

Chapter 21

■ **Software Quality Assurance**

Slide Set to accompany

Software Engineering: A Practitioner's Approach, 8/e

by Roger S. Pressman and Bruce R. Maxim

Slides copyright © 1996, 2001, 2005, 2009, 2014 by Roger S. Pressman

For non-profit educational use only

May be reproduced ONLY for student use at the university level when used in conjunction with *Software Engineering: A Practitioner's Approach, 8/e*. Any other reproduction or use is prohibited without the express written permission of the author.

All copyright information MUST appear if these slides are posted on a website for student use.

Comment on Quality

- Phil Crosby once said:
 - The problem of quality management is not what people don't know about it. The problem is what they think they do know . . . In this regard, quality has much in common with sex.
 - *Everybody is for it.* (Under certain conditions, of course.)
 - *Everyone feels they understand it.* (Even though they wouldn't want to explain it.)
 - *Everyone thinks execution is only a matter of following natural inclinations.* (After all, we do get along somehow.)
 - *And, of course, most people feel that problems in these areas are caused by other people.* (If only they would take the time to do things right.)

Elements of SQA

- **Standards**
- **Reviews and Audits**
- **Testing**
- **Error/defect collection and analysis**
- **Change management**
- **Education**
- **Vendor management**
- **Security management**
- **Safety**
- **Risk management**

Role of the SQA Group-I

- **Prepares an SQA plan for a project.**
 - The plan identifies
 - evaluations to be performed
 - audits and reviews to be performed
 - standards that are applicable to the project
 - procedures for error reporting and tracking
 - documents to be produced by the SQA group
 - amount of feedback provided to the software project team
- **Participates in the development of the project's software process description.**
 - The SQA group reviews the process description for compliance with organizational policy, internal software standards, externally imposed standards (e.g., ISO-9001), and other parts of the software project plan.

Role of the SQA Group-II

- **Reviews software engineering activities to verify compliance with the defined software process.**
 - identifies, documents, and tracks deviations from the process and verifies that corrections have been made.
- **Audits designated software work products to verify compliance with those defined as part of the software process.**
 - reviews selected work products; identifies, documents, and tracks deviations; verifies that corrections have been made
 - periodically reports the results of its work to the project manager.
- **Ensures that deviations in software work and work products are documented and handled according to a documented procedure.**
- **Records any noncompliance and reports to senior management.**
 - Noncompliance items are tracked until they are resolved.

SQA Goals (see Figure 21.1)

- **Requirements quality.** The correctness, completeness, and consistency of the requirements model will have a strong influence on the quality of all work products that follow.
- **Design quality.** Every element of the design model should be assessed by the software team to ensure that it exhibits high quality and that the design itself conforms to requirements.
- **Code quality.** Source code and related work products (e.g., other descriptive information) must conform to local coding standards and exhibit characteristics that will facilitate maintainability.
- **Quality control effectiveness.** A software team should apply limited resources in a way that has the highest likelihood of achieving a high quality result.

Formal SQA

- Assumes that a rigorous syntax and semantics can be defined for every programming language
- Allows the use of a rigorous approach to the specification of software requirements
- Applies mathematical proof of correctness techniques to demonstrate that a program conforms to its specification

Statistical SQA

**Product
& Process**

**Collect information on all defects
Find the causes of the defects
Move to provide fixes for the process**

measurement

***... an understanding of how
to improve quality ...***

Statistical SQA

- Information about software errors and defects is collected and categorized.
- An attempt is made to trace each error and defect to its underlying cause (e.g., non-conformance to specifications, design error, violation of standards, poor communication with the customer).
- Using the Pareto principle (80 percent of the defects can be traced to 20 percent of all possible causes), isolate the 20 percent (the *vital few*).
- Once the vital few causes have been identified, move to correct the problems that have caused the errors and defects.

Six-Sigma for Software Engineering

- The term “six sigma” is derived from six standard deviations—3.4 instances (defects) per million occurrences—implying an extremely high quality standard.
- The Six Sigma methodology defines three core steps:
 - *Define* customer requirements and deliverables and project goals via well-defined methods of customer communication
 - *Measure* the existing process and its output to determine current quality performance (collect defect metrics)
 - *Analyze* defect metrics and determine the vital few causes.
 - *Improve* the process by eliminating the root causes of defects.
 - *Control* the process to ensure that future work does not reintroduce the causes of defects.

Software Reliability

- A simple measure of reliability is *mean-time-between-failure* (MTBF), where

$$\text{MTBF} = \text{MTTF} + \text{MTTR}$$

- The acronyms MTTF and MTTR are *mean-time-to-failure* and *mean-time-to-repair*, respectively.
- *Software availability* is the probability that a program is operating according to requirements at a given point in time and is defined as

$$\text{Availability} = [\text{MTTF}/(\text{MTTF} + \text{MTTR})] \times 100\%$$

Software Safety

- *Software safety* is a software quality assurance activity that focuses on the identification and assessment of potential hazards that may affect software negatively and cause an entire system to fail.
- If hazards can be identified early in the software process, software design features can be specified that will either eliminate or control potential hazards.

ISO 9001:2008 Standard

- ISO 9001:2008 is the quality assurance standard that applies to software engineering.
- The standard contains 20 requirements that must be present for an effective quality assurance system.
- The requirements delineated by ISO 9001:2008 address topics such as
 - management responsibility, quality system, contract review, design control, document and data control, product identification and traceability, process control, inspection and testing, corrective and preventive action, control of quality records, internal quality audits, training, servicing, and statistical techniques.