Volume	6	Support and Development
Chapter	03	Quality and Processes
Section	03	Quality Actions
Kew Word Tags	Design Considerations, Developer Trade-offs	
Stage	Has a demo product or service	

FITNESS FOR USE DECISIONS

File No. 6.030304 | 2017-06-01

Quick Summary: Developing any product or service requires trade-offs that must be made consciously.

Abstract:

Quality has been defined as Fitness for Use as expected by the user. This definition is, indeed, very broad with different users having different expectations. Developing a product or service requires the consideration of many factors that, at times, may be at odds with each other. To address the design considerations and simultaneously attempt to address all of the issues that a user may encounter, all involved parties need to consider all of the product's or service's characteristics and prioritize them accordingly.

In the early 1950s, the world-famous quality pioneer and expert Dr. Joseph Juran, defined quality as "fitness for use." The fundamental notion behind this seemingly simple definition is that a product (or service) should do what a customer expects it to do. Notice that the metric is based on the customer's expectation and not what the supplier's specification sheet might say. For engineers and others who develop products and services to meet specific written requirements, meeting expectations rather than specific measurable criteria can be troublesome. However, focusing on the customer's expectations is critical to building and maintaining a long-term successful customer relationship. The article in this collection, "*A Simple Definition of a Defect,*" drives home this point that a defect is defined as "*any deviation from a customer's expectations*."

With a startup company's focus on releasing their MVP, it is not unusual for them to feel that quality issues can be dealt with later, after the wave of initial customers have accepted the product or, at least, provided positive feedback. It is easy to fall into the trap of thinking there is not enough time, money, or resources to dwell on quality – it can be added later. In far too many cases, "later" never comes! The alternative need not be a full-blown, rigorous approach to quality. In most cases, reviewing the many aspects of quality and providing direction to the development team is enough. With that guidance, the development team can make the necessary trade-offs between features, resources, and time while keeping quality constant.

Even with the very first customers, it is crucial to view quality as table stakes. Those customers may accept limited feature sets, less-than-optimum documentation, and other "growing pain" issues. However, they rarely will accept poor quality as they define it.

As significant a challenge as it is to meet customer's expectations, Fitness for Use must take on a larger, more encompassing meaning. To be truly fit for use, a product or service must meet other criteria that may not be immediately or even important to customers but are critical to the supplier of the product or service. If, for example, a product cannot be supplied profitably, a supplier simply cannot be expected to keep offering it. The lack of profitability could be caused by a variety of factors including product cost, support costs, distribution costs, maintenance issues, or continued parts availability, plus many other similar issues. The development of a product or service always represents a compromise of several factors. Product cost is probably the first factor that comes to mind. However, many other factors are also present, and many of them are interrelated and require careful consideration. Unfortunately, in many cases, the weighing of alternatives or the emphasis of one factor over another are made either unconsciously or are made by a few individuals as part of the normal course of development. Developers constantly make decisions based on what they perceive is correct. Often, the impact of those decisions are not obvious until some future event takes place.

To avoid this unwittingly created problem, a technique has been developed to raise the awareness of the characteristics that should be considered in developing a product or a service. By consciously reviewing and prioritizing the characteristics, all involved parties can develop the same level of expectations and understanding. Over the years, the list of characteristics that should be considered has grown to thirty-two items. Certainly not all items are of equal importance, and not all items are even applicable to all products or services. But, by considering all of them and prioritizing some and discarding others, a clear picture of what constitutes superior Fitness for Use for a particular product or service can be developed. Further, understanding the overall priority of the characteristics, developers are provided guidance as compromises are needed.

The table below lists thirty-two characteristics quality characteristics along with a short description of each. Some of the characteristics overlap, and the descriptions are certainly broad enough to allow for interpretation or misinterpretation. The list is shown in alphabetical order to avoid any assumed biases as to which characteristics are more important than others. With today's ever-increasing reliance on software components, many characteristic descriptions appear to be more software-specific. However, most of the characteristics are equally applicable to hardware-only products.

A Fitness for Use Tool and a description of its use is included in Chapter 8.02, Assessment Tools, Volume 8 in this collection. As a quick overview, the tool involves reviewing and prioritizing the list of the thirty-two characteristics. Once complete, the characteristics should be divided into three categories that roughly equate to "Must Have," "Nice to Have," and "Low Priority." To be fair, virtually all items could be classified in the "Must Have" category, but working with thirty-two items is practically too cumbersome to be effective. However, even the items in the "Low Priority" category should be reviewed from time to time to make sure that the design elements are not in direct conflict with any of those items that could have an adverse impact on the product or service. Rank ordering the top third and, perhaps, the middle third will provide the guidance required. Although this ranking and prioritizing task sounds easy and straightforward, there are bound to be disagreements between individuals based on their particular responsibilities in the company. For example, a finance person may rate Financial Viability far higher than a developer who might rate Performance higher. A customer service rep might rate maintainability as higher than any of the other characteristics.

