

IMPROVING SOFTWARE QUALITY

Sharon Wheeler and Sheryl Duggins
Computer Science Department
Southern Polytechnic State University
Marietta, GA 30060
sawheel@ix.netcom.com
sduggins@SPSU.edu

Abstract -- This paper examines how organizations are pursuing software quality. The utilization of Software Quality Assurance departments as a means to improving software quality is investigated. A ten step program on building an effective SQA department is presented. The research investigates whether organizations with SQA departments produce better quality software products than organizations without SQA departments.

1 Introduction

The phrase "Total Quality Management" (TQM) is used to describe the management style of organizations ranging from all types of businesses to non-profit leisure affiliations. It may be applied to individual fields, like software engineering, resulting in specialized versions of the phrase, such as "Software Quality Assurance". Thus many companies that produce software now have departments, usually called Software Quality Assurance (SQA) departments, designed to ensure that "quality" procedures are being utilized in the organization.

But does having an SQA department really mean that the software being developed is actually of higher quality? Furthermore, how does one know what quality is? Is quality in software different than quality in other fields? How do you measure quality? What do you need to know or do to build an effective SQA department?

In 1993, information systems expenses represented the third largest corporate disbursement for American businesses, and approximately 40% of the US capital investment [4]. Software development is a global business producing an estimated \$120 billion yearly [24]. As a result of global and domestic competition, software quality is a key issue for maintaining competitive and strategic advantage. In *The Decline and Fall of the American Programmer*, Edward Yourdon [23] questioned the fate of

the American programmer due to poor quality software, productivity, and cost issues. Yourdon presented results from a CASE survey which found that only 54% of U.S. companies were practicing any methods of quality control. He also estimated that 85% of all US software organizations rated lowest when measuring quality using the Capability Maturity Model.

This paper will investigate the utilization of Software Quality Assurance departments as a means to improve software quality. Furthermore, the role of software engineering in relation to SQA organizations will be investigated. The goal is to discover the effect on software quality resulting from establishing and employing effective SQA organizations. A brief summary of TQM views on improving quality from well known "quality experts" Deming, Crosby, Juran and Ishikawa is provided. The results from an internet survey created to assess how organizations are pursuing software quality are discussed. The research demonstrates that those organizations with SQA departments have a better foundation for producing superior software.

2 Previous Related Research

The concept of product quality became the focus of Japan business leaders in the early 1950s [1,15]. It was during this time that American quality experts were invited to Japan to advise them on quality. Results from Japan's implementation of the recommendations from American quality experts led to an industrial revolution that eventually left the American industry lagging behind. The 1990s has now been characterized as the quality era. However, it was during the late 1980s that American industry began to finally look to their quality experts for methods to improve quality. In the late 1980s, an NBC documentary called "If Japan Can... Why Can't We", brought national attention to the needs for quality improvements for global competition. The program focused on W. Edwards Deming's approach to improving quality. The documentary validated Deming's philosophy and led to many organizations consulting him on quality issues. As other industries focused on quality, so did the software industry. Improvements in quality were seen as a means to

Permission to make digital/hard copy of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage, the copyright notice, the title of the publication and its date appear, and notice is given that copying is by permission of ACM, Inc. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

© 1998 ACM 1-58113-030-9/98/0004 \$3.50

increase productivity, lower costs, and provide the customer with the desired product.

As American organizations move towards a total quality management approach to software development, what actually constitutes software quality remains subjective. The idea of software quality is not a new one; quality has been an issue since the software crisis of the 1960s. Over the years a number of definitions have originated from the "quality experts". These experts are identified as Kaoru Ishikawa, Joseph M. Juran, Lennart Sandholm, W. Edwards Deming, and Philip Crosby.[19,10,23]. Many of these experts have dedicated the last 30 years to consulting, researching, teaching, and publishing works on quality improvement. Although the experts' research is not specifically in the area of software product quality, their approaches can be used to improve the quality of any product.

Quality is defined as a "characteristic or attribute of something"; "property"; "a feature" "excellence"; "superiority", "degree or grade of excellence"[2]. Quality can be different for each customer. Some customers consider a quality product as one that is low in cost, has many features, or is easy to use. The quality experts disagree both on the definition of quality and how to achieve it.

3 The Quality Experts

3.1 W. E. Deming

Deming is considered by many as the father of quality. In the 1950s, Deming visited Japan to advise their business leaders on quality issues. Deming's work with Japan lead them to revolutionize quality and productivity, leaving the U. S. lagging behind. The U.S. is now beginning to apply Deming's principles to improving quality in an effort to retrieve some of the product market held by Japan.

