Lesson Proper for Week 16

QUALITY ASSURANCE

The quality assurance team or organization for software provides product evaluation processes and specific quality assurance for effective software engineering methods and use of software tools. The quality team does ensure compliance to software design/development standards and control of work products and changes. The practice of quality management is applied throughout the software design/development processes. It is important that quality attributes become the responsibility of everyone supporting development environments in software companies and military and aerospace programs and projects. The management of quality for software activities is summarized in engineering reviews, change control, or subcontractor audits and compliance to standards, verification, and validation.

Ø **Software Quality Plan**

The software quality plan (SQP) describes and documents the software quality assurance roles and responsibilities to ensure that programs and projects are following procedures and processes defined per development plans and other applicable standards.

This plan provides a documented process for assessing software life-cycle processes and their outputs to obtain assurance that objectives are satisfied; deficiencies detected and evaluated, tracked, and resolved; and software work products and software life-cycle data conformed to intended requirements.

Quality reviews/internal audits are performed to ensure compliance with released processes and AS9100C for measurement, analysis, and improvement activities to be conducted; senior managers must adhere to this review and audit. These activities include examining:

·   Program and project artifacts

·   Released processes and procedures

·   That organizations meet the configuration management requirements of Electronic Industries Alliance (EIA) standards

·   That organizations meet the requirements of AS9100C

Ø **Software Engineering Process Group**

A quality assurance internal organization for software has a relationship with the software engineering process group (SEPG). In programs and projects, this is a group or team with assigned responsibilities for monitoring software process activities.

This group operates with appropriate functional areas and software personnel and serves as the software process improvement center for systems and software engineering. The program and project managers, focal points, configuration management, quality team, and team members make up the group and perform the following functions:

·   Evaluate company and program best practices to promote these best practices to document software engineering processes and procedures

·   Establish and use a process for receiving, evaluating, and acting on and reviewing results for proposed processes, procedures, and technology changes

·   Develop software processes and procedures for the entire life cycle that comply with software engineering standards, comply with contractual requirements, support ISO (International Organization for Standardization) 9001 and AS9100 requirements:

§ AS9100C quality management system (QMS) software requirements

§ AS9100D QMS audit requirements

§ Obtain software program and project manager approvals of software engineering processes and procedure changes released for use by engineering

🞧   **PRODUCT EVALUATION SCHEDULE**

The quality organization or team performs product evaluations to ensure software design/development. Test and integration phases are conducted per a product evaluation schedule (Figure 11.1).

Ø **Senior Managers**

Senior managers lead lower-level program and project managers. All teams are affected when the pressure is on to produce, show results with no mistakes, and deliver work products on time. The customer is always right because if programs and projects do not deliver work products on time, panic and chaos occur. That is when you see the

