Nº3 September 2020

SQS NEWSFLASH

Integration Test Camp

Architecture for IDSA Components

Tester Mindset

The tester's posture and attitude

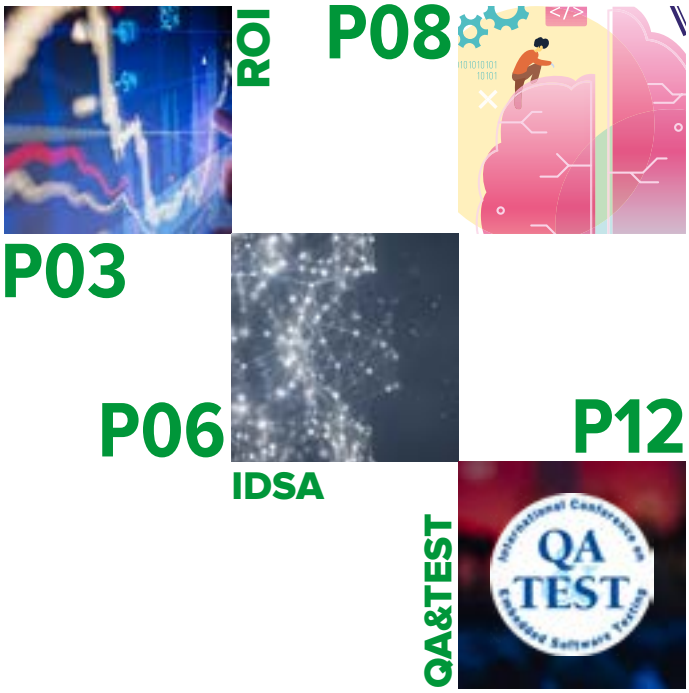
TEST AUTOMATION

Calculating the ROI

SQS Newsflash

Nº3 September 2020

Summary



Editorial

Do the testers have a different mindset? Definitely. We face our challenges with critical thinking, so that we can find possible mistakes. Our analytical mind is essential to evaluate risks, and our willing to learn and solve puzzles is the ideal complement in our search for excellence.

In our day-to-day lives, we focus on “breaking” the software rather than building it, and we need to be able to put ourselves in the shoes of the end user and imagine all the possible scenarios in which the software does not work.

From SQS Newsflash we want to create a community of testers, where we can share experiences and our different way of understanding the world. If you want to collaborate with us, you can write to us at **info@sqs.es**.

Automation ROI

Ali Khalid
Test Architect
Emirates Airlines, UAE

Learning ROI (Return on investment) for an automation project is a common activity teams often do. From the business's perspective it makes sense they need to know how much they are getting back on their investment from spending on automation.

Automation is usually sold as a cost saving exercise, and the whole ROI calculation is then around man hours of manual testing effort saved. While that is true in a very specific context, this method generally gives a very biased view of what to expect. When teams try to match the expectation with the actual cost saved, there isn't usually much to show for. Let's dissect what wrong with this approach and why things don't add up.

Manual hours saved

The calculation of man hours saved usually goes like this:

"Savings per test cycle = Tests automated x Execution effort (man hours) per test"

And then we'd calculate the break-even point when the savings equal the initial investment in preparing that automation suite plus any other costs etc. For an accountant this would make perfect sense, except the "effort per test" cost does not exist!

The way 99% of the teams do testing, they have a defined time in which they have to complete the testing and hardly ever get to execute all regression tests. So, when we say – time taken to execute all test cases, that's not true because no one ever does execute them all.

In fact, if we go by this formula of cost saving, then we might not even see a cost saving at all! The question is then, how to calculate automation ROI? To answer that, let's first have a closer look at what automation actually helps with.

Quick feedback

Automation does not directly help with testing time saved, it does help with quicker feedback and testers able to focus on 'more important' things. This is of tremendous value which is not easily captured in numbers. The cost of fixing a bug weeks after the developer pushes their code is quite high, compared to if the same issue is fixed within the next few hours after a code commit.

This is also one of the key principles of DevOps which is how teams are able to deliver faster, it helps eliminating waste. In this case, fixing an issue immediately after the code is pushed.

Help testers to spend time TESTING

Most testing teams are seen as a bottle neck, standing between pushing the feature into production. That's because they are spending most of their time 'executing' written tests and then have to also find some time to explore the application and find out risks.