Deming does not provide a single definition of quality. In *Out of the Crisis*[9], Deming states that quality should be aimed at the needs of the consumer and it begins with the intent by management. He also notes that quality has a predictable degree of uniformity and dependability at low cost. His prevention approach is achieved by process, analysis, control and improvement. He states that improved quality leads to lower costs, higher productivity, greater market share and more jobs. This continuous process is called the Deming Chain. He charges management with the responsibility of quality improvements which can be accomplished by adopting his 14 points and leading quality improvements by example. His approach to improving

quality focuses on statistical principles and techniques applied during all phases of development.

3.2 Joseph M. Juran

Joseph M. Juran is considered one of the pioneering authorities in the field of quality management. He was among the first to recognize that product quality does not happen by accident -- it must be planned [15]. Juran's definition of quality centers around two concepts: "fitness of use" and "a product free from deficiencies" [15, p5]. According to Juran, companies should spend time focusing on quality planning because of loss of sales, quality costs, and threats to society. Juran lists three processes (The Juran Trilogy) through which quality management is achieved: quality planning, quality control and quality improvement. Quality planning relates to identifying customers and their needs, creating a product to meet the needs, and developing a process through which the product can be produced. Quality control is concerned with maintaining current state and not letting the process worsen. Quality improvement involves improving processes thereby leading to lower costs for higher quality. To meet quality challenges, Juran prescribes the following:

- Structured annual improvements in quality;
- Massive quality-oriented training programs; and
- Senior management leadership in each company's approach to product quality.

3.3 Kaoru Ishikawa

Ishikawa emphasizes that training and education must be conducted company-wide to introduce quality assurance in software. He states that "Quality control begins with education and ends with education" [14, p123]. In *What Is Total Quality Control* [14], Ishikawa suggests: quality control training at all levels of employment, long-term quality control education, education and training within the organization and permanent education. Through company-wide quality control, all departments adhere to a defined set of quality standards provided by upper management. Ishikawa [19, p27] identifies six attributes of quality work. These attributes include company-wide quality control, quality control audits, education and training, quality circles activities, application of statistical methods, and nationwide quality control promotions and activities.

3.4 Philip Crosby

Philip Crosby is an internationally known quality expert. He began his study in quality assurance at ITT in the early 1970s. He defines quality as "conformance to requirements"[6]. Conformance to requirements implies the product does what the producer said it would do. Crosby is the author of such books as *Quality is Free*[6], *Let's Talk Quality*[8], and *Quality Without Tears: the Art of Hassle*

Free Management [7]. The theme of Crosby's message on improving quality is prevention. He focuses on doing it right the first time, zero defects, the absolutes of quality, and the six C's. Crosby identifies the six C's as: comprehension, commitment, competence, communication, correction, and continuance [22, p7]. Crosby presents the approach to quality improvement through a five stage quality management maturity grid which is used for assessment and improvement. The five stages in the maturity grid are uncertainty, awakening, enlightenment, wisdom, and certainty.

4 Specific Research in Software Development

Now that the views from the quality experts have been discussed, this paper will concentrate on research specifically directed towards software development. Many organizations are beginning to take a Total Quality Management (TQM) approach to improving software. Total quality management is a style of management in which quality and customer satisfaction are the main objectives. There are many approaches to TQM as represented by the views of the quality experts formerly discussed. Many organizations realize that improving software quality will truly bring about the Deming Chain of improvement.

In accordance with IEEE standard P729, Software Quality Assurance is defined as "a planned and systematic pattern of all actions necessary to provide adequate confidence that the item or product conforms to established technical requirements" [21, p28]. The aim of the SQA organization is to assure that the standards, procedures and policies used during software development are adequate to provide the level of confidence required for the process or product. Policies are general statements or understandings concerning development. Procedures are guidelines to accomplish tasks in a structured manner. Standards are mandatory requirements that are enforced to constrain a controlled and uniform approach to software development. Software quality assurance is an independent process, composed of planning and control.

An effective SQA program should consist of the following four components.

- Training and Planning;
- Audits, Inspections and Reviews
- Standards and Procedures; and
- Metrics.

Each of these components is briefly described below.

4.1 Training and Planning

To be effective, SQA planning must begin early in the development process. Since quality must be planned, budgeted and a part of the total software engineering process, SQA planning must begin when initial project planning begins. SQA planning involves the totality of all functions necessary to implement a pre-determined level of quality. According to Emanuel Baker and Matthew Fisher [3, p47] it involves: establishing software quality requirements; establishing and enforcing procedures for production and maintenance; and establishing and implementing procedures to evaluate the software product's quality and corresponding documentation, processes and other activities that affect quality. Some of the plans required for assuring software quality include: Audit and Review Plan, Software Verification Plan, Acceptance Plan, Documentation Plan, Configuration Management Plan, and Staff Development Plan.

