

# THE TESTER

## SIGiST Summer Conference Friday 22<sup>nd</sup> June 2018

63

### IN THIS ISSUE

#### Conference

Agenda .....2

Venue .....6

Abstracts.....7

#### Articles

E2E API Automation  
Testing Hybrid  
Solution..... 13

Dev / Ops for  
Testers ..... 22

#### Features

Fifteen minutes with  
..... 29

### From the Editor

Welcome to the *SIGiST* and *The Tester*. Following our recent member survey, the aim of the June programme is to cover as many of the highest rated topics as possible, presented by excellent speakers. All eight of the highest-rated topics from the member survey are covered in *six keynote* presentations. Check out the abstracts covering Testing topics ranging from the Internet of Things to autonomous cars and bitcoin!

Most of our speakers for June are 'top-rated'. Several have been recognized for being the best speaker at international conferences, and most have been awarded for their contribution to the software testing industry.

In *The Tester* review the abstracts for June, but also read articles on API testing and DevOps, and read an interview which could help with your career.

The committee are grateful for the contributions from the sponsors of the SIGiST.

Phill Isles  
The Tester Editor  
[phill.isles@bcs.org](mailto:phill.isles@bcs.org)

### Conference Booking Instructions

To register online, please use the link below, or scan the QR code with your smart device. The booking system accepts multiple and third-party bookings.

[Book Now!](#)



Join our Linked-In Group:



[LinkedIn/BCS SIGiST](#)



Follow us [@SIGiST](#)



# Conference Agenda

**BCS SIGiST – Summer 2018 Conference – Friday 22<sup>nd</sup> June 2018  
BCS 1st Floor, Davidson Building, 5 Southampton Street, LONDON. WC2E 7HA.**

*Keynote Day – Six popular speakers covering the most requested topics*

Time	Session
09:00	Registration
09:40	<b>Welcome – Stuart Reid, Chair, SIGiST</b>
09:45	<b>Agile Test Leadership - Helping the Team put Quality First</b> Geoff Thompson, Planit Testing
10:35	<b>Networking Session</b> Phil Isles, Editor of The Tester, SIGiST Committee
10:50	Coffee, Tea & Refreshments
11:10	<b>Blunders in Test Automation</b> Dot Graham, Independent Consultant
12:00	<b>How to be a Better Tester? Make Things!</b> Graham Thomas, Independent Consultant
12:50	<b>Lunch in networking area – Videos and Vendor presentations in main room</b>
14:00	<b>Testing Self-Learning Systems for Autonomous Cars</b> Stuart Reid, STA Consulting
14:50	<b>Understanding the Test and Risk in Bitcoin</b> Declan O’Riordan
15:40	Coffee, Tea & Refreshments
16:10	<b>Devices and Desires: as humans how do we experience software?</b> Isabel Evans, Independent Consultant
17:00	<b>Close of the day meeting – Stuart Reid, Chair, SIGiST</b>

The SIGiST committee reserve the right to amend the programme if circumstances deem it necessary.

# Conference Sponsors

## Ten10

Our expert-led, flexible and scalable approach to testing enables our clients to embrace technological transformation with absolute confidence.

Specialisms include:

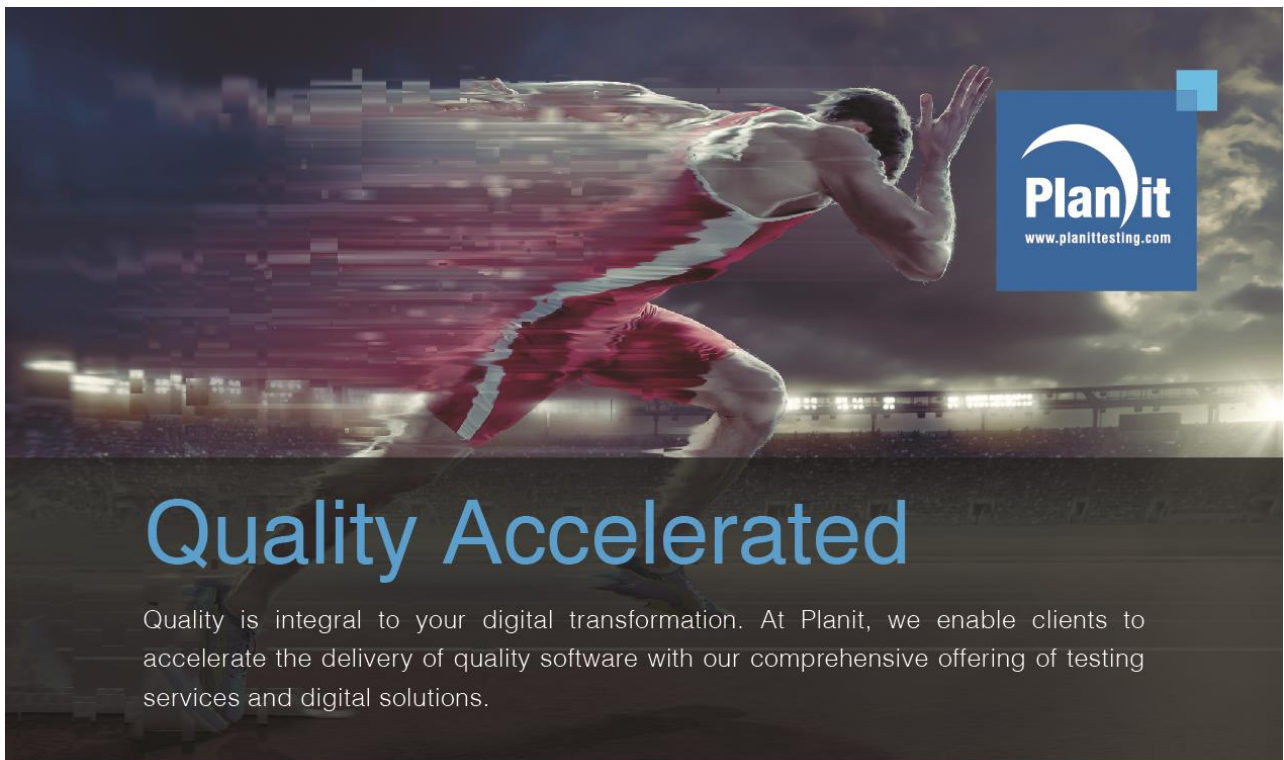
- Test strategy
- Functional testing
- Automated testing
- Performance testing
- Mobile testing
- Agile testing
- Accessibility testing



Call +44 (0) 203 697 1444 or email [contact@ten10.com](mailto:contact@ten10.com) for an initial chat to find out how Ten10 can help with your testing challenges.

Offices in Brighton,  
Leeds and London  
[www.ten10.com](http://www.ten10.com)

# Conference Sponsors



**Planit**  
www.planittesting.com

## Quality Accelerated

Quality is integral to your digital transformation. At Planit, we enable clients to accelerate the delivery of quality software with our comprehensive offering of testing services and digital solutions.