With automation built properly, that takes care of laborious work of executing regression tests, saving testers time and energy to focus more on exploring & thinking about risks instead of just executing written tests. This significantly helps with quality of the product since we are able to spend more resources on exploring new risks instead of just keep on checking for risks identified before.

Calculating automation adoption not ROI

Automation does save money – but in the long run, and certainly not by saying we have 5 testers today, in 6 months' time we'll have just 2 because automation will do their job. Therefore, the concept of calculating a return on investment is not a great idea for automation to begin with. It's going to be very hard to prove cost savings, it's like proving sponsoring a charity event has brought in x amount of revenue. While we can make some assumptions, but they might not be very accurate.

Calculate automation adoption

Instead teams need to calculate how good are they at adapting automation. As teams build automation, which is 'actually' beneficial to the team, we'll be able to ship quickly and with more quality. To measure adoption, there are two key metrics teams can use.

Ownership of failures in the pipeline

Adoption also means decisions of merging code from feature branch into master branch, deployment on dev & staging environment all need to be based on results of the automation scripts. If there is no consequence of a failing automation result, no one cares to maintain them or improve them. If a test fails, it shouldn't be just the automation engineer's problem, it's everyone's problem. The whole team should focus and make it a priority to fix the failing issue which is stopping the pipeline from pushing code.

Usage of automation results

Often automation script results are viewed only by automation engineers, which means hardly anyone has stake in these results. Developers should have stake in failing automation scripts, exploratory testers should know what's automated, what's passing so they can have a better picture of where to explore for more risks in the new build. Product owners should be interested in the health of the product, for which one source should be the automation results.

When everyone on the team is relying on these results, and take decisions based on these results, that's when they start to bring in a positive change.

Calculate KPIs instead of ROI

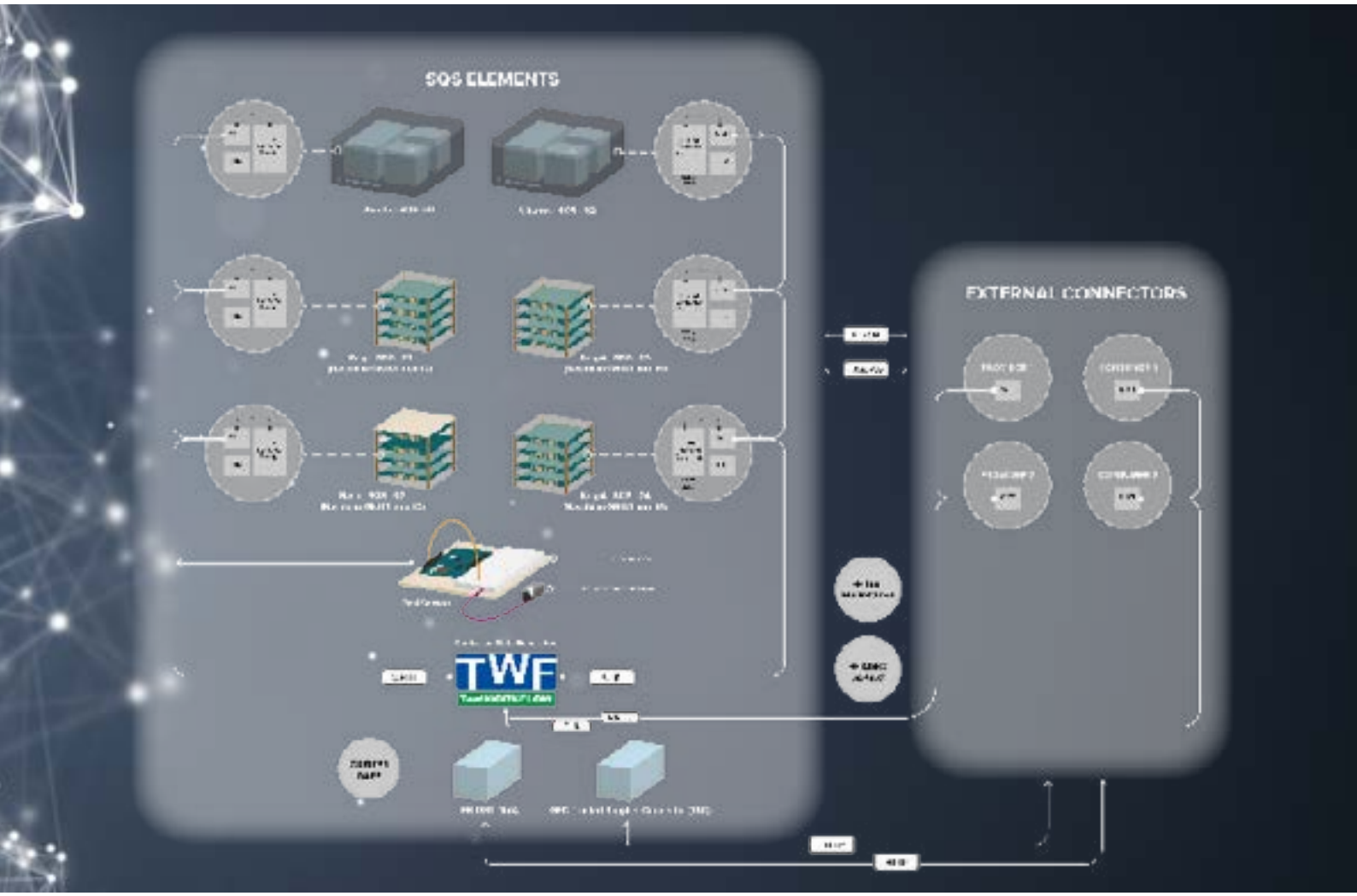
Adoption also means decisions of merging code from feature branch into master branch, deployment on dev & staging environment all need to be based on results of the automation scripts. If there is no consequence of a failing automation result, no one cares to maintain them or improve them. If a test fails, it shouldn't be just the automation engineer's problem, it's everyone's problem. The whole team should focus and make it a priority to fix the failing issue which is stopping the pipeline from pushing code.

