

# Quality: 2020

**Brian Gaudreau**

[bg2066@outlook.com](mailto:bg2066@outlook.com)

## **Abstract:**

Quality in software delivery and sustainment will always have competing pressures between capability desired and the cost incurred, time taken and quality delivered.

As we approach the year 2020 the technology landscape will have additional competing pressures we need to understand to be most effective keeping quality high. For example:

- Client/Customer expectations will change and impact product roadmaps in new ways
- There will be new risks and assumptions to identify
- What is meaningful to measure in a Quality program will be more varied and dynamic than ever before
- Quality Organizations and Programs will need to re-assert their position to add even greater value

This paper will introduce new perspectives on how to plan and be ready for the future of software quality. Expanding DevOps culture with greater involvement by QA and adding customer/product owners to the model is one perspective. I will also highlight how to identify, manage and loop data and metrics back into the requirements and continuously improve the client/customer experience.

## **Biography:**

*Brian was most recently Group Manager of Quality at Avanade on the Cloud Solutions team. Brian brings to PNSQC over 20 years of software testing and quality assurance experience relevant to telecom, enterprise metadata, government, multimedia, health care and CRM platforms. In addition to local involvement in software quality assurance interest groups, Brian's other interests include playing live music and supporting animal rescue shelters in the Pacific NW.*

## Introduction

Whether it is the steadfast construction of an ocean vessel or producing enormous pyramids and temples, Quality has been part of our thinking and delivery for thousands of years. Software quality is relatively new adjective to our industrial journey. Moreover, most technologists would agree that the 1972 Trident submarine project involving over a million lines of code was the first time an iterative quality feedback approach was used in large scale software delivery. Software as a focus of quality and the rate which we need to iterate and change to be most effective is overwhelming and exciting at the same time.

As technology advances, ways in which requirements and business value are determined and measured also changes. For most people in developed nations acquiring specific goods or services relies on the Internet. It is easier than asking your neighbor or a good friend, and even more accessible wherever you are. The data points that allow a transaction acquiring items of value to be possible are increasing- even the way we seek and store information has changed.

## Strategy: Understanding Each Piece

Building the right thing and building it right the first time means you have to embed quality into every part of the software development life cycle. By integrating a quality focus from planning, requirements, design, release and beyond, testing can deliver more than simply quality assurance. The resulting framework delivers a high-value quality roadmap that will support and sustain the product or program.

Einstein's theory of relativity can help us put the new quality landscape into perspective:

*Two events, simultaneous for one observer, may not be simultaneous for another observer if the observers are in relative motion. [Einstein, Albert (2009), Relativity - The Special and General Theory]*

What does this have to do with software quality? Cloud computing, Internet of Things (IoT) and connected Devices as a Service (CDaaS) are expanding technical possibilities similar to the big bang theory. Accelerating outward on multiple tangents and exponentially growing in size and impact with every moment. And these new components exist on top of the technology infrastructure already established. As professionals immersed in technology we may not be able to see how all these elements impact each other, but to add value and be successful with Quality going forward we must learn this skill.

To understand and implement a relevant quality management strategy with the future's ever growing scope, we should understand smaller pieces of the puzzle in more detail:

### Testing

The term "testing" (and with that Quality) is getting lost in today's framework and methodology. Creating test assets and then monitoring the health of those tests seems to be accepted by the majority nowadays as the best case scenario, but testing is so much more than that. Make sure Testing and Quality are defined and effective, even as they apply to nonfunctional requirements.

### Strategy

Don't always assume that when test or quality teams are invited to strategy or planning meetings, that is the time to tell everyone that a concept cannot succeed or has too many missing criteria to be sustainable. Even though the test and the quality effort may be a separate team, the effectiveness of our input may require follow up after team meetings.

