

Personnel Requirements to Make Software Quality Assurance Work

Kenneth S. Mendis

4.1 Introduction

This chapter has a primary focus of software quality assurance (SQA) personnel who perform as evaluators/auditors of the software development and management processes and work products. There is recognition of the validation/testing activities many SQA personnel perform, but this is not the primary focus in the chapter. Obtaining qualified engineers and keeping them motivated in what they are doing is a problem most of the engineering disciplines have been facing for some time. The problem is compounded when we focus on the software engineering discipline. At the level of SQA, we find ourselves battling with the software developers for the few software engineers who are available.

To be effective and contribute to a project's success in a manner that is professionally acceptable, the SQA organization must be staffed with qualified software engineers. In addition, these individuals must also possess the credentials that make them good quality assurance representatives. Achieving any of the promised benefits of SQA is directly related to an organization's ability to staff the operation. Some of the issues that the manager will be confronted with are engineer motivation, career training, and recruiting techniques.

The commercially available *Software Life Cycle Process* standard, ISO/IEC 12207, is a well-known standard containing sections on software quality as well as auditing techniques. Additionally, ISO 9001:2000 and the ISO 90003 *Guidelines for the application of ISO 9001:2000 process requirements to the development, supply and maintenance of software* are also influencing the software development and quality decisions of many organizations. Also along these lines is the Code of Federal Regulations that direct the Food and Drug Administration (FDA), as it monitors food production, the development of medical devices, the medicines we take, and other FDA relevant consumer products and services. Each standard defines a structured approach for developing software, and with that approach comes the need to staff positions within the organization to enforce the plans that have been set in motion. Unfortunately, the glamour and challenges provided by a developing environment attracts the interest of the majority of software engineers.

This leaves SQA with a limited number of qualified personnel from which to choose.

The Capability Maturity Model Integration[®] for Development (CMMI[®]-DEV) has had a significant impact on SQA through its Process and Product Quality Assurance process area. Personnel working in a CMMI[®]-DEV environment must be knowledgeable to perform the practices in the Model. Particularly essential is how to perform, record, and promulgate process and work product evaluations/audits.

At a high level, the personnel requirements that make SQA work are as follows:

- Approximately 3 to 5 years developing software;
- Experienced software engineer who has seen it all and has survived the software battles;
- Individual seeking to advance to management or a program manager's position;
- Good communication skills;
- Computer science academic background;
- Willingness to meet and accept new challenges.

There will be further discussion later of these important attributes.

The Defense Logistic Agency has developed an approach to software quality assurance. The procedure, known as the Single Process Initiative for In-Plant Quality Assurance makes use of continuous improvement tools and problem solving techniques to examine the adequacy of a contractor's process to continuously produce conforming products and to identify opportunities for product improvements.

The concept of Single Process Initiative includes management commitment, people development, quality excellence, and user satisfaction. Implementation of Single Process Initiative embraces techniques that use process and product quality to evaluate the quality of an organization's software products. Single Process Initiative focuses on working with the software developers, working with the software users, and working with contracting agencies to produce a product that meets the users' needs. It means working as a team to measure and continuously improve the process.

4.2 Facing the Challenge

Why consider software quality assurance? A review of warning letters issued by the U.S. FDA to firms in the United States, Europe, and Asia, as a result of formal inspections, highlighted the weakness of firms to adequately institute a SQA program that contributes to a more complete and consistent software design, development, testing, documentation, and change control. In most cases reviewed, FDA actions usually had a negative impact on the validation status of the computer system and on a firm's bottom line.

Imagine the impact on a war if the United States Army had to wait to validate the performance of its missiles after installation in a war zone. What if the computer control and guidance system had not been validated to do what it was supposed to do every single time—that is, intercept and destroy incoming enemy missiles? Or in another instance, picture yourself several hundred feet below sea level sitting in front

of a computer terminal in the submarine command and control center ready to fire a missile. What if the installed guidance software used to deliver the missile had not been validated? Think of the destruction it would cause if it ended up somewhere hundreds of miles from its intended target! SQA plays an important part in the outcome of recent wars, where software-intensive systems are used.

If the SQA process helps to assure systems that land men on the moon and bring them back safely, defend a country against external attack, help pilots land safely at our busiest airports during a blinding snow storm, then why not apply such a trustworthy methodology to all software development efforts?

Our academic institutions today still do not provide the required training for SQA engineers. An SQA engineer is a software engineer trained in the disciplines of software quality assurance. Today, little or no training is provided in the techniques of software design review, good software documentation, and software reliability and maintainability. Training is also inadequate for software attributes such as the use of program design languages, top-down development, object oriented methods, and structured programming techniques, which are used to assess and measure the progress of software development. About the only way an individual becomes knowledgeable of SQA principles and disciplines is by hands-on work-related experience, which only makes the SQA staffing problem even more difficult.

Individuals involved in developing, staffing, and maintaining an SQA organization within their company are familiar with the daily battles of SQA staffing. It is not uncommon to search through countless résumés and interview many applicants in an effort to find those individuals who would make good SQA engineers. In many instances, if an applicant is technically acceptable, then he or she still lacks those special attributes that turn a software engineer into an SQA engineer. Therefore, recruiting and hiring qualified personnel to staff SQA positions is expensive and time consuming.

Thus, before the organization proceeds on a recruiting campaign, it is necessary to define and set priorities for those issues and positions that are critical to the success of the SQA function. The organization must consider the following factors:

- Is it possible to promote from within and train individuals to fill the openings?
- Can contract employees help fill the organization's needs, and if so, in what capacity should these employees be utilized?
- Should the recruiting effort be national, regional, or local?
- What does it take to attract qualified trained individuals to your company?
- What does an organization do to retain its qualified staff?