As world leaders in testing, we have built a strong client-base through our thought leadership, unique cloud-based offerings, responsiveness, excellent customer service, guarantee of testing efficiencies and proven track-record of delivering quality outcomes. We can help you:

- **Accelerate Quality Outcomes:** Realise a positive ROI sooner and increase your speed to market with our best-in-class consultancy, services and solutions.
- **Enhance Customer Experience:** Delight your customers by optimising system quality and performance to provide a smooth, speedy and seamless user experience.
- **Discover Innovations & Efficiencies:** Through continuous improvement and constant innovation we are able to find ever improved solutions to our clients' challenges, achieving quality, efficiencies and cost savings for every project.

With over 1,300 highly skilled testing and digital specialists, we have the right people for your project. We have experts in the following areas:

Functional Testing	Test Automation	Performance Testing	Mobile & App Testing	Service Virtualization	Tools & Cloud Solutions	DevOps Solutions	Agile Services	Training & Coaching
Website Testing	Accessibility Testing	Integration Testing	Compatibility Testing	Acceptance Testing	Security Testing	<b>ENSURE QUALITY FOR YOUR PROJECT</b>		
Test Management	Process Improvement	Test Strategy & Planning	SAP Testing	Application Monitoring	Operational Testing	On-site Testing	Testing Consultancy	Managed Services

+44 203 457 5020 infouk@planittesting.com www.planittesting.com

# Conference Sponsors



## Qsome

Fast Forward Your Software QA

<http://www.qsometech.com/>

Contact Us For Software Testing Services & Testing Tools

Global

[solutions@qsometech.com](mailto:solutions@qsometech.com)

+61 8 8312 1287

Australia

Tel: 1300 28 44 92

L28, 303 Collins Street, Melbourne, VIC 3000

India

+91 9845 00 86 96

201, Green Glen Layout, Bellandur, Bangalore - 560103

United Kingdom

+44 20 3769 2460

# SIGiST Conference Venue

The June 2018 SIGiST conference will be held at the BCS London office.  
Travel details and location below.

## London office guide

### How to get to the BCS London office

**First Floor**  
The Davidson Building  
5 Southampton Street  
London WC2E 7HA  
Telephone 01793 417666

These area and inset maps have been simplified in the interests of ease of understanding. Not all roads are shown. The inset map below is more accurate.

Access by car is very difficult due to the local one-way system. There are no car parking facilities at BCS London. The nearest car park is located on Drury Lane, Parker Street, Parker Mews, London, WC2B 5NT.

The rear door in Exeter Street is to be used for deliveries only and is normally locked.

The main entrance is fully accessible to wheelchair users and should be used by all staff and visitors.

On arrival, report to the Davidson Building Reception who will direct you to the first floor.

### Travel tips from major London stations

**Charing Cross** – 6 minutes walk

**Waterloo** – 12 minutes walk across Waterloo Bridge, or buses 139 or 176 to Stop **C**

**London Bridge** – onward rail link to Charing Cross

**Kings Cross or St Pancras** – Piccadilly Line to Covent Garden tube, or bus 91 to Stop **B**

**Euston** – West End Branch of Northern Line to Charing Cross, or bus 91 to Stop **B**

**Victoria** (rail and coach stations) – Circle Line to Embankment, but the most direct journey is via bus 11 to Stop **A**

**Paddington** – Circle Line to Embankment or Temple, Bakerloo Line to Charing Cross or buses 15 or 23 to Stop **A**

**Liverpool St** – Circle Line to Embankment or Temple, or buses 11 or 23 westbound

**Fenchurch St** – Walk to Tower Hill, then District or Circle to Embankment



---

# Presentation Abstracts and Speaker Biographies

## Keynote One

### Geoff Thompson, Planit Testing

#### Agile Test Leadership - Helping the Team put Quality First

Agile methods put a strong emphasis on quality and getting the team focused on delivering a quality product that the end user wants. But how do we support the team to enable them to do this? Part of that role falls to the Agile Test Leader. This is a person who will foster self-managing teams while mentoring and coaching them so that they can deliver work of the highest quality.

In this presentation we will look at the testing and quality activities that take place in an Agile project and examine the ways that an Agile Test Leader can help drive these initiatives forward.

***Geoff** is the UK Director of Testing Services for Planit Testing, part of the global Planit Testing group. He has a real passion for software testing, test management and process improvement. He is a founder member of the International Software Testing Qualification Board (ISTQB), the TMMi Foundation, and the UK Testing Board and is currently the Vice President of the ISTQB and Chairman of the UK Testing Board.*

*He co-authored the BCS book Software Testing - An ISEB/ISTQB foundation and is a recognized international speaker, keynoting in many conferences, and was the chair of EuroSTAR 2011.*

## Keynote Two

# Dot Graham, Independent Consultant

## Blunders in Test Automation

In chess, the word blunder means a very bad move by someone who should know better. Even though system-level functional test automation has been around for a long time, people still make some very bad moves and serious blunders. The most common misconception in automation is thinking that manual testing is the same as automated testing. And this thinking accounts for most of the blunders in system level test automation.

Dorothy Graham takes us on a tour of these blunders, including: Testing-Tools-Test, Silver Bullet, Automating the Wrong Thing, Who Needs GPS, How Hard Can It Be, the Stable-Application Myth, Inside-the-Box Thinking, the Project / Non-Project Dilemma, and Isolationism. Different skills, approaches, and objectives are needed, or you'll end up with inefficient automation, high maintenance costs, and wasted effort.

Join Dot to discover how you can avoid these common blunders and achieve valuable system-level test automation.

*Dorothy Graham has been in software testing for over 40 years, and is co-author of 4 books: [Software Inspection](#), [Software Test Automation](#), [Foundations of Software Testing](#) and [Experiences of Test Automation](#). With Seretta Gamba, she is working on the wiki [TestAutomationPatterns.org](https://www.testautomationpatterns.org)*

*Dot has been on the boards of conferences and publications in software testing, including programme chair for EuroSTAR (twice). She was a founder member of the ISEB Software Testing Board and helped develop the first ISTQB Foundation Syllabus. She is a popular and entertaining speaker at conference and events worldwide. She has been attending SIGIST meetings since 1989. She was awarded the European Excellence Award in Software Testing in 1999 and the first ISTQB Excellence Award in 2012.*



## Keynote Three

# Graham Thomas, Independent Consultant

### How to be a Better Tester? Make Things!

In this session I want to explore how to become a better software tester by making things. Small computer powered devices to be specific.

How does this make me a better tester? Well I believe that by making things you will gain a deeper insight into how to test them. And maybe learn a few things about yourself along the way.

I will bring along several small devices I have made, and we can have a go at testing them. Some of the devices will be IoT (Internet of Things) capable. All of them will be based around a Raspberry Pi computer. I will share some of the experiences I had building them, highlight where I learnt valuable testing lessons, and what I also learnt about myself in doing this.

And, if making things really doesn't make you a better tester, then at least you will have some fun!

Warning, this session will be highly interactive!

