CHAPTER 1

Organizing for Quality Management

Emanuel R. Baker and Matthew J. Fisher

The relationship between the quality of a product and the organization responsible for the development of that product is multidimensional. The relationship depends upon many factors such as the business strategy and business structure of the organization, available talent, and resources needed to produce the product. It also depends upon the combination of activities selected by the organization to achieve the desired product quality. The ultimate focus of this chapter is how organizations could be structured to implement a *Quality Program* and achieve the project's quality objectives. The Quality Program is a framework for building quality into a product, doing the evaluations necessary to determine if the framework is working, and evaluating the quality actually achieved in the product.

In Sections 1.1 and 1.2, we establish the context for the organizational discussions, and, in fact, for other chapters as well. We describe a *Quality Management Framework* first articulated in 1982 [1] to help clarify and place into context the multiple dimensions of this organizational relationship. The framework addresses the conceptual elements necessary for building quality into products, or any entity, and evaluating the quality actually achieved. This framework consists, in part, of a set of definitions and their attendant concepts that impact quality. These definitions constitute the structural members of the framework. Next we use these definitions to explore a Quality Program, which, in combination with the definitions, comprises the Quality Management Framework (QMF).

In Sections 1.3 through 1.6, we use the context of a Quality Program to examine various organizational aspects of implementing the tasks in the QMF. Two examples of organizations are described: one for a large organization and one for a small organization.

1.1 The Quality Management Framework

The original goal of the QMF that was developed in 1982 was to place quality in proper perspective in relation to the acquisition and development of products, including software products. Over time many of the concepts and principles articulated in the original framework have been implemented in various forms not only in

venues such as the U.S. Department of Defense (DOD) standards, but also in process models of today, such as Capability Maturity Model Integration® (CMMI®). In such manifestations, the framework concepts have been applied to software and to other types of products and processes. Although the concept of relating process quality to product quality was first articulated in 1982, it was not until the development of the Process Maturity Model[1] in 1987 [2], and eventually the Capability Maturity Model® for Software (SW-CMM®) [3] and CMMI® [4], that these principles finally were codified in a concrete manner, making them easier to implement.

We describe here definitions and associated concepts that constitute the structural elements of the framework that will lead to the concept of a Quality Program.

The following terms and their definitions are provided to establish a basis for the definition of "quality" and to provide the foundation for the Quality Program:

- Object (entity);
- Process;
- Requirements;
- User;
- Evaluation;
- Measure and Measurement;
- Quality.

In presenting the definitions, we avoid any specific organizational connotations and relate only to activities and interrelationships.

1.1.1 Object (Entity)

The types of objects (entities) to which quality can be applied include:

- Product;
- Process;
- Service;
- Resource;
- Artifact;
- Activity;
- Measure or metric;
- Environment;
- Collection of entities or objects.

For conciseness, in this chapter we will focus on the quality associated with a product.

1. The Process Maturity Model, developed by the Software Engineering Institute (SEI) in 1987, is the forerunner of the SW-CMM®.

1.1.2 Product

First, we define a *product* as any tangible output or service that is a result of a process [4, 5]. A product itself may include hardware, software, documentation, or a combination of these; also note that a service is included in the definition. Accordingly, even though the discussion focuses on products, it is important to remember that the same principles apply equally as well to a service, or to anything else included under our definition of object.

1.1.3 Process

Ultimately, what one is interested in is the quality of the delivered product or service. The quality of a product or service is dependent on the quality of the process used to create it [3]; consequently, we need to establish a definition of process, which will enable us to develop a definition of quality. As part of this development, we view *process* as a set of activities performed for a given purpose, for example, a software acquisition process [5].

1.1.4 Requirement

In defining the elements of a Quality Program, a definition of requirements is needed. There are various definitions of a requirement. We include here a general definition of requirements that we use for the remainder of this document. The references cited define a *requirement* as a needed capability, condition, or a property [attribute] that must be possessed by an entity to satisfy a contract, standard, specification, or other formally imposed documents [4, 5].

Simply put, a requirement is a way of characterizing a user's need in a way that allows a development team to implement that need in the product in a concrete way. Put another way, the achievement of the requirements are the yardstick by which we measure the quality of a product or a service. A user—for example, an airline—may need an aircraft that is capable of flying 600 passengers 10,000 miles nonstop with high fuel economy. To actually develop the aircraft, more specific characterizations of the need must be expressed. For instance, the aircraft must be supplied with four engines each with 110,000 pounds of thrust.

We must also be aware that as in the Software Acquisition Capability Maturity Model® (SA-CMM®) [5], there are several types of requirements such as technical, nontechnical, product, allocated, users development, and so on. As a caution, we must be cognizant of the type of requirements being discussed or specified. For example, fixed regulations may also be requirements and may be interpreted as a contract provision, an applicable standard or specification, or other contractual document. Thus, as defined in IEEE-STD-610, *IEEE Standard Glossary of Software Engineering Terminology*, a product requirement is a condition or capability that must be met or possessed by a product or product component in order to satisfy a condition or capability needed by the user to solve a problem. As noted earlier, these conditions may consist of a number of diverse types of attributes.

1.1.5 User

For our purposes, we define user as either the customer or the end user. Typically, three kinds of situations may be encountered in a development or maintenance effort. In the first situation, the customer (either internal or external) and the end user are one and the same (Figure 1.1). In the second situation, the end user is represented by a buyer, and all contact with the client organization is through the buyer (Figure 1.2). In this case, the buyer represents the user. The face presented by the buyer, therefore, is that of both buyer and user. The third situation is where both the buyer and the user community are accessible to the development or maintenance organization (Figure 1.3).

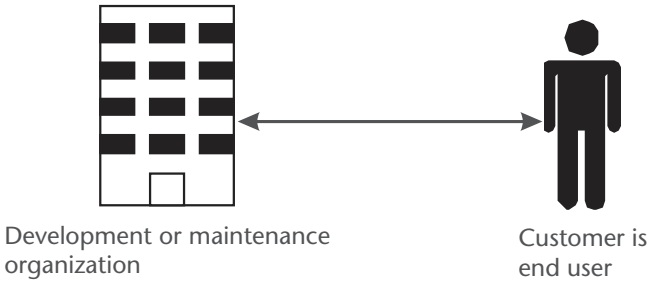

Figure 1.1 Customer is end user.

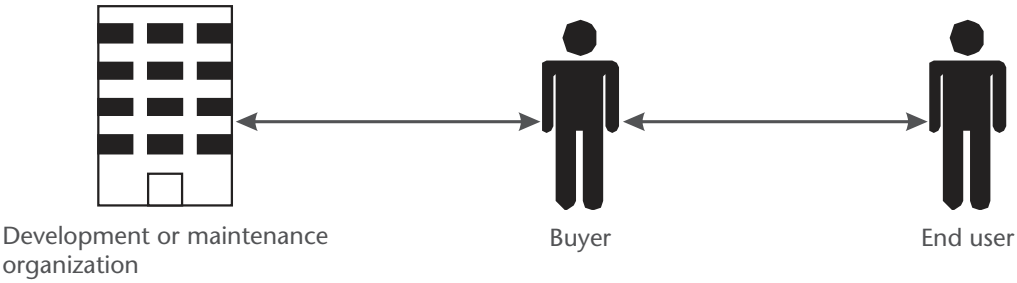

Figure 1.2 Customer represents end user.

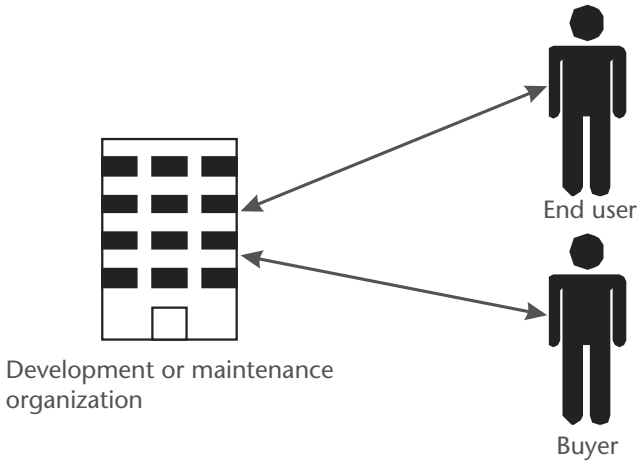

Figure 1.3 Customer and end user are accessible.

For ease of reference, we will use the term "user" in this chapter to represent all three situations. The considerations for these three user situations become prominent when we discuss the first element of the Quality Program, *establish requirements*.

1.1.6 Evaluation

As part of a Quality Program, the concept and definition of evaluation is critical, particularly that of evaluation of product quality. As with requirements, the definitions of evaluation are varied. To place *evaluation* in the context of quality and a Quality Program, evaluation is defined as the process of determining satisfaction of requirements [5]. Kenett gives a more precise definition stemming from this basic definition: Evaluation is "a [process] to evaluate the quality of products and to evaluate associated documentation, processes, and activities that impact the quality of the products" [6].

According to both definitions, evaluations may include methods such as analyses, inspections, reviews, and tests. In this context, evaluation can be applied to acquisition, services, documentation, process, or any product or work product resulting from a process. In this technical note we are interested in evaluation of quality, specifically, determining process and product quality.

1.1.7 Measure and Measurement

As part of evaluation discussed above, we need to include in the QMF the ability to measure the quality of processes and products, and be able to assign actual quantitative values to the item's quality. Toward this end, the definitions and concepts of *measure* and *measurement* are as follows: Measure (v.) is to ascertain the characteristics or features (extent, dimension, quantity, capacity, and capability) of something, especially by comparing with a standard [5]; measurement (n.) is a dimension, capacity, quantity, or amount of something (e.g., 300 source lines of code or seven document pages of design) [5].

Other definitions for measure and measurement exist that convey similar meanings but reverse the definitions. That is, since the term "measure" can be used as a verb or a noun, and "measurement" can mean a process or an item of data, the community tends to use them interchangeably. For example, Fenton [7] describes measurement as the process by which numbers or symbols are assigned to attributes of entities in the real world in such a way as to characterize the attributes by clearly defined rules. This implies an action such as "to measure" in the above definitions.[2]

In addition, use of the terms "metric" and "measure" has become somewhat confused. In most cases they are used interchangeably, but be aware that in some cases various authors have made attempts to distinguish the two terms (and they have done so in different ways). For evidence of this, see [7, 8]. The *IEEE Standard*

2. There are numerous reference works on the subject of measurement: Krantz, D. H., et al., *Foundations of Measurement,* Volume 1, New York: Academic Press, 1971; and Ghiselli, E. E., J. P. Campbell, and S. Zedeck, *Measurement Theory for the Behavioral Sciences,* San Francisco: W. H. Freeman and Company, 1981. Some of these (Ghiselli, for example) provide excellent discussions of important issues such as measurement reliability and validity.

