

# Chapter 4: Project Management Quality Control Mechanism

The IT project management quality control mechanism involves the establishment of efficient system development practices and monitors those practices throughout the life cycle. This results in the development of high-quality, cost-effective IT system products. The IT project manager should plan to provide and enforce the quality principles. The manager establishes the quality assurance section (QAS) to control the established quality principles. The manager makes it clear that the quality is the responsibility of all those who are involved in the system development and maintenance and not just the QAS.

## System Quality Management

System quality management is the prime responsibility of the IT project manager. The manager establishes a quality assurance (QA) team during the initial stages of project management planning. The QA team forms a section and assigns duties and responsibilities to their members who will measure the quality parameters at various stages in the system development and maintenance. The QAS members control the quality according to the established standards and procedures. Their responsibilities include the following:

- Prepare the QA plan
- Attend system hardware and software meetings, walk-throughs, and reviews to ensure that the products do what they are supposed to do and conform to the established quality parameters
- Measure the products with the established standards and procedures
- Evaluate tests

Test plans

Test procedures

Test scenarios

Use cases for testing

- Evaluate deliverable documents
- Submit recommendations to improve the quality
- Report the findings to the project manager
- Establish and follow the QAS checklist and guidelines
- Evaluate quality

## System Quality Characteristics

*Quality* is defined as the presence of desired characteristics and the absence of undesirable characteristics in the IT system's products. The manager decides about these characteristics when consulting with customers, users, and stakeholders.

The manager must identify and measure the quality characteristics of each system product during the IT system development phases. This starts during the system requirement analysis stage, continues through each successive phase, and culminates in the operation and maintenance phases. Quality deals with how well the system product performs its functions in the operating environment over time. Each characteristic needs metrics to measure quality. Metrics include indicators that provide an early warning for detecting the system's

quality problems.

### Quality Metrics

Quality metrics provide a quantitative measure of the degree to which the system possesses and exhibits certain quality characteristics. QAS metrics are as follows:

- Management metrics, which include quality, cost, and schedule
- Process metrics, which include effectiveness and efficiency
- Product metrics, which include defects, complexity, and maintainability

The overall system quality is a function of the metrics values. The product functions are summarized as hardware and software failure, reliability, recovery, modifiability, availability in time, and maintainability. Suggested quality metrics is the degree to which the system is portable, accurate, complete, robust, consistent, accessible, and structured:

- IT system software is portable if it can be operated easily on a computer configuration other than its current one. Its portability consists of hardware independence, initiation of its own storage area, and use of data types that are hardware independent.
- A system is accurate if it meets the customer's demand for accuracy. The system does what it is supposed to do in accordance with the requirements. This confirms that the development of hardware and software followed established IT standards, data typing, timing constraints, and runtime parameters.
- System software is complete if all of its components and units are present in a computer software configuration item (CSCI) and hardware configuration item (HWCI). Each component is fully developed with the established standard and well documented. This requires that the system software be well-commented and include a graphic model, interfaces, and an interoperability network.
- System software is robust if it continues to perform despite a violation of assumptions in its specification. Some of the precautionary measures for robust system software are error-handling concepts, error exceptions, input data for range errors, and availability of default values.
- System software is consistent to ensure uniform notation, terminology, and symbology. It is consistent internally and externally to the extent that the content is traceable to the requirements, forward and backward. The manager achieves consistency by use of pretested generic packages, name-of-data dictionary elements, objects, database files, reusable modules and packages that are defined in the previous phases of the system software development, sets of global variables, a common area, and real and integer data types.
- System software is accessible to facilitate the selective use of its components for reusability, efficiency, and testability.
- System software is structured when it involves a definite pattern of organization of its components, follows a systematic approach, follows set standards, and is uniform and easy to understand and follow.