Integration Test Camp

Olatz Mediavilla
Senior Tester
SQS, Spain

The Integration Test Camp is a remotely accessible infrastructure developed by SQS, where the interoperability of pre-commercial IDSA components could be tested in a production-like scenario. It is opened for everyone in a monthly event, which is usually announced two weeks before.

In these monthly events, participants have a slot of time, of two hours, where they have all the infrastructure available and dedicated for them. The communication with SQS team is continuous withing this time, and teams and webex meetings are used to share the screen so both sides could see what happens in the other side. To guide the sessions, a document with the test steps that are thought for each test camp is shared.



Integration Test Camp Architecture

The goal is to have an architecture with all the components that take part of the IDSA ecosystem. But, to have a trustful infrastructure, SQS decided to go step by step, developing first a basic architecture, and adding new elements in each edition.

In the architecture developed for the First Integration Test Camp there were:

• Connectors based on the Trusted Connector from Fraunhofer, supporting IDSCP communication protocol, installed in SQS facility. Some acted as providers and others as consumers. There was also DIVA connector, from Fraunhofer, supporting HTTP communication protocol and acting as provider, installed in SQS facility.

• Three different data sources:

o Real temperature sensor, sending a float with the temperature measure every five seconds.

o Real motor making random movements, sending a json with the movements made every five seconds.

o Then we have a way to send Controlled data using TestWorkFlow, a tool SQS developed, able to send any type of data participants need in a controlled way.

• A Broker and a DAPS were offered as a service.

