

VERSION 1.0
DATE 07/09/2024



WHAT VALUE DOES A LEADER OF SOFTWARE QUALITY ASSURANCE (SQA) ADD TO A COMPANY.

PRESENTED BY: SIMON READ

INTRODUCTION

To answer this question, we need to first understand that unless the SQA Leader has already implemented Testing as a Service (TaaS) and is generating a sizeable supplementary revenue stream (an SQA 'side-hustle' if you will) via the sale of their test assets as a service to paying customers, then the services the SQA function provides to their own company are most likely to come at a relatively sizeable cost, and one which the company executive management, with their weighty responsibility for controlling the company's cash flow, instinctively prefer to minimise, because more cost means less profit - right?

After all, procuring test equipment isn't always easy or cheap; developing test infrastructure can be complicated and time-consuming; SQA engineers' salaries are increasingly comparable to those of software developers'; and all that investment in SQA could be spent instead on employing more developers, to produce more lines of code, and create more software products - right?

But let us pause and reflect as to what the fundamental role of the SQA Leader is and how they add value to a company's saleable software product portfolio. Popular definitions for SQA are to be found in [1], [2], and [3] and can be summarised as:

To create an SQA function that provides a continuous information service to the company executive management that will enable them to objectively and predictably establish the quality of the company's software products, decide whether this quality meets the customer's actual requirements, and take any corrective actions in a timely manner that will increase both the company's and the customer's confidence in the quality of the company's software products.

Furthermore, and hiding in plain sight are the multitude of essential ancillary roles the more diligent SQA Leaders continuously perform, many of which go beyond the boundaries of only adding value to discrete saleable software products, and which are in fact integral to adding value to the company itself by establishing and underlining its most profitable asset: **its brand reputation for software product quality.**

So, let us now rewrite the fundamental role of the SQA Leader as:

To create an SQA function that provides a continuous information service to the company executive management that will enable them to objectively and predictably establish the quality of the company's software products, decide whether this quality meets the customer's actual requirements, and take any corrective actions in a timely manner that will increase both the company's and the customer's confidence in the quality of the company's software products and its brand reputation for quality.

*

To define it more succinctly:

The proper investment in the SQA function will not only increase a company's software product quality but will also add value to its brand reputation for quality, which will lead to increased customer confidence, which will lead to more sales.

And therefore:

The proper investment in the SQA function ... will generate more company profit.

*

THE QUALITY ADVOCATE

Within a company, the SQA function ideally has a degree of independence sufficient for it to perform its software product quality assessment duties without subjective bias, undue pressure, and with objective accuracy, honesty and integrity. In a mature software company, software product quality is inherent - permeating through all job grades, disciplines, and functions - and is everyone's responsibility: quality practises are integrated into every stage of the software development lifecycle (SDLC) and software test lifecycle (STLC). But it is ultimately the SQA function who measure and communicate software product quality.

Consequently, the SQA Leader's role can be an isolated one and one for which they must quickly become adept at being self-reliant and disciplined. On the one-hand, independence allows them the freedom to choose their test policy, test strategy and test methodologies, where and when to focus their test efforts, how to balance manual versus automated test coverage, which metrics will most accurately represent quality and so on.

A word of caution: it is neither possible nor necessary to test every permutation, every event, and every user scenario conceivable in a software product and the SQA Leader must carefully manage the deliberation process that decides not only what to test, but equally what not to test. It is still very possible and indeed desirable to be a perfectionist – but only for those aspects of the role the SQA Leader can truly control.

But on the other hand, independence can make tight integration and close collaboration with the company's software development function and their SDLC practises more challenging. The SQA Leader must work hard to cultivate trusting relationships which reassure software developers that the aim of the SQA function is to provide a service which mutually improves software product quality through measurement, information and communication, and not the proverbial stick of detected defects with which to berate them. The sum of the SDLC and STLC combined must be greater than the sum of these two individual functions.

However, a company that does not fully understand the fundamental role of the SQA function or which views it only as a 'tick-box exercise' for a sought-after accreditation with a wall plaque to hang in its reception, or a cost to be minimised at all costs, will present the greatest challenge to the SQA Leader's role, and one which cannot be overcome quickly or easily.

If the SQA function is perceived to be a singular activity that occurs in the latter stages of an SDLC (as in the 'waterfall' SDLC model) or if the SQA engineers themselves are perceived to be interchangeable and identically skilled 'clones' that can be readily applied to any task requiring testing, then the SQA Leader truly has a mountain to climb to educate the company in the benefits of SQA best-practises, processes and methodologies such as 'Shift-Left' and 'Shift Right', Continuous Integration (CI) / Continuous Deployment (CD) / Continuous Testing (CT) pipelines, Behaviour Driven Development (BDD), Test Driven Development (TDD), static and

dynamic analysis, Agile and 'W' SDLC models, Artificial Intelligence and Machine Learning, and so on.