***Graham** currently works in two key areas of software testing; program test management and testing change & transformation. His current focus is on helping testers and the organisations he works with in transitioning to more agile ways of working.*

*Graham has extensive experience in IT across a number of industry sectors including; Finance (Banking, Treasury & Insurance), Utilities and Retail. This has been gained in software house, consultancy and end user environments. He has specialised in software testing since the early 1990s. And for the last 6 years Graham has had loads of fun making small devices based on the Raspberry Pi computer.*

## Keynote Four

# Stuart Reid, STA Consulting

## Testing Self-Learning Systems for Autonomous Cars

The testing of traditional (deterministic) safety-related systems is well-understood, but the advent of autonomous vehicles has introduced the need for Artificial Intelligence in the form of self-learning systems. These systems are creating new challenges for testers, both for the vendors of autonomous systems and for the regulators. This presentation explores four major challenges. The process of creating a self-learning system opens up new opportunities for errors and defects, which testers and quality assurers must both consider. The complexity of self-learning systems has meant that most testing has been black-box, but their criticality when used for autonomous vehicles means that white-box testing must also now be considered. The difficulties of assuring these systems and the balance of costs between development and testing means that developers must consider the architectures needed to make the testing and assurance possible. Finally, the costs inherent in testing probabilistic systems will mean that virtual testing will have to play a far more prominent role.

*Stuart is a software testing consultant with over 35 years' experience in the IT industry, working in development, testing and education. Application areas range from safety-critical to financial and media. Stuart supports the worldwide testing community in a number of roles. He is convener of the ISO Software Testing Working Group, which has already published several software testing standards and is currently developing new standards in the areas of Performance and Automotive Testing. He is also leading the ISO Software & Systems Study Group on the standardization of Autonomous Systems. He founded the International Software Testing Qualifications Board (ISTQB) to promote software testing qualifications globally.*

## Keynote Five

# Declan O'Riordan

## Understanding the Test and Risk in Bitcoin

The term proof of work was adopted in 2008 by 'Satoshi Nakamoto' in the Bitcoin cryptocurrency proposal. Proof of work is a test, and understanding this test underpins any attempt to adopt blockchain technology in all variations.

There are many risks to be considered, but the reductionist 'impact x likelihood' pseudo-calculation has multiple limitations. Complex, opaque, non-linear, and subtle risks don't fit abstract labels, simplified matrices, or linear scales. Time undermines fixed-state approaches that are unable to amplify and dampen feedback, yet appear deterministic. Let's take a fresh, deeper look at risk assessment using Bitcoin as an example.

*Declan is an NCSC certified Security & Information Risk Advisor. In 2014, he won the EuroSTAR prize for best conference paper and his debut talk was voted the 'do over' session. In 2015 Declan won the best conference paper prize at STAREAST and was on the EuroSTAR programme committee. In 2016 Declan was a joint EuroSTAR best paper prize winner and implemented the UK's first real-time Interactive Application Security Testing solution. He also built his own house that year! In 2017 Declan was co-chair of UKSTAR and a contributor to the EuroSTAR 'Little Book of Testing Wisdom'.*

## Keynote Six

# Isabel Evans, Independent Consultant

### Devices and Desires: as humans how do we experience software?

We consume and still we desire more. More devices, more apps, more data, more bandwidth, more connectivity. The more we have, the more we want... We assume that to be true – those of us who work in the software industry. But is that true? To understand what is really required of our products, we need to design and test a pyramid of interlocking quality attributes, that build together to make an optimum experience for the people who use our products, matching their needs, and their desires. It is not enough to test functional suitability, performance, and reliability. People also require usability, accessibility, and safety. These profoundly affect the user experience (UX) which also considers trust, flow, and excitement. To test the UX, and so deliver a good experience for people, we first need to understand them.

Isabel discusses why and how we can test and deliver better UX. She will look at the pressures from the business and government environment, the commercial imperatives, and the needs of society, along with the needs of the individual software user.

*Independent quality and testing consultant **Isabel Evans** has more than thirty years of IT experience in the financial, communications, and software sectors. Her work focuses on quality management, software testing, and user experience (UX). A published author, popular speaker and storyteller at software conferences worldwide, Isabel is a Chartered IT Professional and Fellow of the British Computer Society, and received the 2017 EuroSTAR Testing Excellence Award. In parallel with her consultancy and teaching in industry, Isabel Evans has recently started as a part-time PhD student at the Department of Computer Information Systems, University of Malta, working with Dr Chris Porter and Dr Mark Micallef on research in human factors for Software Testing. Within that, her current research project is to examine human factors around test tools and the automation of testing, in particular, the UX of test tools for testers.*

# E2E API Automation Testing Hybrid Solution

Sandeep Panghal

Test Automation Engineering Associate Manager at Accenture, United Kingdom

## Abstract

API stands for Application Programming Interface. APIs have become the primary channel for reaching application users. Social media companies like Facebook, Twitter and e-commerce giants like Amazon typically serve more user requests through API calls than through the front ends. As a result, API quality is becoming increasingly important to companies and their customers and subsequently thorough API testing is crucial.

API testing can often be challenging and requires highly skilled and trained testers who can write code to test specific API functions and can understand complex variables in the desired outputs.

Also, poor API testing can lead to longer development cycles and higher defect rates which can increase overall project costs exponentially.

To address these challenges, there is a need of highly reliable, easy-to-use, automated end to end solutions to perform functional, performance and security testing of APIs.

**API Automation Testing Hybrid Solution** has been designed to meet this requirement with a focus on three critical testing areas – Functional, Performance and Security. In this whitepaper, high-level design and architecture of hybrid solution has been discussed along with some common myths about APIs and how this solution stands when compared with existing tools in the market.

## Introduction

In today's globalised world where same applications are being used across the globe through localization, most enterprises have recognized the need and power of API automation testing and are looking to leverage automated testing solutions for validating end-to-end web-based and mobile applications.

**API Automation Testing Hybrid Solution** provides a scalable and flexible automated testing service aligned to CI/CD environment in the NEW IT world. There are three types of API Automation Testing which have been discussed as part of this solution to cover End to End API Testing:

- API Functional Testing
- API Security Testing
- API Performance Testing

An API, once ready for testing, must undergo through various test phases before deployment to LIVE:



\* The logos used in this image refers to marks owned by third parties. All such third-party marks are the property of their respective owners. No sponsorship, endorsement or approval of this content by the owners of such marks is intended, expressed or implied.

Additionally, there are lot of confusion floating around the globe about API Testing and Unit Testing. Below table describes some high-level differences between API Automation Testing and Unit Testing.