SQA begins with training. According to Yourdon, in *The Decline and Fall of the American Programmer* [24, p202], approximately 75% of software companies in the U.S. have independent SQA departments. However, the problem with many of them is a lack of trained and certified personnel in software quality. Research from Ishikawa shows that software quality assurance begins with massive training.

There are specific skills the SQA staff should possess. Again, it is also important to note that Software Quality Assurance is an organizational approach. However before preceding with this discussion on training, it must be emphasized that Software Quality Assurance starts with senior level management. It cannot be built from the bottom-up (software engineers) or the middle-down (SQA departments), it must be built from the top-down [17]. Quality begins in the boardroom. With that said, there are a number of areas in which the SQA personnel must be trained to be effective. In addition to understanding the software development process, training is required in analysis, statistical modeling, and independent verification and validation (IV&V) techniques. Some areas of analysis that the SQA members should be trained in include: process, Pareto, cost-benefit, and risk analysis.

4.2 Audits, Inspections, and Reviews

Audits, inspections, and reviews are conducted at various times in the development cycle. The purpose of the software quality audit is to examine the conformance of "process to procedures" and "product to standards". There are two flavors of software quality audits: internal and external. Internal audits are used to detect and improve problems internally in the process. They are conducted by in-house auditors or teams. On the other hand, external

audits are conducted by outside auditors. There are four steps involved in an external SQA audit[22,17,20]:

- Audit planning and preparation;
- The site visit;
- Audit reporting; and
- Follow-up.

Audit planning addresses four areas: product, process, project, and organization. In the planning and preparation stage, the SQA auditor gains knowledge about the project. This is accomplished by reviewing the procedures and documents of the process being used. After the auditor receives and reviews the quality manual and process documentation, he decides if an audit should take place.

The audit report details the findings related to the status of the development process and conformance. Based on the auditor's findings, the auditor makes a recommendation to the registration board on the certifiability of the organization. If the process does not meet the standards of the registration board, the non-conformances are noted and a follow-up audit may be required.

A review is a planned event concentrating on a particular sub-product or process [5]. Reviews are components of I V&V. Reviews are defined as part of the Audits and Review plan. The purpose of a review is to evaluate a specific element to determine if that element or process is being developed according to the standards, guidelines, and policies set forth in the SQA plan.

4.3 Standards and Procedures

The purpose of standards and procedures is to bring uniformity and control to the process of developing a software product. These standards may include specification, documentation, reviews, audit, and software engineering standards. Procedures include practices to be followed when tracking and resolving software problems, rules for conducting reviews and inspections, and guidelines for updating documents into the configuration management system. In general, procedures are directions to be followed to accomplish a specific task. A number of standards must be provided in the SQA plan. These include standards for documentation, programming, and SQA.

SQA standards are concerned with those principles and procedures necessary to produce the SQA plan. There are a number of standards that were developed over the years by national and international bodies. These standards bodies include: DoD, NATO, ANSI, IEEE, and ISO9000. Each standard contains a list of elements that must be addressed by the quality system.

ISO 9000 provides three standards that can be used for software development: ISO 9001, ISO 9000-2, and ISO 9000-3. ISO 9000-2 and ISO 9000-3 are subsets of ISO 9001. ISO (International Organization of Standardization) 9000 is a organization that establishes worldwide standards for developing quality systems. Stated simply, the ISO 9000 philosophy can be characterized as "say what you do, do what you say, and prove it" [16, p48]. A detailed discussion of the ISO 9001 standard in relation to software development can be found in [13] and [20].

ISO 9000 registration is not an easy standard to fulfill. Many companies are applying for registration but Kan [16, p47] reports 60 to 70% fail the initial audit. Compliance to ISO 9000 and other standards is used by some customers to judge the competence of an organization.

4.4 Metrics

Metrics are the final component in the SQA program. Metrics are used as predictors of quality and as a means to target areas that are experiencing quality problems. Since TQM centers around continuous improvement, there must be a means to gauge the progress in both process and product. In order to use metrics successfully, the SQA personnel may require extensive training. Several categories of metrics are identified by Kan in *Metrics and Models in Software Quality Engineering* [16]. He divides software quality metrics into two classes: end-product quality and in-process quality metrics. He also discusses software maintenance metrics.