Another problem that one has to face is how to hold the qualified individuals' interest and motivation in job assignments. Developing and outlining career paths is another important factor in the problems facing SQA staffing. Indeed, in my opinion, the most serious problem a manager faces is to prevent his or her department from becoming a stepping stone to other opportunities within the organization. The SQA department, if it is to develop into an effective organization dedicated to assuring a product's software quality, must consist of professionals both seasoned in software and dedicated to quality, and capable of providing guidance, training, and quality-consciousness within the organization.

Lastly, top management support is of prime concern. The lack of top management support or lack of a clear understanding of SQA's needs is perhaps the major issue confronting most SQA organizations today. To properly staff the organization, management must clearly understand the problems of personnel and assurance goals and be willing to address them. Support and understanding must go hand in hand; one without the other is ineffective.

4.3 Organization Structure

In April 1979 the Software Management Subgroup of the Joint Coordinating Group on Computer Resource Management (JCG, CRM) sponsored a Joint Logistics Commanders software workshop. One of the key findings of the workshop stressed the difficulties facing the implementation of SQA, such as the lack of a well-defined and consistent set of requirements, differences in SQA approaches across the various branches of the services and industry, and the unavailability of a good source of experienced personnel. Nearly three decades later, this requirement continues to be a concern within organizations.

Experience has shown that independence is the key to success in implementing SQA programs. The SQA organization should be situated in the overall organization so that it always reports to the same level that the department which it must evaluate and audit reports to. The quality organization must have the organizational status and access to top management as do the other functions. Figure 4.1 shows how this concept may be instituted within an organization and how the assurance function can use its position within an organization to achieve its goals and objectives.

An industry survey conducted by the National Security Industrial Association in 1983 and a 1995 assessment by the American Society for Quality (ASQ) reveal that the typical profile of most SQA organizations possesses the following attributes:

- The SQA organization is located within the quality assurance organization.

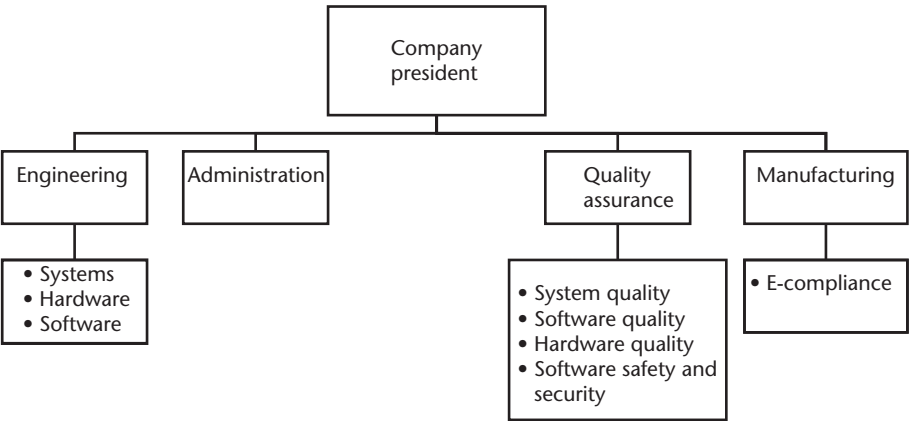


Figure 4.1 Organization structure.

- SQA is staffed with people who possess approximately 1 to 5 years of software engineering experience.
- The person in charge of the SQA organization has more than 5 years of software experience and is a middle manager within the organization.
- The career path for these individuals is into development and management.

Johanna Rothman advises the following [1]:

Product development organizations focused on developing quality products that their customers will buy need to consider the organizational structure that works best for them. It is not an organizational requirement to separate the tasks and people into functional groups. The product may require a project-focused group, focused on the product, not the organizational hierarchy.

When product development groups consider their organizational needs, they need to consider their staff, quality requirements, the process requirements, and the kinds of projects they have. In many cases, integrating the product developers and product verifiers onto a project team will have multiple positive effects in terms of project schedule, product quality, and increase in team knowledge. As the team increases its knowledge, management can trust the team more to meet schedules and deliver the promised product.

The SQA function, established to evaluate the software development effort, must possess the objective and authoritative controls required. An SQA function that reports to the development organization lacks the independence needed to get the job done properly. Moreover, the members of the development organization are by first love software designer/programmers, therefore making their quality tasks secondary in nature. An organizational structure of the type shown in Figure 4.1 allows itself to develop the SQA engineers into a position of responsibility, leadership, and independent management reporting. It is from here that the SQA engineer derives a perceived responsibility that allows him or her to translate that into getting the job done right.

A common problem of many organizations arises from appointing project-related software engineers as SQA personnel. These individuals function as senior staff members within the project organization, directing and managing the SQA effort of the project, while reporting to project management. A shortcoming of this approach is that, if SQA is relatively new and not completely defined, its implementation varies sharply from individual to individual trying to enforce it—particularly those with a loyalty divided between the project and assurance.

Experience has shown that to establish any new discipline within an organization, a central motivating force is needed. Fragmented efforts are diluted and end up being ineffective. In my own experience, project-related SQA activity generally lacks depth and maturity. All too often, the SQA activity functions as a workhorse of the developer performing tasks for the developer.

SQA will work effectively only if all project SQA personnel report to a single SQA manager. This organizational posture allows for specialization—that is, all personnel meeting the needs of the project as well as uniformity, and all projects meeting the same minimum acceptance criteria. Members of the SQA staff should have relatively high technical expertise and a thorough knowledge of good software

and quality assurance practices. The manner in which the staff is organized depends largely on staff size, estimated workload, and personnel skills.