API Automation Testing	Unit Testing
<ul style="list-style-type: none"> <li>• Testers Perform it</li> </ul>	<ul style="list-style-type: none"> <li>• Developers perform it</li> </ul>
<ul style="list-style-type: none"> <li>• End to end functionality has been tested</li> </ul>	<ul style="list-style-type: none"> <li>• Separate functionality is created</li> </ul>
<ul style="list-style-type: none"> <li>• Testers cannot access the source code</li> </ul>	<ul style="list-style-type: none"> <li>• Developer can access the source code</li> </ul>
<ul style="list-style-type: none"> <li>• Only API functions are tested</li> </ul>	<ul style="list-style-type: none"> <li>• UI testing is also involved</li> </ul>
<ul style="list-style-type: none"> <li>• All functional issues are tested</li> </ul>	<ul style="list-style-type: none"> <li>• Only basic functionalities are created</li> </ul>
<ul style="list-style-type: none"> <li>• Broader in scope</li> </ul>	<ul style="list-style-type: none"> <li>• Limited in scope</li> </ul>
<ul style="list-style-type: none"> <li>• Ran after build is created</li> </ul>	<ul style="list-style-type: none"> <li>• Usually ran before check-in</li> </ul>

## High-Level Design - E2E API Automation Testing

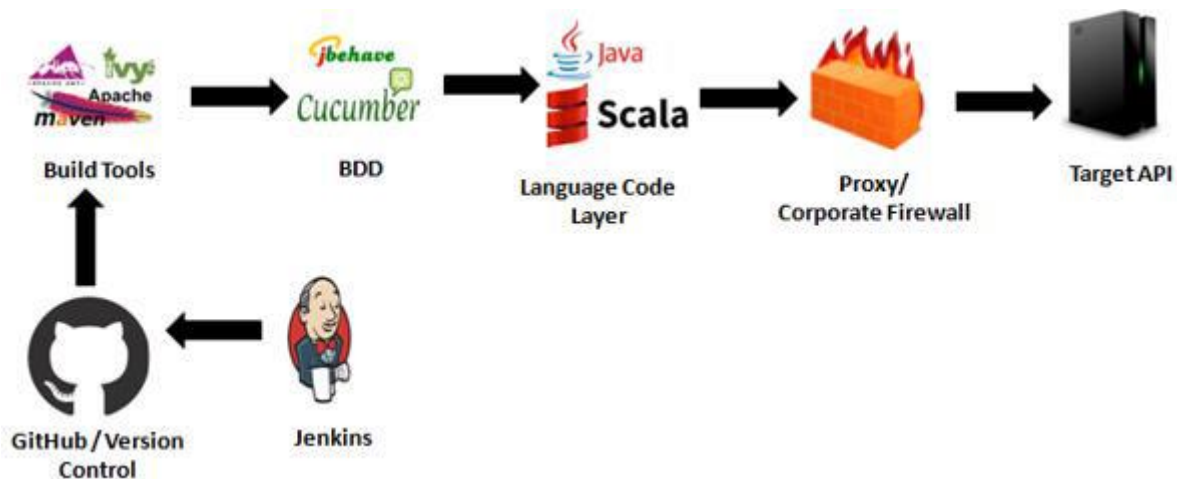
In this section, high-level design of proposed API Automation Testing Solution has been discussed with respect to three flavours – Functional, Performance and Security.

Some of the prominent features of below designs are:

- Supports Agile / Behaviour Driven Development(BDD) delivery approach with continuous/iterative regression test activities.
- Fully automated API functional testing based on acceptance criteria defined in each test (*i.e.* features and scenarios defined in Cucumber).
- Granular control over scope to allow individual tests and test suites to be included in an execution cycle, based on tags defined in Cucumber.
- Test Suite execution can be triggered either manually via IntelliJ / Eclipse IDE, command line or Jenkins UI or triggered as part of a CI/CD delivery process using Jenkins jobs.
- Graphical and tabular reporting with detailed results at Cucumber feature, scenario and test step level.
- Version control via GitHub, *etc.*

### HLD - E2E API Automation Testing (Functional)

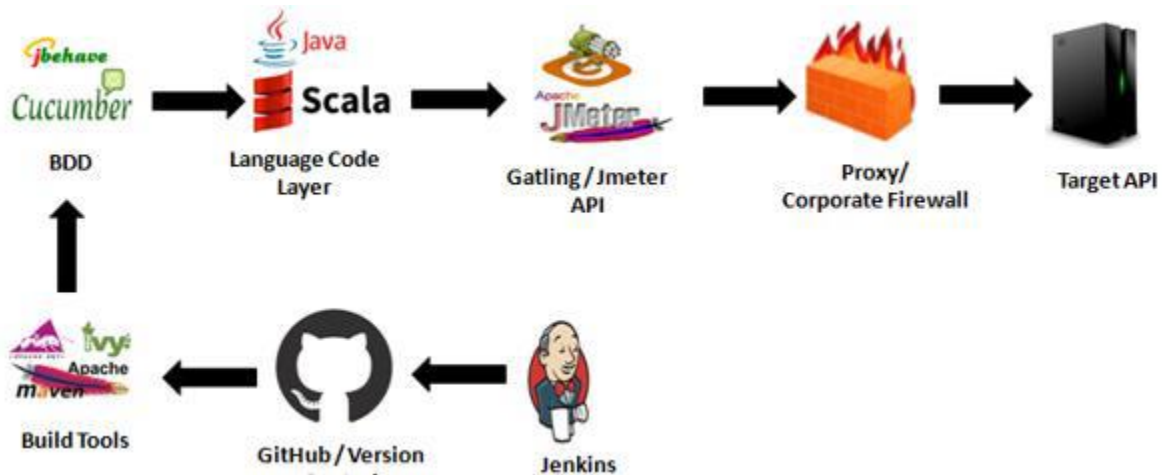
API can be tested functionally by sending requests and receiving responses using Java/Scala Framework along with testing tools like Cucumber, which in turn is integrated with other tools as displayed in the proposed design:



\* The logos used in this image refers to marks owned by third parties. All such third-party marks are the property of their respective owners. No sponsorship, endorsement or approval of this content by the owners of such marks is intended, expressed or implied.

### HLD - E2E API Automation Testing (Performance)

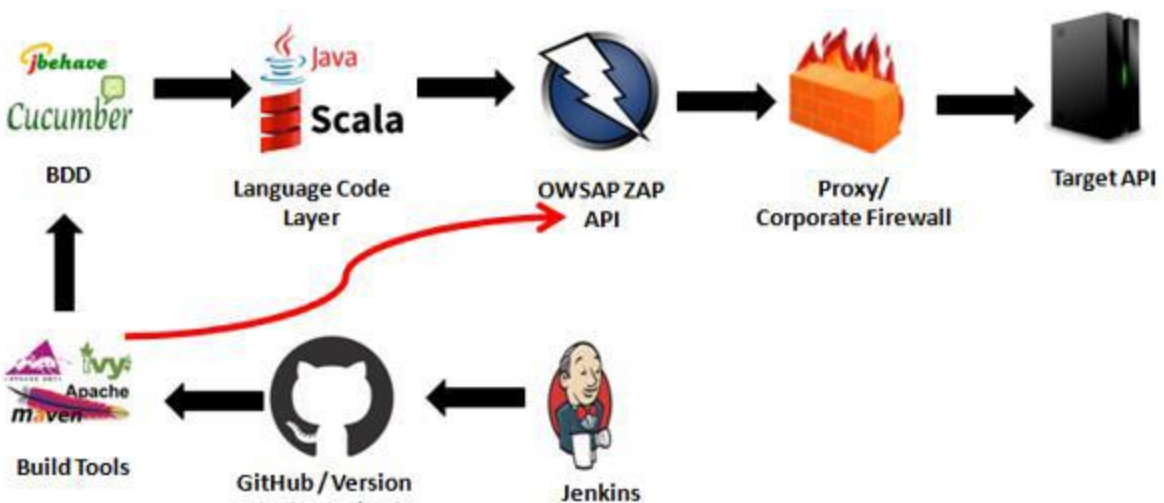
Performance testing can be defined as the technical investigation done to determine or validate the speed, scalability, and / or stability characteristics of the product under test. As displayed in the below proposed design, Gatling / JMeter has been integrated with other tools as part of solution.