Glossary of Software Engineering Terms defines "metric" as follows: "A quantitative measure of the degree to which a system, component, or process possesses a given attribute." Note that the word "metric" is not used in ISO 15939, *Software Engineering—Software Measurement Process*. See Chapter 16 on further definitions of SQA measurements.

Measurement plays an important role in the QMF. As we will later see, an element of the QMF is quality evaluation. We are interested in evaluating the compliance of the product and processes with applicable standards and requirements. One means of achieving that is through measurement. Consequently, we are concerned with the act of measuring, as well as the measure itself (both the verb and noun forms of the word). Anyone familiar with the discipline of measurement and analysis knows the importance of operational definitions for the measures selected. Without them, there is ambiguity about what is being measured, and how it is being used. Likewise, an operational definition for measure and measurement is important in explicating the QMF in order to avoid ambiguity about what it is that the QMF is intending to accomplish.

1.1.8 Quality

A prime focus of the QMF is, of course, the definition of quality and its implication to a quality program. The definition discussed here has broad implications, especially in terms of implementations of a quality program. In the QMF, *quality* is defined as in this reference: Quality is the degree to which an object (entity) (e.g., process, product, or service) satisfies a specified set of attributes or requirements [5]. However, it is important to point out that a product possesses many quality attributes that are intrinsic to the product and that exist regardless of what is desired, specified, or measured, and only depend on the nature of the product [1].

Thus, the definition of quality includes two aspects:

- The concept of attributes;
- The satisfaction or degree of attainment of the attributes.

1.1.8.1 Attributes

An attribute is "a property or characteristic of an entity that can be distinguished quantitatively or qualitatively by human or automated means," from ISO 15939, *Software Engineering—Software Measurement Process*. The word "attributes" includes all specified requirements governing functional, performance, and other specified characteristics such as adaptability, maintainability, and correctness [1, 9]. The attributes (i.e., requirements and other specified characteristics) are considered the determinants of product or process quality.

1.1.8.2 Specifying Product Quality Using Attributes

The word "specified" implies that definitions of the needed quality attributes are documented. Without clear articulation of the quality attributes, it is impossible to develop a product or determine whether the finished product has the needed quality.

A specification is required to communicate to others which attributes constitute the product's quality. Contractually, this specification is critical [1].

In addressing product or process quality, it is therefore necessary that the specification or definition of the attributes is expressed quantitatively. This quantitative expression allows a determination of the degree to which a product or process satisfies or attains the specified attributes. For example, saying that a hardware product has high reliability does not tell us how reliable the hardware is. Stating that the reliability of a product is 0.9999999 mean time between failure (MTBF) expresses a characteristic that can be measured, which means there is a method used to determine if the specified attribute has been attained.

For implementation of a product, one selects those attributes most significant to the user community and evaluates, rates, or measures the product's quality on how well, or to what degree, the selected attributes meet those criteria for excellence. Often, these attributes address only functionality and performance and ignore the other attributes, often referred to as *-ilities*. The -ilities can be considered as attributes that address fitness for use. Conceivably, "if we look at the issue of the software meeting its requirements and if those requirements are solely functional and prescribe no -ilities, then clearly the software can meet the requirements but could be unable to fulfill any reasonable purpose" [10].

Thus, we recognize that just as beauty is in the eye of the beholder, so is quality [9].

Consequently, a set of attributes that one user community deems important as a measure of quality may not be deemed important by another user community. Rather, each user community is likely to have its own set of attributes with which to measure quality.

1.1.8.3 Considering User Needs

It is difficult to satisfy users if they cannot articulate what quality they are expecting. In many cases, users default to "give me something and I will tell you if I like it or not." Such a paradigm wastes time and resources in trying to satisfy the illusive user expectations. Clearly, vague notions such as "user needs," unless they are articulated, cannot be used to determine the quality actually achieved in a product. Something concrete, such as a user specification document, can be used. Obviously, the requirement to accurately capture the user needs in such a document is crucial. Typically, the documents are operational needs documents, which are then decomposed into system requirements documents and then further decomposed into software and hardware component requirements documents. All of these start from the documented user's needs. Codification of this set of activities, for example, is documented in the process area (PA) of Requirements Development in the CMMI®.

The fact that product quality requirements include the functionality and performance requirements and may include requirements for maintainability, portability, interoperability, and so on, leads us to the key point that product quality requirements stem from many sources, above all, from the stakeholders of the project, and this leads us to the idea that quality is everybody's business [11].

However, if we consider how product development projects are organized, the implication is quality is affected by many, but implemented by few [11].

What we will see in later sections (for example, Section 1.2.2) is that the actual activities of the Quality Program are distributed among a number of entities within an organization. No one organization has the capabilities to perform all the functions; consequently, the activities must be assigned to those entities most capable of performing them. As we shall also see later in Sections 1.3 through 1.6, it is necessary to have a central point of responsibility for coordinating all the elements of the Quality Program.

Finally, one must realize that the final quality of a product results from activities performed by the project developing the product. Everything that occurs within a project during development affects some attribute of the product and, therefore, the total product quality. However, all possible attributes may not be of equal relevance. Furthermore, all actions may not affect the specified attributes to the same extent and, therefore, the specified quality. In any event, quality is affected by activities such as requirements definition, design, coding, testing, and maintenance of the product, activities associated with the Quality Program for the product, and the interaction of these activities.

1.2 Quality Program Concepts

The foundation of the Quality Program stems from the definition of quality and the precept that many people supporting the project affect the quality of the product. The interaction of the Quality Program with the other parts of the project elements is necessarily complex. The involvement is at all levels of the project organization and takes place throughout the project's life. In some cases, the Quality Program directs the other activities; in other circumstances, it can only influence those activities. In any case, all the project activities, in some way, affect product quality. The Quality Program is defined as the overall approach to effect and determine the level of quality achieved in a product [9].

1.2.1 Elements of a Quality Program

The Quality Program incorporates three elements that cover the activities necessary to:

1. *Establish requirements and control changes:* Establish and specify requirements for the quality of an product.
2. *Establish and implement methods*[3]*:* Establish, implement, and put into practice methods, processes and procedures to develop, operate, deploy, and maintain the product.
3. *Evaluate process and product quality:* Establish and implement methods, processes, and procedures to evaluate the quality of the product, as well as to evaluate associated documentation, processes, and activities that have an impact on the quality of the product.

3. Methodology is a system of principles, procedures, and practices applied to a particular branch of knowledge. As used here, the organizations' processes and procedures in a development are instantiations of methodologies.

Figure 1.4 illustrates the interaction of these elements and the interaction of the Quality Program with a product's design and implementation activities to produce quality products. This interaction is continuous with the design and implementation activities affecting the Quality Program activities. The Quality Program addresses both technical and management activities. For instance, ensuring that quality is built into a product is a management activity, while specifying the methods used to build in the quality is considered a technical activity.

Given the precept that quality is everybody's business, it follows that a Quality Program covers both technical and management activities. For instance, if we look at the element of the Quality Program concerned with methodologies or product development, enforcing these methodologies (in order to build quality into the product) is a management activity, while the specification of the methodologies is a technical activity. The following discussion expands on the elements of the Quality Program.

One of the foundational aspects of the Quality Program is how well quality can be built into a product, not how well one can evaluate product quality. While evaluation activities are essential activities, they alone will not achieve the specified quality. That is, product quality cannot be evaluated (tested, audited, analyzed, measured, or inspected) into the product. Quality can only be "built in" during the development process [11].

Once the quality has been built in, the deployment, operation, and maintenance processes must not degrade it. Unfortunately, the very nature of maintenance and bug fixes for software often degrades the quality of the code. What was once structured code becomes messy "spaghetti" code with all the modifications resulting from bug fixes and enhancements.

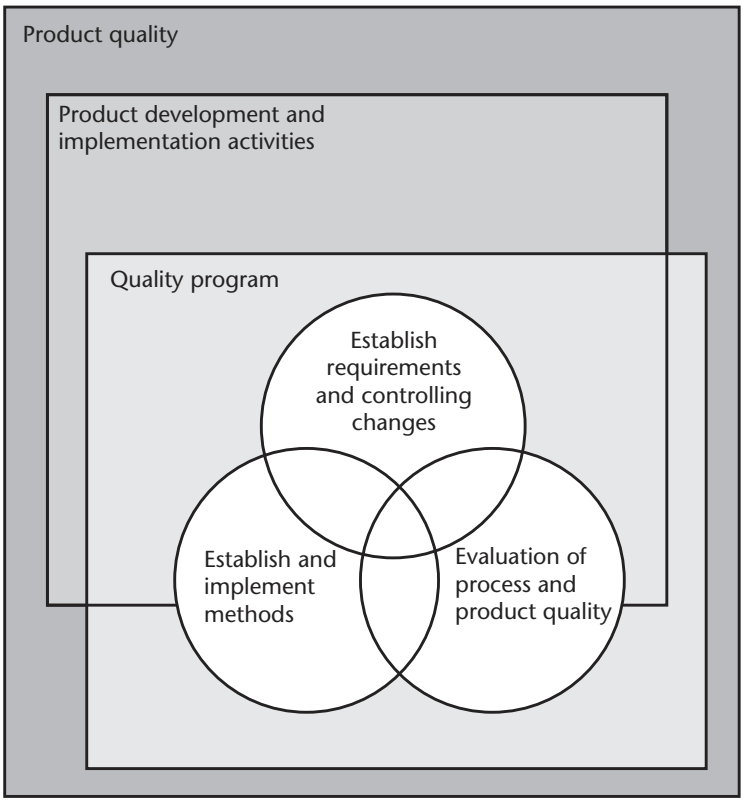

Figure 1.4 Interaction of the elements of a quality program.

For example, Figure 1.5 provides an indication of when the quality of the software likely has been compromised and qualification test procedures should be reexecuted or the software should be reengineered. A threshold level of 30% of the modules changed per software maintenance request was set as the point at which requalification of the software should take place.

