
Appendix D

Software Quality Management

Software Quality Management

- **Quality planning:** Identifying which quality standards are relevant to the project and how to satisfy them.
- **Quality assurance:** Periodically evaluating overall project performance to ensure the project will satisfy the relevant quality standards.
- **Quality control:** Monitoring specific project results to ensure that they comply with the relevant quality standards

1

Software Quality Plan

1. *Defines the Quality Goals*
 - A goal is SMART: Specific, Measurable/Testable, Attainable, Relevant, Time-bound
2. *Realistic about where defects come from*
3. *Selects appropriate detection and prevention methods*
 - **Danger of Lack of Goals:** Repair-service behaviour: Without any clear idea of what the benchmarks are, we go in search of things that are broken and our goal becomes fixing them
 - **Know-how behaviour:** We often don't solve the problems that need to be solved but the ones we know how to solve. There is No justification for any quality action.

2

ISO/IEC 9126-1 - Quality Model

- Functionality
 - Suitability, Accuracy, Interoperability, Security
- Reliability
 - Maturity, Fault tolerance, Recoverability
- Usability
 - Understandability, Learnability, Operability
- Efficiency
 - Time behaviour, Resource utilization
- Maintainability
 - Analysability, Changeability, Stability, Testability
- Portability
 - Adaptability, Installability, Conformance, Replaceability

3

Items in a Software Quality Plan

- "Standards" are instruction documents that detail how a particular aspect of the project must be undertaken. There can be no deviation from "Standards" unless a formal variation process is undertaken, and approval granted.
- "Guidelines" are not compulsory. They are intended to guide a project rather than dictate how it must be undertaken. Variations do not require formal approval.
- "Checklists" are lists that can be used as a prompt when undertaking a particular activity. They tend to be accumulated wisdom from many projects.

4

Table of Contents for a Quality Assurance Plan

- 1.0 Draft Quality Assurance Plan
 - 1.1 Introduction
 - 1.2 Purpose
 - 1.3 Policy Statement
 - 1.4 Scope
- 2.0 Management
 - 2.1 Organizational Structure
 - 2.2 Roles and Responsibilities
 - 2.2.1 Technical Monitor/Senior Management
 - 2.2.2 Task Leader
 - 2.2.3 Quality Assurance Team
 - 2.2.4 Technical Staff
- 3.0 Required Documentation

5

Table of Contents for a Quality Assurance Plan

- 4.0 Quality Assurance Procedures
 - 4.1 Walkthrough Procedure
 - 4.2 Review Process
 - 4.2.1 Review Procedures
 - 4.3 Audit Process
 - 4.3.1 Audit Procedures
 - 4.4 Evaluation Process
 - 4.5 Process Improvement
- 5.0 Problem Reporting Procedures
 - 5.1 Noncompliance Reporting Procedures
- 6.0 Quality Assurance Metrics
- Appendix
 - Quality Assurance Checklist Forms

6

Quality Assurance (QA)

- **Quality Assurance (QA)** is fault prevention through process design and auditing
- **Examples:** Templates, checklists, guides
- Quality assurance includes all the activities related to satisfying the relevant quality standards for a project.
- Another goal of quality assurance is continuous quality improvement.
- **Benchmarking** generates ideas for quality improvements by comparing specific project practices or product characteristics to those of other projects or products within or outside the performing organization.
- A **quality audit** is a structured review of specific quality management activities that help identify lessons learned that could improve performance on current or future projects.

7

Qa versus QC

- **Quality Assurance (QA)** is fault prevention through process design and auditing
- **Creating processes, procedures, tools, etc. to prevent faults from occurring**
- **Examples:** Templates, checklists, guides
- **Quality Control (QC)** is fault/failure detection through static and/or dynamic testing of artifacts
- **Examining the artifact against pre-determined criteria to measure conformance**
- **Examples:** Code testing, peer reviews, beta trials

8

Quality Control

- The main outputs of quality control are:
 - Acceptance decisions
 - Rework
 - Process adjustments
- Some tools and techniques include:
 - Pareto analysis: involves identifying the vital few contributors that account for the most quality problems in a system.
 - Statistical sampling
 - Six Sigma
 - Quality control charts

9

Benchmarking

- A *“benchmark”* is a reference or measurement standard used for comparison.
- *“Benchmarking”* is the continuous activity of identifying, understanding and adapting best practice and processes that will lead to superior performance.
- Benchmarking measures an organisation’s products, services and processes, to establish targets, priorities and improvements, leading to competitive advantage and/or cost reductions.
- Benchmarking is a tool to identify, establish, and achieve standards of excellence;
- a structured process of continually searching for the best methods, practices, and processes;
- the practice of measuring your performance against world-class organisations;
- a disciplined method of establishing performance goals and quality improvement projects based on industry best practices;
- a continuous process of measuring products, services, and practices against the company's toughest competitors or those companies renowned as industry leaders;

10

Benchmarking

- There are four basic types of benchmarking:
- **Internal** - a comparison of internal operations and processes
- **Competitive** - specific competitor to competitor comparisons for a product or function
- **Functional** - comparisons of similar functions within the same broad industry, or to industry leaders
- **Generic** - comparisons of business processes or functions that are very similar, irrelevant of the industry

Bibliography

- 1 Bill Curtis, William E. Hefley, Sally A. Miller, *People Capability Maturity Model (P-CMM) Version 2.0*, Carnegie Mellon Software Engineering Institute, 2001.
(https://resources.sei.cmu.edu/asset_files/MaturityModule/2001_008_001_435287.pdf)
- 2 Jeff Tian, *Software Quality Engineering*, Wiley-Interscience, 2005.
- 3 Michael Fagan, *Design and code inspections to reduce errors in program development*, 1976.
(<http://www.mfagan.com/pdfs/ibmfagan.pdf>)
- 4 James Whittaker, Jason Arbon, Jeff Carollo, “How Google Tests Software”, Addison Wesley, 2012.