

Quality = Business Value

Part 1: Why common definitions of Quality fall short

How does your organization define “Quality”? Has a definition of “Quality” ever been formally adopted? Is there even a common understanding of what is meant by the word among all of the various people in your organization?

What about *you*? How do *you* define “Quality”? How many others in your organization would agree with you?

“Quality” is one of those words that is used all the time, but rarely defined. We are supposed to produce “quality” products (or to be more correct, “high-quality” products). But the precise meaning of that mandate is left to the imagination. Is it any wonder that we have so many disagreements about whether or not we have succeeded?

Even among those who have undertaken to define “Quality”, there is more variety than consensus. For example, a product or service is “high-quality” if it ...

- Conforms to Specification
- Satisfies Users’ Needs
- Implements “Best Practices”
- Is produced using defined processes
- Is acceptable to the customer
- Is driven by the customer
- Fits cleanly into the operational environment
- Does not fail in any serious way

By examining the ways each of these traditional definitions falls short, we can appreciate an emerging new definition:

Quality is delivering Business Value

As we do, we will see that it borrows the high points from each of the common definitions while leaving the baggage behind.

How Common Definitions Fall Short

Conforms to Specification – This is perhaps the oldest of the definitions of Quality. It has been around since we began writing specifications decades ago. Since a specification is an agreement between producer and customer, the expectation of conformance is implied. And of course, when we promise something, ethics demands that we do what we said we would do!

The problem with using the specification as the definition of quality lies in the difficulty of writing specifications. Has everything that needs to be said been captured? And has it been written in sufficient clarity and detail to lead to the correct product? Can the customer and producer working together accurately forecast what will be needed? In what ways will the technology or business environment change between when the specification is signed and the product is delivered? (And many other problems.)

Although defining Business Analysis as a professional specialty has helped us to write better specifications, it has not proven to be a panacea (nor does it promise to be in the foreseeable future). Conformance to specification is generally a good thing, but sorry experience has proven it to be far from sufficient.

Satisfies Users' Needs – This definition came about primarily as a reaction to the problems identified above. In extreme cases, the concept of specification is abandoned altogether. But even where there is a specification, its focus is on the users' needs, and the project is more about those needs than the spec itself. What could be more appropriate than a people-oriented approach? Meet people's needs, and you have done your job.

The problem here starts with the definition of "User". Identifying all of those people is non-trivial, and subject to the same completeness issues as the specification itself. But the real problem here is that the users as a group do not represent all of the dimensions of quality. Many things that should be included in the product come from other Stakeholders who will never use the system itself (either directly or indirectly). So we are likely to end up with a myopic view of the needs and an insufficient product.

Beyond that danger is an even more important issue: Is satisfaction of every need of every user really appropriate? In many situations, this approach could be costly and result in a never-ending project that never quite reaches the goal of user satisfaction.

Implements "Best Practices" – As the decades have passed and more organizations have experienced various forms of success, we have seen a plethora of "Best Practices" being identified and promoted. These run the gamut from how systems should be developed to how various business activities should be done, and how technology should support those business processes. Following the patterns of successful organizations seems like a safe path to success.

The problem here is with the "Best Practices" themselves. I have adopted the rule of putting the term "Best Practices" in quotation marks because it has become clear to me (and I am not alone) that what is best for one organization may be a train wreck for another. And even if it does not constitute a train wreck, a practice may be merely passable and inferior to other options. As a friend and colleague of mine loudly proclaims, "There is no such thing as 'Best Practice'! There are only *good* practices that are appropriate in certain contexts." Applied to the wrong context, a "Best Practice" is a recipe for poor quality.

Produced using defined processes – The process movement of the last decade or more has resulted in this definition of Quality. The Software Engineering Institute's (SEI) CMMI® and the Project Management Institute's (PMI) PMBOK® are the best-known (though far from the only) examples of methodologies, models and BoKs (Bodies of Knowledge) purporting to tell us how to solve the difficulties we encounter on our projects. Although they are helpful, they can only take us so far.