Note that in Figure 1.5, the software is approaching the threshold level at which it should be reengineered (approximately 67% of the modules changed). Clearly, in this situation, there is a great potential for the quality of the software to be degraded as a result of maintenance activities. This quality degradation is shown in Figure 1.5 that is based on the following equation:

$$Volatility = \frac{\text{Number of modules changed due to a software maintenance request}}{\text{Total number of modules in a release over time}}$$

The Quality Program does not impose any organizational structure for performing the activities. Organizations are responsible for assigning resources to accomplish the Quality Program activities. We do suggest that organizations, especially at the corporate level, avoid assigning certain roles to carry out the Quality Program without clearly understanding the concept of product and process quality and how those are affected and implemented.

The idea of many people affecting the product quality should be obvious from the fact that so many disciplines are involved in accomplishing the array of quality requirements. Virtually everyone working on the project, from the project manager (PM)[4] to the most junior member of the staff, affects the quality of the product. However, only those actually producing the product (performing tasks such as

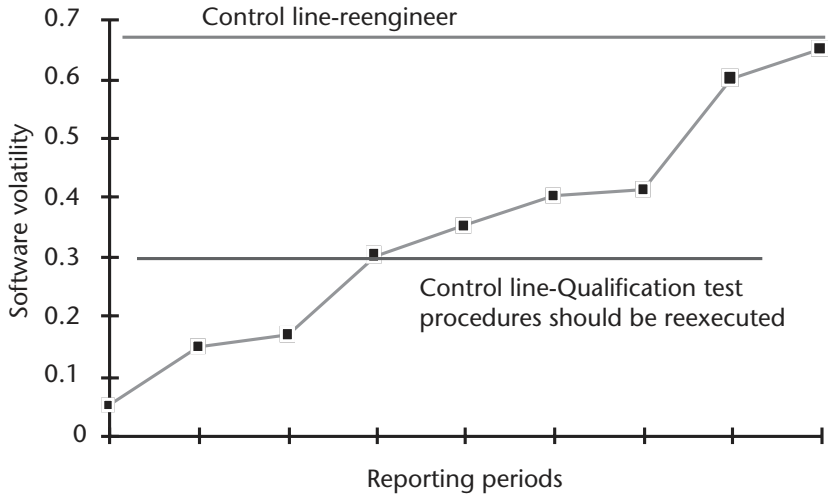

Figure 1.5 Software volatility indicator.

4. A project manager is an individual assigned the management responsibility for acquiring, developing, or providing a product or service. The project manager has total business responsibility for an entire project and is the individual who directs, controls, administers, and regulates the project. The project manager is the individual ultimately responsible to the customer or end user (see [5]).

requirements analyses, design, and manufacturing/coding) build quality into the product. Even though, for example, code reviewers do affect the quality of the resultant product, like testers, they do not actually produce the code.

Thus, it is important to understand that the person ultimately responsible for the quality of the product is the PM. It is the project manager's responsibility to integrate the efforts of "the many" and "the few" to accomplish the project's quality objectives. The PM may, of course, delegate authority for any part of this function, but ultimately he or she is responsible. This is reflected in the often used phrase that, while authority may be delegated, responsibility cannot be.

1.2.1.1 Establish Requirements and Control Changes

The first element or set of activities of the Quality Program is to establish requirements. Product requirements must accurately reflect the product's desired overall quality, including functionality and performance, and must be documented and baselined (formalized). As noted previously, the requirements must accurately reflect the needs of the user community. The process for defining the requirements must ensure that the needs of all the stakeholders involved with the end product have been accurately captured. Thus, a process for establishing and controlling the requirements must be established. This indicates an interface with the second element of the Quality Program: *establish and implement methods*.

One problem associated with specifying a product's quality requirements is the inaccurate perception that the quality cannot be stated quantitatively or in a way that permits objective evaluation. As noted earlier, communicating the quality of an entity to others becomes difficult because people tend to interpret the quality of the same entity from their own perspective. The result is that verifying achievement of the desired quality can be quite subjective. Consequently, the methodology established must ensure that ambiguity is reduced and verifiable quality criteria are specified.

Simply defining and formalizing product requirements are insufficient. The baseline product requirements must be strictly adhered to and fully implemented. Failure to implement the requirements as specified can result in products that do not meet user needs and derived requirements. The resultant impact on product quality, such as functionality and performance, will range from negligible to severe. It follows that any changes to the product requirements must be controlled and documented, and the effects of those changes must be understood.

The activities of defining and establishing the requirements and controlling changes to them necessarily involve interfaces with the other two elements of the Quality Program: *establish and implement methods* and *evaluate process and product quality*.

To illustrate the interface between these two elements of the Quality Program, an organization may establish methods such as the use of data flow analysis, use cases, or object-oriented analysis for performing requirements analysis. Whatever method is selected and used must provide high confidence that the users' needs have been captured accurately; consequently, the evaluation of the requirements analysis process must demonstrate that it was followed and is effective in capturing the users' needs, and the evaluation of the requirements must indicate that the users' needs were captured correctly for the instances examined.

As another example, when establishing baseline requirements and controlling changes, a configuration management method must be selected and later implemented in order to:

- Establish a baseline as a reasonably firm point of departure from which to proceed into the other phases of project activity knowing that there is some reasonable degree of stability in at least the set or subset of requirements that were established.
- Prevent uncontrolled changes to the product baseline.
- Improve the likelihood that the development effort results in a quality product and that processes in subsequent life-cycle phases will not degrade it.

Again, the process evaluation must demonstrate that the process was followed and is effective, and the product evaluations must demonstrate the correctness of the outputs.

A second interface between elements of the Quality Program exists. It is between the *establish requirements and control changes* element and the *evaluate process and product quality* element. It is concerned with two things: the evaluation of the product against the requirements, and the determination that the process was adhered to for defining requirements. Total compliance with requirements does not guarantee a quality product if the requirements are not properly defined and errors exist in them. Then compliance with requirements produces a product that does not satisfy the intended end use. Clearly, evaluations/audits for compliance with the process for establishing requirements must be performed. Furthermore, the process and method by which the requirements are developed must be evaluated for adequacy in this regard during the development or maintenance process.

1.2.1.2 Establish and Implement Methods

The second element or set of activities of the Quality Program involves selecting, implementing, and putting into practice the appropriate processes, practices, and methods *to build quality into the product* and achieve the specified quality requirements. This is typically accomplished by codifying these processes, practices, and methods as standards and training the organization and project teams to use them. These standards may be tailored to meet the unique needs of the project in accordance with established tailoring guidelines (or as defined processes, in the context of the CMMI®).

Implementation of the methodologies may be facilitated by tools compatible with the methodologies and the standard practices and procedures.

The act of getting these standards into use is accomplished by corporate management, who can consistently and unequivocally require the application of the selected methods from project to project even under conditions of schedule pressure. The enforcement can be through various means, for example, assignment of appropriately trained personnel, or monitoring and controlling the project against a project plan, or both. The important point is that requiring compliance with standards is the responsibility of management and not some other organizational entity, like, for example, a quality assurance group.

Enforcing compliance does not preclude tailoring the methods to be consistent with the unique characteristics of the various projects. Tailoring should be permitted to account for the fact that there may be considerations that would militate against full compliance with the organization's standard process for some projects, or that perhaps might require additional, more stringent process steps. Guidelines should exist to cover those cases, and the resultant tailoring should be subject to review and approval. Reviews and audits of project activities for adherence with the established processes are performed to provide visibility or insight to management. Reviews or audits for adherence do not necessarily constitute enforcement; they can only determine if compliance has occurred. Management can use the result of audits to exercise its leadership responsibilities by active monitoring and control of project activities.

Typically, we believe there is a strong link between product quality and the processes used to develop (and maintain) it. If the processes used by the development organization are not well defined or organized, the quality of their products will not be predictable or repeatable from project to project. Based upon the maxim that the quality of a product is highly influenced by the quality of the processes used to produce it [3], the development community, in conjunction with the Software Engineering Institute (SEI) at Carnegie Mellon University, developed a process maturity model and associated appraisal methodology called Capability Maturity Model Integration® (CMMI®) and Standard CMMI® Appraisal Methodology for Process Improvement (SCAMPI℠), respectively [12]. This process model and the appraisal methodology are used to characterize the maturity of the development processes and associated procedures and methodologies. Five levels of maturity are described by this model. The levels, their names, and the characteristics that describe when the organization has reached that level are shown in Figure 1.6.

Characterizing the process maturity in this way is an attempt to show one link between the quality of the product and the processes employed in its development. Note that process maturity is one way the development community describes the quality of the processes. While this quality attribute may not be the only link to

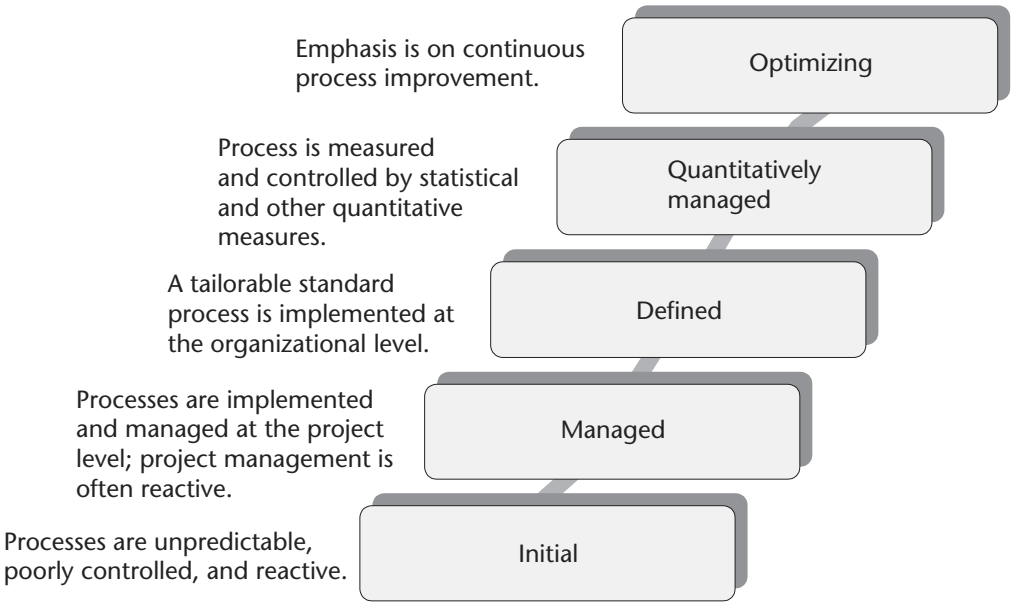

Figure 1.6 CMMI® maturity levels.

product quality, it does indicate the importance the community places on process (the establishment of methodologies) and why this is such an important element of the Quality Program.

