Quality and Cost – It's Not Either/Or: Making the Case With Cost of Quality

George Webb
45th Range Management Squadron, Patrick Air Force Base

LTC Nanette Patton
Office of the Surgeon General

Today's organizations must be committed to the continuous pursuit of quality improvement as a requirement for survival. Traditionally, quality and cost have been perceived as a trade-off decision. For this reason, the main purpose and benefit of measuring quality costs has been to demonstrate that improved quality and lower costs go hand-in-hand. Through collection and analysis of these quality costs, improvement is translated into a language management listens to and responds to: money. This article provides tools and techniques to help infuse cost of quality (COQ) concepts into the project team activities to promote quality improvement throughout the full project life cycle.

There have been many changes in how DoD project management applies quality to software projects. In days gone by, there was a separate cost for quality attached to the items identified within an acquisition, whether for software or for hardware. As acquisition, project management, and system engineering have evolved, many companies have replaced the term *quality* with other terms, such as *performance* and *best practices*. Current best practices standard-bearers—such as the International Organization for Standardization (ISO), the Software Engineering Institute's Capability Maturity Model® Integration (CMMI®), the Project Management Institute (PMI), and the IEEE—integrate quality into everyone's work ethic and processes. This entails creating integrated product/process teams (IPTs) and letting the individuals, such as engineers, logisticians, configuration managers, testers, and project mangers (PMs), assume responsibility for quality within their functional processes.

ISO and CMMI argue for a separate quality group to maintain objectivity. While the PM has the final responsibility for quality, a quality manager can be responsible for day-to-day quality by developing and implementing a quality effort. Experience has shown, however, that when funds are tight and time is short, the quality group is among the first cut because they do not produce a *physical product* for the customer. Another short-term cost-saving measure is for the PM to select someone untrained in quality assurance, from the ranks of the engineering staff, to be the quality manager. In other cases, there may be no identification at all of a separate quality process owner to oversee this critical area; consequently, the COQ remains hidden in other project costs. When quality is just another sub-process, it has to fight for attention, priority, and funding like all the others. When this occurs, the opportunity to identify and correct problems early in the life cycle is often lost.

This maladaptive application of quality principles can be attributed to a lack of training, management support for quality processes, and adequate quality cost measurement systems. It is more likely to occur when cost and schedule become more important than or equal to quality.

Figure 1 (taken from [1]) shows the basic cost of good and poor quality.

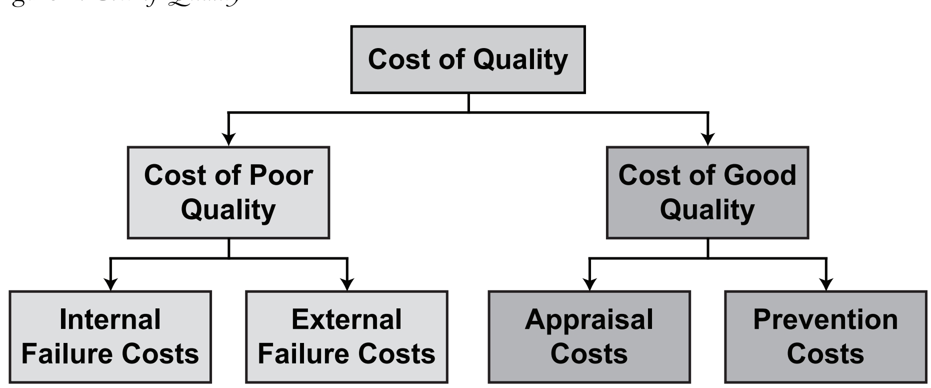
Evolution in the Approach to Quality

In decades past, the focus of quality was merely finding problems at the end of an assembly line and removing the defects before shipping to the consumer. If the product did not meet specifications, it was either reworked or scrapped—both expensive options. This approach is prone to human error and rarely finds all defects. Furthermore, this quality control approach only identified the defects found through a random sampling, but actually did nothing to determine the root cause of the problem for resolution.