### Quality Indicators

Quality indicators provide insight into the adequacy of the system software specifications. The manager can use quality indicators effectively throughout the system life cycle. They lay the foundation for quality software development, which is crucial to obtain a reliable and maintainable IT system. The manager can use the values determined from the components of the completeness indicator to identify specific problem areas within the system software specifications and design.

## Quality Parameters

Quality indicators assist in identifying the following:

- How well system level requirements have been translated into software requirement specifications and hardware requirement specifications
- Any hidden or missing linkages in the system design
- How well the customers', users', and stakeholders' requirements have been translated into the design
- The degree of implementing testable software products
- System stress during testing
- The percentage of hardware and software errors detected during system testing
- System documentation meeting the need of the customer, user, and stakeholder in maintenance

Quality indicators can be summarized as follows:

- **Completeness.** Depends on the number of defined functions, data items, referenced functions, decision points, condition options, calling parameters that do not agree with the defined parameters, and the data references that have no destination
- **Design structure.** Depends on the number of system software packages dependent on the source of input or destination of output, the number of system software packages dependent on prior processing, and the number of database items
- **Detection of defects.** Depends on analysis of the system requirement, system design, hardware components, software requirement, software design, and code inspection process
- **Detection of faults.** Depends on test results and the analysis of the system requirement, system design, hardware components, software requirement, software design, and code inspection process
- **Test coverage.** Depends on the system structure and system requirements
- **Test sufficiency.** Depends on the number of faults, system components integrated, software packages in a CSCI and HWCI, and faults detected to date
- **Documentation.** Depends on system software and hardware documentation in accordance with established standards

## Quality Parameters

The IT system manager establishes quality parameters to measure the quality at various phases and stages of the development and maintenance of the system. The manager states the methods and techniques used to maintain quality during the development of the system plan and in response to the request for proposal. The manager establishes the quality parameters to measure the quality of the following items:

- Hardware development phases
- Software development phases
- Established standards
- Guidelines and procedures
- Products developed at each phase
- Walk-through
- Reviews
- Inspection
- Monitoring of the progress
- Enforcement of standards and procedures
- Deliverable hardware, software source, and object code
- Deliverable document

Quality is a relative term. The quality changes with changes in the quality parameters. The following are the quality parameters that the system developers should check:

## Technical Reviews

- Accuracy
- Correctness
- Reliability
- Testability
- Maintainability
- Portability
- Reusability
- Efficiency
- Modularity
- Ease of use
- Understandability
- Adherence to standards
- Error-free products
- Traceability

## Technical Reviews

A technical review is a disciplined examination of a document that is extensive and provides critical evaluation. The technical review helps developers do the following for an IT project:

- Find errors in the system development in early stages
- Track the progress of the project
- Provide feedback
- Educate the reviewers
- Pass the products of each phase to the next phase

The manager can base technical reviews on either the established standards or a methodology defined by his or her organization. A checklist in the standard guides the preparation of review participants and helps the professionals organize and conduct reviews. The participants in a technical review consist of a review leader, a recorder, a group of reviewers, and producers of documents. The reviewed document is either accepted or rejected in a review. The experiences of the participants attending a review provide feedback concerning the correctness of the document. The QAS attends the review to ensure the quality and correctness of the document in accordance with the system's requirements.

The QAS also ensures that all action items on the review report are addressed. The QAS analyzes the data on the review forms and classifies defects to improve the system development and review processes. These data include error-detection efficiency, cost-effectiveness, reliability assurance techniques, and tools supporting the reviews.

## Technical Walk-Through

The technical walk-through determines and ensures the correctness and quality of the documentation. The manager should conduct it at all phases of the system development and maintenance. Whether formal or informal, technical or nontechnical, a walk-through creates a friendly, professional atmosphere different than that of a meeting and review. The result of a walk-through is the creation of an error-free, cost-effective document that performs its required functions.

The QAS ensures that technical walk-throughs are conducted as many times as needed to create a quality product.