Even with this attempt of linking product quality to process quality in terms of process maturity, at present, no formalized techniques exist for organizations to select and specify optimal methodologies necessary to achieve product quality; the selection is based on experience, intuition, literature search, common knowledge, and, to some extent, trial and error. In general, it is only the highest maturity level organizations that have the capability of quantitatively evaluating how effective their processes are, and this evaluation is typically made with reference to the processes already selected by the method described next.

Generally, the basis for determining which processes to implement when establishing standards is to look to past history. If a process produced "high-quality products" on some past projects, it is believed that properly implementing the process on an organizational basis will result in a high-quality product now. Establishing standards in this manner is somewhat misleading. The link between methods selected for product development and the resultant product quality has not been demonstrated quantitatively because establishing links is more heuristic or intuitive than analytical. For example, Ada was touted as promoting information hiding, which, in turn, should make the product more adaptable. However, to our knowledge, the actual quantitative cause and effect link has never been documented.

1.2.1.3 Evaluate Product and Process Quality

The third element or set of activities of the Quality Program involves both evaluating the implementation of processes and evaluating the quality of the resulting product(s). Evaluations are used to assess:

- The quality of the product;
- The adequacy of the processes and activities responsible for product quality;
- Compliance with established processes.

Evaluation of Product Quality

Quality evaluations help to define the "health" of the product and hence the project. Through evaluations, the project can determine whether its product satisfies the specified quality requirements within cost and schedule constraints. Because of the number of organizations typically involved in quality evaluation activities, coordinating the results of this process should be performed by the PM. Whatever assignment decisions are made, management must be sure that all quality evaluation activities are assigned to competent, unbiased reviewers.

Evaluation activities include reviews, assessments, tests, analyses, inspections, and so on. Depending on the action taken and the processes or products being evaluated, the results may be qualitative or quantitative in nature.

Any evaluation that requires reasoning or subjective judgment is referred to as an assessment. Assessments include analyses, audits, surveys, and both document and project performance reviews. On the other hand, measurement activities constitute quantitative evaluations such as tests, demonstrations, metrics, and, in some

cases, inspections using checklists (although some checklists tend to be written only to allow a subjective evaluation). Accordingly, measurements can include tests for unit level, integration, and product or application level performance, as well as the output of a compare or a path analyzer program.

Evaluation activities will vary with each phase of the development cycle. Furthermore, they can be performed by individuals independent of the project, or one or several independent organizational units. Evaluation activities to be performed and responsibility for them are generally defined and documented in project management plans, product development plans, project-specific product quality procedures, and/or company quality plans and related product quality procedures.

Evaluation of Established Processes

Another form of process quality evaluation is doing reviews and audits for compliance to or adherence with the process. It is one thing to specify a process, but if that process is not being followed, the quality of the resultant product can be adversely affected. Periodic audits for compliance with the process need to be performed to ensure that the established process is being implemented. For example, external appraisals, such as a SCAMPI^{SM} appraisal, are helpful in this regard.

It is also important to evaluate the processes used to determine if these processes are producing products that yield the required quality. Using a concept from the CMMI® to explain how processes are evaluated, low maturity organizations will do qualitative evaluations supported, in some cases, by rudimentary quantitative methods. High maturity organizations will use metrics and statistical analysis to determine how effective the processes have been. In the low maturity organization case, the evaluation will in most cases be experiential. For example, did the process performers experience making a number of mistakes and doing rework by following the process? In other cases, the organization might collect simple data, for example, defect data. If the number of defects appear to be large (a subjective determination at lower CMMI® maturity levels), then investigations will be performed to figure out where in the overall scheme of things the defects are being introduced. In the high maturity cases, process performance goals are established, processes will be quantitatively monitored (using statistical analysis for the most critical processes), and corrective action implemented when process performance goes off the track.

1.2.2 Considerations

1.2.2.1 Quality Evaluation Versus Quality Assurance

A point of confusion, especially related to organizational aspects of the Quality Program, is the role of quality assurance (QA) groups and the quality evaluation activities that group may perform. Part of this confusion stems from a misunderstanding by many project teams about the word quality and the belief that anything to do with quality is (or should not be!) the purview of a QA group. This belief flies in the face of the precept that quality is everybody's business, and that quality cannot be injected into a product, say, by a QA audit. Part of this confusion is also the blurred difference between QA functions and QA organizational entities. In fact, QA functions can be performed by many groups, not only groups designated as QA by corporate decree.

Thus, as far as an organizational perspective goes, if a QA entity exists in a corporate structure and the capability of this group is limited, let us say, to a checklist approach, a project manager may mistakenly conclude that the effort performed by this group is sufficient to satisfy all the quality evaluation needs of his or her project. This situation may preclude vital measurements (or tests) on critical parts of the product.

The Process and Product Quality Assurance (PPQA) process area of the CMMI® provides an excellent approach to QA functions. In doing so, the process area defines the QA function as providing insight into the implementation of the process and products against established standards. As noted above, this is part of the overall quality evaluation activities. In addition, this CMMI® process area does not relegate these functions to any specific group but allows the organization to assign these responsibilities. More detailed analysis and discussion of this is in Chapter 11.

What is crucial to any product development project is the definition and implementation of the activities necessary to assess and measure the quality of the products developed by that project and the processes used to develop them, in accordance with company and project requirements, such as quality goals, established for the project. When the quality evaluation activities have been defined, the assignment of these activities to specific organizations is a management prerogative. Where QA organizations have the capability to perform many or most of the quality evaluation activities, these can be assigned to the QA organizations.

1.2.2.2 The Concept of Independence

As part of quality evaluation activities, we discuss the concept of independence.

Relative to quality evaluation, independence implies performing product quality evaluations by an "outside" organization (or individuals).

In this case, *outside* means different from those that produced the product or those that executed the processes and activities being evaluated.

The concept of independence relates not only to performing the evaluation, but extends to establishing the evaluation criteria. The need for independence arises because the persons performing the process or creating the product may have a conscious or unconscious need to make the process or product look good. They might also have a biased expectation of what the result should be and consequently would fail to perform certain checks or could miss anomalies because of that expectation. Such evaluators can hardly be considered independent. By removing from them the responsibility for establishing the evaluation criteria and performing the evaluations, such problems will be substantially reduced.

Independence, as a concept, has two aspects:

1. Independence exercised within an organization, such as the use of a test team composed of individuals different from those who designed and developed the product;
2. Independence exercised by establishing a separate group outside the project, such as an independent verification and validation[5] agent from outside the

5. Verification and validation can be characterized (based on the CMMI® [4]) as follows: verification determines if the product is being built right, and validation determines if the right product is being built.

organization that is producing the product. This form of independence is perhaps the most stringent.

Either way, the notion of independence is applied to reduce errors resulting from extensive familiarity with the product being evaluated. Decisions as to the application of independence, the degree of independence to apply, and the types of independent agencies to employ are a function of a number of variables, such as size and complexity of the product/project, corporate policy, available funds, and criticality of the product to its end use (human safety, destruction of equipment, severe financial loss, and so on).

1.3 Organizational Aspects of the Quality Program

Using the context for the Quality Program established in Sections 1.1 and 1.2, we address here organizational concepts in implementing the Quality Program.

Each of the elements of the Quality Program (QP) discussed above involves a number of organizations or functional entities within an organizational structure (e.g., a company or a department of the government). The discussion that follows describes the functions or activities these entities perform in the implementation of the QP. It also explores how these entities interact to implement the QP.

Sometimes the functions that will be described are not necessarily performed by separate organizations, but rather may sometimes be performed by different individuals within a single organization. In other words, when a company organizes to implement the QP, it is not a requirement that there be separate functional entities established to perform these activities. For instance, within an information technology (IT) department, the responsibility for some of the QP functions may be shared between a database administrator and a quality administrator.[6] Quality management in IT is further discussed in Chapter 14. For convenience in the following, the word "organization" will be used to refer to both actual organizations and to the situation where the functions are performed by separate individuals within an organizational entity, rather than separate organizations.

1.4 Quality Program Organizational Relationships

Many of the concepts that we discuss here have been addressed in previous editions of the *Handbook of Software Quality Assurance* [13]. These concepts have been codified in various process improvement and quality models, such as the CMMI® and ISO 9000. Since the CMMI® has become a process improvement model adopted by a large number of organizations worldwide, references will be made in what follows to the CMMI®, as appropriate and without additional references, to emphasize the importance of the concepts [4].

In what follows, we discuss the organizational aspects of the Quality Program in terms of:

6. The names for these roles are not standardized and will typically vary from organization to organization.

- Type of systems;
- Mapping quality program functions to project organizational entities;
- Example implementations.

The following discusses the organizational relationships for the Quality Program by using the Quality Program elements as the structure for the discussion; that is:

- Establish requirements and control changes;
- Establish and implement methods;
- Evaluate process and product quality.

1.4.1 Establish Requirements and Control Changes

A number of organizations participate in establishing, implementing, and controlling the product quality requirements (including, for example, the functional and performance requirements). The kinds of organizations that are involved will depend on the type of product under development. To illustrate, the kinds of organizations that will be involved in this effort for data-intensive systems, such as information management systems (IMS), will be very different from the kinds of organizations that will be involved for engineering applications, such as an avionics system for a commercial airliner. Nonetheless, the activities that occur in establishing and controlling the requirements, and the sequence in which they occur, will be essentially the same for all types of applications.

1.4.1.1 Information Management Systems

For IMS or similar applications, the development of the user requirements should be performed by the using organization and specified in a user specification, which defines the overall functionality of the product (and does not specify how it is to be implemented in the product). In many instances, the using organization should obtain assistance from the product development organization in order to ensure that the requirements are expressed correctly and unambiguously, in a manner that the users can concur with as being correctly representative of their needs and developers can understand and from which generate detailed processing requirements. This is an issue that is addressed in the CMMI® in the Requirements Management (REQM) process area under specific practice (SP) 1.1. The participation of the user is essential in order to ensure that the requirements are responsive to the user's needs. In parallel to this, as the user specification is being developed, preliminary processing and database design requirements should be developed by the development organization. Within IT organizations, this process typically involves product analysts, data analysts, and the database administrator. The user organization is also involved, insofar as they have a role to play in verifying that the (processing) requirements reflect the functionality they want in the product.