\* The logos used in this image refers to marks owned by third parties. All such third-party marks are the property of their respective owners. No sponsorship, endorsement or approval of this content by the owners of such marks is intended, expressed or implied.

### HLD - E2E API Automation Testing (Security)

For security testing, OWSAP ZAP, a security testing tool has been integrated in the solution as displayed in the below proposed design.

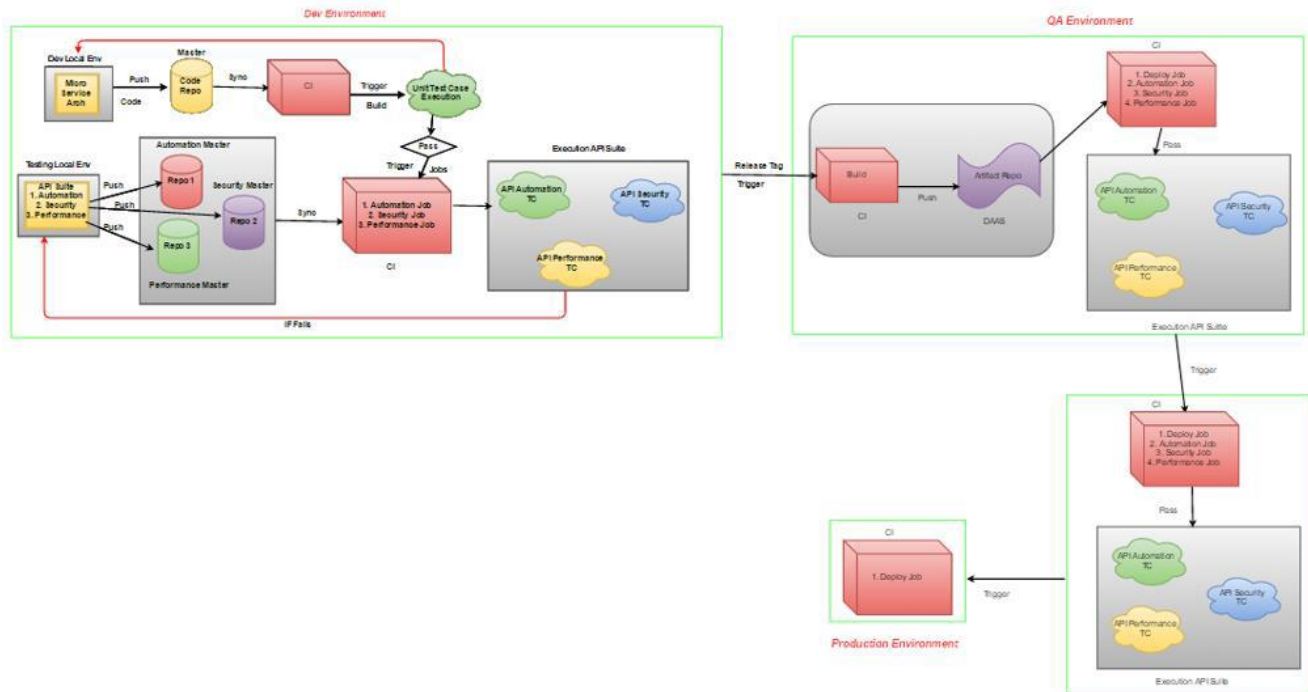


\* The logos used in this image refers to marks owned by third parties. All such third-party marks are the property of their respective owners. No sponsorship, endorsement or approval of this content by the owners of such marks is intended, expressed or implied.



## E2E API Automation Testing Hybrid Solution Architecture

In this section, detailed architecture hybrid solution has been displayed in the below figure covering all the three testing areas: Functional, Performance and Security testing.



### E2E API TESTING HYBRID SOLUTION WITH CI/CD

## Tools used for API Automation Testing

Some of the commonly used tools for API testing are:

### API Automation (Functional) Tools

- SOAPUI
- Postman with jetpacks / Newman
- Curl

### API Performance Testing Tool

- Gatling
- JMeter

### API Security Testing Tool

- OWSAP ZAP
- Burp Suite

## Why API Automation Testing Hybrid Solution over other API Automation Tools?

Now the question arises: “Why this solution over other existing tools in the market?” The below table describes advantages which this proposed solution offers over existing tools.

API Automation Tool Comparison Chart					
S. No	Parameters	API Automation Hybrid Solution(Proposed)	Soap UI	Postman/Newman	curl
1	License Type	NA	Proprietary	Proprietary	MIT/X derivate
2	Price	Free	€539 / Year / License (standard)*	\$6.10/user/month*	Free
3	Integrates with Version Control tools	Yes (GitHub)	Yes (GitHub)	No	Yes
4	Integrates with Continuous Integration (CI) tools	Yes (Jenkins)	Yes (Jenkins)	Yes (Jenkins)	No
5	Integrates with Build Tools	Yes (Maven / SBT)	Yes (Maven / Ant)	No	No
6	Supports Behaviour Driven Development(BDD)	Yes (Cucumber / Jbehave / Behat)	No	No	No
7	Integrates with Functional Scripts	Yes	No	No	No
8	Supports Continuous Deployment	Yes	Yes	No	No
9	Integrates with Selenium (Web based Automation)	Yes	No	No	No
10	Integrates with Appium (Mobile Automation)	Yes	No	No	No
12	Integrates with Travis CI	Yes	N/A	No	No
14	Supports Multiple Environments	Yes	Yes	Yes	No
15	Supports http/https Header Configuration	Yes	Yes	Yes	Yes
16	Supports End Points Configuration	Yes	Yes	Yes	Yes
17	Supports Ports Configuration	Yes	Yes	Yes	Yes
18	Supports Programming Languages	Java, Scala, Python, Ruby, Php	Groovy, JavaScript	Json	40+ languages
19	Can Test Rest APIs	Yes	Yes	Yes	Yes
20	Can Test SOAP APIs	Yes	Yes	Yes	Yes
21	Can validate Response (Code/Body)	Yes	Yes	Yes	Yes
22	OS Supported	Linux, OX, Windows (10, 8, 7, XP)	Window, Linux, OS	Chrome Extension (OS Independent)	Window, Linux, OS
23	Test Execution Report	HTML	HTML	HTML	No
24	Supports Data Sources	Excel, CSV, XML, JSON, JDBC	Excel, XML, JDBC	CSV, JSON	No
35	Supports Command Line Interface	Yes (Shell Script, BAT File)	Yes	Yes	Yes
26	Can Integrate with Security Testing Framework	Yes	Yes	No	No
27	Can Integrate with Performance Testing Framework	Yes	Yes	No	No
28	Scalable	Yes	No	No	No

\* The price mentioned in the table is publicly mentioned on their official website.

