

From Afterthought to Lynchpin: The Evolution of The QA Role

BY AMIR GHAHRAI

The last decade has seen a significant shift in the way we go about assuring the quality of development. From baton passing to collective responsibility, our thinking and practice has evolved to embed quality assurance practices across the project lifecycle.

But how has the role of the Software Tester changed in this context, and do we need to be thinking differently about how we recruit, recognise and reward these specialists?

Historically, taking a lead in the final phases of a waterfall project, testing would sit firmly on the right of the project lifecycle. After up-front requirement definition, the testers would take the baton from the development team at the close of the development phase and run lengthy, detailed test scripts, often manually, and usually through siloed teams and groups of SMEs.

And we got the job done. Tests were meticulously planned in advance, scripts were executed by specialists and tracked with precision through software: defects were detected and reported, and test cycles were run and rerun until the predefined quality levels were achieved.

Most notably there was always a clear separation between developers and testers, with no overlap of responsibilities or activities. Indeed, during the distinct, ring-fenced phase of testing, activities were purely focused on the functional validation of software with the core aim of finding and reporting defects.

The Age of Agile

The emergence of agile methodologies and ways-of-working fused the activities of development and testing to such an extent that software testing was no longer a standalone phase. Instead, testing became an implicit activity during the coding and development of software. In some cases, it would be hard to see the distinction between a 'tester' and a 'developer' as each would have the ability to seamlessly undertake each other's activities.

'Quality' ceased to become the sole responsibility of the testers and became a shared responsibility of everyone involved in developing and delivering the product.'

Along with this evolution came a shift of test responsibilities to the left of the development lifecycle, with activities underway right from the beginning, even before a single line of code was written.

Focus moved from finding defects in built software to preventing defects from getting into the software in the first place. With a shared goal to ensure not only that the product or feature was functional and met requirements but was also fit for purpose and provided a high level of user satisfaction. Testers' involvement in story refinements, peer code reviews, unit testing and practices such as the Technical Definition Document and Business Definition Document, and Continuous Testing, ensured testing and quality were at the forefront and was embedded into the development.

But, while Agile went a long way to combine the activities and practices of development & testing, the operations team was still siloed. The two streams of work (Dev & Ops) were often unaware of each other's activities. If anything went wrong in production, investigation would take a long time. Developers didn't have the insight into how their application was performing in production in the longer term; there was no transparency or clarity of collaboration between the two teams.

Welcome to DevOps

DevOps refers to the collaboration of Development and Operation teams across software creation, delivery, maintenance and support. It refers to a continual union of resource, processes and the product itself. DevOps enables methods of continuous integration and delivery of value to the end user.

'The DevOps movement has driven a new perspective on testing and created new opportunities for testers themselves.'

In this new era testers need to be aligned with both development and operations. The remit of testing is no longer limited to the product but also the testing of the infrastructure where the product is ultimately executed. Continuous Integration (CI), and Continuous Delivery (CD), has become the de facto standard in the development and delivery of software, and therefore much of testing effort is now spent on ensuring the CI/CD pipeline, environments and infrastructure.

This is the spine that supports both development and delivery.

If testing of these are neglected, it could result in flaky environments, much effort being wasted investigating repeated infrastructure issues and, ultimately, a high risk to development and speedy delivery.

Quality Driven Development and The Modern QA:

Although much has been done to embed quality in every stage of development and, as a result, testing has a much wider scope, I still believe that QAs are spending most of their time looking for functional issues and focusing on the verification of software.

Most QAs don't realise the importance of their role and the impact they can have on development and delivery.

Despite the considerable shifts in development practices over the past ten years, I feel that testers still take an old-fashion view of their role and are, thus, remaining entrenched in the old era of testing.

Testing as a profession and the role of a tester has been under fire for some time with the rise of “automated testing”. And indeed, many industry professionals still believe that the role of a tester is simply to test the application that the developers build, all of which can be automated.

If developers are better suited and more savvy to write the code needed for automated testing, then what need is there for a tester on the team at all?

It's about time we changed that perception. We have to acknowledge difference in value and skills between ‘testing’ and ‘quality assurance’ as, where ‘testing’ is functional verification and validation of software, ‘quality assurance’ is not a single activity. QA is a series of processes, including testing, and best practices to ensure a quality product is delivered for users.

We have to strive for quality driven development and look at the QA profession as the central and core function in the development and delivery of software.

Hence **Modern Testing** now has a very different shape. QA is now a key component of development from start to finish working across the entire process. And, although common parlance says that everyone in a delivery team is responsible for the delivery of a quality product, I firmly believe that it is the responsibility of a QA to ensure that quality is being held as a KPI of every deliverable, that quality is the responsibility of all.



Who is the Modern QA?

Where the test profession was often seen as an access route into development, project management or other – usually more lucrative – disciplines, the new QA is a highly-skilled role that demands a holistic knowledge of development practices.

It requires a broad understanding of the challenges of coding practices and management, an appreciation of deployment methods and environments, and an increasingly complete appreciation of performance and security standards, methods and challenges. This is a T-shaped role with the resource not only able to apply their deep expertise and experience to deliver their core remit, but to apply a wider contextual knowledge across architecture and development.

Sitting in the centre of any project, the modern QA should have a good understanding of architecture, performance, security, and cloud offerings, be technically sound and have a thirst for learning new technologies to stay in the game.



As an industry, in order to build on and fully benefit from this testing evolution, we need to be attracting the best talent into the QA resource market, by drawing attention to the value of the Modern QA and the vital role they play in the delivery of high-quality development.

Although salaries don't vary wildly between software engineer and test roles, perceptions do. In order to attract and retain the highly skilled resources required to deliver effective, consistent QA in any organisation, this perception needs to change.

Emphasis needs to be given to the role a QA plays across the entire project cycle, and the value of the context they bring to a project. And most importantly, potential QA specialists should be attracted by the learning opportunities the career affords in and of itself, and the satisfaction and personal reward that comes from acknowledged accountability of delivering a quality product as an equal member of the team.

Amir Ghahrai is a Senior Consultant at Amido