## System Quality Improvement Process

The system quality improvement process continues during the life cycle of the IT system development and maintenance. The members of QAS are quality professionals with many years of experience in hardware and software development. They should be trained to know more about IT systems than other developers. The project manager should select for QAS those professionals who believe in quality and keep on learning and practicing quality. Sometimes it is better to rotate developers from the QAS to other sections of the system development to gain a variety of experiences. This also gives other professionals a chance to learn more about quality. The following are the characteristics of a good QAS professional who can improve the quality process:

- Believes in quality
- Improves QA discipline
- Shows experiences in system development and maintenance phases
- Has experience to introduce quality in documenting the IT system
- Has a positive and innovative attitude
- Supports the quality organization's QA program
- Interprets guidance and directives for the QAS
- Prepares QA policies and procedures
- Applies imagination and creativity for improving the system's quality
- Shows responsibility and dependability in improving quality
- Complies with quality policies and guidance
- Is sound in technical accuracy and completeness
- Has good interpersonal relations skills
- Is good at representing the organization at system reviews and meetings
- Has good communication skills
- Performs walk-throughs, reviews, inspections, and internal audits

The system quality process starts during the planning and understanding of the system requirements and cumulates at the completion of the project. The QAS members conduct research and study parameters to improve the quality of the system.

## System Software Quality and Interoperability

The project manager forms a system software quality and interoperability working group. This working group considers how to best apply electronic commerce (e-commerce) off-the-shelf products to enhance their operations as part of their e-business initiative. The working group develops series of questions that provide the IT manager with information for better understanding of e-commerce and e-business software and interoperability. The working group covers the following questions for software quality, interoperability with hardware and software, and management of user expectations ([www.E-Interop.com](http://www.E-Interop.com)):

### *A. Software quality*

1. What are the most important attributes that define quality in your software product?
2. What is your organization's system software development process in the following areas?
  - a. Requirement determination (How do you decide what functions will be included in the software product?)
  - b. Testing and validation program
  - c. QA program

## System Quality Improvement Process

- d. Creation and validation of technical documentation
- e. Relation of product function to user training
3. How does your organization Ensure [InsideOut] software quality?
4. Do you measure quality by Using any of the following indicators or benchmarks?
  - a. Certification by independent laboratories
  - b. Number of calls for technical support
  - c. Number of errors found by customers
5. How does your organization address risk management for a software product?
  - a. How does your organization identify potential risk factors and assess their possible influences?
  - b. What is the tolerance for potential risk factors? Are there general guidelines for the amount or level of risk or liability that your organization is willing to accept?
  - c. c. What are your risk mitigation and/or risk control activities?
6. How does your organization measure customer satisfaction for your software product, documentation, support services, and user training?
7. When you organization makes purchasing decisions for system software, what do you look for? Rank the following criteria in order of importance:
  - a. Price
  - b. Robust functionality
  - c. Reputation for quality
  - d. Interoperability

### ***B. Interoperability with hardware and software***

1. What are your organization's interoperability needs to support e–business? How are they expected to change in the future?
  - a. Characterize the communities with whom interoperability is required (e.g., internally, externally).
  - b. Characterize the nature of the interoperability required with each community. If appropriate, consider interoperability requirements from two perspectives: minimal level (must have) and desired level (would like to have). How important is it to get beyond the minimal level?
2. To what extent and how has your organization solved interoperability (internally and externally) requirements to enable e–business with respect to the following?
  - a. Data and document interoperability (discovery, access, and semantics)
  - b. Application interoperability
  - c. Security
  - d. Network and platform interoperability
3. What are the major residual interoperability issues?
4. To what extent has interoperability been provided within product lines, across product lines, through middle–ware solutions, and through standards?
5. To what extent have you found that the following are important in solving interoperability issues?
  - a. Software standardization (within and/or across product lines)
  - b. Platform standardization
  - c. Open system standardization

## Software Development Capability Maturity Model

- d. Architectures and testing
  - e. Middle-ware
  - f. Extensible markup language (XML)
6. To what extent do you believe technologies such as Java, common object request broker architecture (CORBA), and component-based architectures will solve interoperability issues? What is the likelihood? What is the time frame? Are there other key technologies?