A formal review should occur after the product requirements have been defined and documented, to baseline the product requirements specification (in accordance with established configuration management procedures). For IMS systems, the review would involve project management and user personnel, the IT development

organization (product analysts, data analysts, and database administrator), and IT configuration management and quality assurance administrators. After the formal review is successfully completed, the configuration management group is then charged with the responsibility for overseeing the control of the documented requirements to prevent unauthorized changes to them (also addressed in the CMMI® under the REQM PA in SP 1.3).

1.4.1.2 Engineering/Scientific Systems

In developing the requirements for a system,[7] there are also a number of organizations involved. For such systems, the product engineering organization should take the lead for developing the system requirements. In some cases, the starting point for that may be a customer or user statement of needs. The product engineering organization should be involved in the effort to ensure that the product requirements have been correctly captured, stated, and allocated to the product components and are being implemented, and to satisfy all concerned that the product requirements are traceable to the user or customer requirements. The using organization should also be involved in order to make sure that the product requirements reflect what is needed in the deliverable product. [Where the product is developed under contract, the using organization becomes the customer—or is represented by another agent acting on their behalf, for example, purchasing. In this case, such involvement in the requirements definition process may be difficult to achieve without affecting contract costs and/or schedule. To surmount these kinds of issues, creative innovations, such as integrated product teams (IPTs), have been implemented.]

As with IMS, a formal review for the product or system requirements should be held. It should occur after the product requirements have been defined and documented. The product requirements specification should then be baselined in accordance with procedures established by the configuration management group. For engineering/scientific systems, this review may involve the customer; project management and personnel from the product engineering, configuration management, product test, product engineering, and quality assurance groups; and various groups concerned with operating, fielding, and supporting the product. After the formal review is successfully completed, the configuration management group is then charged with the responsibility for overseeing the control of the documented requirements to prevent unauthorized changes to them.

The baselined system requirements become the point of departure for developing the requirements for all the major components (e.g., subsystems). The product engineering organization should take the lead for ensuring that the requirements stated at the component level (e.g., the software requirements) are compatible and consistent with the system level requirements.

1.4.1.3 All Systems

The manager and subordinate managers responsible for product engineering (or component development) are accountable for implementing the requirements as

7. We define a system as a product comprised of two or more interacting components that can be separately developed and controlled. These components may be software, hardware, and/or personnel.

established and for assuring that they are not changed in an unauthorized manner. The quality assurance group may be responsible for monitoring the configuration management process to verify that no unauthorized changes have occurred. The quality assurance group may also be responsible for conducting audits to verify that the established requirements development process was followed [see Generic Practice (GP) 2.9 in the REQM and Requirements Development (RD) PAs in the CMMI®].

For this element of the Quality Program, then, we see that at least the following organizations are active in establishing and controlling the product quality requirements for engineering/scientific applications: user/customer organizations, product engineering, product test, configuration management, quality assurance, project management, and various support groups, such as a logistics group, field maintenance group, and the like. For IMS systems, it may involve the users, the IT development organization, quality assurance, and configuration management.

Within the structure of the CMMI®, there is a GP that exists within each process area, GP 2.7, "Identify and Involve Relevant Stakeholders." One of the roles of the quality manager (QM) clearly is to ensure that the relevant stakeholders, such as illustrated here, are properly identified and involved in the Requirements Definition and Requirements Management processes.

1.4.2 Establish and Implement Methods

Establishing and implementing methodologies to develop the product and maintain its quality include establishing the methodologies themselves and institutionalizing them in the form of standard practices, procedures, tools, and methodologies. These methodologies, practices, and procedures cover a wide number of areas. They include requirements analysis, documentation, design, coding, test, configuration management, installing and operating the product, and product maintenance.

In implementing this element of the QP, interactions occur with a number of organizations. Product engineering must be involved in the definition process since they will be the ultimate users of the methodologies, standards, procedures, and associated tools (if applicable). An interface with the quality evaluators exists. First, when the methodologies are initially developed, the points in the process where quality evaluation tasks must be performed need to be identified, along with the methodologies for performing the quality evaluations. Second, from time to time, changes are made to the specified methodologies and implementing documentation and tools. Consequently, it may be necessary to change the corresponding quality evaluation process. These changes may occur under two conditions: (1) the specified methodologies, documentation, or tools are not producing the required levels of quality; or (2) new methodologies have become available that will materially improve the quality of the product. Once the changes have been made to the processes, they must be monitored to determine if, in fact, improvements have been made. The determinations of methodology adequacy result from product and process evaluations. The personnel performing product quality evaluations typically provide the raw data for evaluating existing, new, or modified methodologies and tools, while product engineering personnel generally do the analyses of the data or of the methodologies. At the highest maturity levels on the CMMI®, such

evaluations are typically performed on a quantitative basis. The project manager must be consulted regarding the adoption of new methodologies and/or tools to determine if such changes will negatively impact productivity, schedule, and/or cost for that project. Operations personnel, such as product librarians and database administrators for software or equipment operators for hardware, must be consulted to determine the effect on operations. Personnel must be assigned the task of producing standards and procedures for implementing the methodologies and using the tools in a manner compatible with the established standards. Clearly, company management must be involved in this element of the QP because of the investment in personnel to staff the function, as well as approval or disapproval for the acquisition of new methodologies, and tools to implement the methodologies.

Again, the multidisciplinary nature of the QP is evident. One can deduce from this that many organizations are involved in establishing and implementing the methodologies for development and maintenance and producing standard practices and procedures for these functions. In organizations that have adopted the CMMI® as the model for process improvement, the function of coordinating these activities is often assigned to a centralized function, sometimes referred to as an Engineering Process Group (EPG), Product Engineering Process Group (PEPG), or in organizations that are primarily software development organizations, a Software Engineering Process Group (SEPG). We will discuss this group in more detail later in this chapter. It should also be noted that the lower maturity level organizations tend to follow a more heuristic and qualitative approach to process change, whereas Maturity Level 5 organizations follow a structured and quantitative approach for implementing process change (see the description for the Organizational Innovation and Deployment process area [4]).

1.4.3 Evaluate Process and Product Quality

Finally, we come to the element of *evaluate process and product quality*, or Quality Evaluation (QE) activities. Activities involved here cover the establishment of standard processes and procedures for performing evaluations and also for implementing these evaluations in order to determine (1) the quality of product and (2) the quality of the processes and activities impacting the quality of the product.

The number of organizations involved in performing the QE activities can be large. Considering that QE includes analytical as well as measurement activities, it is easy to see that QE is a discipline that encompasses engineering as well as support groups. For example, analyses may be performed by systems engineering or a product engineering group. Tests may be performed by an integration test team or an independent product test group (or both), possibly with a quality assurance group monitoring. In some companies, a quality assurance group does testing as well.

Project reviews may include project management, and the system engineering, product engineering, configuration management, and quality assurance groups. Certainly a quality assurance entity would participate in and conduct audits. The configuration management and quality assurance groups would be involved in document reviews as would the system and product engineering groups.

In any event, it can be seen that the activities involved in QE requires the talents of almost all groups participating in the development process. See Chapter 13 on development quality assurance for an in-depth discussion of this point.

To complete this discussion of QE, it is imperative to introduce the concept of independence. As discussed earlier, relative to QE, independence implies performing product quality evaluations by an organization (or individuals) different from the organization (or individuals) that produced the products or documentation, or that execute the processes and activities being evaluated. Independence extends to establishing the evaluation criteria. The performer may have a conscious or unconscious need to make the process or product look good. Evaluators so inclined can hardly be considered independent. By removing from them the responsibility for establishing the evaluation criteria and performing the evaluations, such problems cannot arise. The criteria for the evaluations must be based on the requirements for the product, hence the importance of establishing good requirements, and ensuring that the user's or customer's needs are accurately reflected in the requirements documents.

Independent QE requires the collection of objective evidence that technical requirements have been established, products and processes conform to technical requirements, and that the products meet the specified quality requirements. This may mean that one organization does a specific evaluation, but another organization establishes the criteria for the evaluation, verifies that the evaluation has been performed, and impounds the data for eventual use in certifying the product or service. "Objective evidence" includes such items as measurement data, audit reports, certified test data sheets, verification and validation (V&V) reports, resolved product trouble reports, and the like.

1.5 Mapping Quality Program Functions to Project Organizational Entities

Numerous organizational structures can be applied to implement the Quality Program. The important point at the project level is allocating the related tasks to corporate organizations available to the project manager. This allocation of these tasks depends upon several interrelated factors. Obviously, one factor is the business structure and guidance established by the corporation or by the project manager to accomplish the project. The structure and guidance given to the project manager eventually reduces to authorized funding and permissible execution control within the corporate structure, both of which limit the flexibility the project manager has to conduct projects. Another factor is the extent and complexity of the tasks and the availability of personnel to perform them.

In many cases, the corporation has predetermined the responsibilities for these tasks, thereby predetermining the allocation of them. This a priori assignment of tasks may restrict the project manager in how he or she mobilizes a particular project and structures the Quality Program (which involves the coordination of so many disciplines). One way a project manager can help insure proper coordination is to appoint a quality manager to his or her staff. (But it must be remembered that even with the appointment of a QM, the project manager is still ultimately responsible for the Quality Program.)

If we assume that most, if not all, necessary resources and talent are usually available for the project manager's execution of the Quality Program, the project manager's task reduces to coordination of assigned activities. The project manager can choose to assign a QM to coordinate the Quality Program activities. If the necessary resources and talent are not available, the project manager must secure these through negotiation with company management and company subordinate entities from which the resources will be obtained.

The purpose of assigning a QM is to support the project manager in providing a quality framework for the project and, more important, making the Quality Program more visible to the rest of the project manager's organization. The quality manager does this by insuring that the Quality Program is planned as part of the overall product development process, by insuring that the Quality Program is implemented, and by keeping the project manager informed and on track with the overall product development. Based upon the definition of quality (i.e., product attributes including functionality, performance, and so forth), the quality manager has the tasks of planning and coordinating all the disciplines involved in the project. In this context the quality manager is the technical lead for the project. Again, based upon the definition and implications of product quality noted earlier in this chapter, the term quality manager does not imply that the individual is from the QA group, or that the individual is only managing the QA portion of the project. The quality manager has a much broader responsibility, especially in the coordination of all the activities that "build" quality into the product, not just simply testing for it. Note that there are some overlaps between the functions that a PEPG/EPG/SEPG and the QM would do. However, the PEPG/EPG/SEPG has responsibility for these activities across the entire organization, whereas the QM has responsibility for applying these activities to the project only. In the context of the CMMI®, there is a number of process areas that implement the various elements of the Quality Program described herein at the project level. The function of the project's quality manager is to ensure that these process areas are implemented as a cohesive whole, rather than as a set of unrelated, independent process areas.