### **Areas of Quality & Test coverage**

We enjoy looking at coverage models and pie graphs that remind us as quality professionals all the potential areas to prioritize and account for in our plan. Now it is becoming harder to segregate these coverage areas based on platform interdependencies. App testing and service testing need to be part of the same equation. Extensibility, Live Service Agreements and Capability are rising as new coverage areas that rank high in priority for planned coverage and reporting.

### **People**

People continue to be the glue that makes delivering products and solutions rewarding and scalable. More than ever we need to invest and train staff to be T-shaped in their thinking. A T shaped person has depth of related skills and expertise in a single role, while at the same time are able to collaborate across roles and platform experts and contribute in areas of expertise other than their own. The “T” represents this simultaneous depth and breadth concept with its perpendicular lines.

### **Test Driven Development**

Hours could be spent digging deeper into this one practice: the technique of having teams write tests that fail before new code is written. One key benefit is self-documenting the product or platform under test. Even with refactoring and working iteratively towards a correct final outcome, the resulting test suite will increase in value as a source of record. This allows our understanding of how Production currently works and enables planning for future capability.

### **Data**

Data is another area that is massive and a person could devote days to the ways Data and Quality intersect. More than ever, we need to be intentional about the ways data impacts our need to have the product or platform function as well as understanding how it is being used (versus how it was designed or planned for). There are four key data areas that need to be understood and planned for:

- *Determine and Classify.* It is amazing even in this day how often this step is glossed over for the mechanics of creating the data itself. Not only sources, types and variants but security policy and data flow need to have expectations set and understood.
- *Develop and Get.* Quantity of data, what qualifies for mock & stub, conversion and encryption strategies are all important. Don't forget researching the performance of extracts and acquisition even before the first validation run is done.
- *Populate and Use.* How loaded, who owns it and storage needs. So many people organizations miss the disconnect between identified data and dependencies to store, migrate and host data. Make sure you take a step back and ensure scenario E2E flows are represented.
- *Maintain.* Refresh, retention, baselining and reporting all are needed to maintain data quality. Without reusability there is very little value in test data management. Consider a self-service model where data can be refreshed on demand as well as how to measure effectiveness of data cleansing.

### **Metrics**

Metrics are one of the keys to evidence decision making, a core principal of Quality Management. We need to have the courage to re-examine metrics we are collecting and how applicable they are going forward, even if we have developed months or years of trends based on the current criteria. Here are metrics I see needing the most attention for organizations to be ready for the Quality agenda:

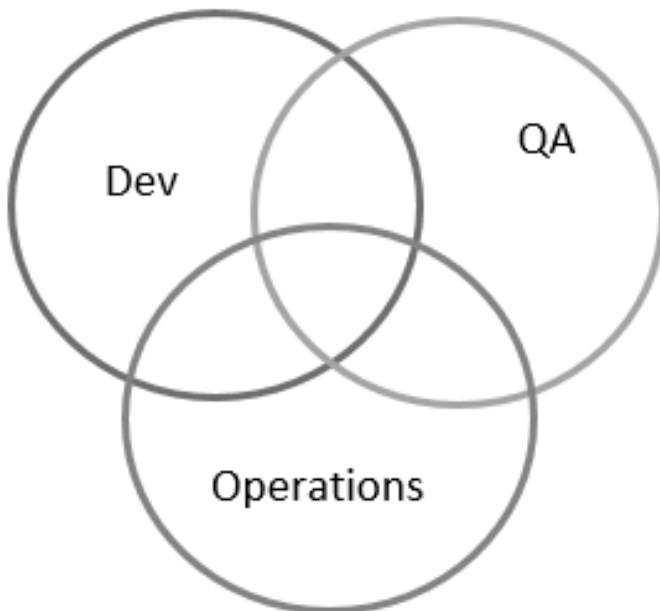
- Define a definition of done (“doneness” relative to the bigger picture). Even when using waterfall. Make sure this is understood, agreed to and relevant to the delivery model and number of concurrent teams developing features.
- Develop metrics that show the cost of test including rework, test data delivery and usage.
- Metrics that show Production incidents by feature as well as metrics that show effectiveness of delivery. And for added value, measure the manual effort for processes associated to features and delivery teams.