If preventive quality measures and rework are deferred until the testing phase, the cost of change is 40 to 100 times greater than if the defect was fixed when it was created [2]. The testing stage has the least recovery time for *show-stopper* problems or unexpectedly large amounts of rework. This unpredictability becomes a large contributing factor to why projects miss their schedules. Furthermore, this type of approach, which assumes that doing more testing leads to shipping a better product, only works up to a point. With pure testing, one can get to something approaching 5-Sigma quality (0.2 defects per thousand lines of code); however, a product shipped at 5-Sigma is perceived as inadequate by today's manufacturing standards [3].

In the post-World War II reconstruction years, Dr. W. Edwards Deming introduced a quality program that simultaneously controlled the production and quality processes. Unfortunately, the United States did not adopt these principles until the 1980s with the introduction of the Total Quality Management System. Deming's core message—that we should stop inspecting defects out of products and start building quality in—has remained. The common thread of various quality methodologies is that the project team will build quality into the system design and will address quality continually throughout

Figure 1: *Cost of Quality*



® The Capability Maturity Model and CMMI are registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.

the life cycle. The goal is to identify problems up front and early, allowing corrective action and quality prevention to take place to reduce the number of critical defects found at the end of the assembly line.

This goal can be met through software quality surveillance, which includes walk-throughs, peer reviews, inspections, testing, IPT structures, as well as any method that identifies quality problems, risks, and operational capability weaknesses as early as possible. Approaching quality in this manner provides early corrective action and promotes lower quality costs upfront and early, thereby reducing end-of-program cost overruns [4].

Today, we have gone a step further by identifying risks which may have the potential to change engineering requirements, operational capabilities, and the quality of the product. In the spirit of risk management, software developers can help prevent one of the most common causes of defects—ambiguous requirements—by writing comprehensive acceptance tests when recording each requirement. Furthermore, automating these tests and running them as part of frequent integration builds will help detect defects when they happen.

While common sense says that preventing defects or finding them when they are cheapest to fix is preferable to finding them at the end when they are many times more expensive, several software development projects fail to write tests upfront, do inspections, or perform frequent integration—despite the benefits.

Why We Don't Implement Preventive Quality Processes

Implementing quality processes is tedious, time-consuming work in most environments. And, time is money. There is document inspection (usually several hundred pages) and writing early tests for critical requirements at the beginning of a project. It is hard to keep the tests up-to-date as the requirements change and even harder when you realize that you have to inspect the tests. These strategies increase the cost of implementing quality and the return on investment is not always predictable with a high degree of reliability, especially when the requirements and design have not been locked in. Thus, it is mind-wrenching work to determine which of all the possible strategies for implementation will bring the best value to the project.

Carolyn Fairbank, CEO of the Quality Assurance Institute, said:

> We're far too focused on product delivery, not process capability. We're too busy trying to get the product out the door. Granted, this is a market-driven phenomenon, but we'll have to change that deadline-driven attitude to one of good processes. If you get the process right, the product will have a far better chance at success. Unfortunately, many IT professionals still don't quite understand the concept of process management. [5]

According to Karl Wiegers:

> We do far too much pretending in software. We pretend we know who our users are, we know what their needs are, that we won't have staff turnover problems, that we can solve all technical problems that arise, that our estimates are achievable, and that nothing unexpected will happen. Risk management is about discarding the rose-colored glasses and confronting the very real potential of undesirable events conspiring to throw our project off track. [6]

"Today, we have gone a step further by identifying risks which may have the potential to change engineering requirements, operational capabilities, and the quality of the product."

Risk identification requires a look into the future as to the potential success of the program. The challenge lies in the identification of risk versus current problems. The PMI defines risk as "an uncertain event or condition that, if it occurs, has a positive or negative effect on a project's objectives" [7]. Current problems require attention and action even if the immediate remedy is to defer corrective action until later. Risks realized may require actions that lead a project team to proceed in a different direction altogether or canceling the project entirely. Project teams must accept some risks due to other requirements, conditions, assumptions, or constraints; however, if a project team chooses to completely ignore risk, they greatly increase the probability of project failure.