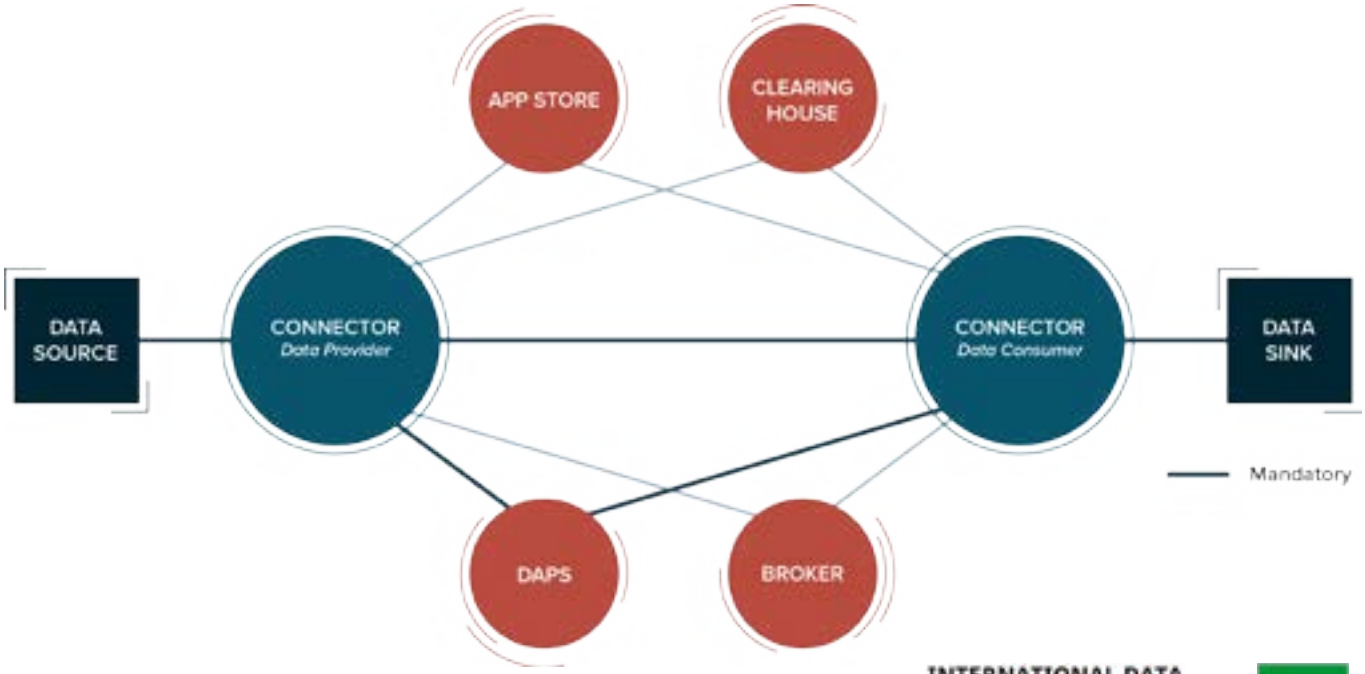
With this architecture, the participants were able to test the interoperability between connectors, mainly the handshake.

In the second edition, a DAPS, developed by Orbiter, and another Connector supporting HTTP communication protocol, developed by German Edge Cloud, have been installed in SQS facility. With this architecture, the participants were able not only to test the interoperability between connectors, also to test certificates and Dynamic Attribute Token management.

For the next edition, a Broker will be installed in SQS facility, and participants will be able to also test registration at a broker and searching of other connectors information.

Then, an App Store will be installed in SQS facility, so participants will be able to search for, download, and upload different apps and try the usage withing their solutions.

After each edition, SQS publishes a set of test cases that could be used as reference for Interoperability Testing. A review of what happened in the integration Test Camp is also published, explaining how was it managed, what were the participants able to do and what difficulties have been encountered in the interoperability of the elements.



INTERNATIONAL DATA SPACES ASSOCIATION

Participate!

	Date	Scope
ITC1	8th, 9th & 10th June	Handshake
ITC2	16th & 17th July	DAT Management
ITC3	Mid September	Broker Interactions
ITC4	Mid October	App Store Interactions

If you want to participate in the next edition, fill in the application form in SQS **web page** or send an email to **idsa_qaas@sqs.es**.

Tester Mindset

Ariel Cymberknoh
Firmware Tester and
Validation Manager
Intel, Israel

The concept “Tester Mindset” clearly and powerfully reflects the tester’s posture and attitude