Starting with the critical aspect of project planning, the following addresses organizational considerations in the mapping of Quality Program functions in terms of the Quality Program elements.

1.5.1 Planning

The quality manager must be an integral part of the project planning to insure that the Quality Program is addressed. He or she must play a very active role in this effort, setting up all the steps to follow in executing the Quality Program, including those in the evaluation effort.

Important in performing this role is the development of the Quality Program Plan. This can be either a major subset of the project management plan, or may be a separate document that is referenced within the project management plan. In any event, the vital task of the quality manager during the planning phase is to produce the Quality Program Plan.

The quality manager must work very closely with all participants in the project in order to generate the Quality Program Plan, specifically, to ensure that:

- The plan is produced.
- The plan is complete and the elements of the plan are integrated into the project management plan.
- The activities to be performed are integrated with each other to the extent that they should.
- The plan contains realistic schedules.
- The plan describes assignment of responsibilities and designates necessary authority to the appropriate performing organizations.
- Expected Quality Program outputs for the project are specified.
- Criteria for successful completion of tasks are stipulated.

During development, the quality manager uses the results of the evaluation efforts to track the progress of the Quality Program against the Quality Program Plan. A primary concern is not simply to determine compliance with the plan, but, more important, to determine if application of the planned activities of the Quality Program will achieve the desired quality, or, if the plan must be changed to effect the desired quality.

1.5.2 Establish Requirements and Control Changes

During the process of establishing the project and product quality requirements, the quality manager must have the authority to represent the project manager. Here, the quality manager ensures that the appropriate process is followed, and that the process is properly managed. As indicated earlier, a number of organizations (or functions within an organization) are typically involved in defining and establishing functional and performance requirements. These may include product engineering, user organizations, system engineering, and so on. Other groups, such as those representing human factors or maintenance, must have a chance to participate in the requirements definition process in order to ensure that their needs are also reflected in the requirements documentation. The kinds of groups involved will depend on the type of application under development. As the number of these groups increase, the job of establishing the requirements becomes more and more difficult. Having the quality manager coordinating and managing this process for the project manager and ensuring that the process is followed simplifies control and ensures that requirements are established and that they are quantitative, testable, and complete.

The quality manager can use several methods of accomplishing this process, orchestrating the various groups involved. For example, for software, he or she may depend totally upon the product engineering group or IT development group to both specify the processing requirements and perform checks (assessments) as to their adequacy. On the other hand, the quality manager may use some groups to define the requirements, and other groups to perform the evaluations. In some cases, the evaluations may be split between the developers and the evaluators. For instance, the assessment for traceability might be performed by the product developers, instead of other designated evaluators, utilizing the traceability capabilities embedded within the software engineering tools being used to develop the requirements.

Whoever is assigned to making these evaluations is designated in the Quality Program Plan.

As pointed out previously, there is an interface between the requirements definition element and the quality evaluation element of the Quality Program Plan. Requirements development involves a strong interplay between requirements analysis and QE. The requirements must be evaluated as they are being developed to make sure that the job is being performed completely and correctly. The quality manager utilizes those personnel designated in the Quality Program Plan to make such assessments (perform Quality Evaluation) and provide some independence. The quality manager uses the outputs of the assessments to:

- Ensure that the evolving requirements are modified where necessary.
- Ensure that requirements become baselined, when stable.
- Assist in revising the process of establishing requirements.
- Assist in changing methodologies used in this process.
- Enforce the procedures originally planned for this part of the Quality Program Plan.

Communication (see Figure 1.7) between the two elements can be conducted totally through the quality manager.

1.5.3 Establish and Implement Methods

As with the requirements portion, the second element of the Quality Program is easy to accomplish within the project structure by assigning responsibility for this function to the quality manager. There are really three parts to this job: (1) establishing the methodologies to be used for the project, (2) enforcing the methodologies, and (3) modifying the selected methodologies, when necessary.

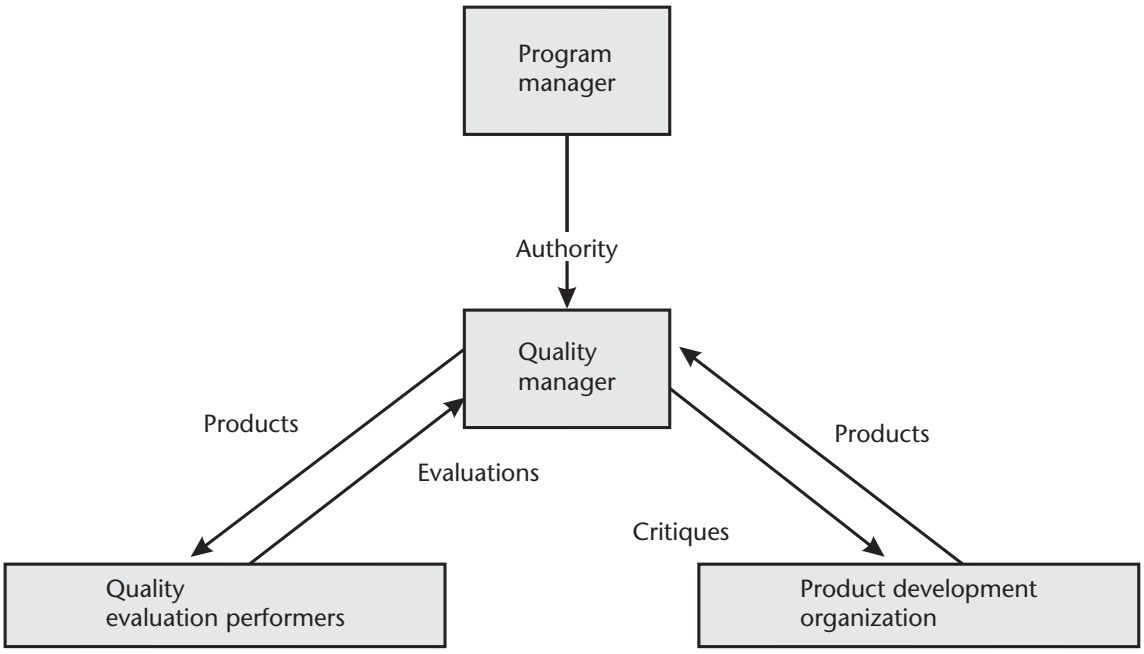

Figure 1.7 Quality manager communications.

One way in which the accomplishment of the first part of this element of the Quality Program can be facilitated is by establishing a Product Engineering Process Group (PEPG) [4] at the organizational level. A PEPG typically is a corporate asset that evaluates and selects methodologies for use by the organization and supports each project in selecting appropriate methodologies. It is the focal point for the methodology element of the Quality Program. Its main function is to serve as the initiator, sustainer, and evaluator of process change. In terms of the CMMI® model, this is the focus of the Organizational Process Focus (OPF) and Organizational Process Definition (OPD) process areas. The PEPG establishes a set of process assets and tailoring guidelines that are used by the project's quality manager to tailor or adapt the organizational process for use by the project. This is one of the intents of the Integrated Project Management (IPM) process area in the CMMI®.

At the very outset of a new project, the applicability of the established methodologies, techniques, and tools for the product to be developed is determined. If a PEPG exists within the company, the quality manager must consult with that function in order to adequately carry out this assignment. The methodologies, which are established by the PEPG, are established for use throughout the entire organization according to the different types of products produced by the organization. It may be necessary to modify these methodologies to suit the unique characteristics of the product to be produced on this project, utilizing the aforementioned tailoring guidelines. The QM, in conjunction with the PEPG, makes this determination and oversees the modifications, if required. These modifications will be reflected in the form of project-specific modifications to the standards and procedures. The tailoring of the standard processes for the unique characteristics of the project must go through an approval process, and the resultant modifications identified in the project plan.

When a PEPG does not exist, the QM then must assume much of the responsibility and coordination effort that the PEPG would have performed. In establishing methodologies to use in order to achieve the desired quality attributes for the product, the quality manager must bring to bear a wide range of disciplines, not just product engineering. The intent of this effort is to select those product engineering methodologies that offer the best promise of producing a product meeting all the specified requirements—an extremely difficult process due to varying maturity in available product engineering techniques. The quality manager must further assure that the interfacing disciplines (e.g., product engineering, testing, configuration management, and so on) are communicating with each other and coordinating on the methodologies to be employed on the project to assure that they are mutually compatible.

Once the project is started, the QM is responsible for enforcing the implementation of the methodologies (the second part of the job). This is accomplished by setting policy and monitoring the development, operation, and maintenance activities to verify that policy is being followed. Enforcement often depends upon an assessment or measurement of products and development, operation, maintenance activities, creating an interface between this element of the Quality Program and the QE element of the Quality Program. Products include preliminary and final versions of documents and preliminary hardware and/or software product releases. Since methodologies are procedural in nature, other kinds of products may be used to evaluate whether the processes are being properly implemented in the development activities.

These may be interim work products or work products resulting from other activities. For example, reports of peer reviews may be used to determine if the developers have followed the prescribed methodologies.

The methodologies established for the project must also be evaluated during the life of the project to determine if they are, in fact, achieving the desired results. They must be modified, and corrective action must be initiated if they are not. The basic information on which a decision to modify the methodologies is based depends on the CMMI® maturity level of the organization. At Maturity Level 3 and below, the decision is based primarily on the results of product quality evaluations, and process performer subjective perceptions of the processes, based on their own experience in using them. At Level 4 and above, the decision is based on quantitative process measurements. These adjustments (or corrective action of the processes) are initially at the project level; however, if it is determined that the corporate process is deficient, the corporate PEPG would take on the task of enterprise-wide corrective action or long-term improvement of the process.

Because of the interfaces that exist between this and the Quality Evaluation elements of the Quality Program, it becomes readily evident that the quality manager is the most logical individual to assign as the one responsible for ensuring that this job is properly coordinated and accomplished.

1.5.4 Evaluate Process and Product Quality

The QM is also responsible for the implementation of the Quality Evaluation program. The Quality Program Plan should have defined the totality of assessment and measurement activities and assigned these to the appropriate performing organizations. Clearly, the QA organization can be a major performer, and as indicated previously, a number of other organizations are likewise involved. Accordingly, it is essential that the QM completely and totally define the tasks and performers.