A company's economic status or condition are significant factors when deciding what process to implement to track quality cost [8]. Pursglove and Dale suggest that the profitable nature of the business can make it more difficult to convince management of the need to track COQ [9]. For example, having more engineers and fewer quality assurance people on a project can be great for a company's short-term financial success. However, if project staff members do not build in quality from the start, a greater reliance on product rework results. The organization will eventually pay for the inadequate quality as customers identify problems with the product or service. Engineering changes must take place before the customer deems the product usable. These engineering changes late in the development process may result in a product or service that does not quite meet the original intent on the capabilities delivery; this, in turn, can lead to lost business. If this happens on a recurring basis, the company may experience competitive and financial difficulties. If so, a company may be more open to performing an assessment in an attempt to get back on track. Then, after collecting and analyzing data that reveals a quality problem, the company finally decides to track quality costs. This may also be the time when the company experiences total failure. They know something has to be done, but don't have a well thought-out plan. They make *knee-jerk* decisions, such as simply canceling the project and not addressing the underlying quality problems in their processes; that, in turn, causes unintended conflict within the organization.

Not having a clear understanding of the actual value of COQ also hinders the adoption of quality processes. There has been a persistent misconception in the business community that the COQ is a cost over and above that of developing and producing a project to meet a specific and required outcome and schedule. The COQ, regardless if it is software or hardware, is the price of not creating a quality product or service. If the development process was perfect with no problems and there was no possibility of substandard service, failure of products, or defects in their manufacture, then organizations would have no expenditures on COQ.

Prevention	Appraisal
Represents everything a company spends to prevent software errors, documentation errors, and other product-related errors. • Staff training • Requirements analysis • Early prototyping • Fault-tolerant design • Defensive programming • Usability analysis • Clear specifications • Accurate internal documentation • Pre-purchase evaluation of the reliability of development tools	Includes the money spent on the actual testing activity. Any and all activities associated with searching for errors in the software (and associated product materials) fall into this category. • Design reviews • Code inspection • Glass box testing • Black box testing • Beta testing • Test automation • Usability testing • Pre-release out-of-box testing by customer service staff
Internal Failure	**External Failure**
The cost of coping with errors discovered during development and testing. These are bugs found before the product is released. • Bug fixes • Regression testing • Wasted in-house user time • Wasted tester time • Wasted writer time • Wasted marketer time • Wasted advertisements • Direct cost of late shipment • Opportunity cost of late shipment	The costs of coping with errors discovered after the product is released. These are typically errors found by your customers. • Technical support calls • Answer books (for support) • Investigating complaints • Refunds and recalls • Interim bug fix releases • Shipping product updates • Warranty, liability costs • Public relations to soften bad reviews • Lost sales • Lost customer goodwill • Supporting multiple versions in the field • Reseller discounts to keep them selling the product

Table 1: *COQ Data Points*

COQ is the sum of costs incurred in maintaining acceptable quality levels plus the cost of failure to maintain that level (cost of poor quality), and typically ranges from 15-25 percent of total cost [10].

Philip B. Crosby's "Quality Is Free" concept [11], identified two main categories of quality costs: conformance costs (cost of good quality), and nonconformance costs (cost of poor quality).

Conformance costs include prevention and appraisal costs; nonconformance costs include *internal failures* as well as *external failures* (Table 1, from [12]). A defect found early in the project prior to customer delivery is termed an internal failure. A defect identified after the product has been deployed to the customer is an external failure. External failures can also include incompatibility of the software with legacy software installed in the field, or a lack of commonality between redundant systems.