4.4 Identifying Software Quality Assurance Personnel Needs

A 10-step process to identify SQA personnel needs is shown graphically in Figure 4.2. The process is presented sequentially with each step using the results of the previous step to build upon the next. What is particularly important about this pro-

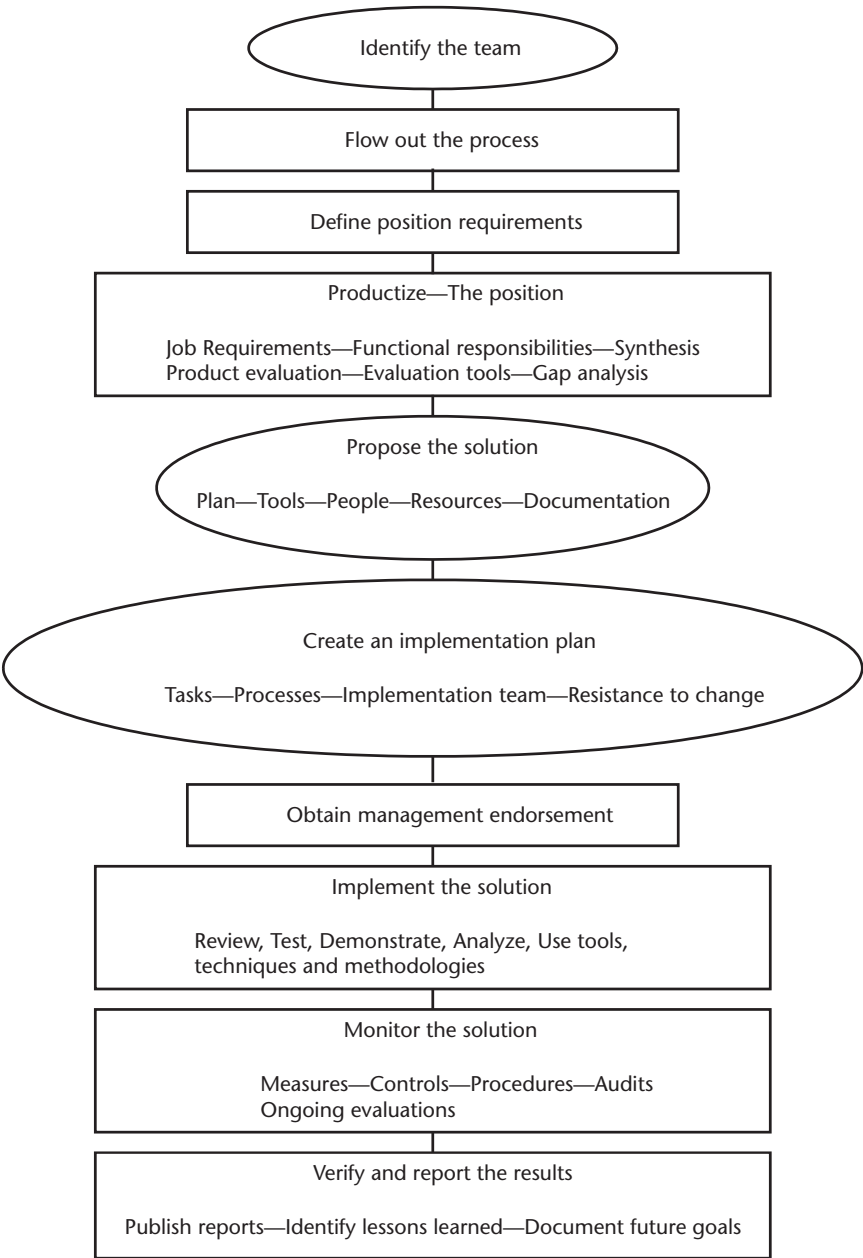


Figure 4.2 Ten-step process to identify SQA personnel based on Deming's Circle.

cess is that it is established on a Total Quality Management (TQM) foundation. Using a team from within the organization, together with continuous improvement techniques, and buy-in by the organization and those involved in developing the system, personnel requirements to make software quality assurance work is treated as an integral part of the organizations staffing activity.

The 10-step process will assure the organization that the individuals selected will require minimal effort to train and integrate into the organization. The process is based on W. Edward Deming's circle (Plan-Do-Check-Act) and on the fact that a successful computer system is achieved only if QA is built into the design, development, and release process.

Step 1: Identify the Team

When formulating this approach, first consider how you intend to answer the following questions. What is the person suppose to do in SQA? What are the required qualifications for the position? How will you know if the solution will meet the needs of the organization and the project? Who is the responsible individual(s) for monitoring progress? How will you know if the results were a success or a failure? To get the answers to these and other questions, use the team approach to develop solutions. Do this by convening past and present software development process owners. The team leader should be someone who has prior experience in computer system development combined with a strong background in software quality assurance and software/system engineering and expertise in post-development software support. This person must understand today's computer software development, testing, quality assurance, and change management methodologies. He or she must possess skills that foster teamwork, and he or she must understand the requirements for computer systems development. The technique of team dynamics depends on a working relationship that each team member brings a specialized expertise during the selection process and that these team members are capable of critiquing and providing the needed expertise to develop the solution.

Step 2: Flow Out the Process

Before beginning the selection process tasks, it is helpful to first flow out the computer system development process and the quality assurance tasks associated with the process, if not already done so. Interview developers and process owners to determine which processes and functions are under computer control and monitoring and which are not. Prioritize each task, assigning a high priority to those tasks which are critical to product quality and personnel safety. Using the process, develop a plan that outlines those processes that need to be implemented in order to assure a quality software product.

Step 3: Define Position Requirements

The next task that will be handled by the team is to develop the position requirements—that is, job descriptions. As a minimum, address the following topics. Identify and define qualifications and position tasks. Evaluate the design process and determine current results, compare current results to requirements and expectations, and identify problems to solve and opportunities for improvement. Process

capability requirements and problem identification and management are the building blocks that help define if the position requirements can be met.