In-process quality metrics should be taken during each phase of the software development process. The quest to engineer quality into processes brought about the field known as process management. The original idea contributing to the field of process management came about through the works of W. E. Deming. According to Kan [16, pp99-102], the minimal metrics required for in-process quality management include:

- Defect density during machine testing;
- Defect arrival pattern during machine testing;
- Phase-Based defect removal pattern; and
- Defect removal effectiveness.

End product metrics are usually the only metrics the customer cares about. The categories of end-product metrics include: software reliability metrics, defect density metrics, customer problems metrics, and customer satisfaction metrics.

The components that are most critical for defining an effective SQA program have been discussed. There are a number of standards and much research that provides

additional information on building SQA organizations. Based on the previous research presented, a brief 10 step overview to get started in building an SQA department follows [5,11,12,18,13,17].

5 Getting Started: Steps to Build an Effective SQA Department

1. Top level management commitment and leadership is first and foremost followed by establishing a quality policy. The quality policy describes the organizational goals and objectives relating to quality. The quality policy also defines how the organization is going to accomplish the quality goals and objectives. If this step is not accomplished, the additional steps are not required, since an effective SQA department cannot be built without commitment and leadership from the top.
2. Hire, assign or train a Quality manager. The Quality manager will be responsible for getting the initial guidelines, training and procedures for establishing the SQA department in order. In essence, there needs to be someone in charge of managing the quality effort.
3. Research past or current literature on Software Quality Assurance. This should be followed by educating all about the need for and benefits from software quality assurance. Increase awareness.
4. Adopt a life-cycle model that is consistent with your business and software development environment. Evaluate your current process for developing software. Take the SEI Quick Assessment test.
5. Decide on a standard to use for your SQA plan. A number of standards were noted in this section. Each standard provides a list of elements that must exist in the quality system. Use those elements as a guide for building the SQA plan.
6. Develop a Quality manual. Determine the elements you want to address in your SQA program. At a minimum the following should be addressed: Project and Process management; Reviews, inspections, and audits; Standards and procedures; Configuration management; and Metrics.
7. Define roles and responsibilities for all involved in SQA. Utilize negotiation.
8. Tailor the plan to your organization.
9. Test the plan. Institute reviews, inspections, and audits. Document everything. Implement the configuration management system. Utilize metrics.

10. Repeat for the next project, using knowledge gained from previous projects to improve processes, product, and your SQA program.

6 Research Methodology

Two instruments were used to gather data for this research. The instruments were an internet survey and an interview. The target audience was anyone involved in the development of computer software.

The purpose of the interview was to gather data on how the use of standards improves the software development process. The interviewee was an ISO-9000 auditor. She was contacted via an instructor at my university. A list of 20 open-ended questions was prepared for the interview. The interviewee granted permission to allow the interview to be taped.

The Software Quality Assurance Survey was developed to determine the methods organizations were using to improve software quality. The survey considered 6; categories of questions. The categories focused on:

- Software Quality Assurance Organizations;
- People Issues
- Software Processes;
- Software Metrics;
- CASE Tools; and
- Reusable Components.

Each category of questions was developed to determine software quality and/or software engineering techniques being used by organizations. Questions in the section on software quality were created to determine what type, if any, software quality department was in an organization. Specific questions were asked to assess the qualifications of those working in the software quality arena. These questions were designed to determine the level of formal training and quality analysis techniques utilized. The questionnaire also covered demographics including current position, type of industry, and years of experience in software development. The questionnaire was composed of 31 questions. There were 29 closed-ended and 2 open-ended questions. The open-ended questions asked the respondents to describe specific personal or company approaches used to improve software quality. The closed-ended questions were multiple-choice and yes-no questions.

The Software Quality Assurance survey was created as a web page. The web page was posted on the internet. In order to obtain respondents to the survey, the web site had to be publicized. The only requirement for respondents was to be involved in the software development process. The objective of using the internet was to ensure a cross-section of world-wide responses. Announcement of the

survey was composed of various means. After completing internet research on publicizing web pages, the web page was registered with a number of national and international search engines including: Alta Vista, Yahoo, Web Crawler, HotBot, Lycos, InfoSeek, Excite, and Yellow Pages. The web site was also posted with many computer and software related user groups, and an announcement of the web site was sent to contacts at various Atlanta high-tech associations.

After completing the internet survey, the respondents utilized the send survey button on the survey to e-mail their responses in the form of an ASCII file. The survey results were collected for data analysis. The respondents were provided a section for comments on the survey. The survey was posted for approximately 3 months.