### C. *Managing user expectations*

1. What are the user expectations to be satisfied in your project by the system software product? What is your process for determining the requirements for the software product? What is the acquisition and marketing strategy for the software product to meet your requirements?
2. What are the major quality and interoperability deficiencies you experienced? Can these deficiencies be classified as one of the following?
  - a. Bugs (functions that did not work)
  - b. Functions that did not work as expected
  - c. Functions not included as expected or promised
  - d. Poor and inadequate documentation
  - e. Inadequate training

## Software Development Capability Maturity Model

The Software Engineering Institute (SEI) has developed a capability maturity model (CMM) to guide organizations in system software process improvement. This model has contributed to widespread success by helping organizations improve their efficiency in developing quality system software products.

The SEI has defined five levels of CMM as shown in Figure 4–1.

- Level 1 is chaotic.
- Level 2 is repeatable.
- Level 3 is defined.
- Level 4 is managed.
- Level 5 is optimized.

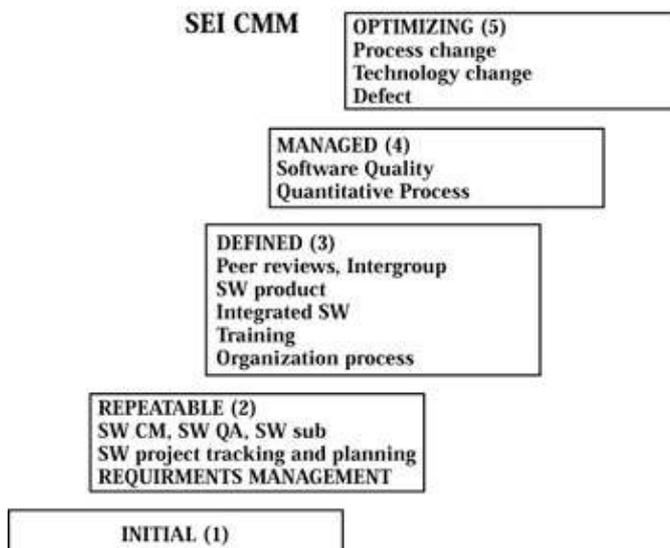


Figure 4–1: Capability maturity model levels

## Software Development Capability Maturity Model

The SEI CMM is used as the standard to assess the maturity of an organization's software development process.

Lately, the people capability maturity model (P-CMM) by SEI is a maturity framework that describes the key elements of managing and developing the work force of an organization. Figure 4-2 shows a model of P-CMM. The P-CMM describes that an organization must focus on people, process, and technology to improve performance. The P-CMM includes practices in such areas as work environment, communication, staffing, performance management, training, compensation, competency development, career development, team building, and culture development in an organization.



Figure 4-2: P-CMM model

The success of the system software (SW-CMM) spawned other CMMs that address a wide range of subjects. An SW-CMM provides an organization with a conceptual framework within which specific processes (e.g., configuration management and quality) can be optimized to efficiently improve the capability of the organization. An SW-CMM provides state-of-the-art practices to do the following:

- Determine the maturity of an organization's processes
- Establish goals for process improvement
- Set priorities for immediate process improvement actions
- Plan for a culture of product or service excellence

By focusing on specific processes, an organization can best leverage the resources for their improvement activities while rallying the organization around specific goals. An SW-CMM can be a road map showing an organization how it can systematically move to more mature levels of performance and do it in more effective and efficient ways. After an objective assessment, an organization can set its goals for increasing the capability of its processes.

An SW-CMM can include processes that span the entire life cycle. Starting with requirement management, the processes can span the breadth of product development, ensuring quality, lean production concepts, and support to the field. Each individual process includes elements that provide basic practices and additional practices that add incremental benefits and maturity. When these processes are sufficiently matured, the organization increases its performance or maturity.