Step 4: Productization

This task is associated with determining what engineering documentation will be required to support the position. Productization can be grouped into four steps, requirements analysis, functional analysis, synthesis, and solution. Although represented sequentially, these steps are interacting and interdependent. Each step not only feeds the next step but also provides new considerations for the previous step. The outcome of this step will be a job description and the tasks associated with the requirements of SQA.

Step 5: Propose the Solution

This step requires that the plans, tools, people, and resources be documented. Proposing the solution also means determining the cost and benefits associated with implementation. In proposing the solution, also consider the following:

- Describe the proposed solution;
- Create the new process flowchart;
- Identify appropriate in-process and outcome measures;
- Identify change in techniques, resources, information;
- Determine people aspects, roles, training, interactions;
- Cost and benefits;
- Estimate the costs;
- Estimate the savings;
- Identify performance improvements;
- Identify customer benefits.

Step 6: Create an Implementation Plan

In addition to those items discussed in step 5, the team should develop an implementation plan that also addresses the following:

- The SQA process to be implemented;
- Identify and plan for resistance to changes;
- Establish implementation team.

Step 7: Obtain Management Endorsement

For the solution to be successful, management buy-in is crucial. In order to secure this approval, consider the following tasks during this step:

- Prepare a presentation for management;
- Present the solution to management;
- Present the solution to those affected after obtaining management approval.

Step 8: Implement the Solution

The implementation phase begins with a briefing that explains the goals and objectives of the SQA solution. Implementation is the responsibility of the team (quality, development, and manufacturing) responsible of producing and using quality software. A team approach is recommended because SQA should be viewed as a task that involves both developer and user.

Step 9: Monitor the Solution

The responsibility of the team during this step is to assure that the solutions that have been developed are consistent throughout all phases of the development effort and that in process product and process audits, evaluations, controls, and procedures are being used to assure adherence to the plan. The solution monitoring process may consist of three activities: a capability audit, a product compliance audit of documentation, and problem identification and management.

Step 10: Verify and Report the Results

At the conclusion of predefined intervals, management should be briefed on the findings and should be given an opportunity to offer evidence or to refute any finding. Following the completion of each validation task, an exit interview shall be conducted with process owners to debrief them of the validation team's findings. The findings and recommendations should be published in a validation report and made available to process owners and cognizant personnel affected by the validation findings. Follow-up investigations of validation findings should be conducted by the validation team or their representatives to verify that corrective actions have been implemented.

4.5 Characteristics of a Good SQA Engineer

As mentioned earlier, the shortage of software professionals makes recruiting software engineers into the quality assurance profession a difficult task. Two factors appear to work against SQA:

- Developing software is far more attractive to the software engineer.
- The career path for someone in the development environment is clearly more attractive.

A salary survey from 2000 shows average full-time salaries by various IT related technical skills. According to this survey, SQA personnel received a very competitive salary [2]. The subtler prestige and glamour aspects can be addressed, so long as salary is not a major issue in the IT industry. If an organization is willing to take the time, it can probably find suitable candidates within its wage and salary guidelines. But time is money, and the longer it takes to hire the talent required, the longer it takes to bring the SQA function up to the engineering level now required to produce quality software products.

More recently the new incentive by companies to have SQA engineers certified by the ASQ as Certified Software Quality Engineers (CSQE) makes selecting

qualified SQA engineers easier. Built on topics that constitute the “Body of Knowledge for Software Quality Engineers,” CSQEs are certified on their knowledge in software quality management; software process; software metrics; measurement; and analytical methods; software inspection; testing, verification, and validation; software audits; and software configuration management. Refer to Chapter 10 in this book for more details about the certification process.

What makes a good software quality assurance engineer? Consider the following characteristics:

- The individual who seems to work the best appears to have spent approximately 3 to 5 years developing software. This individual recognizes the limited involvement he or she has had in the total developmental effort, and now wants a bigger piece of the pie. SQA clearly will provide this opportunity. This is an opportunity to get system and managerial exposure in a relatively short period of time. It is to the SQA manager’s advantage to point this out to a prospective new hire. However, finding such an individual with the necessary qualities, as listed below, can be difficult and may require a national recruiting policy.
- The experienced software engineer who has seen it all and has survived the software battles is a good candidate. This individual can truly contribute to improving software development methodologies, being inherently familiar with the existing developing techniques and capable of assuming a position of leadership in a very short time. The reader is cautioned to be aware that a lack of motivation on the part of these individuals may sometimes be a problem.
- The individual seeking to advance to management or a program manager’s position clearly is a good candidate. It is within the SQA organization that one learns how to deal with people, learns about design and development approaches and techniques, and learns how to manage and report on software development projects, which are some of the attributes one would look for when recruiting for a management position.
- A good SQA engineer must possess good communication skills. This is especially true if he or she is to be effective in performing SQA duties. As we are well aware, software engineers at times can be an unfriendly breed of professionals, very possessive of their work, and often protective of what they have designed as confidential. An SQA engineer has to be able to deal with this and win the trust and respect of software design engineers. Communication skills play a vital role in this regard; the individual should be skillful in expressing ideas both orally and in writing.
- An academic background in computer science is essential. Over the years many individuals who possess a degree in education or the liberal arts have made the switch to software. They were hired as programmers and function in this capacity. I have found that for the most part their ability in the software engineering field is limited to that of being programmers rather than being effective in design. These individuals work well under the supervision of a good software designer, but they make poor SQA engineers.

- The individual who will succeed as an SQA engineer must be willing to meet and accept new challenges and be able to carry out independent research and analysis assignments that deal with analysis of the techniques used to develop software. Such an individual must be capable of evaluating software development methodologies with an eye to improving software productivity and performance.
- The introduction of the SQA person into the CMMI®-DEV appraisals and ISO 9001:2000 audits provides a position of great importance and influence to the organization. For the first time SQA engineers are now being called upon to help establish and manage the cultural environment and monitor performance improvement goals set by management, therefore requiring the SQA engineer to sharpen the needed people development skills.