A software tool was written to read the text-based user responses and help analyze the data. The survey results were categorized into two groups. The categorization was based on the response to the following survey question: *"Does your organization have an SQA department that enforces standards and procedures that define how quality is achieved?"* Based on the response to this question, the SQA and NON SQA groups were defined to evaluate the survey. The data was further analyzed using chi-square analysis. The chi-square analysis tests assume a critical probability of $p=0.05$.

7 Discussion/Interpretation of Results

As discussed above, a software quality assurance survey was utilized to gather information on the SQA function in today's organizations. This section contains the summarized results of the study. There were eighty-three responses to the survey questionnaire. Three of the responses were duplicates and only the last survey received from the respondent was considered valid. The number of usable responses was eighty.

In order to assess the qualifications of the respondents to the survey, the questionnaire examined current position, years of experience and education. The pie-chart in Figure 1 shows the respondents' job classifications. The current positions of the eighty respondents are listed below.

- Software Engineers 21
- QA Engineers/Auditors 14
- Programmer Analysts 12
- Project Leaders 7
- Managers, Software Develop. 7
- Managers QA 5
- Software Testers 5
- OO/GUI Developers 4

- Systems Analysts 3
- Others 2

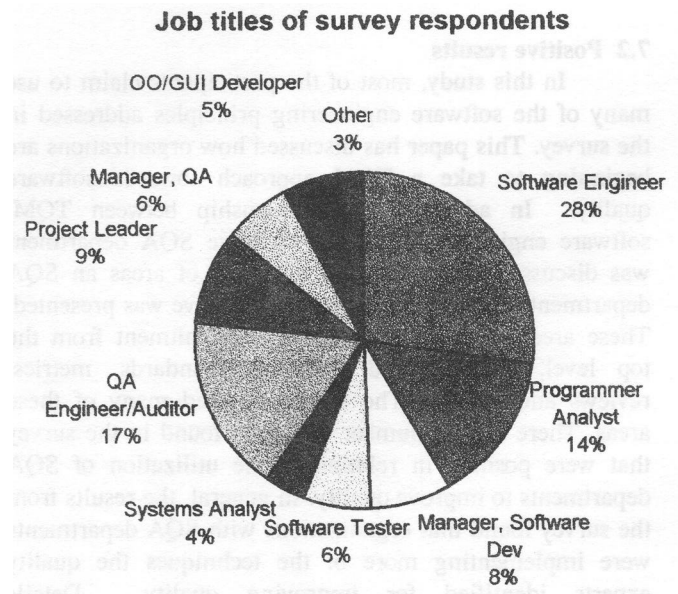


Figure 1 Classification of survey respondent job title

The profile of the average participant in this survey sample had more than 5 years of experience as a software developer. The respondent was involved in software development as a software engineer, programmer analyst or quality assurance person and had a degree in Computer Science, Engineering, Physics or Mathematics.

The following question was asked on the survey

Does your organization contain an SQA department that establishes and enforces organizational standards and procedures that define how quality is achieved?" Based on the response to this question, two groups were formed. The remaining results will be presented based on the two groups. These groups are identified as SQA and NON SQA. The group breakdown is shown below.

- SQA 47 respondents identified SQA departments in their organization
- NONSQA 33 respondents identified no SQA department in their organization

7.1 Results of the Survey

The objective of this research was two-fold. One of the objectives was to determine how organizations were pursuing software quality. The research goal was to assess the use of software quality assurance organizations and software engineering practices in improving software quality. The expected outcome was that organizations with SQA departments produce better quality software than organizations without SQA departments. The other

objective of the research was to determine other methods individuals and organizations were using to improve software quality. The survey results are categorized as: Positive, Negative, and Encouraging.

7.2 Positive results

In this study, most of the participants claim to use many of the software engineering principles addressed in the survey. This paper has discussed how organizations are beginning to take a TQM approach towards software quality. In addition, the relationship between TQM, software engineering, and the effective SQA department was discussed. Furthermore, a variety of areas an SQA department must master to become effective was presented. These areas included management commitment from the top level, training and planning, standards, metrics, reviews and audits. The survey covered many of these areas. There were a number of results found in the survey that were positive in relation to the utilization of SQA departments to improve quality. In general, the results from the survey found that organizations with SQA departments were implementing more of the techniques the quality experts identified for improving quality. Details concerning these techniques are presented below.