Beyond not clearly understanding COQ concepts, key decision makers in an organization may lack knowledge in determining quality costs and the principles for collecting quality costs. Without knowing what quality principles are, an individual or organization may have no idea where to place their focus to obtain quality costs. The organization can remedy this either by ensuring that a quality curriculum is included in the training for project staff and senior leadership or by hiring a quality consultant to guide the organization.

Getting a COQ System in Place

Herb Krasner and Dan Houston explain that companies need to answer three questions [13]:

1. How much does poor software quality cost?
2. How much does good software quality cost?
3. How good is our software quality?

Once these questions are answered, the project team can compare quality costs to overall software production costs and software profits, and to benchmarks and norms. They can also better analyze product quality to improve their competitive situation, measure improvement actions and the bottom-line effect of quality programs, visibly see previously hidden costs related to poor quality, and more clearly see the economic tradeoffs involved with software quality.

Even if the project team cannot measure all of these costs with a high reliance, a COQ model quantifies (for management and executives) the amount of money being lost on fixing defects and delivering poor-quality products. This, in turn, negatively affects their bottom line. That provides motivation and impetus for implementing preventive quality measures.

In order to feed decision makers those costs with some degree of legitimacy, a life-cycle model has been developed to guide this work (Figure 2, next page).

Note: If no data source exists for the collection of costs, then the project team will have to use some type of analytical technique to develop a cost estimate model. With this model, the team should be able to establish the cost estimates for the appropriate quality categories.

Capturing COQ

This step in the life cycle is twofold: (1) identify the costs and (2) determine a method for entering costs into an accounting system that tracks them throughout system/service development.

When quality costs are initially determined, the categories included are the visible ones.

Oftentimes it is what you do not know that can hurt a project. Software quality costs are not always easy to identify within programs. Software has many hidden costs that may not be readily apparent to the project manager. These are shown below the water line in Figure 3 (next page, adapted from [1] and [14]) and in the expanded Figure 4 list on page 27.

As an organization internalizes a broader definition of poor quality, the hidden portion of the iceberg becomes apparent. Identifying these costs opens a