Quality Evaluation is the major instrument defining the health of the product and hence the project. Through the evaluations performed, the PM can determine if his or her product will satisfy the customers' or users' needs within cost and within schedule. Because of the number of organizations involved in the Quality Evaluation process, coordination of the results of this process is an essential role to be performed by the QM.

Whatever decision management makes, it must be sure that all Quality Evaluation activities have been assigned to an organization competent to perform that function and, where independence is specified, to an organization with the proper detachment as well.

1.6 Example Organizational Implementations of a Quality Program

A major determinant as to how the Quality Program is to be implemented is the size of the organization. A small organization, comprised of a number of small projects, cannot implement the Quality Program in the same way that a large organization can. The next section examines some approaches that organizations have used in

implementing a Quality Program. We also describe the implementation of the PEPG concept.

1.6.1 Project Engineering Process Group

Many companies have adopted the PEPG concept. It is an important factor in successful implementation of the second element of the Quality Program, *establish and implement methods*. The PEPG is typically the focal point for methodology selection and evaluation. This has come about with the recognition that it is difficult to begin the process improvement journey without a centralized function responsible for it, regardless of the application domain in which the organization specializes.

Fowler [14] describes strategies for the implementation of PEPGs into the organizational structure. Organizational size is taken into account in the strategies discussed. We refer you to that technical report for a more comprehensive discussion of the organizational considerations in forming a PEPG.

1.6.2 Quality Program Structures in Large Projects

1.6.2.1 Large Development Project

The easiest organization structure to describe is that which exists for large organizations producing engineering or scientific applications. Figure 1.8 illustrates an organization chart from an actual project, although somewhat disguised to protect the identity of the actual organization. In the figure, the acronym APM means assistant project manager. In this structure, the quality manager, or, in this case, the project quality manager (PQM), as this person was called, was responsible for planning the performance of the Quality Program and documenting the output of the planning effort in the appropriate plans, coordinating the activities of the performers of the Quality Program activities, and monitoring their performance to verify that they were being performed properly.

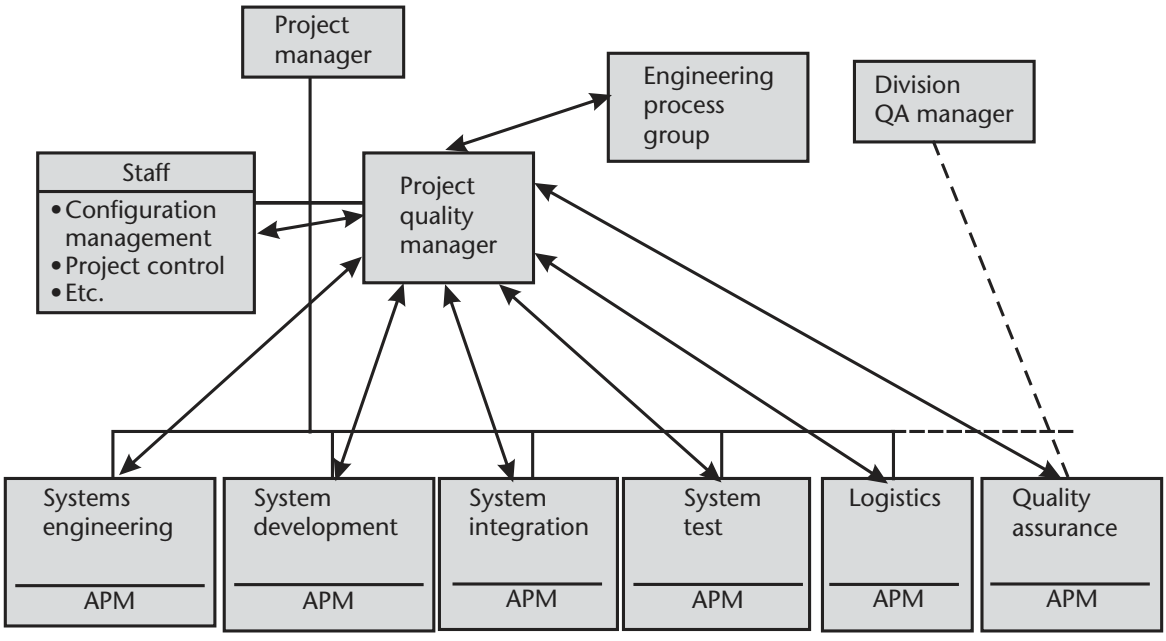

Figure 1.8 Example of a large project organization.

For the requirements element of the Quality Program, the organizations involved in the requirements definition effort included Systems Engineering, System Development, and Logistics. The Logistics organization participated in the definition of the maintainability and product supportability requirements for the operational product. In this structure, the PQM was responsible for coordinating and integrating the requirements definition activities of these areas of the project. The PQM, as can be seen from the figure, also coordinated with the configuration manager with regard to establishing the baseline for the requirements.

To establish and maintain the methodologies to be utilized on the project, the PQM coordinated with the EPG. The EPG was responsible for coordinating with the other organizations within the company with regard to establishing the methodologies in general usage and for determining their effectiveness. (Its position on the chart has no significance with respect to hierarchy or importance. Its position is only intended to show that, as an enterprise-wide resource, it was outside the organizational structure for the project.)

Product quality evaluation was performed by Quality Assurance, Product Test, System Integration and Test, and System Development. The PQM coordinated and monitored the performance of the product quality evaluation elements of the Quality Program. The functions that each organization performed in support of the quality evaluation element of the Quality Program were documented in the Product Quality Evaluation Plan (PQEP). Feedback of the evaluation results into the development, operations, and maintenance activities and products was provided for in the PQEP. The coordination and monitoring of the feedback process was another function performed by the PQM.

Because the PQM was a staff function to the project manager, he had a direct line of communication to him to ensure that all project staff members complied with the requirements of the Quality Program. In the event of a noncompliance that could not be resolved directly with the individual or organization involved, the PQM could call on the project manager to enforce compliance.

1.6.2.2 Integrated Product Team: A Special Case

Another organizational structure (shown in Figure 1.9) that has been effective is the integrated product team (IPT). This is sometimes used on large projects involving multiple contractors. Often, concurrent engineering is also involved. The intent of the IPT concept is to ensure effective communication of project-critical information between all members of the team, and all stakeholders involved in all aspects of the product life cycle. This is often accomplished through colocation of the team members. IPTs will often include customer representatives, prime contractors, and subcontractors to encourage rapid resolution of contractual issues, as well as speedy clarification of requirements-related questions.

IPTs may exist at various levels. For instance, in Figure 1.9, we see that IPTs exist at the system, segment, and subsystem level. Since a product exists at each of these levels, a PQM could exist at each level shown. For instance, one would exist at the space segment level, and one could likely exist for each one of the subsystems comprising the space segment. Furthermore, if the lower level subsystems were

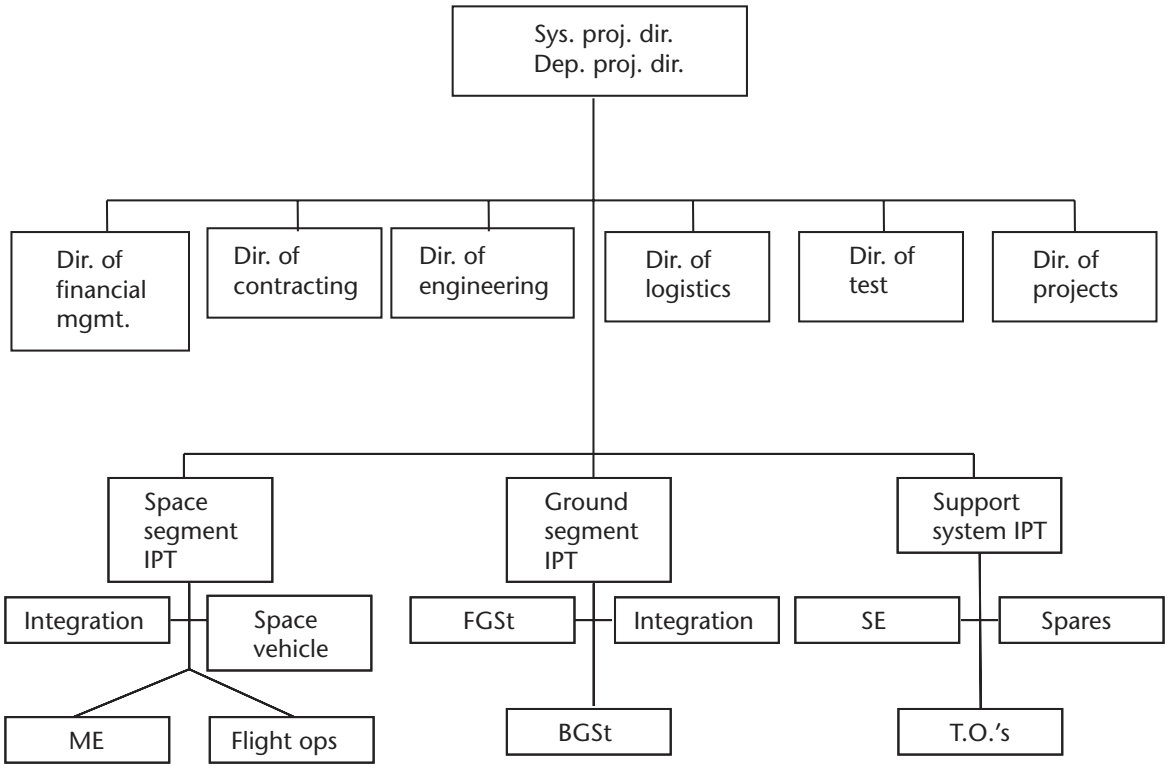

Figure 1.9 Example of a large project organization with IPTs.

sufficiently large and complex, IPTs could exist at lower levels. A PQM would be a member of each of these IPTs, as well, if each had a significant product component.

1.6.3 Quality Program Structures for Small Projects in Large Organizations

For small projects in large organizations, a PQM serves several small projects in a part-time capacity in each. For really small projects (three people or less), the project manager is undoubtedly performing some development roles, as well as the project management functions. In this case, the PM will be more dependent on a PQM to ensure that all the quality functions are being performed. Tailoring guidelines should exist to ensure that the Quality Program activities are commensurate with the size and criticality of the projects to avoid placing an onerous burden on the projects in complying with the Quality Program.

