The Role of the Quality Group in Software Development

Douglas Hoffman
Software Quality Methods, LLC.
124646 Heather Heights Place
Saratoga, CA 95070
(408) 741-4830

Abstract

This paper describes the role of the quality organization in software development as observed in dozens of commercial organizations. It views the different charters and purposes the quality groups have. The potential benefits and drawbacks for various charters are presented, along with the organizational structure and typical activities in each. The charter for the quality group changes over time, and observations of progressions in organizations are made. The various possible organizations, charters, and roles are described and related briefly to quality systems described in both the SEI Capability Maturity Model and ISO 9000 Standards (ISO 9001 and ISO 9000-3). The impact on product quality of the different types of development process, and possible roles for the quality group are also covered.

About the Author

Mr. Douglas Hoffman has been in the software engineering and quality assurance fields for over 20 years. He is currently an independent consultant with Software Quality Methods, and specializes in identifying the appropriate development processes and tools for software quality based upon organizational requirements. Currently, he is Chairman of the Santa Clara Valley Software Quality Association (SSQA), a Task Group of the American Society for Quality Control (ASQC), and Program Chairman for the Third International Conference for Software Quality. He is also active in the local section of the ASQC and the ISO 9000 Task Group, and is registered as an ISO 9000 Provisional Auditor. He received his MBA from Santa Clara University, and his MS in Electrical Engineering and BA in Computer Science from UC Santa Barbara.

The Role of the Quality Group in Software Development

I. Introduction

The purpose, or mission, of software quality organizations ranges from testing products to providing information and expertise about the product and development process. The group may also provide knowledge and training on product testing, process creation and management, toolsets, and metrics. Various tasks for the software quality group are covered, and the order in which they typically appear as the organizations grow and mature is outlined. This provides a foundation for understanding the value of the quality assurance organization and the contribution they make to the product and process quality.

The roles of software quality assurance can be described with the tasks they undertake. The roles range from acting as an extension of development for debugging software products, to development process definition and control. Verification and validation, acceptance testing, measurement and metrics, and process consulting are also roles that software quality groups sometimes assume. The various charters that the organization may assume are described, and the impact on quality is addressed for each charter.

As the organizations grow and change, the needs and roles also change. Depending on the type of product and organization itself, the life cycles may differ and the tasks done by the quality organization evolve. The evolution takes familiar tracks, following patterns based upon the maturity of the organization and other factors. The SEI Maturity Model and other standards are relevant in understanding the importance and roles for the quality group.

II. Quality Group Purposes and Roles

Figure 1 shows the basic purposes, roles, and activities established for software quality groups. Although an organization may fit the description well at one time, it is likely to change as the organization evolves and matures. It is also likely that any given quality group has characteristics of many of the organizations. Generally, however, there is one primary or predominant theme in the group.

When the goal of the organization is to test products, the group usually acts as an extension of the development organization in performing a debugging function. This is shown as the Type 1 row in Figure 1. The group's primary activities are to develop and run tests, with the primary emphasis on reporting defects. The majority of time is generally spent running the tests and reporting the results. This role gives the most tangible and obvious results, yet is the least leveraged and thus the least productive. Most organizations encountered in industry begin with this goal, and often even well established groups are relegated to being only testers.

This first type of group is doing quality control; acting as a filter for products to ensure that only products with acceptable levels of errors are delivered. The activities are reactive, responding to past

events, and the only real power in the organization has is in stopping products from being delivered. In the most extreme case, the group is acting as a debugging function for development, with little or no effect on the cause of quality in the software. The functioning of the group as quality assurance doesn't begin until the group is operating as a type 3, 4, or 5.

Another, more subtle version of the testing goal is to measure the quality of products. this is shown as Type 2 in the table. This differs from simply testing in the group's arms-length relationship with the testing and the data, and emphasis within the organization on achievement of specific levels of quality. The quality group often does not do the primary testing, but rather oversees and reports on the results of other groups' activities. Although they may act in a more advisory capacity, the focus in Type 2 is still on testing and measuring the quality through product defects. Both forms of organizations have been observed, performing arms-length tests and recording the results from other groups' testing.