importance of quality factors come into play, and milestones are achieved and are a success.
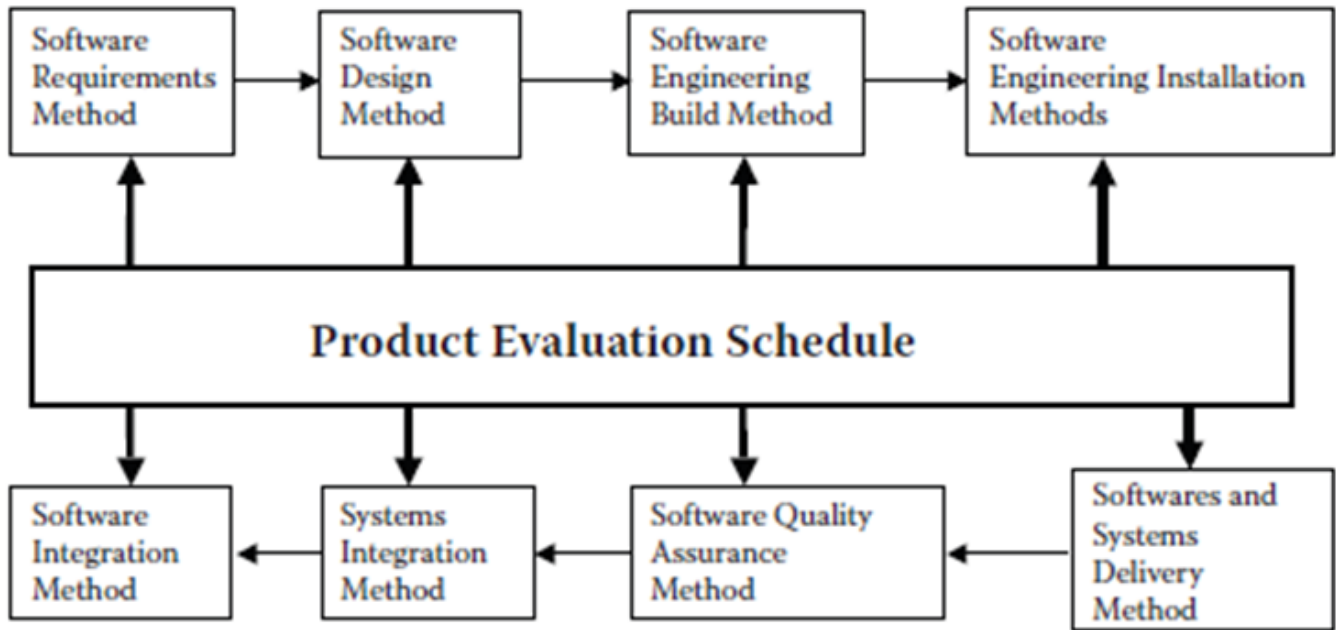

FIGURE 11.1: Product evaluation schedule.

Ø **Program and Project Managers**

The program and project managers are required to provide a vision statement and describe the benefits, goals, and objectives for developing a work product or system for delivery. The work product or system should be the highest priority for customer requirements to identify the importance of business criteria and requirements. When communication or coordination is not provided to the responsible team, the failure point occurs. A defined vision statement is an example list that program and project managers are responsible for producing; it includes issues related to the following:

·   Team objectives

·   Risk mitigation

·   Issues and concerns

·   Root cause (RC) analysis

·   Corrective action (CA) plans

·   Significant accomplishments

The focus of successful program and project managers is the team, processes, and work product. A manager who fails to communicate early in the software development process could pay a heavy price by making wrong decisions. The manager who pays little attention does run into risks if competent methods and tools are not made available to the team. Having a solid program and project plan does not hinder the success of the program and project. Team members need to be highly skilled software people. The talents that the team should apply are:

- Motivation

- Organization skills

- Attention to business goals

- Work ethics

The objectives for product evaluations are established, and solutions, technical constraints, and alternative solutions are always a consideration. The system and software/design developers, along with test, configuration management, and quality teams, define objectives. The product objectives identify the goals for design and development of data, which provides functionality in a quantitative manner. The program and project managers select the best approach with consideration of constraints imposed by delivery deadlines, budget issues, available team members, and technical solutions to problems.

Processes that are understood provide the framework to implement effective plans and procedures for software design/development activities. The framework details the number of tasks, milestones, and applied quality factors to enable activities to adapt to requirements for programs and projects. Aspects of configuration management and quality assurance are important to the independent nature that occurs during the process.

Ø **System and Software Team Participation**

Independence of quality support is ensured by the separation of reporting chains to a level that is independent of a program and project. This independence for the quality team increases the objectivity of the product evaluations, which allows the team to provide a better oversight function and involves the system and software teams. The quality team participation in formal reviews does provide team support for product evaluations, and reviews are scheduled and coordinated within team and life-cycle activities.

Each evaluation or review conducted by the quality team generates reports containing the status of the audit, any findings, observations, and recommendations. Compliance, noncompliance, and opportunities for improvements are documented in quality reports and tracked for closure with support from program and project managers.