1.6.4 Quality Program Structures in Small Organizations with Small Projects

Small organizations face a totally different picture when it comes to implementing the elements of a Quality Program. In this situation, a number of conditions may exist. Two example situations are as follows: (1) the company is a one-project company, or (2) the company is working entirely on a number of small projects. Figure 1.10 is an example of how one IT department organized to implement the Quality Program. Again, the structure is somewhat disguised to protect the identity of the actual organization

Within the IT department, the IT standards committee fulfilled the function of the PEPG. It was comprised of key members of the department including the IT

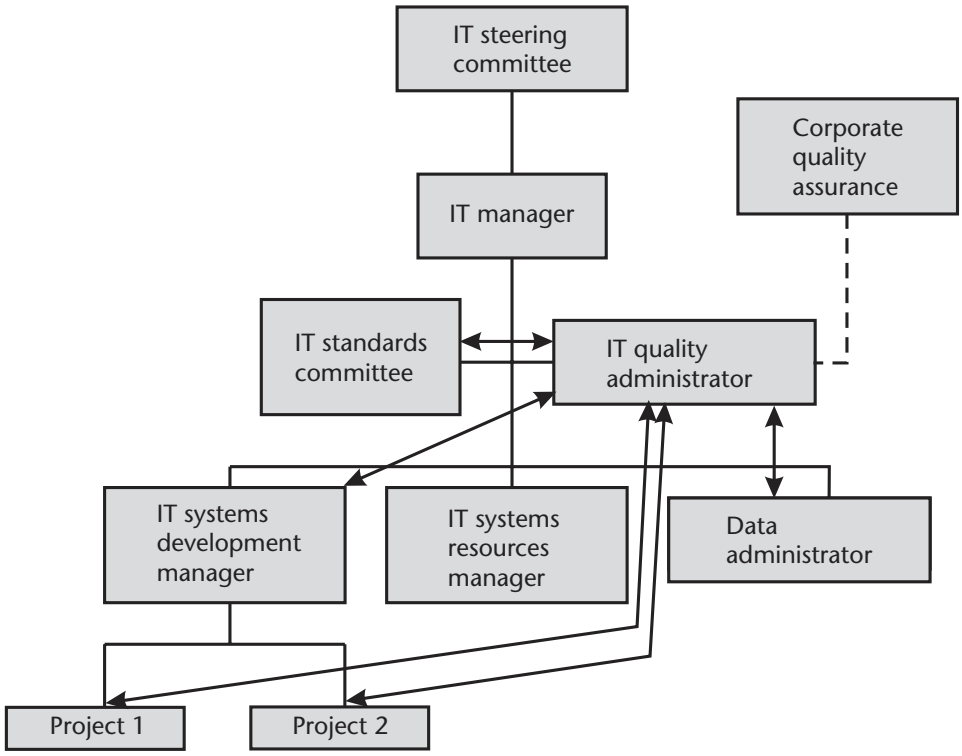

Figure 1.10 Example of a small IT organization.

quality administrator and representatives of the development, system resources, data administration, and configuration management areas of the department. Because of the size of the department, none of the members were assigned full time to the standards committee to do its work.

The IT quality administrator reported administratively to the IT manager, and was an employee of that department. By company policy, the IT quality administrator was deputized to act on behalf of corporate quality assurance to ensure that the provisions of the corporate Quality Program were carried out. The IT quality administrator had a responsibility to corporate quality assurance to provide periodic reports on the activities of the IT Quality Program. Note that in this case the quality administrator was not independent. The intent of independence was achieved, however, through a reporting channel to corporate quality assurance and periodic audits by corporate quality assurance to ensure that the provisions of the applicable policies and procedures were being correctly implemented.

In this structure, the IT quality administrator acted more as a coordinator and monitor with respect to the Quality Program functions. The responsibility for defining requirements was shared between the user community and the project. Requirements definition was performed in accordance with the procedures defined in the IT standards manual, and the individuals responsible for performing this task, the outputs they produced, the informal and formal reviews to be held, and the schedule for the entire activity were documented in the software development plan (SDP) for the project. The IT quality administrator monitored the activity to ensure that it was being performed as prescribed by the IT standards manual and the SDP. Any conflicts regarding implementation that could not be resolved directly with the development or user project leaders were raised to the IT systems development manager for resolution.

The responsibility for the methodology element of the Quality Program was vested in the IT standards committee. They performed the function of the PEPG. The IT quality administrator was a member of the IT standards committee and ensured that this function was being properly executed. The project-specific modifications, if applicable, to the standardized methodologies were documented in the SDP. Project-specific adaptations to the standards and procedures were also identified in the SDP. The IT quality administrator was a signatory party to the SDP, and consequently could coordinate and monitor the application of this aspect of the Quality Program for the project.

The QE element of the Quality Program was handled in a unique way by this company. The typical project size was approximately three to four developers. Because of the size of the entire organization, and the size of the projects, only two people—the IT quality administrator and one assistant—were dedicated full time to Quality Program tasks. Each project had their own part-time quality evaluator, and this person was also a part-time developer. He or she was responsible for performing the quality evaluations. Where necessary, the quality evaluators could call on other resources elsewhere within the IT department or within the affected user community to assist in the quality evaluations. For instance, in performing a quality evaluation of a requirements specification for a payroll program, the quality evaluator could call on personnel within the accounting department to assist in the review of a document.

The results of each evaluation were documented on a quality evaluation record. These were entered into a log and into a database. Both were available online. A major function performed by the IT quality administrator was auditing each individual project for compliance with the software quality evaluation plan (SQEP) and the standards and procedures specific to QE contained in the IT standards manual. Since the SQEP contained the definition of the QE tasks to be performed, the person responsible for performing it, and the schedule for its performance, the IT quality administrator could use it to determine when to perform the audits. The database was queried to determine if a record existed of a given evaluation's performance. The IT quality administrator had the authority to review the record and spot check the product itself to ensure that the review was performed in accordance with the approved procedures. The IT quality administrator could also participate in a review performed on an activity or product.

Another responsibility assigned to the IT quality administrator was the audit of the configuration management functions. The configuration management functions were distributed to various projects. The development baseline was under the control of the development project leader, which resulted in another interface with the project leader, and the production baseline was under the control of the IT software configuration control board, which was chaired by the IT manager. Changes to the applications product, corporate and project data dictionaries, and databases were handled by the librarian, data administrator, and database administrator, respectively, and their functions were audited by the IT quality administrator.

Audits performed on the IT area by corporate quality assurance determined if these functions were being properly performed by the IT quality administrator. Other related methods for small projects are covered in Chapter 12.

1.7 Summary

In organizing to implement a Quality Program, several concepts must be kept in mind.

First, it must be emphasized that the foundation for the organization is tied to achieving the requisite product quality. One must understand what product quality is and the technical aspects of specifying, developing, and evaluating it. Product quality is achieved with proper product design and implementing appropriate processes and methodologies. Quality cannot be achieved by "assuring" and "testing" the product.

Second, the ideas associated with product quality lead to the Quality Program. General principles of such a program have been discussed. Three elements of the Quality Program were described in some detail; these elements interact not only with each other but also with all other project activities. This interaction is extremely complex, occurring at many levels within the development project and throughout a project's life.

From the perspective of the Quality Program, an organization can be derived based upon corporate structure (controlling policies) and available talent. It is recommended that the project manager be allowed to structure his or her own project organization without the restriction caused by a priori corporate organizations. The project manager needs to recognize and understand the Quality Program. Given this understanding, the project manager allocates tasks of the Quality Program to those with appropriate talent. Because of its broad nature, the Quality Program requires a range of disciplines including product engineering as well as evaluation expertise. It is recommended that a quality manager be appointed who is steeped in this expertise and in the methodologies needed to achieve product quality.

References

[1] Baker, E. R., and M. J. Fisher, "A Software Quality Framework," *Concepts—The Journal of Defense Systems Acquisition Management*, Vol. 5, No. 4, Autumn 1982.

[2] Humphrey, W. S., "A Software Process Maturity Model," *IEEE Software*, Vol. 10, No. 4, July 1987.

[3] Paulk, M. C., et al., *The Capability Maturity Model: Guidelines for Improving the Software Process*, Reading, MA: Addison-Wesley, 1995.

[4] Chrissis, M. B., M. Konrad, and S. Shrum, *CMMI®: Guidelines for Process Integration and Product Improvement*, 2nd ed., Reading, MA: Addison-Wesley, 2006.

[5] Cooper, J., and M. Fisher, (eds.), Software Acquisition Capability Maturity Model (SA-CMM®) Version 1.03 (CMU/SEI-2002-TR-010, ADA399794), Pittsburgh: Software Engineering Institute, Carnegie Mellon University, 2002, available at http://www.sei.cmu.edu/publications/documents/02.reports/02tr010.html.

[6] Kenett, R. S., and E. R. Baker, *Software Process Quality: Management and Control*, New York: Marcel Dekker, 1999.

[7] Fenton, N. E., and R. Whitty, "Introduction," in *Software Quality Assurance and Measurement, A Worldwide Perspective*, N. Fenton, R. Whitty, and Y. Iizuka, (eds.), London, U.K.: International Thomson Computer Press, 1995.

[8] Melton, A., *Software Measurement*, New York: International Thomson Computer Press, 1996.

[9] Baker, E. R., and M. J. Fisher, "A Software Quality Framework," *Fourth International Conference of the Israel Society for Quality Assurance*, Herzliyah, Israel, October 18–20, 1982.

[10] Voas, J., "Software's Secret Sauce: The –ilities," *IEEE Software*, November/December 2004.

[11] Baker, E. R., and M. J. Fisher, "Organizational Aspects of the Software Quality Program," in *The Handbook of Software Quality Assurance*, 3rd ed., G. G. Schulmeyer and J. I. McManus, (eds.), Upper Saddle River, NJ: Prentice-Hall, 1999.

[12] Members of the Assessment Method Integrated Team, Standard CMMI® Appraisal Method for Process Improvement (SCAMPI℠), Version 1.1: Method Definition Document (CMU/SEI-2001-HB-001, ADA3399204), Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2001, available at http://www.sei.cmu.edu/publications/documents/01.reports/01hb001.html.

[13] Schulmeyer, G. G., "Software Quality Lessons from the Quality Experts," in *The Handbook of Software Quality Assurance*, 3rd ed., G. G. Schulmeyer and J. I. McManus, (eds.), Upper Saddle River, NJ: Prentice-Hall, 1999.

[14] Fowler, P., and S. Rifkin, "Software Engineering Process Group Guide," Pittsburgh, PA: Software Engineering Institute, CMU/SEI-90-TR-24, September 1990.