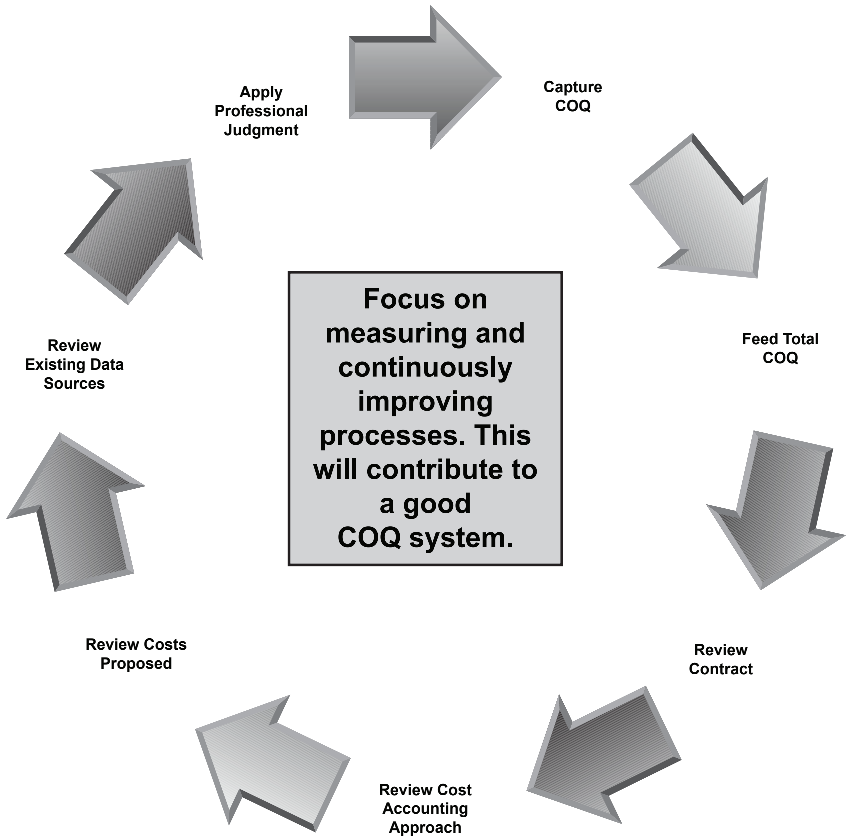


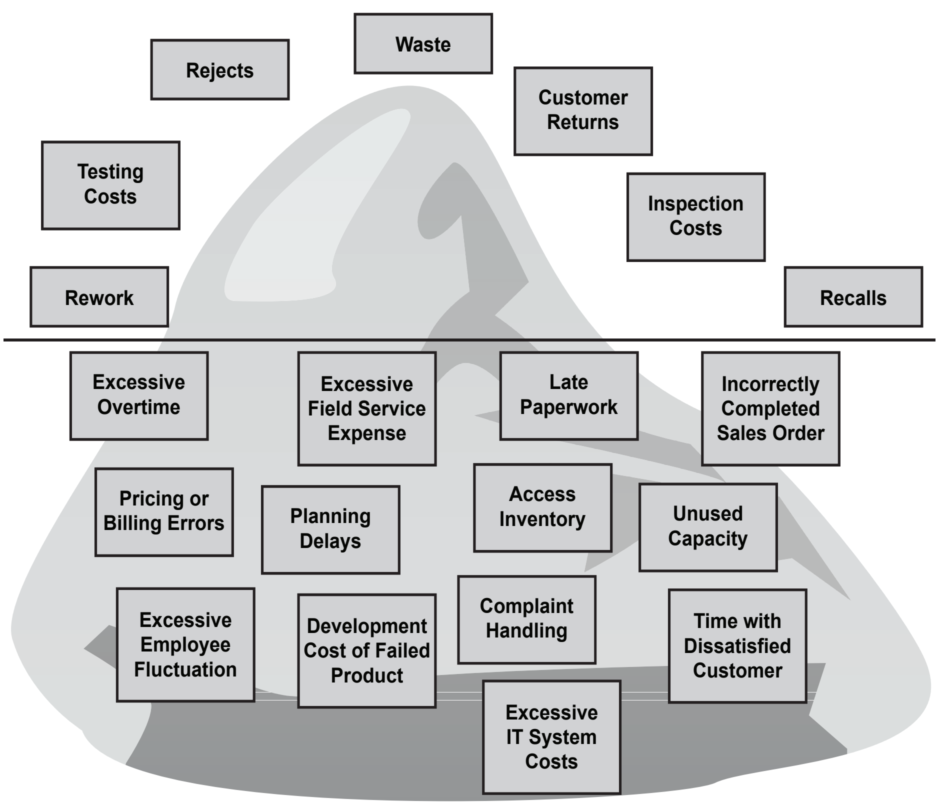
Figure 2: *The Life Cycle of COQ*

door of opportunity and the project team can then implement processes to avoid more costly expenditures later in the project. Furthermore, the team increases the probability that the product/service will be acceptable to the customer within all required elements and functions that the product/service is supposed to deliver.

A proper focus on quality entails identifying and funding quality costs and institutionalizing quality processes at the very beginning. It also requires that quality experts supply to management an estimate of the total quality costs, good or bad. Management uses the information obtained through the quality initiative as a tool to adjust funding allocations. Once the quality experts have gathered data, upper management can determine with a clearer picture where to concentrate quality efforts and funding for eventually achieving a greater positive impact and promoting a successful project.

While the *traditional method* is the most used method in collecting costs related to quality (according to the American Society for Quality), organizations can also use the defect document collection method, the time and attendance collection method, or the assessment method. The PM and the quality manager (if the PM has designated one for the project), will have to do a cost-benefit analysis to determine which method is the best fit for the organization and project, given available resources.