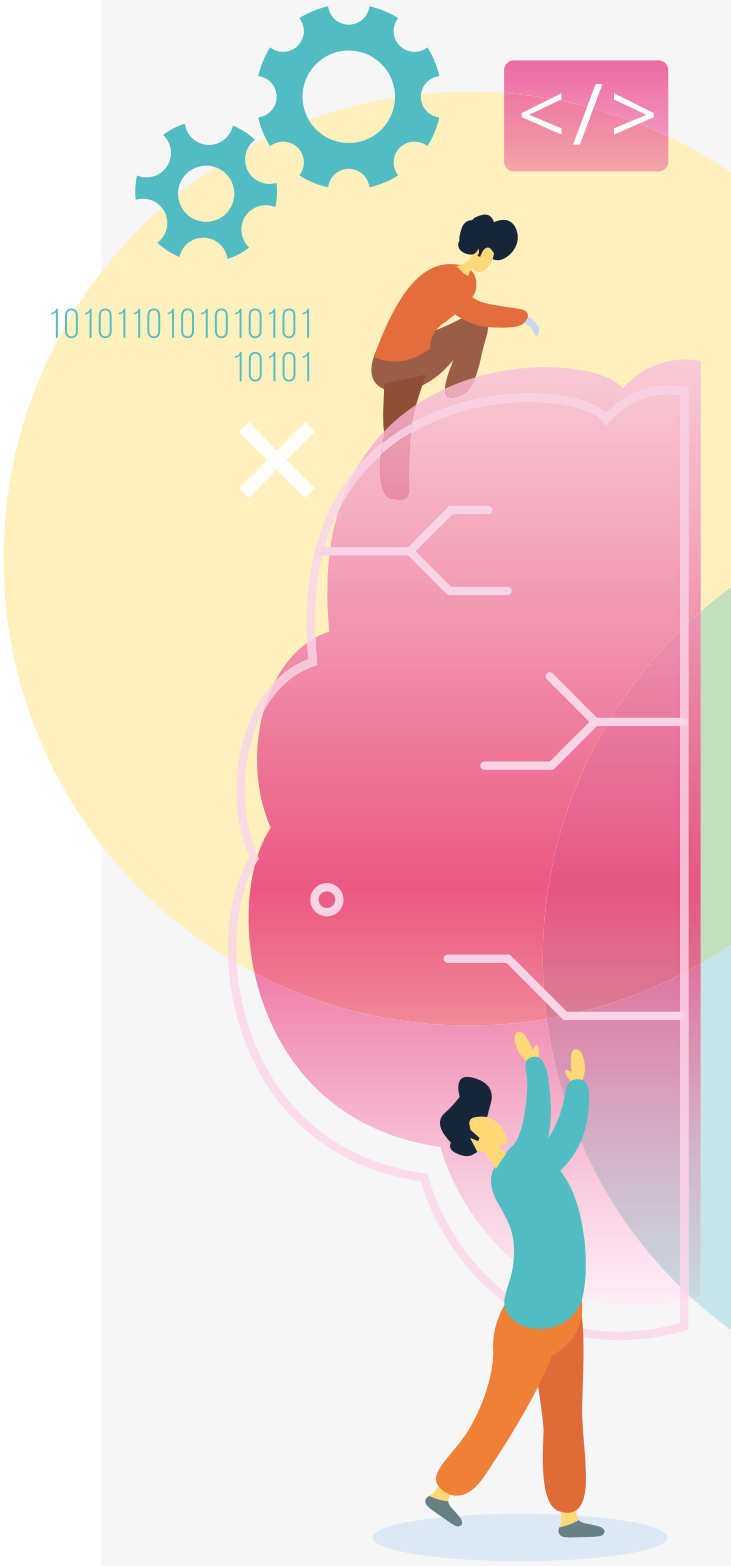
Today I would like to refer to this Tester Mindset and its importance when it comes to the professionalism and quality of the work coming from the Testing and QA departments.

The old reality we had in the ‘90s and early 2000s, where development and testing functions were clearly separated in Software organizations, was transformed: With the incorporation of Agile techniques we have seen how this separation became much more blurred and unclear.

Contrary to what the Agile / Scrum concepts indicate (a team where “everyone does everything”), we actually continue to see Scrum teams where part of their members participate almost exclusively in development tasks and the other part of the team in testing tasks.

Even leaving aside the adoption or not of Agile techniques, we continue to see Software development teams with members dedicated to testing tasks. In some cases, disguised as help to create ULTs (Unit Level Testing) but in other cases they simply fulfill QA and testing functions within the development teams.

Half a year ago, I assumed the position of Manager of Embedded Software Validation and Testing. The group, spread over 4 different geographic locations, had recently become independent as a separate team from the development team. Although I was facing a group of professionals of different seniority and experience, with high motivation and determination to start a new path of excellence, in reality it was a group (or rather, a group of individual people) without strategy, without guide, without identity.



For years, no one could (Could? Knew how to?) invest in increasing the professionalism of this important group of people who played a fundamental role within the organization. A group that knew how to receive a document with Requirements and Specifications and write a test plan (even automated!), but did not know basic concepts such as negative tests, stress and stability tests, performance tests, etc. Even more. Many of them had never known concepts like “Exploratory Testing”, “Pair Testing” or other techniques commonly used in the industry.