ARTIFACTS

Development of system and software work products yields artifacts, including specifications, plans, and procedures. Artifact information associated with quality product evaluations includes software configuration records, testing records, and other artifacts associated with activities, including:

- Audit records (i.e., electronic or paper) associated with product evaluations

- Audit and product evaluation checklists

- Audit results and audit reports

AUDIT FINDINGS

Quality organizations or teams utilize criteria audit finding derived from software plans and internal procedures to perform scheduled product evaluations. Product evaluations include:

·   Review of plans and procedures that oversee programs and projects to determine and select appropriate product evaluation criteria

·   Review and analysis of the results of previous product evaluations

·   An assessment of whether implemented processes are compliant or noncompliant

·   Identification of issues or an opportunity for improvement

·   Additional product evaluations required

The results of each product evaluation are recorded in evaluation plans and are added into databases recording summary information from performing an evaluation. The quality team uses these evaluations to indicate if processes are compliant, noncompliant, or there is an opportunity for improvement. Other information may be developed as needed to address team activities or processes. Information on CAs for follow-up of deficiencies reviewed and discovered during product evaluations are also maintained in the applicable databases. Quality team members generate reports weekly or monthly and are provided to managers.

The weekly/monthly team report data are a record of any issues, noncompliance, opportunities for improvement, and so on that were identified and documented during product evaluations, and the status of all open items requiring a CA is recorded and logged into required databases. Metrics are collected weekly or monthly.

CORRECTIVE ACTIONS

A corrective action is required to eliminate or mitigate the cause of a detected nonconformity or other undesirable situations to prevent recurrences during product evaluation. Two types of a CA initiated are as follows:

·   The root cause (RC) requires RC analysis and actions taken to address the analysis.

·   The immediate action (IA) is taken to address a direct cause and prevent recurrence of a specific nonconformity.

Collective analysis is performed periodically to monitor adverse trends of detected nonconformities and undesirable situations that may not have been addressed by an RC or IA analysis. Results are recorded and provided for senior manager reviews of progress/status and overall process performance. The review may be accomplished through a corrective action board (CAB). Appropriate reviews of the CA and IA should be conducted regularly to:

· Monitor progress/status and overall process performance (e.g., timeliness, efficiency, effectiveness)

· Review adverse indicators and trends

· Resolve issues or elevate them to the proper forum or level of management for resolution Quality gates come into play to ensure process and work products are compliant (Figure 11.2).
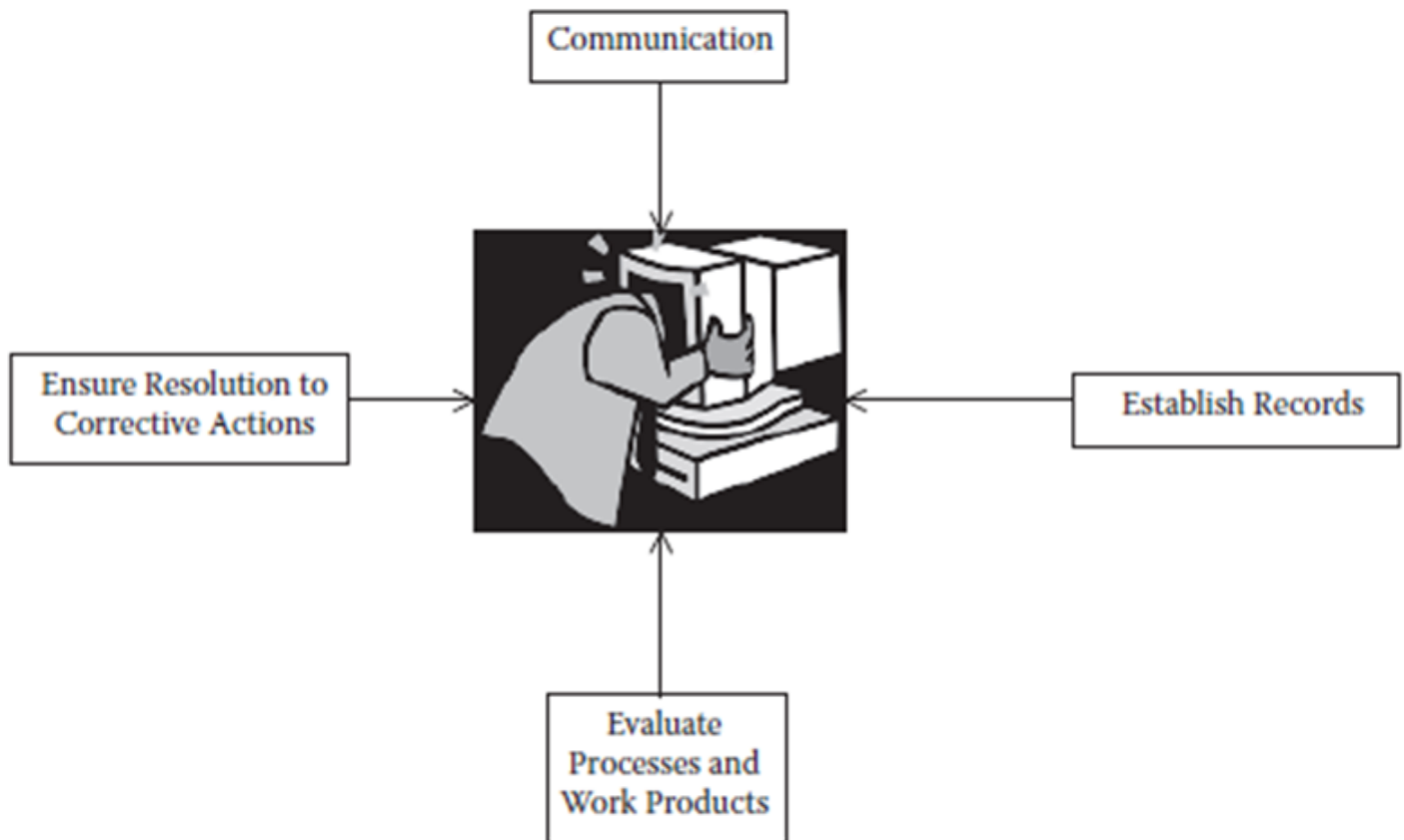