Often the Type 2 organization is gathering enough data on the product that good quantitative decisions and predictions can be made. This sometimes results in the group functioning as a "quality hurdle"; a quality level to be achieved before the product can be released. Depending on how the acceptable quality levels are determined, and the way the group interacts with other organizations, this can be a large step forward from a Type 1 group.

A different focus for the quality group occurs when they concentrate more on the organization's processes, rather than the products. This often occurs when the organization focuses on metrics programs and expands beyond the nebulous "defects happen" theory into an understanding that "defects are built in". This is the Type 3 organization. The role of the quality group becomes more general, that of information brokers seeking insights from whatever data is obtainable. Metrics programs change the role of the quality group to information engineers; applying the data to understand and improve the organization. Few quality organizations have progressed to using sophisticated metrics, and even fewer have been successful in applying them to predict program and process behavior.

Armed with the information provided by the metrics, the quality group can begin to assure that the software is of good quality. Even without planning the processes, the knowledge of what works and what doesn't is powerful enough to change what is done. This information also is the foundation of quality improvement programs. The group is not purely reactive at this level.

Type	**Purposes**	**Activities**	**Roles**
1	Test Products	Test development, test execution	Testers; extension of development
2	Measure Products	Test oversight, reporting results	Measurers; Quality hurdle
3	Measure Processes	Metrics	Information Engineers
4	Define Processes	Process and Risk management	Quality and Process Engineers
5	Guidance Resource	Quality Reference	Quality Engineers

Figure 1: Types of Quality Groups

When process definition is the goal, the quality organization performs more of a role of quality and process engineering. They assess risks and design processes to reduce the risks and increase the quality. This becomes more of a management approach than a technical one. Organizations often form a separate group for process engineering, but this is different from the quality group concentrating on ensuring that the process is properly defined, measured, and controlled.

As a guidance resource, the quality group provides expertise and reference information so others in the organization can effectively do their jobs and improve quality. This reference information includes how-to as well as what-to guides, and measures of quality and their meanings. This is more than arms-length measurement and reporting, but provides guidance, training, and assistance in issues relating to quality throughout the organization.

III. Software Development Life Cycles

Several Software Development Life Cycles (SDLC) are described in Figure 2, and situations where they are most applicable and effective are shown. The SDLCs used in organizations include those shown and variations created to address specific organizations' needs. The appropriateness for the life cycles is shown in relation to the stability and understanding of product requirements. It is these requirements and their characteristics that appear to be the underlying driver of life cycle selection.

For example, the classic waterfall approach to software development is most appropriate when the requirements can be fully known before beginning development, and they don't change substantially during the product development. If they change substantially, a spiral approach is more likely to fit the organization's needs. Variations are usually developed for customization to the unique organizational requirements. Variations are required for situations such as when the requirements are initially known but subject to change.

Product Requirements	Life Cycle
Known, unchanging	Waterfall
Unknown, changing	Prototyping
Unknown, unchanging	Spiral
Known, unchanging	Decomposition/Integration
Known, provable	Cleanroom
Unknown	Fourth Generation Techniques

Figure 2: Most Appropriate Software Development Life Cycles Based Upon Product Requirements

The role of the quality group is independent of the life cycle involved. The role is primarily a function of the maturity of the organization and the appropriateness of the life cycle model. The specific activities differ on a technical level, but the various possible roles remain the same, and the progression and evolution usually occur in the same ways. Problems occur in an organization when the wrong life cycle is chosen for a project, and in these instances the quality of the product is generally low, and the role of the quality group is relegated to testing.

The incorrect choice of life cycle causes wasted effort and frustration among the project members. It also results in lower quality for the product. For example, changing requirements frequently when using a waterfall approach causes respecification, redesign, recoding, and retesting. The fixes to accommodate the changes cause the quality to drop and the code to age quickly. In these situations, it is also impossible to properly plan for the testing. This means continual rewriting of test cases and rerunning of the test suites. The quality group does not have time to properly plan and create stable measures of quality. Similar problems occur whenever a poor fit is chosen for the life cycle approach.