4.6 Training the Hardware QA Engineer

Training the hardware QA engineer is one method of obtaining and retaining good SQA engineers. Some hardware QA engineers of yester-year now may be in a very unmotivated position because of obsolescence in hardware engineering. Selecting those individuals willing to be retrained in software engineering is the first step. Such individuals will tend to stay within the SQA field the longest. Furthermore, they bring to the function the needed expertise to deal with designers and managers, a quality that is learned over years of on-the-job training.

A hardware QA engineer requires a number of years of training to become an effective SQA engineer. However, the return on this type of investment is, in my opinion, the surest method of developing a staff of highly qualified engineers in software quality. This approach to SQA staffing allows for a permanent core within the function, which is essential if the SQA activity is to survive as a long-range objective of the company.

The training of hardware QA engineers in software should follow one of two paths. The engineer should be encouraged to pursue a degree in computer science. Also, in-house training and learning by example should be pursued, with job assignments utilizing newly learned skills. Today's highly technological advances in the computer engineering field mandates that these individuals obtain the required academic training before releasing them to perform SQA work.

4.7 Training the Software Engineer

The optimal approach—the one the author has found to work the best with software engineers—is the mentor approach to training in the SQA discipline. A mentoring SQA is a teacher or an advisor, someone who has worked in SQA for a number of years. This individual is charged with teaching the software engineer the principles of SQA engineering. This technique to training works well with recent college graduates. Selective training is needed if this approach is to be applied to the more experienced software engineer.

The basic principle under the mentor approach to staffing is to hire software engineers into the SQA organization and to assign them to an experienced SQA engineer. The mentor's responsibility will be to outline a program of task training and to closely monitor the new hire's work output. This mentor–new hire relationship gives the new hire access to someone who will provide guidance and leadership during the learning phase. For this approach to training to work, it is imperative that a training plan exist. An example of such a plan is outlined next.

The following steps will be taken to indoctrinate new personnel into the SQA team:

- Describe the organization surrounding the project to which the new hire is to be assigned, and explain what each department does and how it interacts with the other departments. As a minimum, the departments to be discussed should include the following:
 - System engineering;
 - Software engineering;
 - Software configuration management;
 - Data management;
 - Software integration;
 - Software quality assurance;
 - Software test.
- Indoctrinate the new hire in the use and availability of existing tools and how to utilize tools to their full potential.
- Assign as reading assignments project-related software development plans, software quality assurance plans, and software configuration management plans. The objective in these assignments is to orient the new hire in the company's software development and quality assurance process.
- Define SQA's involvement in the development process and monitor compliance by the establishment of entry and exit criteria associated with the respective development phases as outlined in Figure 4.3. The reviews of the program technical approach with SQA personnel involvement include (1) a requirements review, (2) design reviews that include preliminary and critical reviews, and (3) documentation reviews. Also, SQA personnel involvement is beneficial during the build and test phase reviews that include test readiness and test exit reviews, and prior to release and use, the conduct of physical and functional reviews.

The benefits to be derived from such a program are twofold. The new hire has easy access to someone who is capable of guiding him or her in the performance of work assignments. Most important, the new hire is able to learn first-hand from someone who has been through the process and knows all of its ups and downs. The organization must be willing to devote a minimum of one calendar year to such a training program before the individual can be utilized effectively as a junior contributor within the organization.

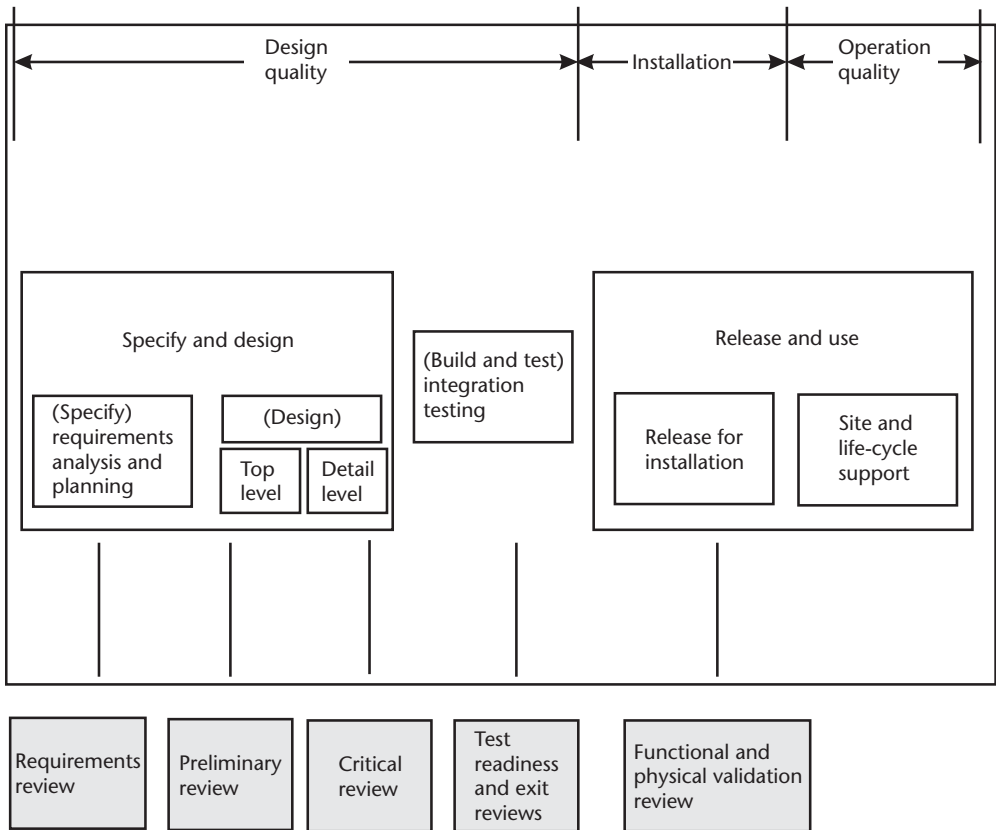


Figure 4.3 Software development with in-process SQA reviews.