Arguably the best place to start is at the top, with the executive management's quality policy (hopefully they have one) from which the SQA Leader can derive and formulate the company's test policy, test strategy and master test plan documentation, all of which together embody the SQA Quality Management System (QMS).

*

The SQA Leader must develop a suite of documents explaining the 'who', 'what', 'how', 'why' and 'when' of SQA best-practises which are embodied into SDLC and STLC processes and procedures and with which the SQA function can educate, inform, and assure both the company and the customer and thus increase the company's and the customer's confidence in the quality of the company's software products and its brand reputation for quality.

*

THE SKILLS ADVOCATE

Just as it would be unwise to generalise the entire profession of software development as simply ‘programming software’, the same is also true of software quality assurance: it is not simply ‘testing software’. Just as a nurse is not just a nurse, and a football player isn’t trained to play in every possible field position, each and every profession has its many specialisations for very good reasons, and this includes SQA.

The SQA role is a matured professional discipline with established career development paths and job grades, and global industry-based qualifications and certifications for its numerous specialisations [4] that recognise an individual’s skills, experience, and seniority. The SQA role has practical applications in almost every industry – from those massively scaled corporations running 24/7/365 operations that keep the lights on, the water flowing, and the internet streaming, to those entrepreneurs starting their first ‘side-hustle’ from the nearest café offering free Wi-Fi. Where software permeates into another aspect of our modern lives, then SQA closely follows.

Whether you are defrosting a frozen car windshield via a mobile phone message transported over a Wi-Fi to telecommunications to internet to (again) telecommunications transport network from the comfort of your bed to your car outside, or trusting your life to a machine in an environment where humans are not naturally to be found, you’ve probably never given much thought, if any, to the quality of the software that’s meeting your needs and sustaining your lifestyle or even your life support. You’ve probably never considered how that software quality was measured and improved by a dedicated SQA function operating within the framework of a QMS and fulfilled by a team of professional and uniquely skilled and specialised SQA engineers determined to make their software products meet the customer’s requirements and be the best products that they can be.

Actually, most software products have to be so much more than the best-products-that-they-can-be. Whether it’s a ‘fly-by-wire’ airliner cruising at 10,000 metres, or a telecommunications base station transmitting radio frequency energy across a community, or the medical equipment in the operating theatre of a hospital, all of them must conform to rigorous industry standards and comply with regulatory and legal requirements (nationally, internationally or both) that ensure their correct functional operation and performance whilst at the same time keeping the users and the wider public safe, and preventing hijacking, malware infections, system degradations and consumer data theft by hackers.

Furthermore, not only does an SQA engineer specialise and hone their skills for a specific industry (e.g. gaming, medical, automotive, telecommunications, security, mobile applications, web applications, energy supply, education, and networking) they must also intelligently apply a suite of different test types to comprehensively measure all the capabilities of the software product and thereby its quality.

Such techniques include functional, performance (i.e. load, stress, endurance, and reliability) scalability, interoperability, and (again) security and these are necessary to not only test that it does what it's supposed to, but it will do it 24/7/365, for maybe up to many thousands or millions of users, for many thousands of hours, without failing (or if it does, then it doesn't fail with catastrophic consequences, but gracefully, and then hopefully recovers soon afterwards), in a deeply-connected environment and whilst keeping your data safe at all times.

*

The SQA Leader must recruit talented SQA engineers with the right blend and balance of competency, experience and diversity to create an SQA function skilled in SQA best-practises and with whom the SQA function can educate, inform, and assure both the company and the customer and thus increase the company's and the customer's confidence in the quality of the company's software products and its brand reputation for quality.

*

THE PRODUCT ADVOCATE

Within the combined frameworks of the SDLC, STLC and the CI / CD / CT pipeline, software developers integrate individual units of functioning software into larger, more complex functioning aggregations until finally a fully functional system is created that can truly be defined as the software product.

Typically, it is the development function who perform the earlier unit and integration-level tests, and the SQA function who perform the latter system-level tests (together these are known as verification). The final stage concludes with the release of the software product to the customer to perform the acceptance-level tests (this is known as validation). Job done, and another satisfied customer (one hopes)!