IV. Organization Growth and Maturity

Organization maturity is not an indication of the age of the group. It can be defined as a measure of the formality of the processes used by software development. For healthy organizations, the role slowly evolves through the maturity levels. The level of maturity of the organization roughly correlates with the role of the software quality group.

The relationship of the software quality assurance group role to SEI's Capability Maturity Model (CMM) is shown in Figure 3. For each level of maturity, the roles for the software quality group is shown. The majority of organizations are at level 1 or 2, with a few delving into metrics and process and risk management.

The relationship between the organization maturity and the role of the software quality group is worth understanding. Although they seem closely correlated, I believe there is a chicken-and-egg problem in trying to determine which causes which. The more mature the group, the more the responsibility for quality is distributed throughout the organization.

The role of the quality group evolves from testing to process definition and control as the organization evolves. Trying to control and optimize the development process in an organization at the Initial Level does not make sense. The process at the Initial level is poorly defined and inconsistent. Control and optimization in this situation is not definable; there is no way to know a process is controlled if one does not know what the process is. On the other hand, paying no attention to process does not make sense either. A balance must be struck between definition and control of process now and improvement of the process for the future.

The most effective role for the quality group is the one that best supports the organization today, while preparing to improve it in the near term. Without necessarily advocating any particular methodology for organization development, the quality group must understand and support some SDLC

model - what ever model the organization agrees fits its needs. When the appropriate SDLC is applied, development goes smoothly and good quality practices can be more easily applied. When a poor choice is made of the SDLC (or no choice is made), the effort by all members of the organization is increased and the quality decreased. This forces the quality group to act as testers and gate keepers to avoid release of seriously faulty products.

SEI Maturity Level	Role of Software Quality Assurance
1- Initial	Testing
2- Repeatable	Quality hurdle
3- Defined	Oversight, Metrics
4- Managed	Process and Risk management
5- Optimizing	Reference, Oversight

Figure 3: Organization Maturity and SQA Roles

V. Models for the Quality System

Other models and standards for quality organizations and organizational evolution are also applicable. ISO 9000 is a framework for a quality system, rather than a process methodology or prescription for the software quality organization's charter or function. The relationship of the quality system to the business system and development methodology is graphically described in Figure 4. The quality organization charter in this model can be as controller of the process or the product.

The quality framework described by ISO 9000 is one which sits on top of, and becomes part of the normal business systems. It is directed at the production process rather than the product directly. In the case of software development, the business system is the development methodology and support mechanisms. The quality system is the set of monitors and controls applied to the business system to ensure that it is working, rather than testing of the product itself to see if it works.

Neither SEI's Capability Maturity Model nor ISO 9000 describe in detail what the right process is, who should do what, or how things should be done. ISO does not begin to prescribe these things, but rather provides rules for knowing if a given quality system might qualify under its guidelines. Neither system addresses product quality directly.