4.8 Rotating Software Engineers

Rotating software engineers through the SQA function is an approach that brings to the software QA function bright and capable software professionals. They should be expected to serve a minimum of at least one year within the SQA function. But the following problems will have to be worked out before such an approach to SQA engineering will benefit the organization:

- There exists a shortage of qualified software professionals within the software development environment. A software manager would be hard pressed to release a good software engineer to SQA, if he or she is facing a manpower problem that could impact schedule completion of a project. In many instances the tendency is to release those individuals who are poor performers.
- A rotating SQA policy requires support from upper management to become effective and not to end up as a dumping ground for the bad programmer or software engineer. The choice of who makes the rotation into the SQA section should be mutually agreed to by all concerned. Motivating the software engineer to participate willingly in such a program is necessary, and the only sure means of accomplishing this is to institute a promotional policy that gives special consideration to individuals who have already served as SQA engineers.

The same policy should also hold true for one being considered for a manager's position in software engineering.

The author knows of three organizations that have tried such a program: IBM, Raytheon, and ITT. The benefits these organizations have derived from such a program have been limited to how successful they have been in retaining the services of the individuals that participated in the rotation program. Because the majority of those participating in the program were recent college graduates, all companies reported that many participants had left for assignments in other companies. For a rotation policy of this type to succeed as a means of increasing the awareness of SQA within an organization, what is needed is a core of resident SQA experts to learn from so as to continue the smooth operation of the SQA function. Furthermore, individuals selected to participate in the program should have between 3 and 5 years of industry experience, therefore reducing the possibility of their departure after the rotation assignment.

4.9 New College Graduates

The recent college graduate is ideal for certain specialized tasks within the SQA organization. Many SQA organizations are evolving from a labor-intensive approach to a more computerized approach to quality software. Such a transition requires developing SQA tools to perform tasks that were once manually performed, hence the transition from a labor-intensive approach to QA to an automated approach. Based on the author's experience, the recent college graduate is an excellent source of expertise to perform some of the tasks needed to orchestrate such a transition.

It is a well-known fact that the interest and professional development of recent college graduates in computer science tends to follow these broad guidelines. First, the graduate seeks out programming tasks and appears to find satisfaction in the activities associated with such a task. After a short period, from 3 to 6 months, his or her interest focuses on the challenges provided with being involved in software design. Some time later in terms of career growth, task assignments in software architecture become appealing. Employing a recent college graduate to perform SQA tasks at the onset has been proved to be a poor management decision, and the organization runs the risk of losing that employee because of a lack of interest in the work assigned.

The procedure that appears to work best is to combine programming tasks with SQA tasks. Obviously the mixture of programming and SQA tasks must be tailored to the needs of both the organization and the individual. During the first 6 months of a college graduate's employment, a 60/40 ratio of programming tasks to SQA tasks seems to work well. The benefits to be derived from such a mixture are many, but most of all the recent college hire's perception is that of performing a constructive task. He or she is also able to observe the benefits of these efforts, while the benefits to be derived from purely SQA tasks are more subjective and therefore a demotivator.

Orienting the new hire to the SQA methodology employed by the organization is an important training procedure that must not be ignored. Typically the SQA training process takes somewhere between 1 and 2 years before such an individual should be allowed to make independent SQA decisions without supervision. This orientation should involve exposure to overall company policies and procedures as well as the specific software tools and techniques employed by the organization to develop software.

4.10 SQA Employment Requisitions

Recruiting, hiring, and training software engineers in quality assurance can be very expensive and time-consuming. It would be wise for the SQA organization to define and set priorities for key positions in the form of job descriptions and responsibilities (see Appendix 4A). Before proceeding, the organization should consider these issues:

- Is the organization located in an area that can provide a local pool of software quality people? Should recruiting policy be national, regional, or local, considering that the degree of staff turn-around is directly related to where the new hire comes from? Can contract employees help and how best can they be utilized? Can the company promote from within and train individuals to fill SQA openings?
- What tasks should be assigned to a new SQA organization?

Whether SQA personnel are acquired from within or from outside of the company, care must be given to distinguishing between software professionals and paraprofessionals. Job descriptions for these individuals should be documented to inform the placement office of the specific tasks they will be called on to perform and the backgrounds needed to sustain these tasks. Furthermore, the careful allocation of tasks between professionals and paraprofessionals will determine the attrition rate the organization will experience. Typical professional job titles within the SQA function are:

- Software quality assurance manager;
- Engineer software quality assurance;
- Software reliability engineer;
- Software configuration management specialist;
- Software safety engineer.

The SQA manager is typically responsible for supervising the operation of the SQA section through planning and directing the utilization of personnel. This position in the organization also requires that counseling and guidance be given to company management in matters pertaining to the quality and reliability of software products developed or purchased. From a global viewpoint, the SQA manager must set the framework that will dictate the use of a software development methodology that lends itself to quality software. The engineering reliability and configuration

staff supporting this effort will provide the technical expertise necessary to assure that the objectives of the QA effort are achieved and maintained. Specific duties of the SQA function should include, but not be limited to, the following:

- Provide SQA support and improve upon the existing SQA system;
- Develop SQA tools that sense software problems during the design, development, and life-cycle phases;
- Keep management aware of the quality status of software development projects during the design, development, and life-cycle phases;
- Monitor the continuing needs and requirements of the SQA program and implement them;
- Participate in software design reviews, testing, configuration control, problem reporting and resolution, and change control;
- Provide inputs to technical and cost proposals relative to the company's participation in computer software quality;
- Audit, monitor, evaluate, and report on subcontractor software development efforts.