But the most surprising thing was the total lack of “Tester Mindset”: It is really necessary to ask the developer for permission to open an incident (bug) in the system? Are we able to claim the correction of the specification documents because they do not correctly describe the product requirements? Is it correct to detect a change in the product code just because the last run of the regression cycle failed (or would it be better to demand better communication from the developers to the testing group)? Is it correct to distribute the latest version of the code, even without having been tested? Like these, countless other questions existed.

Fortunately, and as a result of hard work, all these attitudes are changing. We have a long way to go but the first results are already in sight.

For me, this was just another example that shows that, in my understanding, there is no replacement to having independent testing and QA teams where its members (from the most experienced managers to the most novice engineers) breathe this unique air called Testing . It is the only way to ensure maximum professionalism and the best performance of the testing team.

Guarantee the security of our systems following these tips from the Spanish National Police Corps

Carlos Loureiro
National Police Corps, Spain

The Remote Desktop Protocol (RDP) was included by Microsoft for the first time in 1996 and since then it is the mostly used to establish remote communication with Windows computers.

This protocol allows not only remote access to office equipment but also to provide technical support without the need for travel. On the other hand, its easy and fast implementation caused a boom for teleworking during COVID-19 lockdown.

We must bear in mind that although large companies had previously implemented solutions, this did not happen with the vast majority of entities and other services. They had to face this need for the first time, and, as we have previously indicated, due to the characteristics of this protocol (easy and fast implementation) it was the solution they mostly opted for.

Since the appearance of the RDP, it has been the target of cybercriminals and its rapid implementation led to a notable increase in attacks by cybercriminals against them. A basic configuration of these services leaves them exposed on the internet, so an attacker could compromise it and thus have access to the network in which it is located.

Thus, for this attack vector, the methodology would be based on exploiting vulnerabilities in said protocol or carrying out brute force attacks against it.
This situation is nothing new if we remember that in 2018 the Dharma ransomware was already exploiting different vulnerabilities in said protocol as an infection vector.

We can have as an example the attack carried out by the SamSam group which encrypted around 8,900 computers in LabCorp using ransomware after carrying out a brute force attack against the RDP server. Thus, we observe that the profits of cybercriminals are quite high if we take into account that the requested ransom does not fall below 40,000 euros.

In Spain we find about 32,111 computers exposed on the Internet according to the Shodan search engine using the default communication port. It should be noted that on the 19th March (this year) a peak of this type of attack was reached in Spain using brute force out of a total of 1,332,796 according to Kaspersky sources.

Therefore, the most basic keys that we must not forget in order to protect our equipment are to have a recommended minimum configuration that must always include:

- do not expose these services to the internet
- use of strong passwords
- update these services in order to reduce their vulnerabilities.

QA&TEST 2020: a virtual edition focused on (cyber)security

As we all know, health circumstances have forced us to professionally reinvent ourselves in many aspects. This has also been the case for conferences such as QA&TEST, the international conference dedicated to Software Testing and Quality Assurance for embedded systems which, during its 18 face-to-face editions in Bilbao, has always rewarded interaction, contact and facilitated networking possibilities so the learning experience at the conference was as complete as possible, allowing attendees to return to their companies with knowledge of new techniques and tools and very useful contacts for the future.

This 2020 edition of QATEST is going to be, of course, different, being the first time it will be held virtually but always guaranteeing the most complete experience in terms of training and networking. Presentations will live and we'll have plenty of time for debate, both general debate, among all the attendees and speakers, and private meetings to solve specific doubts.