But let's again remind ourselves of the fundamental role of the SQA Leader:

To create an SQA function that provides a continuous information service to the company executive management that will enable them to objectively and predictably establish the quality of the company's software products, decide whether this quality meets the customer's actual requirements, and take any corrective actions in a timely manner that will increase both the company's and the customer's confidence in the quality of the company's software products and its brand reputation for quality.

Within the lifecycle of the software product, it is the SQA function who are uniquely positioned at the critical juncture between the company and the customer, between the system and acceptance-level test stages, and whose information of the software product quality forms the basis from which the company executive management decides whether or not the software product is to be released to the customer.

This juncture is arguably the most critical of all the SDLC quality gates. It's the time to pause and check your answers one last time before raising your hand and submitting your test paper in an exam; it's the moment to reflect upon your decision to take up sky-diving before stepping out of the aeroplane; it's the final opportunity for the SQA Leader to ask themselves the question "has enough test coverage been performed, of the right test-types, and at the right test-levels, and has all of the necessary objective information been amassed and collated" to be able to look the company executive management in the eye and declare "you now have all of the information you need to establish the quality of this particular software product and decide whether it meets the customer's actual requirements".

A word of caution: company execution managers generally have very little time to spare so the SQA Leader must communicate this information objectively and succinctly, and always back it up with evidence. Remember: always objective, never subjective.

But a software product is never only about software: software needs a physical medium on which to be installed and to operate, and a host operating system from which it can request the services necessary for it to perform its role, such as memory to be able to fully function, suitably powerful processors to meet its performance requirements, and network connectivity to communicate beyond its own physical confines in a multi-technology environment. It may even need to be able to do all of this for a range of different proprietary operating systems and types of hardware.

Therefore, in order to assess the software product at the system-test level and implement the combined frameworks of the SDLC, STLC and the CI / CD / CT pipelines successfully, the SQA function must specify, design, procure, build, and maintain 'test benches' comprising both the hardware and operating system to host the software product and the test equipment essential to stimulate and measure the software product as a complete and deliverable system, ideally identical to that to be delivered to the customer.

Consequently, in the lifecycle of the software product, it is also the SQA function who are uniquely positioned to observe and assimilate the system-level knowledge of the software product operating as a complete and deliverable system, and with which they can review and revise their QMS and their recruitment strategy, and inform of any discrepancies between the software product's expected behaviour and its actual behaviour.

Furthermore, not only will the SQA function's information of the software product quality be communicated internally to the company executive management, but it can also be communicated externally to the customer as demonstrable evidence of the software product's progress against any contractual milestones.

*

Now the SQA Leader must manage the creation of test environments that host the software product as a complete and deliverable system and with which the SQA function can educate, inform, and assure both the company and the customer and thus increase the company's and the customer's confidence in the quality of the company's software products and its brand reputation for quality.

*

CONCLUSION

So, what value does an SQA Leader add to a company? Beyond the traditional boundaries of adding value to discrete saleable software products, a diligent SQA Leader performs a multitude of essential ancillary roles that are integral to adding value to the company itself and which include: the implementation of the Quality Management System; the recruitment of the right blend and balance of suitably skilled SQA engineers (and the management of their on-going professional development and retention); and the creation of test environments capable of enabling the assimilation and sharing of the system-level knowledge of the software product behaviour when operating as a complete and deliverable system.

*

To conclude:

The proper investment in the SQA function will not only increase a company's software product quality but will also add value to its brand reputation for quality, which will lead to increased customer confidence, which will lead to more sales.

Ergo:

The proper investment in the SQA function ... will generate more company profit.

*

REFERENCES

[1] IEEE – Software Quality Assurance: *A set of activities that define and assess the adequacy of software processes to provide evidence that establishes confidence that the software processes are appropriate for and produce software products of suitable quality for their intended purposes. A key attribute of SQA is the objectivity of the SQA function with respect to the project. The SQA function may also be organizationally independent of the project; that is, free from technical, managerial, and financial pressures from the project.*

[2] ASQ (American Society for Quality) – Quality Assurance: *... part of quality management focused on providing confidence that quality requirements will be fulfilled." The confidence provided by quality assurance is twofold—internally to management and externally to customers, government agencies, regulators, certifiers, and third parties. An alternate definition is "all the planned and systematic activities implemented within the quality system that can be demonstrated to provide confidence that a product or service will fulfil requirements for quality.*

[3] ISTQB (International Software Testing Qualifications Board) – Quality Assurance: *Activities focused on providing confidence that quality requirements will be fulfilled.*

[4] ISTQB portfolio of [software testing] certifications: <https://www.istqb.org/#certifications-diagram>