### **Tying It All Together:**

We should remind ourselves that clients and customers focus on the experience, not the features. Features in concert influence and drive a positive experience. The following strategies are key to having a successful quality management program in the future:

#### ***Take DevOps to the next Level***

Most DevOps teams operate in a 3 circle model sharing Development, QA and Operations:



Having a data driven DevOps approach will result a larger Quality impact:



So how is this wider circle achieved? Developing quality heuristics that measure how the product or solution is *actually* being used in both preproduction and production can dramatically reduce critical defects and ongoing customer support issues surrounding usability and data compatibility. This evolution of DevOps culture allows more direct feedback to influence and understand what the solution and product is actually delivering (not just what the specifications say it should deliver)

### Test Data Management 2.0

Another strategy that needs attention is data management. The idea of Test Data Management (TDM) has been around for years and applied in many different ways. To be most effective in the future TDM will need to evolve even more.

The Quality Program of the future will need a matrixed perspective on the data it is using to create the maximum value for the effort:

PEOPLE – The user community and providers using the product or platform

TECHNOLOGY – The expectations and dependencies of the product or platform

PROCESS – Delivery and Service of the product or platform

By deliberately understanding production data delivery teams can spend less time justifying what data to use and more time validating a delivered solution. Dependencies within test and production environments become intentional and not blocking or reactive. Having the Quality program lead this effort reinforces the value of this discipline to the overall product or solution.

In future Quality programs data management metrics will distinguish departments that document defects as a reactive response and reinforce being a cost center ...from those that proactively detect flow/data errors and are a value add to the organization. We need to get Quality teams comfortable with exploring how customers are using the delivered product rather than measuring clients using the system only in the way we expected them to.

## Removing the Label of Quality as a Cost Center

Time to market expectations and delivery complexity is not making it easier to sustain a quality program. To be best prepared we need to assert the value of quality planning and tasks whether they are part of a standalone team or embedded into the development cycle. Here are four areas I believe are a solid start to meeting that goal:

1. Reduce handoff costs and measure effectiveness of continuous delivery. Having the process established is a great milestone and I do not want to minimize this, but measuring the effectiveness of this process takes value and efficiency to the next level
2. When the Quality program understands and proactively supports the product roadmap - more products/services will be sold. Use the strategy planning, data and metrics elements discussed earlier to allow execs and product owners to make informed prioritization choices that will position your features best in the marketplace
3. Modify attributes and codes so a Quality event gets traced back to providing value and not just absorbed into the Engineering delivery effort
4. There will always be bugs, people will make mistakes and providing solutions using cloud will introduce scaling and service level challenges never before experienced. Automating repeatable processes identified by a collaborative concert of Architects, Developers, Test and Operations resources will be of paramount importance going forward. As I discussed earlier, remember to track progress towards automating key manual tasks that remain necessary for deployment, support and extensibility

### **Conclusion:**

Whether we are delivering a fully developed system in a consulting model or our packaged software is playing a role in a larger platform ecosystem, we need to broaden our understanding of the impact a Quality program can have.

The very definition of quality is having a grade of excellence. The more distinctive and essential something is, the better able it is to satisfy stated or implied needs. Read that part again: *implied needs*. We are moving to an era when we will develop and implement an idea, test how the customers respond to using the idea based on initial conditions and then evaluate how close we were to building the right thing. Facebook and other service companies have been doing continuous delivery and deployment in Production for years – but the next years of delivering technology solutions are going to expand feasibility of who can use this approach and the speed in which results will be assessed and prioritized for delivery. Quality teams can either choose to embrace and be influencers of this new approach, or risk being seen as less agile with higher cost, but without matching benefits in return.

The value of this approach should be crystal clear by 2020.