## Some common Bugs that API Testing uncovers

- Fails to handle error conditions gracefully
- Unused flags
- Missing or duplicate functionality
- Reliability issues such as difficulty in connecting and getting a response from API
- Security Issues
- Multi-threading issues
- Performance Issues. API response time is very high
- Improper errors/warning to caller
- Incorrect handling of valid argument values
- Response Data is not structured correctly (JSON or XML)

## Some Common Mistakes / Myths about APIs

- Assumption that all functions of an API will always produce expected results once tested
- Poor documentation of API code which doesn't provide low level detail such as behaviour with third parties
- Incorrect anticipation of scaling of APIs with respect to business processes
- Testing of an API includes interaction with the code so testers feel reluctant to do testing
- Ignoring the response time assertions
- Assuming that GUI tests will validate all the API functions so no specific API testing required
- Ignoring the fact that APIs are a core element of successful web / mobile application
- End-to-end tests of APIs are not feasible due to too many dependencies

## Why market shift towards API Automation?

Some of the commonly used tools for API testing are:

Old Market Needs	Current Market Needs
<ul style="list-style-type: none"> <li>• Waterfall Model</li> </ul>	<ul style="list-style-type: none"> <li>• Agile methodology</li> </ul>
<ul style="list-style-type: none"> <li>• Extended release cycles with slow feedback loops: Typically, 3-12 months</li> </ul>	<ul style="list-style-type: none"> <li>• Short release cycles with fast feedback loops: Typically, 2-4 weeks in agile</li> </ul>
<ul style="list-style-type: none"> <li>• Client is less demanding</li> </ul>	<ul style="list-style-type: none"> <li>• Client is more demanding</li> </ul>
<ul style="list-style-type: none"> <li>• Code resides on the presentation layer</li> </ul>	<ul style="list-style-type: none"> <li>• All the business logic in Business Layer</li> </ul>
<ul style="list-style-type: none"> <li>• GUI does not change so frequently</li> </ul>	<ul style="list-style-type: none"> <li>• GUI changes very frequently every sprint</li> </ul>
<ul style="list-style-type: none"> <li>• Service Oriented Architecture</li> </ul>	<ul style="list-style-type: none"> <li>• Micro Services Architecture</li> </ul>
<ul style="list-style-type: none"> <li>• System complexity is less</li> </ul>	<ul style="list-style-type: none"> <li>• System complexity is more</li> </ul>
<ul style="list-style-type: none"> <li>• Tough to maintain and time-consuming GUI test suite</li> </ul>	<ul style="list-style-type: none"> <li>• Easy to maintain API test suite</li> </ul>

<ul style="list-style-type: none"> <li>• GUI test suite takes more time to execute</li> </ul>	<ul style="list-style-type: none"> <li>• API test suite takes very less time to execute and is more reliable in comparison to web test suite</li> </ul>
<ul style="list-style-type: none"> <li>• All functionality can't be tested using GUI testing</li> </ul>	<ul style="list-style-type: none"> <li>• Creating automated API testing will help you to test all the functional paths without the complications introduced when going through a GUI</li> </ul>
<ul style="list-style-type: none"> <li>• Less dependent on third party system / service or API</li> </ul>	<ul style="list-style-type: none"> <li>• More dependent on third party system/service or API</li> </ul>

### Success Factors (API Automation Testing Hybrid Solution)

- Can reduce time significantly for testing cycles thus a reduction in costs
- Robust and extensible framework
- Faster speed to market
- Can reduce dependencies on automation experts (supports BDD)
- Early evaluation of application builds before running GUI tests
- Easily integrated with GUI automation tools like Selenium, Appium
- Can be used extensively in agile projects supporting continuous deployment and other CI tools like Jenkins.
- Can reduce overall risks and increases efficiency

### Conclusion

APIs play a crucial role in the success of any mobile app or web application but testing APIs end to end has become a bottleneck in the delivery lifecycle. To remedy this clients/companies are looking for a complete solution that can overcome some of the core challenges faced when testing APIs.

The **API Automation Testing Hybrid Solution** provides a scalable and flexible automated testing service aligned to continuous integration and continuous development methodologies. It includes a robust infrastructure/solution to carry out Functional, Security, and Performance testing of APIs. A3HS has been designed for easy use by non-technical testers, reducing the dependency of specialist resources and skills.

The solution is highly automated to reduce the overhead of manually intensive testing and to allow for continuous/iterative regression testing. This also increases the speed to market and reduces the overall cost of testing.

## References

- <https://www.soapui.org/>
- <https://www.getpostman.com/>
- <https://curl.haxx.se/>
- <https://github.com/hmrc/api-integration-test-runner>
- <http://www.guru99.com/api-testing.html>
- [https://en.wikipedia.org/wiki/API\\_testing](https://en.wikipedia.org/wiki/API_testing)
- <http://searchsoftwarequality.techtarget.com/tip/A-primer-on-automated-API-testing>
- <https://blog.udemy.com/api-testing/>
- <http://searchsoa.techtarget.com/feature/Digging-into-quality-API-best-practices-problems-and-advice>
- <https://jenkins.io/>
- <https://github.com/>
- <http://gatling.io/>
- <https://cucumber.io/>
- <http://www.scala-lang.org/>
- <https://maven.apache.org/>
- <http://jmeter.apache.org/>
- <http://jbehave.org/>
- [https://www.owasp.org/index.php/OWASP\\_Zed\\_Attack\\_Proxy\\_Project](https://www.owasp.org/index.php/OWASP_Zed_Attack_Proxy_Project)

## Glossary

Term	Definition
API	Application programming interface
HLD	High level design
E2E	End to end
BDD	Behaviour driven development
CI	Continuous integration
CD	Continuous deployment
ZAP	Zed Attack Proxy

**Sandeep Panghal** is an Automation / Performance Testing expert with 10+ years of experience in Software Testing who has worked with Fortune 500 clients across banking, Public service and Ecommerce sector. He specializes in designing and implementing complex Automating Frameworks in different technologies for all three flavours of development methodology – waterfall, incremental and Agile. His portfolio includes certifications like Agile Coach (IC-ACC), Professional Scrum Master (PSM), Project Management Professionals(PMP), PRINCE2 and many others.

His client list includes Deutsche Bank, Lord Abbott Mutual Funds, Fidelity Investments, HMRC, DWP, Snapdeal.

Contact: [panghalsan@gmail.com](mailto:panghalsan@gmail.com)

# Dev / Ops for Testers

Adam Leon Smith, CTO at Piccadilly Group

## Abstract

Dev/Ops is a trend that isn't going away, with the number of people identifying themselves as part of a Dev/Ops team having doubled between 2014 and 2017. This trend is particularly noted in the US, with teams now identifying themselves as working within the tech sector rather than Financial Services. Whether the European FS market will see the full impact of this trend, or if it's going to pass them by is yet to be seen.