- One of the elements discussed in terms of the effective SQA organization was the use of standards and procedures. As previously discussed, the purpose of standards and procedures is to bring uniformity and control to the process of software development. The survey results found that in general, the organizations with SQA departments were more often pursuing standards. Assuming a critical probability of $p=0.05$, chi-square analysis tests support the findings that there is a dependence between the existence of an SQA department and the pursuit of ISO-9000 ($X^2=4.69$, $p=0.031$, $n=80$) and SEI-CMM registration ($X^2=6.28$, $p=0.009$, $n=80$).
- Another positive result from the utilization of SQA departments was the focus on defect prevention, not detection. Specifically, the SQA organization used reviews as their dominant V&V technique ($X^2=9.08$, $p=0.003$, $n=80$). Conversely, the NON SQA organization continued to depend on testing. Recall that the purpose of the review is to ensure that a specific element is being developed according to standards, guidelines or policies.
- The TQM approach was another positive finding in the survey. Approximately 68% of the SQA organizations used customer satisfaction as an indicator of quality. This compares to 58% for the NON SQA group. Also, the SQA group focused more on customer related

problems as a metric of product quality. The results showed that approximately 80% of the SQA groups measured customer related problems compared to 60% for the NON SQA group. These elements are key to the TQM style of management. Since software quality has become key in maintaining competitive advantage, higher quality products will be produced if the customer demands it.

- Planning was another element found to exist more with the SQA organizations. One of the key concepts presented by J.M. Juran was: Quality does not happen by accident - it must be planned. The survey results found that other than the Software Requirements Specifications, the NON SQA organization completed an average of 21% of the other required project planning. In contrast, the SQA organization completed an average of 53% of the other required project planning. Chi-square analysis tests indicate a dependence between having an SQA department and producing the SQA ($X^2=18.62$, $p=0.00$, $n=80$), Software Verification ($X^2=5.78$, $p=0.017$, $n=80$), Acceptance Test ($X^2=11.62$, $p=0.001$, $n=80$), Configuration Management ($X^2=16.57$, $p=0.00$, $n=80$), and Software Maintenance ($X^2=7.58$, $p=0.006$, $n=80$) plans. However, the tests found no dependence between the existence of an SQA department and producing Software Requirements Specifications ($X^2=0.01$, $p=0.913$, $n=80$), System Plan ($X^2=1.93$, $p=0.165$, $n=80$), or Staff Development Plan ($X^2=2.29$, $p=0.131$, $n=80$).
- Although there are a number of other positive results from the utilization of SQA departments, the final one to be addressed relates to a demonstrated commitment to quality from senior level management. As noted in the prior research, the first step in building an effective SQA organization is commitment and leadership from senior level management. Approximately 70% of the respondents from organizations with SQA departments indicated the senior level management at their organization demonstrated a commitment to software quality. This compares to around 45% for the NON SQA organizations. Chi-square analysis tests also support the finding that there is a dependence ($X^2=4.93$, $p=0.027$, $n=80$) between the existence of an SQA department and management commitment to software quality in an organization.

7.3 Negative results

Although there are favorable results from the survey analysis to support the utilization of SQA organization, there are also areas of concern. In general, the areas are in relation to metrics and training.

- A negative result was observed in relation to the training and certification of the SQA personnel. In this examination, roughly 67% of the respondents with SQA departments report the personnel in their SQA department has little or no training in software quality assurance. As noted from the prior research, training is essential for an SQA department to be effective.
- Although it is positive to let the customer be the ultimate judge of quality, a customer mostly determines end-product quality. Software maintenance and in-process metrics must also be measured and used as indicators of quality. In-process and software maintenance metrics were measured by less than 40% of the organizations with SQA departments. Additionally, Pareto analysis, software reliability and complexity modeling were being used by less than 20% of the organizations with SQA departments.
- After making an official release of a software product, 41% of the SQA organizations release a patch in less than 30 days because of a critical bug. For the NON SQA organizations, 33% of the organizations release a patch in less than 30 days. Chi-square analysis tests show that there is no dependence ($\chi^2=0.41, p=0.52, n=80$) between the existence of an SQA department and when a patch is released due to a critical bug in the software.

7.4 Encouraging

If this small sample of respondents is to be used as an indicator of the direction the software industry is heading in relation to improving software quality, there are a number of encouraging statistics.

- In evaluating the results, most of the respondents had a genuine interest in improving software quality. As a result of the 2 open-ended questions on the survey, approximately 25 pages of suggestions and comments were generated on specific methods being used to improve software quality. In addition to having an interest in software quality, the majority of the respondents had related degrees and 5 or more years of experience in software development.
- The fact that approximately 80% of the organizations represented by these participants produced software requirements specifications was encouraging. This indicates that organizations are initially putting in writing the required functionality of the proposed system.
- Due to the ever changing technology in the software industry, keeping one's job skills current has become

essential. Approximately 80% of these respondents reported seeking self-improvement techniques outside of work to help improve their software related skills. In general, the majority of these respondents were satisfied with their work environment, current salary and knew how their job contributed to the overall success of the organization.