FIGURE 11.2: Quality gates.

Ø **Corrective Audit Plan**

The product evaluations performed on deliverable work products produced during each product development phase ensure compliance to requirements. Problem definition, investigation, RC analysis, software design/development, and implementation are verified in the corrective audit plan (CAP).

QUALITY METRICS

In multiple programs and projects, software engineering is required to produce high-quality systems within a defined time frame to ensure reaching customer expectations. To achieve this requirement, effective methods and current software tools should be used to produce quality metrics. Trained senior managers and program and project managers measure if high quality is to be reviewed. The quality of a system, application, or work product is only as good as the requirements that describe problems and test results that are discovered early in the process.

The quality metrics collected by software engineering to ensure program and project's delivery schedules, what is in-work, and completed product evaluations. These types of metrics provide an indication of the effectiveness of a software engineer, test teams, configuration management, and software quality personnel. An example of quality metrics is shown in Figure 11.3.

QUALITY MANAGEMENT SYSTEM

The QMS is required to have processes documented and executed with knowledgeable people and teams. At times, metrics are reviewed and monitored to ensure processes are showing improvement. Customer focus is QMS and provides the framework to say; what you do, do what you say, prove it, and show improvement. The standards for QMS are AS9100, AS9100C, AS9100D, SAE AS9110, and ISO 900, which are the models for:

· Quality requirements

· Design and development

· Production

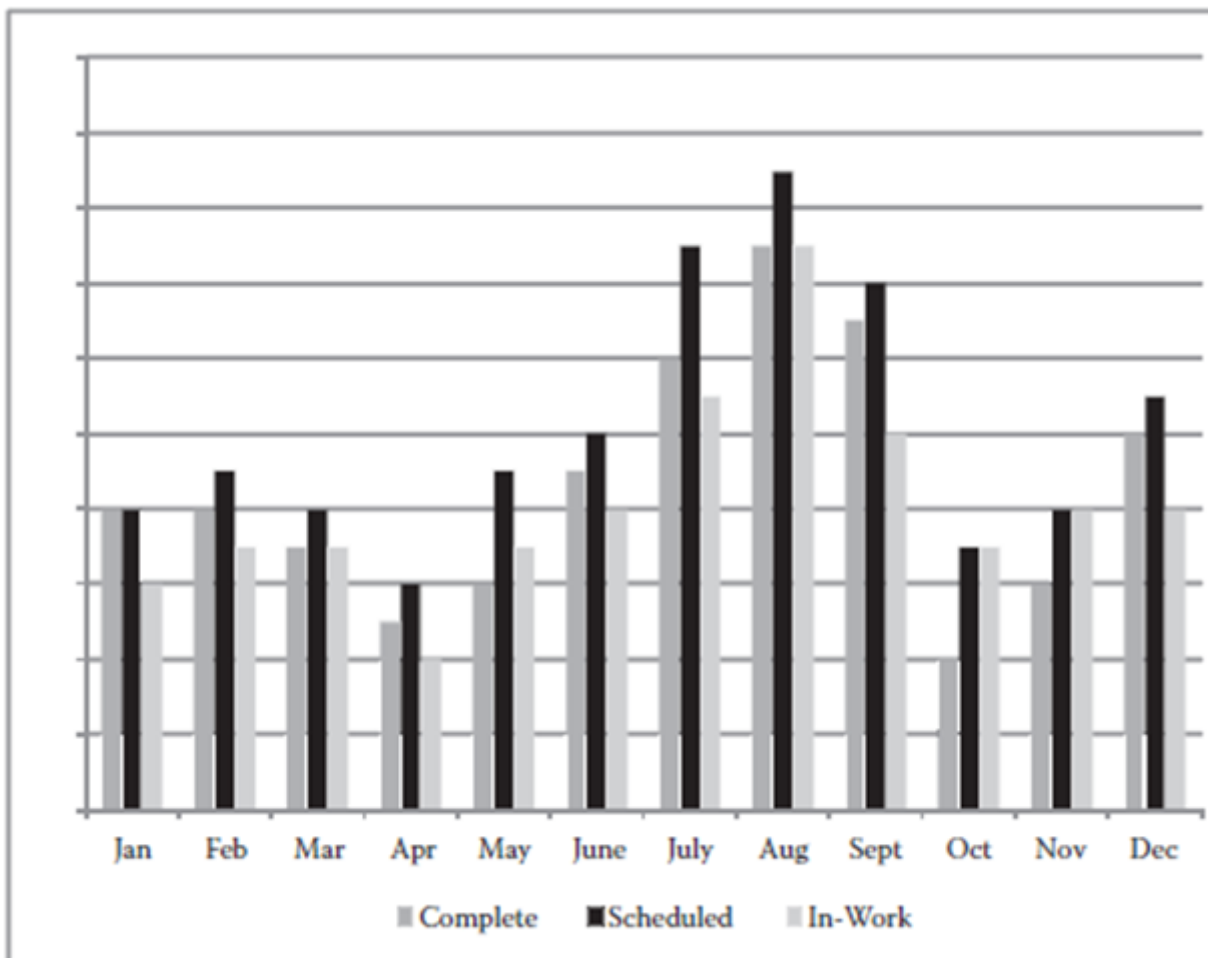· Software and systems installations

FIGURE 11.3: Quality metrics.

The support of CMMI® provides the basis for conducting product evaluations, reviews, and audits to ensure compliance to requirements as shown in Figure 11.4.

Measuring quality does ensure a program's and project's operational goals are successful during the software design/development life cycle. It is so important to measure software engineering processes and determine whether programs and projects are consistently improving. If quality metrics are not used, then there is no way to determine if any improvement is within sight. If there are no improvements, it means you are lost, confused, and out there somewhere during software design/development activities.

