time to solve this lack of software quality. There is obviously no single or miracle solution, but I think that we should at least include a new quality criterion: The "human".

The "Human" quality criterion to frame certain practices

I really like the ISO-25010 standard which lists the different quality criteria. The criteria defined are :

- **Functional:** verifying that the software does what you want it to do
- **Performance:** ability of the software to answer in the given time and under the given circumstances
- **Compatibility:** to check the ability to function in its environment
- **Usability:** focused on ease of use
- **Reliability:** focused on the ability to continue to function
- **Security:** related to protection (especially of data)
- **Maintainability:** ability to continue to develop the software
- **Portability:** ability of the software to be deployed in different environments

As you can see, there is nothing specific about the consequences of the software on its users! Such a criterion, which

could be called “human”, would push testers and IT actors to look into this subject! It could be composed of 2 sub-criteria:

- Ethics

This point on quality would aim to measure the quality in terms of the purpose and direct impact of the software developed. Does the purpose of the software / functionality go against or in the direction of human values? Is the implementation of the software done solely for commercial reasons at the expense of the quality of life of the users or is this quality superior? These questions cannot remain unanswered and have already been the subject of conferences, such as that of Olivier Denoo on AI:

< <https://www.youtube.com/watch?v=zPDpaqCoUyg> >

- Impact

The purpose of this quality sub-criterion would be to study the impact of the software / functionality in the medium and long term. It is often said that “the road to hell is paved with good intentions”. We can also see this drift with algorithms that give us suggestions that are close to what we like and that ultimately lock us into boxes and do not allow us to make new discoveries. The quality sub-criterion on impact would study the developments linked to the software. It would therefore be complementary to ethics which, for its part, also depends on the “era of the times”: things that seemed barbaric or divisive in the 1990s are now considered “normal” and vice versa. Whether these criteria are official or not, they seem to me to be unavoidable!

The essential “human” quality criterion?

Software is and must remain a tool. A tool that helps us and makes our lives easier. Unfortunately, we can see that these tools are currently based solely on criteria and indicators that do not really take into account the impact on humans, and that they are, sometimes, going against our interests. It seems essential that testers, but also all the actors in the software industry, become aware of this problem in order to be able to fight against these abuses. The arts have already taken on these problems, and I am thinking in particular of:

- Series such as *Black Mirror*, which at the end of the 2010s offered dystopias showing how far certain logics could lead. Numerous aspects were addressed, such as social networks and rating systems.
- The novel “*MétaCités*” written by Aude Hage. It follows the life of a family over two generations in an ultra-technological city governed by AIs which gradually drifts towards a digital dictatorship.
- The comic book “*Carbone et Silicium*” by Mathieu Bablet where we wonder about the boundaries of life with robots equipped with an AI very close to the human being or human beings totally transformed by technology to the point of not having some of their organs and limbs.

Conclusion

One of the roles of testing is to protect people from potentially negative impacts of software. It is now clear that these

impacts are not just related to safety software but to a large part of software. This problem has been compounded by the ability of software to affect more and more people and being always more powerful. The combination of these factors has brought us to the current situation and it is our duty to think of ways to positively direct the full power of software.

The proposal of a ‘human’ quality criterion is only a beginning, but it is not enough. The tester can, for example, play the role of “agitator”, who challenges the approaches and choices... within the limits of his or her capacities and responsibilities (many choices do not depend on the development teams). Other avenues can and should be considered... and it also seems obvious to me that this type of subject will sooner or later end up in the realm of legislation, as we have seen with the GDPR.



Marc Hage Chahine

-Methods and tools expert at Sogeti with a strong focus on agile testing

-Animator of the blog “*la taverne du testeur*”
<https://latavernedutesteur.fr/>

-Organiser of events dedicated to testing (JFTL, STLS...)
Software testing teacher

SOFTWARE TESTING PERSPECTIVES

The point of view

When it comes to testing perspectives several factors must be taken into consideration such as regions, locations, organization size, projects, resources, time, and of course the team members' points of view, for instance: how the team sees the testing approach, how the Product Owner or Manager sees testing, how the development team integrates testing, how the tester works within the team and how the testing approach or strategy is being carried out.

Consideration must also be given to the project Framework; nowadays Agile frameworks are everywhere, but just because you are working on these frameworks does not mean you are actually being agile. In this article, we are going to explore some of these perspectives and possible scenarios.

Let us begin with how teams see testers?

Most people take for granted the testing approach. Projects are being managed by frameworks that claim to integrate testers at early stages, but is this really happening? Are testers being informed about managerial decisions, epic changes, stories creation or edition, requirements changes... on many occasions, this is not the case. Testers usually struggle to get as much information as they need to properly work and provide not only a good performance but also to make sure the final product meets the agreed set of requirements. When it comes to teams you may see two different perspectives,

a) The QA or testing team is responsible for the quality of the product or service being delivered. In this scenario, the team will be separate from the other project teams, there may be a division when it comes to decisions, outcomes or issues.

b) The overall project team (developers, testers, POs) is responsible for the quality of the product or service being delivered. In this scenario, the project team works as a unit, decisions, outcomes, and managed by the whole team.