Many of the tasks within the SQA function can be performed by individuals who are paraprofessionals. It would be to the benefit of the organization to use these individuals to perform those tasks. This category may include the following positions:

- Software librarian aide;
- Senior software librarian;
- Software quality assurance engineering assistant;
- Software quality engineering assistant;
- Software quality assurance aide.

The role of these paraprofessionals can be viewed as assisting the professionals in achieving the SQA objectives defined by management. Work assignments are, in many instances, related to performing tasks that have been defined in detail by the professionals assigned to the SQA function. The manager employing the services of such paraprofessionals should realize that these individuals, properly trained and with formal education, could in the future make excellent SQA professionals.

4.11 What to Expect from Your SQA Engineering Staff

Members of the QA staff should have a relatively high level of technical expertise and a thorough knowledge of good software quality assurance policies. The manner in which the staff is organized depends largely on staff size, estimated workload, and personnel skills. Several alternatives are suggested:

- Each SQA staff member could be specialized to perform one task for all software products.

- Each SQA staff member could perform all software QA tasks associated with a particular product.
- The SQA organization could act as a team in which all members would cooperate in performing the QA tasks.

If it is properly staffed and organized, you can expect from your SQA organization, the staff to have the ability to work independently.

If your staff and organization are to grow and meet the demands placed upon them, the individuals assigned to perform SQA tasks must possess an understanding and involvement in their assigned projects. This understanding and involvement is achieved if the system requires their independent involvement and participation. The part-time SQA engineer is ineffective and does little to improve the quality and reliability of the software product. Productivity with minimum supervision is achieved only if the policies and procedures in place lend themselves to a team approach to quality assurance. Productivity with supervision does not permit individuals the freedom to develop into professionals who function independently.

Other qualities that one could expect from an SQA engineering staff include the following:

- *Ability to devise new and improved methods to perform SQA tasks.* The discipline of SQA is relatively new, and the whole process is being transformed from an approach that is labor-intensive to one that can be automated. The SQA engineer must be expected to develop the necessary tools, techniques, and methodologies to accomplish tasks that have been assigned.
- *Good judgment and objectivity in approaching problems.* It is imperative that the SQA engineer be able to apply good judgment and objectivity when dealing with other members of the software development team. These attributes are important, because the SQA engineer must have the support of the software development team to function effectively. If an SQA engineer alienates himself or herself from the development team, the SQA function is no longer contributing to the team's effort.
- *Communication skills for a better understanding.* The SQA engineer must possess good communication skills. Skill in expressing ideas both orally and in writing are crucial, for example, when communicating SQA review and audit findings to the software development team, or making presentations to upper management. Since most findings are of a negative nature, challenge provided by the developing environment to these findings will require considerable skill to get the message across. Furthermore, SQA is frequently called upon to present its needs and requirements to the developers. Good communication can make this job easier.
- *Technical competence and knowledge of the project are imperative.* An SQA staff (or member or portion of the staff) not knowledgeable about software development cannot support the objectives of the SQA organization. Moreover, these individuals cannot provide the expertise needed to perform the software QA tasks that the position will demand. Such individuals will therefore not be able to complete assignments rapidly or at all without compromising standards of quality.

4.12 Developing Career Paths

A software quality assurance organization without engineering-defined career paths will not survive the tests of time and effectiveness. It is essential that SQA engineers have the opportunity to ascend the corporate ladder. One of the disadvantages of some corporate organizational structures is that many of the SQA organizations exist within a hardware matrix organization, which limits the career paths for the software professionals within the product assurance organization. It is imperative that the organization recognize this critical shortfall and take steps to remedy it in order to permit the SQA function to develop into an essential and capable factor in software development.

What should be done? Obviously, if the organization is to survive, career paths from SQA to other disciplines within the organization must exist, such as SQA engineers becoming lead software engineers or SQA management moving into software development management. Specifically, a dual ladder system must exist: this allows highly competent technical employees to continue their career growth without assuming management responsibilities. It also allows management-oriented engineers to climb organizational ladders and assume management responsibilities. (In fact, the editor of this book moved from manager of software quality engineering to manager of software development engineering. Another example, at the same firm, occurred when an experienced software quality engineer transferred into software systems engineering.) This parallel structure bridges the gap between engineering and management. The organization's main goal, however, for such an approach to staff development, must be to allow engineers to progress up the ladder without becoming managers, if they desire not to.

4.13 Recommendations

Remember that the personnel requirements that make SQA work, discussed earlier, are:

- Approximately 3 to 5 years developing software;
- Experienced software engineer who has seen it all and has survived the software battles;
- Individual seeking to advance to management or a program manager's position;
- Good communication skills;
- Computer science academic background;
- Willingness to meet and accept new challenges.

The organizational environment plays a decisive role in how successful the software QA function will be. Success can be measured only in terms of a team of dedicated individuals contributing in a supportive posture to a project. To give this success its best chance, the following recommendations are offered:

- The salaries of the SQA engineers should be generally competitive and specifically in line with those of software development engineers.
- Project-related SQA functions are dysfunctional and present too many problems. A central, independent SQA function driving all projects is a more effective method to achieve SQA goals.
- A rotating SQA policy should be used as a long-range plan and only after a core of experienced SQA individuals already exists. The rotation program is not a recommended approach to starting up an SQA function.
- Responsibilities of SQA must be clearly defined and firmly supported by corporate management.
- The best approach to starting an SQA function is to first create a position within the corporate organization for an SQA manager, then promote or hire an individual to fill that position.
- The SQA organization should be situated in the corporate organization so that it always reports to the same level as the department, which it must evaluate and audit.