By evaluating productivity and quality, teams and management establish goals for improvement of the software processes. Using quality metrics, baselines become more manageable and benefit the program's and project's processes to make sure work products operate at a higher level of consistency.
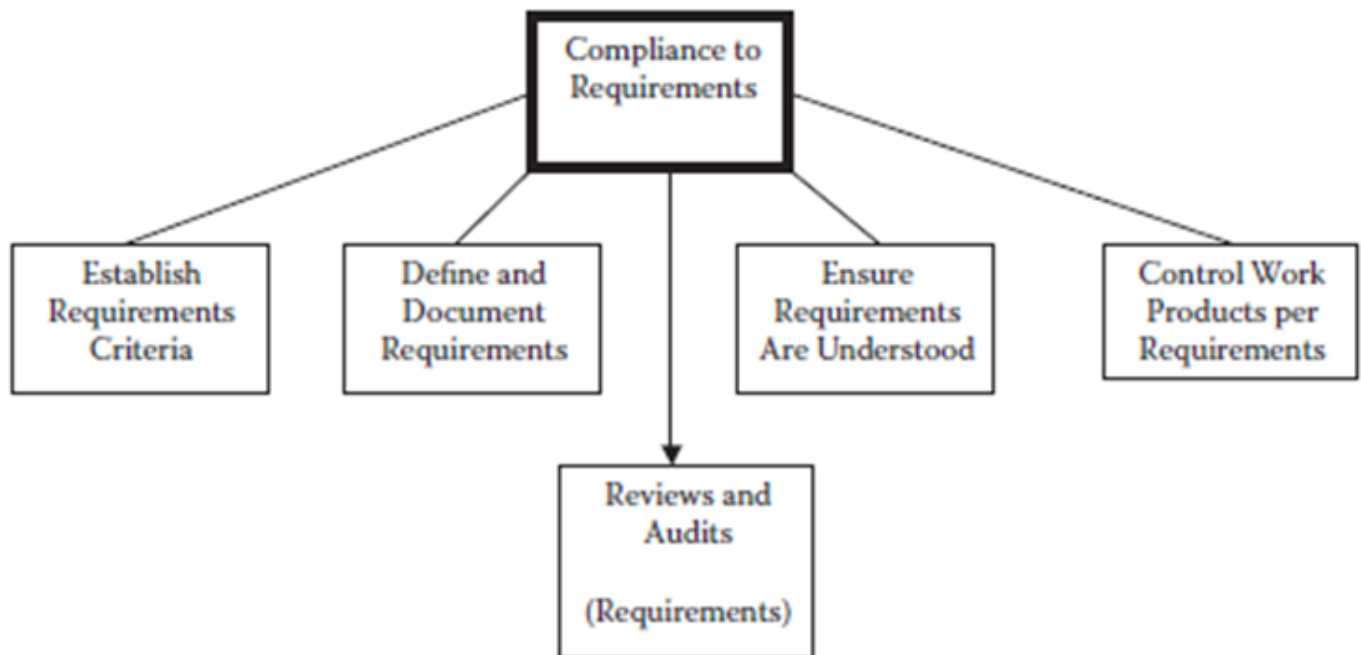

FIGURE 11.4: Compliance to requirements.

Numerous programs and projects run on collected data. The goal is to make software work products and processes better and more cost effective. Data are the key to important decisions. Many problems are linked to using poor-quality data, including:

· Poor estimations of program or project costs

· Not meeting schedules

· Not having effective staffing of personnel

· Flawed software architecture designs

· Poor design decisions

· Ineffective testing decisions

The information on quality measurements is everywhere, and programs and projects have different definitions and requirements when it comes to the quality of data measurements. Mistakes contribute to ineffective information for quality improvement. The most common errors in measuring data are the following:

· Not understanding the measurement goals

· Teams not involved in quality improvement decisions

· Management sponsorship for measurement of quality metrics or data

· The collection of poor data inside program and project development

· Poor data collection and analysis

To help the need for quality and metrics, the Software Engineering Institute (SEI) CMMI version 1.3 for the development of measurement and the analysis method can be used. This method allows programs and projects to perform quality product evaluations at a high level and to establish metrics in standards and best practices.

⊞ SOFTWARE PROCESS

The software process is effective and followed if programs and projects have the discipline to enforce process needs and to follow these processes at all times when product evaluations are conducted. Every software program or project encounters problems as it moves through the life cycle for software design/development.

Proven solutions to these problems should be addressed early and fixed quickly. Established process patterns show a consistent method for explaining problems in the context of software processes.