Subsequent to the success of the SW-CMM, other CMMs were developed with SEI support. These CMMs included the systems engineering (SE) CMM and the integrated product development (IPD) CMM. It became apparent in the development of these and other models that they all contained common processes (e.g., configuration management, quality, and requirements management), supporting the various types of functional engineering and systems engineering. Improvements in these common processes could benefit other disciplines. Furthermore, it became apparent that process improvement resources applied to one functional discipline (e.g., software engineering) could be beneficial to another functional discipline. The common elements used in an SW-CMM appraisal could be used for SE appraisal, and there would be no need to redo the appraisal of common elements. In addition, improvement efforts based on a unique CMM could

## QAS Checklist

result in suboptimization, confusion, and potentially unnecessary expenditure of process improvement resources.

Mark D. Schaeffer of the U.S. Department of Defense says that one of the top-priority projects in the SEI is integration of the CMM products for use in single or multiple functional disciplines. This will greatly enhance the efforts of CMM users and protect the resources already invested. Organizations can use their previous CMM process improvement work and tailor their future efforts to their unique organization. The initial common framework effort will be based on the SW-CMM, the SE-CMM, and the IPD-CMM. Other functional disciplines may be added later. The work accomplished to date in SW-CMM, Version 2.0, and the IPD-CMM have been included in the initial CMM integration (CMMI) baseline. In building these CMMI products, the needs of industry and government partners must be understood and met.

## QAS Checklist

The QAS checklist consists of such standards as the International Standards Organization (ISO) and the Institute of Electrical and Electronic Engineering (IEEE) and complies with any other applicable documents:

- Prepare QAS plans, set the tone for quality, and instill quality attitudes among the practitioners.
- Define the purpose and benefits of QAS.
- Explain the elements of a QAS program.
- Explain the procedure steps required for QAS.
- Identify the roles and responsibilities between the project and the QAS.
- Specify QAS reviews, evaluation, and report.
  - ◆ Define quality processes and procedures.
  - ◆ Conduct quality evaluations.
  - ◆ Review products before release.
  - ◆ Maintain quality records and metrics.
  - ◆ Follow up on corrective actions.
- Train the organization in quality processes and procedures.
- Involve evaluation of the project, and attend technical reviews, products reviews, and process reviews.
- Establish a QAS report form, process, policy, and procedure.
- Follow QAS checklists.
- Prepare, submit, and report QA reports.
- Follow the SEI CMM saying: The purpose of QAS is to provide management with appropriate visibility into the process being used by the software project and of the products being built.
- Understand that QAS is an integral part of the IT system development and management process.
- Understand that QAS is a planned and systematic pattern of all actions necessary to provide adequate confidence that the product conforms to established standards.
- Improve quality by appropriately monitoring the system and the development process that produces it.
- Verify full compliance with the established standards and procedures.
- Verify that any variance in the product, process, and standards is brought to the management's attention.
- Review the following major functions of QAS:
  - ◆ Project planning
  - ◆ Requirements process
  - ◆ Design process
  - ◆ Coding practices

## QAS Checklist

- ◆ Software integration and testing
- ◆ System integration and testing
- ◆ In-process review of the management and project control process
- Understand that QAS prevents defects:

Early error detection for increased cost savings

- ◆ Assurance for users, customers, stakeholders, and management of product reliability and maintainability
- ◆ Repeatable evaluation process
- Understand that improving quality reduces system development and maintenance costs.
- Plan, perform, check, and act for continual improvement of performance.

## Section II: Industry Best Practices

### Chapter List

*Chapter 5: IT Project Risk Management*

*Chapter 6: Commercial Best Practices Standards*

*Chapter 7: System Measurement Techniques*

*Chapter 8: Commercial Items*

*IT project managers establish processes to identify, monitor, and manage risks.*