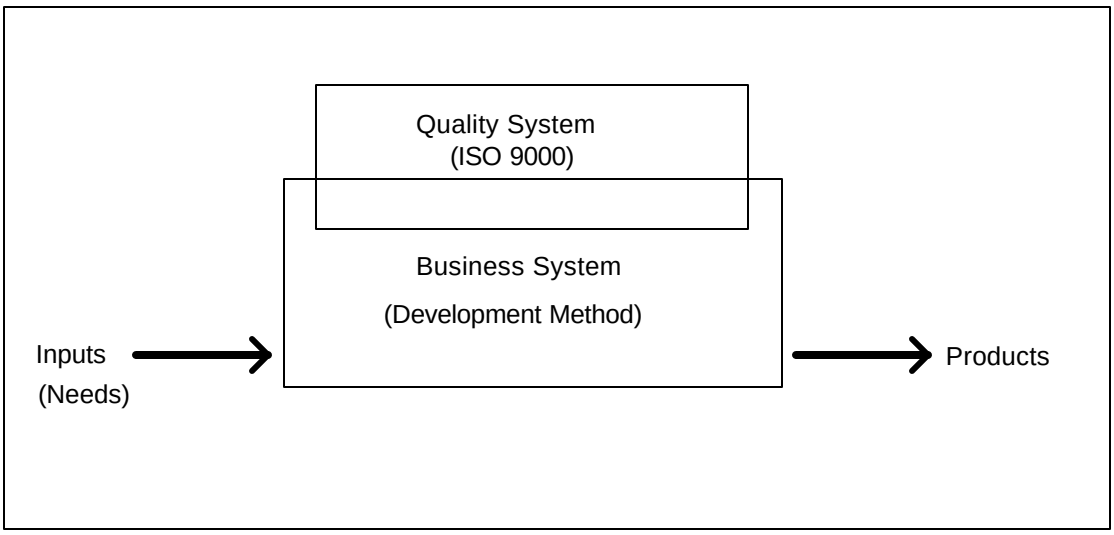

Figure 4: Quality System and Business System Relationship

The models don't prescribe specific techniques or methods because each organization's situation is so different that there is no single solution. In order for generalized models for development organizations or quality systems to be useful, they must be applicable in many situations. If these models prescribed specific methods and techniques, they would not be applicable to the majority of organizations since they have such different needs and characteristics. Organizational requirements are unique, and are based upon such things as the product characteristics, technical development environment, customer expectations, and organizational politics.

The models are also process oriented, not product oriented. They focus on the processes organizations should have, not the products. They do not directly address testing of products or product quality. They describe how the process must be defined, controlled, and improved. Only by controlling the process can the product quality be predicted and controlled.

VI. Continuous Improvement

There are a number of factors that must be in place before a continuous improvement program can succeed. A system must be defined and in place to be improved. Improvement is a relative concept that assumes some knowledge of the quality before and after some event, therefore measurement is required. Continuous improvement requires continuous measurement and comparison to ensure that the processes are under control and the changes implemented do indeed improve things. As described in the CMM, software development organizations usually evolve from less structured to more formally functioning. These changes take months or years, and generally must be planned and nurtured to be successful.

At the lowest level of structure, there is not enough defined process to measure and compare. When an organization has matured beyond trial-and-error approaches to software development, it is

possible to begin to formally monitor and improve the processes. The quality group can be instrumental for both monitoring and defining the processes. The monitoring takes place both on the process directly and on the products to confirm indirectly that the processes are under control. In mature organizations, measurement of product quality is necessary and routine to ensure that the processes are under control and to monitor the outcomes of changes implemented for improvement.

VII. Conclusion

The quality group's role travels full circle as the quality system takes root. When the system is beginning, or the organization is in trouble, the quality group concentrates on testing of product and acting as gate keepers. As the quality system becomes established, emphasis shifts toward process and measures. Intervention is often necessary, and the quality group may take an active role in implementing change. Once the quality management system is well established, only small amounts of attention are needed to monitor and review activities to ensure that the processes remain under control. With a well established quality system, the quality group role moves back more to arms-length measuring and reporting, but with much higher leverage and effect.

How can this be applied? First, we need to establish what the organization is doing. The role of the quality group should be set based upon the needs of the organization. These needs can be predicted by the maturity of the organization and the need to change. The appropriateness of the SDLC can be evaluated and changes made if required. Then goals for improvement of the process and evolution of the organization can be set. The quality group can play a big part in the planning and implementation through understanding of organizational development needs and techniques. Then an improvement program to attain the goals can be begun. This is the foundation of any continuous improvement program, and ultimately should be the goal of the software quality group and all of management in the organization.

References

Humphrey, W.S., *Managing the Software Process*, Addison-Wesley, 1989

Humphrey, W., *"Characterizing the Software Process: A Maturity Framework,"* IEEE Software (March), 73-79, 1988.

Quality Management and Quality Assurance Standards, ASQC, documents ANSI/ASQC Q90, Q91, Q92, Q93, and Q94-1987. (ISO 9000)