Rather like Agile, the meaning of Dev/Ops is very much in the eye of the beholder – or at least the definition preferred by the last tool vendor you met. That said, at least Agile had a manifesto!

So what is actually involved in Dev/Ops, and does it have anything to do with testing?

When boiled down, the sole and obvious purpose of Dev/Ops is to strengthen the relationships and collaboration between development and operations silos - thereby smoothing the delivery of software to production whilst reducing the cost and effort involved in supporting it long-term. Testing typically sits between development and operations in a release process, so, it only makes sense that Dev/Ops should incorporate it. In fact, they rely heavily in it.

## Rapid Feedback

The first principle favoured by Dev/Ops promoters is nothing new. Rapid feedback is preferred, with automated regression tests at every level of the release process. Secondly, kill code branches quickly and develop/deploy from your trunk branch only. This encourages continuous integration and reduces the number of times any changes need to be made. This is obviously great if you can, but many of us live in a world of multi-tenant, multi-deployment monolithic architectures and can't envisage maintaining a single branch. This practice isn't new either, "Release early. Release often." was initially coined in the great essay *The Cathedral and the Bazaar* in 1997 and gained increasing traction under the Agile movement.

Finally – feedback from beyond the release. Test automation and tooling and active reviews of the software shouldn't stop once applications are released. Using log aggregation, instrumentation and monitoring tools – software should be monitored carefully in production to ensure its production characteristics are within expected parameters. Those of us who work in the financial services industry, particularly card and payments, are likely familiar with first occurrence validation – the practice of monitoring production software the first time it is expected to carry out a function. This is a similar concept, but extended to cover non-functional factors.

## Constructive laziness & choosing the right weapon

A huge element of Dev/Ops is automation. Rather than simply automating the most time consuming and repetitive tasks. True Dev/Ops practitioners will aim to automate as many tasks as possible, removing distraction, variation and room potential for error. An interesting development here is an emerging view that teams should be empowered to pick their own tools. This is a contentious view, as many vendors and large enterprises have been pushing the message for years that standardisation and centralisation of tooling is the key to efficiency. No doubt it is, to a degree. But the argument for individual choice is that the efficiency savings from a single “silo” fall away when you look at the full value chain, and include the adverse efficiency impact and risk when releases are done manually because the “central tool” doesn’t support something essential to one team’s application.

I wrote a few years ago about virtualisation and how enterprises had picked up the technology and used it for cost saving via infrastructure silos, without bothering to roll out all the “agility” functionality that would be useful for reasons other than saving direct costs (snapshots, for instance!). The panacea to my gripe is known as containerisation, the next evolution of virtualisation.

At its core, containerisation enables servers to be created and managed as if they were code, rather than as something that requires a dedicated team and set of processes. A server supporting a specific service can be defined in a simple text file, and checked in with the code, and rebuilt automatically whenever a release is available for deployment.

This is great because:

- Development, testing and production environments are never out of line unless you specifically make them out of line. Servers essentially follow the same processes and pipelines as software.
- Because containers are virtual servers, developers and testers can easily spin up their own containers and then an integration test environment all running on their local machine.
- Building the servers can be built into your continuous integration and deployment pipeline - so you push server definitions between environments, not code releases.

## Breaking down silos

A big part of the Dev/Ops practice, and in fact probably the source of the name – is holistic systems thinking and the absence of silos. In the book *The Phoenix Project* by Kim & Behr (a great book, once you pass the fact you are reading about IT transformation) the development manager is awed when he sees how a support ninja can solve a problem with their scripting toolset faster than his developers can with “real” code. In the same vein, a very experienced tester of an application often knows workarounds that can get a failing production system out of difficulties. This is about thinking about solving problems in different ways than may be aligned to the traditional silo you are from, and different ways brought from the experience of other members of the holistic, integrated team.

It isn’t just Development and Ops teams that participate in this cross-silo thinking. A great example given is Security Engineers / Officers / Testers. These roles can be, and are certainly usually represented as, the epitome of a silo. The typical security officer cares greatly about security and threatens anyone that comprises it with hefty penalties, yet doesn’t bear

responsibility for the convenience and usability of the solution. Integrating these representatives directly within a product team “builds in” security from the start of the process, as well as “buying in” security to the software process itself.

The final practice, which is one that grates on the instincts of most modern IT professionals, anyone can deploy. This is of course, the opposite to what we were taught, pretty much by everyone. The principle is that there are so many checks and balances prior to deployment that deployment becomes zero risk. Getting to this point of course demonstrates you have got your Dev/Ops stripes, destroyed your silos, automated your test, release and environment activities to the death, killed any long running branching activities and accelerated your CI process so much that any release is a matter of a few commits.

### What does success look like?

Success in my opinion can only be demonstrated by the quality of the end product and how well it satisfies its users without burning out the IT teams supporting it. Some examples of metrics that indicate success are:

- Less than an hour from code commit to deploy - maybe not in production, but you SHOULD be able to do it that quickly if you need to.
- Multiple production releases every day - Amazon release software every 11.6 seconds.
- Less than one hour to restore service after failure and even better, a very low change failure rate.

The real glimpse of success though is the human side, the way customers are pleased when software can be changed, quickly tested and released; the way IT professionals can leave the office during a release confident in its success, and that they won't receive a call in the middle of the night; and the way that teams work together once bureaucratic silos of enforced “segregation of duties” disappear.

So what's next? Will big organisations in Europe in the heavily regulated financial services sector adopt this trend? We think, probably yes, but only for their most strategic “new build” applications. All of these practices are far easier to adopt when you are starting with a blank canvas. Furthermore, the internal inertia that needs to be overcome before you can integrate silos will take significant “hearts and minds” work from senior leaders. We think we will see this increasingly in 2018 but targeted at a few strategic areas, not wholesale.

*Adam Smith is CTO at Piccadilly Group where he leads the company's technology innovation. Adam has held senior technology roles at Barclays and Deutsche Bank delivering large complex transformation projects. Adam also has extensive experience leading, driving and solutioning across a range of testing disciplines, including test automation, performance and penetration testing as well as the traditional functional testing. Piccadilly Group helps a range of financial institutions with large scale technology change programmes and Adam often leads large teams to reach complex goals.*

[adamsmith@piccadillygroup.com](mailto:adamsmith@piccadillygroup.com)

[@adamleonsmith](https://twitter.com/adamleonsmith)



1 Fore Street, LONDON. EC2Y 9DT.  
+ 44 (0) 20 7851 4462  
[info@piccadillygroup.com](mailto:info@piccadillygroup.com)



---

## Write an article

We are always on the lookout for new content, so if you have a testing story you would like to share, a test technique you would like to evangelise or testing research you would like to publish, then The Tester is the place to do it. Simply email the Editor on [phill.isles@bcs.org](mailto:phill.isles@bcs.org)

---

---

## SIGiST White Paper Scheme

We have set up an area on the BCS website of a searchable repository for white papers and articles on testing and we are looking for contributors. That means you!