References

- [1] Rothman, J., “Software Quality Assurance: Should It Remain a Separate Organization?” <http://www.jrothman.com/Papers/SQAseparate.html>, May 1996, accessed December 2006.
- [2] computerjob.com, “Average Income by Skill,” <http://www.computerjobs.com/salary2000/index.asp?display=skills&emptytype=perm>, December 2006.

Selected Bibliography

- Arthur, L. J., *Measuring Programmer Productivity and Software Quality*, New York: John Wiley & Sons, 1984, pp. 12–35.
- Mendis, K. S., “A Software Quality Assurance Program for the 80s,” *ASQC Technical Conference Transactions*, 1980, pp. 379–388.
- Mendis, K. S., “Software Quality Assurance Staffing Problems,” *ASQC Technical Conference Transactions*, 1983, pp. 108–112.
- Ryan, J. R., “Software Product Quality Assurance,” *Proceedings of the National Computer Conference*, 1981, pp. 393–398.

Appendix 4A Typical Software Quality–Related Job Descriptions

Software Quality Assurance Manager

Experience required: 8 years of software related experience, 3 years in SQA, 1 year management experience.

Education required: B.S., computer science, information technology, or related technical discipline; M.B.A or M.S. in software engineering highly desirable.

Duties: Manage the SQA organization. Provide personnel to support the projects that require SQA activities. Do strategic planning for the SQA organization. Interview and hire SQA personnel. Inform upper management of the status of SQA and its activities across the projects supported. Monitor the SQA portion of proposals and estimates. Provide management interface with software engineering and software process organizations. (Refer to Sections 4.5 and 4.10.)

Engineer Software Quality Assurance

Experience required: 4 years of software related experience, 1 year in SQA.

Education required: B.S., computer science, information technology, or related technical discipline.

Duties: Perform SQA activities on the projects. Participate in software design reviews, testing, configuration control, problem reporting and resolution, and change control. Audit, monitor, evaluate and report on the software subcontractor activities. Assist in the Interviewing of SQA personnel. Produce write-ups and estimates for the SQA portion of proposals. Interface with software engineering, software configuration management and the software process organizations. (Refer to Sections 4.5 and 4.10.)

Software Reliability Engineer

Experience required: 4 years of software related experience, 1 year in SQA or reliability engineering.

Education required: B.S., computer science, statistics, or related technical discipline.

Duties: Perform the reliability calculations for the software projects that require them. Utilize the software reliability tools available on the PC to perform the calculations. Advise the other software quality engineers on what the meaning of and results of the software reliability calculations are for the project.

Software Configuration Management Specialist

Experience required: 4 years of software related experience, 1 year in SQA or software configuration management.

Education required: B.S., computer science, software engineering, or related technical discipline.

Duties: Perform the software configuration management functions for the project. This includes software identification, configuration control, and configuration status accounting and configuration audits. Coordinate these activities with

software development and SQA. Review subcontractor's software configuration management activities. Orient the software related personnel on projects as to the software configuration management requirements. Evaluate and support software configuration management tools for the project.

Software Safety Engineer

Experience required: 4 years of software related experience, 1 year in SQA, software safety, or human factors.

Education required: B.S., computer science, software engineering, or related technical discipline.

Duties: Perform the software safety functions for the project. This includes the evaluation of human factor, human-machine interface, and life critical functions of the software. Coordinate these activities with software development and SQA. Review subcontractor's software safety activities. Evaluate and support software safety tools for the project.

Software Librarian Aide

Experience required: None.

Education required: AA degree in computer related field, or computer technical school diploma, or high school diploma with proven competency in PC software.

Duties: Assist the software configuration management person on the project in performing the software library duties. These duties include both the hard copy library management and the electronic library control. Handling the releases and baselines of the software and documents are an integral part of these function. Keeping access control is a critical function to be performed.

Senior Software Librarian

Experience required: 2 years as a software librarian aide.

Education required: AA degree in computer related field, or computer technical school diploma, or high school diploma with proven competency in PC software.

Duties: Handle for the project the software library duties. These duties include both the hard copy library management and the electronic library control. Handling the releases and baselines of the software and documents are an integral part of this function. Keeping access control is a critical function to be performed.

Software Quality Assurance Engineering Assistant

Experience required: 1 year as a software quality engineering assistant.

Education required: AA degree in computer related field, or computer technical school diploma, or high school diploma with proven competency in PC software.

Duties: Handle administrative activities for the SQA engineer on the project. Place information into the SQA tool used for the project to report deficiencies. Be the interface to the SQA tools in use on the project. Where necessary keep the SQA hard copy Project Book on SQA findings. Assist in test witnessing. Fill out SQA evaluation reports where appropriate. Interface with software configuration management as necessary.

Software Quality Engineering Assistant

Experience required: 1 year as an SQA aide.

Education required: AA degree in computer related field, or computer technical school diploma, or high school diploma with proven competency in PC software.

Duties: Handle administrative activities for the SQA engineer on the project. Place information into the SQA tool used for the project to report deficiencies. Be the interface to the SQA tools in use on the project. Where necessary keep the SQA hard copy Project Book on SQA findings. Assist in test witnessing. Fill out SQA evaluation reports where appropriate. Interface with software configuration management as necessary.

Software Quality Assurance Aide

Experience required: None.

Education required: AA degree in computer related field, or computer technical school diploma, or high school diploma with proven competency in PC software.

Duties: Handle administrative activities for the SQA engineer on the project. Place information into the SQA tool used for the project to report deficiencies. Be the interface to the SQA tools in use on the project. Where necessary keep the SQA hard copy Project Book on SQA findings.