Quality Concepts

LECTURE # 37

Objectives

- After completing this chapter you will understand:
- What is software quality
- Why it is important
- The impact of different quality factors over software product
- The concept of Cost of Quality

Software Quality

- Today, software quality remains an issue, but who is to blame?
- Customers blame developers, arguing that sloppy practices lead to low-quality software.
- Developers blame customers (and other stakeholders), arguing that irrational delivery dates and a continuing stream of changes force them to deliver software before it has been fully validated.

Quality

- The American Heritage Dictionary defines quality as
  - "a characteristic or attribute of something."
- For software, two kinds of quality may be encountered:
  - Quality of design encompasses requirements, specifications, and the design of the system.
  - Quality of conformance is an issue focused primarily on implementation.
  - User satisfaction = compliant product + good quality + delivery within budget and schedule

Software Quality

- Software quality can be defined as:
  - An effective software process applied in a manner that creates a useful product that provides measurable value for those who produce it and those who use it.

Effective Software Process

- An effective software process establishes the infrastructure that supports any effort at building a high-quality software product.
- The management aspects of process create the checks and balances that help avoid project chaos—a key contributor to poor quality.
- Software engineering practices allow the developer to analyze the problem and design a solid solution—both critical to building high quality software.
- Finally, umbrella activities such as change management and technical reviews have as much to do with quality as any other part of software engineering practice.

Useful Product

- A useful product delivers the content, functions, and features that the end-user desires
- But as important, it delivers these assets in a reliable, error free way.
- A useful product always satisfies those requirements that have been explicitly stated by stakeholders.
- In addition, it satisfies a set of implicit requirements (e.g., ease of use) that are expected of all high-quality software.

Adding Value

- By adding value for both the producer and user of a software product, high quality software provides benefits for the software organization and the end-user community.
- The software organization gains added value because high quality software requires less maintenance effort, fewer bug fixes, and reduced customer support.
- The user community gains added value because the application provides a useful capability in a way that expedites some business process.
- The end result is: 1. greater software product revenue, 2. better profitability when an application supports a business process, and/or 3. improved availability of information that is crucial for the business

Why Quality is Important?

- You can do it right, or you can do it over again.
- If a software team stresses quality in all software engineering activities, it reduces the amount of rework that it must do.
- That results in lower costs, and more importantly, improved time-to-market.

Garvin's Quality Dimensions

- David Garvin [Gar87] suggests that quality should be considered by taking a multidimensional viewpoint that begins with an assessment of conformance and terminates with a transcendental (aesthetic) view.
- He suggested Garvin's eight dimensions of quality that were not developed specifically for software, they can be applied when software quality is considered.

Garvin's 8 Dimensions of Quality

- Performance quality.
- Feature quality.
- Reliability.
- Conformance.
- Durability.
- Serviceability.
- Aesthetics.
- Perception