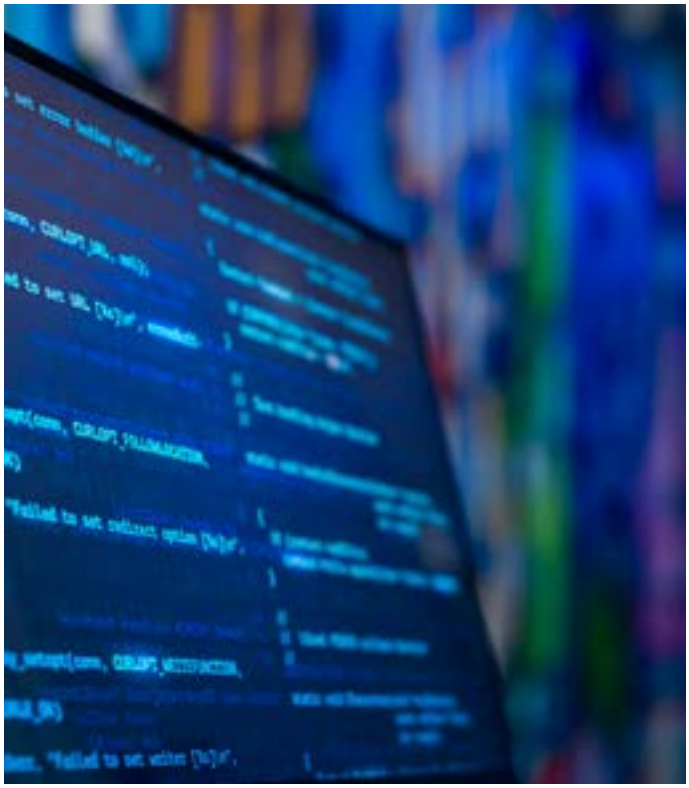
In this year 2020 it is estimated that more than fifty percent of the world's population is connected in cyberspace making transactions, purchases, communications... We use online banking, digital administration, social networks, email etc ... and, of course, within this already everyday-life, we cannot forget the growing presence of the Internet of Things and Artificial Intelligence through all our interconnected smart devices.

This virtual reality in which we also live is huge and very complex and, of course, it is not without risks. In this interconnected paradigm, cybersecurity has a very high impact and developing secure software will be the basis to guarantee the reduction of attacks and the economic and other losses that they could entail.

Cybersecurity is, therefore, a top priority issue for all companies and we have reflected this in the QATEST programme. There are numerous methodologies, techniques, tools and standards that will allow us to address, in the most effective way, the security of our systems from the moment of their conception or design and until its production readiness.

Aware of the importance of this aspect, the 19th edition of QA & TEST will dedicate a full day to security from two different points of view: On the one hand, we will learn how to ensure that a system meets the security requirements for which it has been designed (track Security Testing) and, on the other, how to manage security from the conception of a system (Security Management track).

This way, we will start the day with a presentation by **Koen Yskout** entitled **"What is a security vulnerability?"**, We will also treat DevSecOps in a very complete way, guaranteeing that attendees are prepared to implement it in their organizations. We will have two presentations: In the first one, entitled **"Moving from DevOops to DevSecOps - Practical integration of Security-by -Design"**, Edward van Deursen, from the Securesult company, will give us the keys to guarantee security from the design or conception of a system. Next, John Erik Horn, from the German company BDO Cyber Security, will present his talk: **Pimp your Pipeline**.



In the afternoon, we will enjoy the Security Testing track that will begin with **Kamil Medzikowski**, from the Intel company, showing us in his talk **"Proximity cards under the magnifying glass"**, how to guarantee the security of the different types of cards that companies use as an access system security control and prevent them from being hacked. We will continue listening to Lennert Wouters, from KU Leuven University in Belgium, who, in his talk entitled **"Threats and Security for Embedded Devices"**, will talk about the specificities of embedded systems, adopting a practical approach to address the security of this type of device, taking into account that the threat model in this type of systems is significantly different from that of other information systems.

We will end the day with a final session where we will analyze the gray box tests of electronic components for automotive with Andreea-Ina Radu, from the Center for Cybersecurity and Privacy at the University of Birmingham and his talk **"Gray-box Analysis and Testing of Automotive Electronic Components"**.

We recommend you:

This section aims to be a living one, a section that grows and accommodates the numerous initiatives in the world of testing and QA that are being developed in the world and that may be of interest to our community.

We are looking forward to reading your proposals and sharing them in this publication!

Send your reco mmendation

Ali Khalid's LinkedIn

If you found interesting the article of Ali, we recommend you to follow him in **LinkedIn**.

His posts, which are very useful for the testing community, also generate dynamic debates.

We encourage you to read and participate!

Derk-Jan de Grood's blog

What happens when Agile and Testing meet? Derk-Jan de Grood, author of several successful books and well known speaker at QA&TEST and many national and international conferences, reflects on how do they proceed their journey together.

You will also find information about events, webinars and training.

Discover the blog

HEADQUARTERS
SQS S.A.
Avda. Zugazarte 8 - 1º6
48930 Getxo
Vizcaya - Spain
Tel.: +34 94 480 46 17
email: info@sqs.es

Follow us online

Don't miss the latest news from SQS and follow us on our social media.
www.sqs.es

@sqspain @sqspain SQS