- In terms of reusable components, over 75% of respondents report their organizations encourage the creation of reusable components. Reuse means you don't have to reinvent the wheel each time a new project is started. Also, there is much research indicating improved software quality through reuse.
- Walkthroughs and code inspections were used by approximately 50% of these organizations.
- Approximately 50% of the respondents characterized their development process at a '2' on the SEI CMM maturity level. Edward Yourdon [24] reported in 1992, that approximately 85% of all organizations in the U. S. ranked a '1' in terms of quality on the SEI scale. So, do these results indicate we are improving? Without administering the full SEI process maturity assessment to these organizations, one could only guess.
- Organizations are applying for certification and assessments without SQA organizations or commitment to quality by senior level management. The encouraging aspect to this statement is that organizations can learn from these evaluations, precisely what is required to build a quality system in terms of improved processes, standards, and procedures.

8 Summary

There is no absolute formula that can be used to improve software quality, but there are many guidelines and approaches that have been provided by the quality experts and industry professionals. This research attempted to determine what methods were being used by those in the software industry to improve quality. The goal was to show that those organizations with SQA departments produced better quality products than organizations with no SQA departments. In general, the research found that those organizations with SQA departments did more in terms of what the quality experts indicate would lead to improved quality. It was found that the SQA organizations depended more on reviews for V&V. The SQA organizations also did more planning and pursued standards more often than organizations without SQA departments. The majority of the organizations with SQA departments reported a

significantly higher level of demonstrated commitment to quality from senior level management. Although, the SQA organizations were following more of the techniques described by the quality experts, the majority of the SQA personnel lacked specific training in software quality assurance. The SQA organizations were also found lacking in terms of metrics measured for the software project.

The analysis of the question, "After making an official release of a software product, how soon does your organization release a patch because of a critical defect in the product?", showed that there is no dependence between the existence of an SQA department and when a patch is released. This implies that one cannot unequivocally state that organizations with SQA departments produce better quality software than those organizations without SQA departments. Therefore, further research is required to prove, beyond a reasonable doubt, that organizations with SQA departments produce better quality software. Specific questions concerning statistics on customer satisfaction, customer related problems, software costs, project schedules, and defect patterns will provide more conclusive results. Another factor that appears to affect the quality of the software produced is management commitment to quality. This factor should be researched as a separate topic in relation to software quality. Although the research cannot positively conclude that the SQA organizations produce the better product, it appears that those organizations with SQA departments have a better foundation for producing a superior product.

REFERENCES

- [1] Rafael Aguayo, Dr. *Deming: The American Who Taught the Japanese About Quality*. Carol Publishing Group, 1990.
- [2] American Heritage Dictionary of English Language. William Morris, editor. Houghton Mifflin Company, 1976.
- [3] Emanuel R. Baker and Matthew J. Fisher. "Software Quality Program Organization". In G. Gordon Schulmeyer and James McManus, editors, *Handbook of Software Quality Assurance*. Van Norstrand Reinhold Company Inc., 1987.
- [4] Cathleen Benko, "Outsourcing Evaluation: A Profitable Process", *Information Systems Management*, Spring 1993, PP45-49.
- [5] Tsun S. Chow. Implementation of Software Quality Assurance Programs. In Tsun S. Chow, *Tutorial of Software Quality Assurance: A Practical Approach*. IEEE Computer Society Press, 1985.
- [6] Philip B. Crosby. *Quality is Free: The Art of Making Quality Certain*, McGraw-Hill, 1979.
- [7] Philip B. Crosby. *Quality Without Tears: the Art of Hassle Free Management*. McGraw-Hill, 1984.
- [8] Philip B. Crosby. *Let's Talk Quality: 96 Questions You Always Wanted To Ask Phil Crosby*. McGraw-Hill, 1989.
- [9] W. Edwards Deming. *Out of the Crisis*. Massachusetts Institute of Technology, 1986.
- [10] Robert H. Dunn and Richard S. Ullman. *TQM for Computer Software*. McGraw-Hill, Inc., 1994.
- [11] J. J. Greene, C. P. Hollocker, M. A. Jones, and T. C. Pingel. "Developing a Software Quality Assurance Program Based on the IEEE Standard 730-1981". In Tsun S. Chow, editor, *Tutorial of Software Quality Assurance: A Practical Approach*. IEEE Computer Society Press, 1985.
- [12] G. G. Gustafson, R. J. Kerr. "Some Practical Experience with a Software Quality Assurance Program". In Tsun S. Chow, editor, *Tutorial of Software Quality Assurance: A Practical Approach*. IEEE Computer Society Press, 1985.
- [13] Darrel Ince. *An Introduction to Software Quality Assurance and its Implementation*. McGraw-Hill, Inc., 1994.
- [14] Kaoru Ishikawa. *What is Total Quality Control? The Japanese Way*. Prentice-Hall, 1985.
- [15] J.M. Juran. *Juran on Planning for Quality*. Free Press, 1988.
- [16] Stephen H. Kan, *Metrics and Models in Software Quality Engineering*, Addison-Wesley, Reading, Mass. 1995.
- [17] Lela M. Long. Interview with ISO9000 registrar. February 1997.
- [18] James I. McManus. "The Heart of SQA Management: Negotiation, Compliance, Regression". In G. Gordon Schulmeyer and James McManus, editors, *Handbook of Software Quality Assurance*. Van Norstrand Reinhold Company Inc., 1987.
- [19] G. Gordon Schulmeyer, CDP. "Software Quality Lessons From the Quality Experts". In G. Gordon Schulmeyer, James I. McManus, editors, *Handbook of*

