IDEA GROUP PUBLISHING **ITB12430**

701 E. Chocolate Avenue, Suite 200, Hershey PA 17033-1240, USA
Tel: 717/533-8845; Fax 717/533-8661; URL-http://www.idea-group.com

This paper appears in the publication, *Project Management for Modern Information Systems*
by Dan Brandon© 2006, Idea Group Inc.

Chapter X

Managing Quality

Well done is better than well said.

(Benjamin Franklin)

Quality is defined by PMI as conformance to requirements, specifications, standards and fitness for use (PMI, 2000). Part of this definition is mostly quantitative, namely the requirements, specifications, and standards (assuming that these three things have been carefully and correctly itemized), but "fitness for use" is mostly qualitative. Because of this qualitative portion of the definition, the quality of the product which is the subject of the project may be an area for potential conflict between the performing organization and benefiting organization. *In fact, as a project proceeds, quality is the most difficult area to keep on track, not because of its complexity, but because the project team may compromise it when a crunch arises* (Hallows, 1998).

Quality Management

Quality is more than just conformance to requirements. A good set of requirements is often difficult to devise for IT systems. If there are defects in the requirements (which there usually are), one could conceivably have a high-quality system that is useless. In addition, simply meeting the requirements will not guarantee that the customer or end users are satisfied with the product. Quality must also be distinguished from grade. A low-grade product may have just as much quality as a high-grade product. Different grades of a product are generally created for different classes of service, and they generally have different unit prices.

Most people think of bugs when they think of quality in the context of IT systems. The term *bug* was originally coined by Dr. Grace Hopper, who developed the COBOL language. She had found that a computer crash was due to a moth that lodged inside of the hardware. For large IT systems of 1,000 function points or more, total defects will average five bugs per function point (Jones, 1994). Even with high-quality software development, for every 500 or so lines of procedural C-like code there is one bug (Linger, 1994). Your electric razor has dozens of bugs, your TV may have hundreds of bugs, and your car may have thousands of bugs. Most of these bugs are encountered only when a certain set of circumstances arises. Software bugs cost as much as $60 billion annually, as estimated by the National Institute of Standards and Technology. Inside sources reported that Microsoft Windows 2000 was released with 63,000 potential defects (Foley, 2000; ZDNet, 2000). Software bugs increase in number in our modern world as our dependency on IT deepens and our reliance on automation and embedded software grows.

In addition to bugs, however, the basic definition of *quality* needs to be further extended when we consider completion criteria and satisfaction criteria. A more complete definition of *quality* for IT projects should include:

- Conforms to requirements and specifications
- Meets "customer expectations"
- Is defect-free
- Is highly usability
- Is consistent with adopted standards
- Is reliability (does it do it right all the time)
- Is robust (can handle invalid/unusual data and usage)
- Is testable
- Is auditable
- Is maintainable and readable
- Is secure
- Is recoverable
- Is appropriately documented (external and internal)
- Is efficient (with respect to speed, storage, clicks, keystrokes, and other resources)
- Is platform independent (portability)
- Is flexible and adaptable

A less specific but official list of quality attributes is found in IEEE 83: portability, reliability, efficiency, accuracy, error, robustness, correctness. Another official list of quality attributes is found in ISO 9126: functionality, reliability, usability, efficiency, maintainability, and performance.