Figure 3: *The Iceberg Model of COQ*



Feed Total COQ

Having identified costs and a method for tracking them, diligent data collection is now required. This also includes incorporating previously unidentified costs revealed during ongoing activities that now require tracking.

Review the Contract

The quality group must be conscious of the legal terms as well as the performance of specific tasks within any contract to support product/service development and delivery. They should be familiar with specifics to ensure the contractor is completing required tasks throughout the life cycle of the product. There may be specific tasks that occur sporadically during the development cycle and therefore require a more concerted follow-up.

Review the Cost Accounting Approach

On a periodic basis, it is important to ensure that the cost accounting process and repository adequately track all pertinent cost information collected on work currently performed.

The system may require refinement due to new requirements and/or additional costs not identified at the start of the life cycle. This may also require changes in the data collection method.

Review Existing Data Sources

It is also important to ensure that the sources used for cost data provide the best available data in terms of validity and accuracy. Do not hesitate to switch to a better data source if it provides data that will give a more accurate picture of where the project stands at a particular point within the process.

Review Costs Proposed

As development continues, hidden costs not identified at the start will reveal themselves. Track these costs along with those originally proposed to facilitate budget adjustments and to recalculate the return on investment projections in order more effectively manage expectations.

Studies (such as [9]) indicate that the further along in the process quality is worked into the product or service that the higher the COQ will be. The project team should try to reduce the overall cost of each product or service by establishing the optimum level of preventive and appraisal costs that minimizes resultant error costs. The net result of quality improvement should be a reallocation of costs across the COQ categories resulting in a reduction in the overall COQ. An example of this [15] is shown in Figure 5.

Apply Professional Judgment

This is the time for analysis of all information gathered regarding the health of the program. By pushing this data and analysis to the appropriate project decision makers, they can make informed decisions on how to proceed to ensure project success.

Conclusion

At first glance, an individual might be prone to think that collecting quality costs is expensive, adding unnecessary costs to the product or project. Quality is not free in that you have to make an up-front investment in time, money, and effort. However, if performed properly over the full life cycle of the project, you can recoup the resources expended for quality processes by avoiding rework later in the project life cycle. By communicating the quality story in terms of dollars, you can enlist the help of senior management to infuse quality processes throughout the project life cycle and contain project costs for the long haul. As with many project management standards and guides (e.g., [7]), collecting quality costs is like project planning; it is cheaper to properly plan than it is to plan a little and fail a lot.◆

- Excessive overtime
- Pricing or billing errors
- Premium freight costs
- Excessive field service expenses
- Planning delays
- Low morale
- Excessive employee turnover
- Development cost of failed product
- Overdue receivables
- Excessive system costs
- Complaint handling
- Expediting costs
- Paperwork/documentation is late
- Customer allowances
- Lack of follow-up on current programs (require re-testing)
- Excess inventory
- Unused capacity/scrap cost
- Time with dissatisfied customers
- Incorrectly completed sales order
- Scheduling delays
- Test tool certifications
- Training of operators
- Obtaining user licenses for off-the-shelf software
- Security measures for information assurance
- Designated Approving Authority certifications
- Compatibility issues with legacy
- Government oversight (for government contracts)
- Risk tracking and impacts of external dependencies
- Management reserve for tracking risks
- Training in risk detection
- Possibility for developing more quality gates in development, which will require additional personnel and time.
- Bug fixes
- Regression testing
- Wasted in-house user time
- Wasted tester time
- Wasted writer time
- Wasted advertisements
- Direct cost of late shipment
- Opportunity cost of late shipment