Do you have an existing paper you would like to repurpose and make more widely available through the SIGiST website?

- Then please send us the paper with three keywords for searching.

Would you like to write a new paper?

- Please send us the title and abstract together with the three keywords (or phrases)
- We will review the proposal and guide you through the authoring process
- For those who are thinking of speaking at SIGiST then this might be a good way to prepare a talk and get some useful feedback

If you have been thinking of writing or publicising an existing paper then this is the ideal opportunity. Please email your existing paper (with keywords) or your proposal to The Tester Editor, [phill.isles@bcs.org](mailto:phill.isles@bcs.org)

Past articles from The Tester will slowly be added to the repository as well.

Follow this link to the repository: <http://www.bcs.org/category/18128>

---

# 15 minutes with ....

## Viv Richards – Senior Test Engineer and previously Developer

**Jennifer Wheeler** of [TestingProfessionals.com](http://TestingProfessionals.com) spent 15 minutes with **Viv Richards**, a Senior Test Engineer and previously Developer, talking about his career in testing so far.

Hi Viv, thank you very much indeed for your time, I am really interested to talk with you, one of the reasons being that you are one of the few people I have found who has moved from software development in to testing. I am sure many people will be interested to hear your thoughts, so I will dive straight in if I may.

### How did you get into testing in the first place?

Right at the beginning of my career I spent just 2 months manual testing at the DVLA before moving to Dev, I had initially joined the company in a non-tech role through administration and case handling. I wanted to try something different so applied for the QA role. Another opportunity came up a couple of months later, they had openings in the software development team and I had always been creative and so I thought I would give it a 'go'. In the lead up to the interview I borrowed some books and studied and was fortunate to be accepted for a role.

Later on, the DVLA started a new project putting paper-based systems online, including the vehicle disposal process. I joined this team and was able to use new technologies and techniques such as TDD, pair programming, Agile and test automation using Selenium WebDriver.

Unfortunately, that project came to an end and I had to move back to the previous role not allowing me to use these new skills and knowledge I'd acquired which was frustrating, so I began thinking about moving on.

### What do you think helped you develop your career the most?

Whilst at the DVLA I got involved in setting up and going along to 'meet up' groups for people interested in technology. I met a really nice group of people and it was by getting to know them and them getting to know me that they recommended me to the company they worked for and I moved to a new role.

By putting myself out there in the tech community away from work and investing my own time in learning rather than relying on my employer to help me, it opened up so many new opportunities and helped me build on my passion for tech.

If you can make the effort to get involved, be part of a community and make a contribution as well as learn new skills it is amazing who you can meet and what you can learn.

### What are your thoughts on investing in qualifications in testing?

For me the situation was a little different from the norm, I already had developed development skills so appealed to potential employers in a test automation capacity. I think if someone is keen to get in to testing but has yet to have any exposure to a tech role it would probably be a helpful way to differentiate themselves from others in a similar position.

There are many opportunities available via the internet which can give you the chance to gain knowledge and skills which in turn could be used to help you to perhaps provide a portfolio of knowledge and ability.

### You have worked in the tech market and testing for some time, what advice would you offer people at the beginning of their careers who are keen to travel down a similar path?

My advice would be to get involved in the tech community outside of work as well and trying to add as much value in your role as possible.

Go to Meetups, EventBrite is also a good option. Don't worry if you don't have any experience yet, these communities are very warm and welcoming, you won't be expected to be a 'guru', people are very helpful and it is a great way to learn, build a helpful friendship group and make a contribution to the industry too.

You can also follow people on Twitter to pick up new stuff, Pluralsight is good, I believe they offer 3 months free membership to try it out. Code Academy is helpful for learning and the Ministry of Test has lots of great content and a mentoring scheme which is brilliant.

A good way to drive your growth is to write a Blog on a new skill you are learning, it really motivates you to know what you are talking about and when you have to get a clear message across to others it really cements it in your own mind.

### Staying up to date with new approaches to testing, tools etc., can be a full-time job in itself. How do you stay in touch with what is happening in the market?

It's really a repeat of all the things above. Rather than sitting at home watching a Soap on TV, take the time to invest in yourself, it really helps build motivation and interest in what you do.

### Can you tell us a little about your most recent / current role and key responsibilities?

I have just been promoted to a senior test role in the R&D department at Vizolution working on new innovations of our product which is fascinating. I am involved in all types of testing – tools based, manual, accessibility and non-functional. We are not restricted on tools or approaches at all, at the moment I am working on a new tool for automating some aspects of exploratory testing.

### What are your thoughts on the direction of testing and where it is going next?

I can see development and testing moving further towards automation, but I believe we will always need the human interaction and intelligence in QA as well. I see an exciting future with AI with lots of great opportunities for those passionate about testing and who invest in keeping pace with change.

### Would you be happy for people to contact you with questions or to ask your advice?

Yes indeed, I would be very happy to chat about my experience or help in any way. My email address is [hello@vivrichards.co.uk](mailto:hello@vivrichards.co.uk) and you can also get in touch with me via Twitter [@11v1r](https://twitter.com/11v1r)

For help and assistance in any aspect of Testing career planning or recruitment please contact Jennifer Wheeler at Testing Professionals.

**T:** 01437 532257 or 07733 121897

**E:** [Jennifer@testingprofessionals.com](mailto:Jennifer@testingprofessionals.com)

**W:** [TestingProfessionals.com](http://TestingProfessionals.com)

 [LinkedIn/JenniferWheeler/](https://www.linkedin.com/company/testingprofessionals/)

 [@TestProfs](https://twitter.com/TestProfs)

Testing**Professionals**.com





The image shows the BCS Personal Development Plan (PDP) interface on three devices: a desktop/laptop, a tablet, and a smartphone. The desktop view shows a dashboard with various sections and a list of activities. The tablet view shows a similar dashboard with a pie chart for 'Activities Reported' and a list of activities. The smartphone view shows a simplified version of the dashboard. The BCS logo is in the top left corner, and the text 'Personal Development Plan' and 'Your portable record of professional aspirations and activities' is in the top right.

## Did you get your Personal Development Plan email with suggested potential CPD activities?

The BCS Personal Development Plan (PDP) uptake is going well, with thousands of registered users already actively recording their CPD Development Goals, Activities and preferences. It's not just about recording details though, as there is a Resources section that shows live feeds of potential CPD activities, and a tailored email is sent every 2 months with details of the latest videos, articles, blogs, books and research in your specified field of interest. If you haven't registered yet, you can see the content from the latest PDP bulletin for topics relating to solution development and implementation here <http://www.bcs.org/content/ConWebDoc/50854> or by going to the CPD Portal at: <http://www.bcs.org/pdp/>.

The BCS Personal Development Plan is free to use; BCS members can use their Member Secure Area login and password to access it at <https://pdp.bcs.org/>, and non-members can use most of the facilities (using the same link) and registering to create their own user name and password. You can use it on a PC / laptop or compatible tablet PC or smartphone.