∅ Software Process Assessment

The existence of the software process assessment in programs and projects does not guarantee that software work products will be delivered to the customers on time and will meet their needs. The process itself can be addressed and assessed to ensure that the process meets a set of basic criteria to show successful software engineering practices will lead to effective software and systems integration to improve processes. The approach for software process assessments and audits is proposed as follows:

· Standard CMMI Appraisal Method for Process Improvement

· (SCAMPI)

· ISO 9001

· QMS

· AS9100C

Ø **Software Reviews**

Software reviews provide the framework and detailed requirements for verifying/validating design/development efforts. It is important that performing reviews that are successful will ensure achievement in all specified requirements for software design, test, configuration control, and quality to released configuration baselines. Reviews improve the individual and team efforts in maintaining a professional setting where software is developed for profit, cost reduction, and service quality improvement.
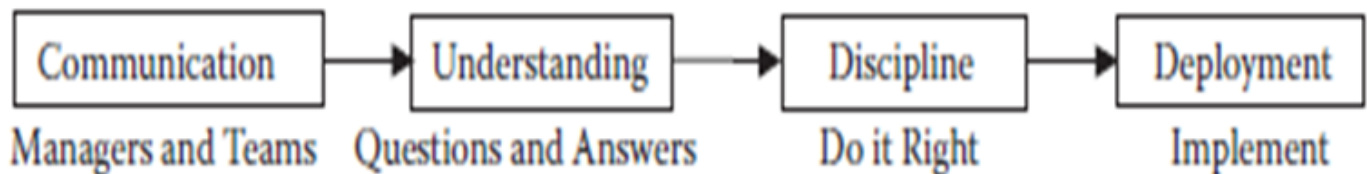
FIGURE 11.5: Process improvements.

There is no clear-cut approach to performing software reviews for multiple companies and military and aerospace programs. At times, ideas, suggestions, standards, and concepts are adopted or implemented to improve the quality of software management, software design/development efforts, subcontractor deliveries, and customer expectations. It is frustrating and confusing when it is time to perform software internal and formal reviews. We always ask what is required, who needs to participate, and what results do we receive for performing software reviews. The answer is quality for delivery to satisfied customers.

Ø **Software Process Improvement**

The process assessment concept proposed in Section 11.9.2 brings order to the stress and chaos for design/development activities, which can lead to failure of software and systems integration. There are no easy answers, but there are alternative options available to system and design/software engineers, test teams, and configuration and quality organizations. Software process improvements become successful if the model for process improvements in Figure 11.5 applies.

🔆 **STRESS MANAGEMENT TECHNIQUES**

Just in case organizations or team members have rough days and are struggling with processes and other team members, there is a seven-step management technique I would recommend, which does work:

·   Picture yourself near the ocean.

·   The ocean is blue and crystal clear.

- Birds are flying by and chirping.

- You are the only one there and in total seclusion.

- There are soothing sounds, and the air is filled with serenity.

- You can easily make out the faces of the team members under water.

- See—you are smiling.

SOLVING QUALITY ISSUES

At times, there are problems with the delivery of work products to software and systems integration facilities and to customers. The quality engineers solve process issues and concerns; they are unable to fix code and test software but participate in peer reviews and witness integration test activities.

The quality engineers provide assistance and help program and project managers look good and become successful, but they need to listen and understand the roles and responsibilities of quality engineering. The quality factors are essential and important to understand.

Jump to...

🔧 Navigation

Home
  Dashboard
    Site pages
    My courses

ℹ️ Fair Warning

NOTICE: Please be reminded that it has come to the attention of the Publishing Team of eLearning Commons that learning materials published and intended for *free use only by students and faculty members within the eLearning Commons network were UNLAWFULLY uploaded in other sites without due and proper permission*.

PROSECUTION: Under Philippine law (Republic Act No. 8293), copyright infringement is punishable by the following: Imprisonment of between 1 to 3 years and a fine of between 50,000 to 150,000 pesos for the first offense. Imprisonment of 3 years and 1 day to six years plus a fine of between 150,000 to 500,000 pesos for the second offense.

COURSE OF ACTION: Whoever has maliciously uploaded these concerned materials are hereby given an ultimatum to take it down within 24-hours. Beyond the 24-hour grace period, our Legal Department shall initiate the proceedings in coordination with the National Bureau of Investigation for IP Address tracking, account owner identification, and filing of cases for prosecution.

🧩 Activities

📄 Assignments
💬 Forums
✔️ Quizzes

📄 Resources

Bestlink College of the Philippines
College Department

Powered by