Now, let's review how developers see testers

Many things have been mentioned throughout the years about how developers see testers. The usual review or comment is always based on differences between these two team members. New perspectives and modern approaches have disclosed and shown that it's not a matter of who is right or wrong but what can be done for the team's sake.

In this case, you may see any of these perspectives,

a) Developers see testers as a possible issue, testing is being treated or taken for granted. Any issues found may be a burden or could lead to a misunderstanding or possible conflict.

b) Developers understand the importance and value of testing, they see testers as team members, and communication is open and improved on a daily basis.



“

A PARTICULAR ATTITUDE
TOWARD THE WAY OF
REGARDING SOMETHING;
A POINT OF VIEW

The other side of the coin, how testers see developers.

Testers have or can show different ways to approach developers. One of my favorites quote is: "Just because you are not writing the code it does not mean you are not influencing how this code is being written." Few understand this and overlook the importance of communication. Many things can happen during testing, the tester can raise a finding or possible bug related to a certain module or component that is not necessarily part of the ticket or requirement being tested, this could lead to confusion and cause delays when it comes to delivering the component itself. Another scenario is when the tester's interpretation of the requirement is different from the interpretation of the developer, this usually occurs when there is a language barrier or testers and developers are in different regions/locations (which is very common).

Working on different projects will give you some insight on how to approach developers, you can either have or create a constructive relationship where bugs are nothing but findings and possible ways to improve or add value to the product, OR, you can start a war where bugs are ways to say "you are not coding well", or even worse "you are not following the requirements". It's up to you (the tester) to decide; but be aware, this will influence the entire team's response and timing, so choose wisely and be humble.

Let's talk about how testers integrate and allocate in Agile frameworks

As mentioned above, Agile frameworks are currently being used on many projects. Some of them use Scrum, others use Kanban, Lean, XP, RAD, others use Scrum at Scale, and some use SAFe. There are also hybrid projects/models that use a combination of these, so it entirely depends on your organization's size and project needs. Now, with so many different frameworks, testers must integrate and adjust based on the selected framework for one particular project. This changes the testing perspective and approach. For some testers having a lot of meetings may not be useful, as they will probably look at these as a possible waste of time. For some testers having to point out or give a possible time to accomplish a task may be stressful or useful depending on the task and testing strategy.

Another scenario is when the tester does not fully understand the framework or has no experience with it. This happens very often, managerial decisions are being made and all of a sudden you go from Scrum to Kanban without understanding the reason why, or even worse, without really understanding how the framework actually works and how this could influence your testing plan or strategy. Regardless of the framework you are working with, make sure to communicate if training or coaching is needed to properly work on a certain project or task, as this will determine how testing will be done.

The Region or Location point of view

When you think about how testing is being done in different regions, is when you actually realize or visualize testing from a broader or global perspective. Different teams in different locations are testing websites and apps based on their project's requirements, needs, timing, resources, budget, tools ... but is there something in common? can we say they all follow best practices? Do we know if they follow ISTQB guidelines? Let's explore Senior QA Engineers' insights from different regions/locations: In this case we are going to review Romania, India and Costa Rica.

Vlad Dragomir from Romania,

"I believe that we are currently witnessing a shift in the QA field, a shift towards a more technical approach to the testing activities. Without a shadow of a doubt, manual testing is here to stay, and there are projects where the focus is mainly on manual testing, but there is a high demand for automation on almost all new projects starting today. When I'm thinking about technical aspects in the QA world, I'm not only referring to API and end-to-end automation, which fall into the usual set of responsibilities, but I'm also referring to activities that, until not long ago, were categorized as "DevOps related responsibilities". From researching CI/CD providers to configuring the CI/CD pipeline for the entire project, from being able to work with virtualization tools and having extensive knowledge about the major cloud providers in the market, the modern QA Engineer must master a wider range of skills than ever before. In my opinion, key to succeeding as a QA Engineer nowadays is to stay up to date with the latest in the field, not be afraid of change, but embrace it, do your best to adapt to change, and always be ready to get out of your comfort zone."

Payal Gogia from India,

"The QAs in the IT industry are well known for ensuring the quality standards of the software at every stage of product development, as ones who validate that the product is as per the requirements. But in a real scenario, the role of a QA is above and beyond this. As a QA, whenever we look at any product we look at it with an approach that questions everything about the product from the business perspective to what value it would provide to the end customers. We all understand how costly a bug can be when found in later stages of Software Development. To catch the bug as early as possible adopting shift left is the need of the time. We have seen many benefits of adopting this approach. It has helped QAs get a good clarity of the product's vision, how is this aligned with the company strategy and understanding of the bigger picture has further enabled them to create better test plans and test strategies. Involving QA early in the project has helped pull QA teams out of silos. Nowadays, QAs are also playing the role of QAOps in the cross-functional teams and are closely collaborating with devs and op-