Case in point: In the original CMM, the “Quality Assurance” Key Process Area was all about *process* quality, not the quality of the product! And although the new CMMI changed the name of that Process Area to “Process and Product Quality Assurance”, the “products” are almost exclusively intermediate work products that result from the process, and “quality” is defined as conformance to standards.

The reality is that good processes can be employed to produce the wrong products. Process quality is an important measure of the development organization and how well it works internally. But as a measure of the quality of the result of our project, process quality has little utility.

Acceptable to the customer – This definition re-focuses on the customer. But in this case, it is not so much about the end users as it is the authority who is requesting and paying for the project. In this case, the “customer” is a manager or executive who has made the decision to request the product, and the final acceptance of the product by this person becomes the focus of the project. What could be more appropriate than this?

The first problem with this definition of quality is much like the problem with the “Users’ needs” definition: Although the “customer” is a critical stakeholder, he or she does not constitute the entire stakeholder population. This person will be optimizing the project from one perspective, and is likely to make decisions that are not in the best interest of other stakeholders, especially end users or the organization at large. In the worst cases, this person may have an inadequate idea of what the true needs are, resulting in a product that he or she deems to be acceptable when it is actually useless.

When this “customer” is truly buying the product (either as a commercial transaction or as a budgetary transfer), the specter of budgetary considerations overriding others is a very real possibility. On the other hand, when there is no money (or funny-money) at stake, you run the risk of having a customer who never finds the product to be fully acceptable and a project that never ends.

Driven by the customer – The Agile methods attempt to remedy many of the problems we have discussed by making the Customer a member of the project team. And this “Product Owner” role is not merely a member of the team, but the primary driver of project activities. The Agile methods “time-box” the project to control costs by limiting schedule, and the Product Owner is given broad latitude to get as much he or she can from the project within those constraints.

This definition of quality has most of the same shortcomings as the prior one, “Customer Acceptance”. While the customer is sure to get the product he or she wants, there is no guarantee that the needs of other stakeholders and the organization at large will be adequately met.

Fits cleanly into the operational environment – In some organizations, high availability and other operational issues are so critically important that they become primary considerations in product design and implementation. This approach remedies problems that many shops experience—operational issues not being sufficiently addressed during specification or development, only to surface during implementation. In a healthy organization, a new product’s fit with the existing operational environment will always be considered.

This is only a problem when operational fit is the overriding or primary measure of quality. The trumping of customer need and other organizational concerns by operational issues fits the classic definition of “the tail wagging the dog”. The Operations function exists primarily to support the organization and its customers, not the other way around. Operational fit should be a consideration in quality, not *the* consideration!

Does not fail in any serious way – Finally, we look at the common focus on defects as the measure of quality. Of course, testing (which is the primary tool employed to detect defects) is usually framed by the specification, so it begins where we did in this article—conformance to specification. But because no specification defines every detail, testers must go beyond the written document as they determine whether they are observing proper functionality or defects.

Defining quality in terms of defects is problematic in two ways. First, as we observed above, personal judgment becomes a significant part of the equation. Many disagreements between testers and developers come up, and it is inevitable that many of those disagreements are won by whoever is the stronger negotiator, more politically potent, or able to yell louder.

The second problem is that this tug of war between testers and developers leaves all of the other stakeholders out of the mix. The quality of the product is determined without input from people outside the project. This means that at the end of the project, when the developers and testers have declared the project to be of “good quality”, other stakeholders could seriously disagree, and often do.

Quality is Delivering Business Value

Defining quality as delivering value to the business takes us a long way down the road to correcting the problems we have discussed in this article. It is not merely a matter of blending these definitions of quality. Rather it is a change of the point of reference from which quality is judged. The project is being undertaken in order to bring specific benefits (or value) to the organization. That value proposition becomes the touchstone for all decision-making in the project, not the least of which are the qualitative judgments about the product.

We will delve into what this looks like next time, in Part 2.