Software Quality Assurance. Van Nostrand Reinhold Company, 1987.

[20] Charles H. Schmauch. *ISO 9000 for Software Developers*. ASQC Quality Press, 1994.

[21] Software Engineering Technical Committee of the IEEE Computer Society, *IEEE Standard Glossary of Software Engineering Terminology*, IEEE-STD729, New York, IEEE, 1983.

[22] Software Quality Assurance Audits Guidebook. April 1996.

URL: <http://satc.gsfc.nasa.gov/audit/auditpage.html>

[23] Gerald Suarez. Three Experts on Quality Management. TOLO Publication No. 92-02. July 1992. URL: <http://deming.eng.clemson.edu/pub/pcsi/files/3expert.txt>.

[24] Edward Yourdon, *The Decline & Fall of the American Programmer*, Prentice-Hall, 1992.

[25] Edward Yourdon, *Rise & Resurrection of the American Programmer*, Prentice-Hall, 1996.

Appendix – The Survey

The complete survey with selections is located at: <http://www.pw2.netcom.com/~sawheel/sqasurvey.html>

1. Select the job title that best identifies your current position.
2. How many years of experience do you have in software development?
3. Select the category that best describes the type of software you are currently developing.
4. What degree(s) do you hold?
5. Does your organization contain a Software Quality Assurance (SQA) department that establishes and enforces organizational procedures and standards that define how quality is achieved?
6. What percentage of the SQA group at your organization is formally trained or certified in software quality assurance?
7. Select each indicator used by your organization or SQA group to quantify software quality.
8. Select each document your organization has developed for your current software project.
9. Has your organization or SQA group provided a Quality or Test Plan which specifies the quality objectives of the current software product you are developing?
10. Select each technique your organization or SQA group uses for V & V (validation and verification) of your current software product.
11. Does the senior management at your organization demonstrate a commitment to software quality?
12. How large is the project team on your current software development project?
13. How many days of training per year does your organization provide to help you improve your software development skills?
14. Do you pursue classes or other techniques (outside of work) that help to improve your software development skills?
15. Does your organization provide a good working environment that enables you to perform your job adequately?
16. Do you know how your job contributes to the overall success of the organization?
17. Does your organization provide you with a competitive salary based on your skills and experience?
18. Does your organization provide incentive programs?
19. Select the software development process that best describes the organizational approach to software development on your current project.
20. Select the characteristics which best describe the software development process of your current project.
21. Select each item your organization has pursued or is pursuing in relation to software quality.
22. Select each software quality metric that is currently being measured by your organization in relation to product quality.
23. Select each software quality metric your organization measures during in-process development.
24. Select each software quality metric your organization measures during software maintenance.
25. Select each CASE tool you currently use in software development at your organization.
26. Does your organization encourage creating reusable components in software development?
27. What percentage of reusable components are you using in your current software development project?
28. Select each item that is currently being reused on your current software project.
29. After making an official release of a software product, how soon does your organization release another version(patch) because of a critical defect in the product?
30. Describe a program or procedure that was implemented at your organization that improved the quality of a software product or process.
31. Describe a strategy or approach you have personally taken that improved the quality of the software you are currently developing.