erations to integrate QA activities with the CI/CD pipeline. This enables our QAs to discover bugs at an early stage, and achieve faster results. They are also participating in improving the processes in DevOps alongside ensuring a high-quality product. As a QA, to stay on top of the game an individual needs to stay updated on the emerging technologies or trends and constantly innovate and reinvent themselves. We shall keep analyzing and checking if these innovations can help resolve any outstanding issues, remove limitations of an existing tool/approach, improve processes or get results faster. We shall not restrict ourselves to a particular technology/tool but shall be open to learning and experimenting with new things and need not be afraid of failures. In a nutshell, we all understand Quality is an integral and most important part of any product development process. The Organizations shall change the outlook of seeing QAs as a separate team, instead shall ensure QAs are not working in silos and are involved early in the process to ensure high-quality products with reduced time to market. From a QA perspective, I think every QA (Manual, Automation or QAOps) shall imbibe the "WHY" attitude for what they do (Why I am doing this?, What purpose does it solve?). As a QA, we also need to shift our mindset rather than seeing ourselves only as 'validators' shall expand our horizons and be able to take up more responsibilities and play bigger roles like QAOps."

Jonathan Blanco from Costa Rica,

"I would say that assuring quality in projects, programs and tools must be the same no matter what is the region, language or traditions of the team in charge of QA, but it is difficult to standardize the knowledge between all the teams. Why is this happening? Lack of resources, seniority of the persons in the team and maturity of the people in management positions can all have an impact. So, it is important for the team in charge of learning and development of the resources, to make assessments and identify what skills can be improved individually for each person assigned to the quality assurance work. Based on the assessment results, training requirements can be identified, sessions scheduled and team members assigned on the proper path according to their skills. Also, management needs to know that paid development of the resources is not only a spend, it is an investment that will help the company to be more competitive in the market and make the team members more specialized to work in complex projects that nowadays are coming more frequently. Finally, for each of the QA's it is important to identify what skills can be improved and how to get the knowledge enough to get them. It will be the personal route that will identify us and make it unique for the company."

Regardless of your location or region, we all share the same testing principles, we are all adding value to the product or service that we support and lastly but not least, we're all moving towards personal and professional growth in highly dynamic and changing environments.

Final thoughts

One of the interesting perspectives or view is how you (the QA professional) see your own contributions. Think about how you normally contribute and how that makes you feel, reflect on the findings or bugs you have documented and how that changed the course of actions, especially when it comes to releases, deliverables and deadlines. Some of the questions you should ask yourself in order to understand the above are: ¿Do you see yourself as a bug finder? ¿Is your main reason/purpose for testing to find bugs? ¿Is finding defects the most satisfying aspect of your career? ¿When working with different projects, do you understand the importance of adding value to the product or service thru testing and how this is being done? ¿Can you openly communicate your point of view and thoughts to your team? ¿Do you seek for approval or the project's success? By doing this retrospectively is how we can improve, not only our perspective may change, but also how we currently approach and deal with every single task. The power of doing small changes on a daily basis is as powerful as you can imagine, this will be your starting point, this is where growth occurs.



Sebastián Viquez López

- Quality Assurance Module Lead at 3Pillar Global
- DevOps Consultant.
- Member of the Hispanic America Software Testing Qualifications Board
- Certified Test Manager ISTQB
- Certified Test Automation Engineer ISTQB
- Certified Dragon at Test Automation University
- Certified Scrum Master & Product Owner.
- Certified DevOps & Blockchain Professional.
- Father of two beautiful daughters
- Jiu-jitsu advance level blue belt fighter



A4Q CX Analysis

CX (Customer Experience) Analysis is critical to business success. The A4Q CX Analysis certification looks in depth at the customer experience, and how an organization can transform its existing processes to offer engaged, responsive and connected customer experiences.

The exams are exclusively available through iSQI & GASQ.



A4Q
Business Service Design



A4Q
Design Thinking



A4Q
Certified Service Designer

**Start your
journey to
A4Q Certified
Service
Designer!**

SQ mag

#13

COMING MAY 2023

PREVIEW

What's in the next issue of SQ mag?

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

PUBLISHER

A4Q Global Services GmbH
Bauerngasse 30
90443 Nürnberg
info@a4q.gs

EDITOR

Editor: Stephan Goericke
Editor-in-Chief: Julia Jasper
Editorial Team: Debbie Archer

contact@sq-mag.com
Friedrich-Engels-Str. 24
14473 Potsdam
(Germany)



PHOTO CREDITS

Cover: @shutterstock
p. 8: @Nejc Soklic (unsplash)
p. 13 - 15: @thisengineering (unsplash)
p. 19: @OmidArmin (unsplash)

LAYOUT

Format Druck und
Medienservice GmbH
www.arnoldgroup.de

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

Get ready for

A4Q TESTING SUMMIT 

Jubilee Edition

20 years
ISTQB®
anniversary

LIVE FROM MARRAKECH

19 OCTOBER 2022

20+ Presentations | 20+ Top Speakers

TRENDS IN SOFTWARE QUALITY ASSURANCE!

REGISTER NOW



WITH THE SUPPORT OF:



GASQ