Characteristic	Description
Analyzability	The product's software code and documentation and hardware schematics and associated design documents allow simple and straightforward analysis to troubleshoot and determine root causes of reported anomalous operations.
Availability	The product maintains a high degree of accessibility for individual users when required.
Buildability	The product can be built by appropriately trained factory personnel with normal skills, tools, and instructions meeting prescribed quality standards with minimum re-work and no required help from development personnel.
Capability	The product meets its functional objectives as committed to and perceived by the customer.
Compatibility	The product is backward compatible with previous releases or other end-to-end system components to ease customer implementation, upgrades, training, and support issues.
Compliance	The product complies with all applicable industry and government standards and norms in accordance with the actual requirements and spirit of those standards when used as specified or as expected by a user within reasonable limits.
Configurability	The product can be set up in the field by appropriately trained installation personnel and can accommodate likely variations and options with no required help from source experts.
Diagnostics	The product supports straightforward fault and problem isolation by people that are tasked with this responsibility and trained accordingly.
Documentation	The product is fully documented and contains accurate and complete descriptions that are usable by all of its intended audiences.
Expandability	The product can "do more of the same" (i.e. “grow”) within its clearly stated limits as the need arises without undue reconfiguration or changes.
Extendibility	The product can address different but related functions as new requirements change or emerge.
Financial Viability	The product’s overall cost of ownership represents a sound value for the customer and meets their required rate of return or other financial objectives while also meeting the suppliers cost goals.
Flexibility	The product's architecture and implementation contain a high degree of flexibility to accommodate logical future requirements or shifts in priorities without undue re-designs of its core structure.
Graceful Degradation	When over-taxed, the product is likely to alert users and go into self-limiting operation or gracefully shut down avoiding sudden and catastrophic failures.
Installability	The product can be installed by appropriately trained personnel with no need for direct support from development or factory personnel using documented procedures.
Integratability	The product can be implemented in conjunction with other components to provide a viable end-to-end solution for the business opportunity that is being addressed.

Characteristic	Description
Interoperability	The product can co-exist effectively with similar systems and provides a consistent experience/usage model for the user that helps them address their overall requirements.
Maintainability	The product can be maintained by designated and trained maintenance personnel following clearly established troubleshooting procedures.
Modularity	The product's design is sufficiently modular so that changes to one portion of the system have known and reasonable impacts on other portions of the system.
Optionability	The product can be easily configured (either in the factory or the field as appropriate) to meet or hide certain features to aid in its overall usability and applicability for its intended use.
Performance	The product meets its readily measured numerical objectives at a satisfactory level.
Predictability	There is a high degree of confidence that the product's performance can be characterized when it is subjected to new or expanded conditions.
Recoverability	The product can automatically resume operation after faults and anomalies are corrected with little or no loss of data.
Reliability	The product maintains a high degree of dependability for all users over a prolonged period of time.
Resilience	The product can withstand minor or uncommon errors or anomalies without causing catastrophic failures, lockouts, or re-starts.
Safety	The product, when used in its intended application, represents no, or very limited, clearly defined and documented risk to the user, observer, or other components and meets all required and implied industry and government standards.
Stability	The product remains operational when planned or unplanned changes occur whether caused by the user, the product itself, or some external event in the operational environment.
Supportability	The product can be remotely diagnosed, configured, and operated with little or no need to travel to the site.
Testability	The product can be easily tested to insure conformance with its stated specifications.
Traceability	The product’s characteristics and operation can be related directly to its stated and agreed upon requirements.
Usability	The product is "easy to use" as defined by each targeted user for their specific needs after following the specified training regimen.
Vulnerability	The product has built-in safeguards to protect it from malicious attacks or unintended commands from users or other interconnected systems.

Although reviewing and prioritizing the list of characteristics may seem to be a burdensome task and difficult to reach consensus across the organization, prioritization will, indeed, occur. If not done on a macro level, individuals, based on their own point of view, will make prioritization decisions as a normal part of their daily tasks. It is highly unlikely that their individual decisions will be in agreement with others.