Figure 4: *More Hidden Quality Cost Data Sets Not Easily Identifiable*

Figure 5: *COQ in Software Testing*

	A	B	C	D
1	Testing Investment Options: ROI Analysis			
2				
3		No Formal	Manual	Automated
4	Testing	Testing	Testing	Testing
5	Staff	$0	$60,000	$60,000
6	Infrastructure	$0	$10,000	$10,000
7	Tools	$0	$0	$12,500
8	Total Investment	$0	$70,000	$82,500
9				
10	Development			
11	Must-Fix Bugs Found	250	250	250
12	Fix Cost (Internal Failure)	$2,500	$2,500	$2,500
13				
14	Testing			
15	Must-Fix Bugs Found	0	350	500
16	Fix Cost (Internal Failure)	$0	$35,000	$50,000
17				
18	Customer Support			
19	Must-Fix Bugs Found	750	400	250
20	Fix Cost (Internal Failure)	$750,000	$400,000	$250,000
21				
22	Cost of Quality			
23	Conformance	$0	$70,000	$82,500
24	Non-conformance	$752,500	$437,500	$302,500
25	Total CoQ	$752,500	$507,500	$385,000
26				
27	Return on Investment	#N/A	350%	445%

References

1. Buthmann, Arne. iSixSigma. "Cost of Quality: Not Only Failure Costs." <http://europe.isixsigma.com/library/content /c070502a.asp>.
2. Boehm, Barry W., and Richard Turner. Balancing Agility and Discipline: A Guide for the Perplexed. Addison-Wesley Professional, 2003.
3. Lahman, H.S. "Testing vs. Defect Prevention." 27 Oct. 2004 <http://pathfinderpeople.blogs.com/hslahman/testing_and_defect_prevention/index.html>.
4. The Data and Analysis Center Gold Practices Web site. "Software Acqui-

Coming Events

January 10-13

6th Annual IEEE Consumer Communications and Networking Conference

Las Vegas, NV

www.ieee-ccnc.org

January 19-20

16th Annual Multimedia Computing and Networking Conference

San Jose, CA

http://mirage.cs.uoregon.edu/mmcn2009

January 19-22

10th Annual Lean Six Sigma and Process Improvement Summit

Orlando, FL

www.site-members.com/lsspi/index.html

January 21-23

Principles of Programming Languages

Savannah, GA

www.cs.ucsd.edu/popl/09/

January 27-30

Network Centric Warfare 2009

Washington, D.C.

www.ncwevent.com/index.php

April 20-23

21st Annual Systems and Software Technology Conference

Salt Lake City, UT

www.sstc-online.org

Coming Events: **Please submit coming events that are of interest to our readers at least 90 days before registration. E-mail announcements to: nicole.kentta@hill.af.mil.**

sition Gold Practice, Formal Risk Management." 2008 <https://www.goldpractices.com/practices/frm>.

5. Smith, Kennedy. "The Software Industry's Bug Problem." Quality Digest. Apr. 2003 <www.qualitydigest.com/april03/articles/03_article.shtml>.
6. Wiegers, Karl. "Know Your Enemy: Software Risk Management." Software Development. Vol. 6, No. 10. Oct. 1998: 38-42 <www.dci.pomona.edu/~markk/cs121.f07/supp/risk_mgt.html>.
7. Project Management Institute. A Guide to the Project Management Body of Knowledge (PMBOK Guide). 3rd ed. PMBOK Books, 2004.
8. Campanella, Jack. "Principles of Quality Costs: Principles, Implementation, and Use, Third Edition." ASQ Quality Cost Committee. ASQ Corporation, 1990.
9. Pursglove, A.B., and B.G. Dale. "The Influence of Management Information and Quality Management Systems on the Development of Quality Costing." Total Quality Management Vol. 7, No. 4, 1 Aug. 1996: 421-432.
10. DeFeo, Joseph A. "The Tip of the Iceberg." Quality Progress Vol. 34, No. 5, May 2001: 29-37.
11. Crosby, Philip B. "Quality Is Free: The Art of Making Quality Certain: How to Manage Quality So That It Becomes A Source of Profit for Your Business." Mentor. 1 Jan. 1980.
12. Pirozzi, Rob. "Understanding Quality Cost." LogiGear Corporation. 23 Oct. 2007 <www.logigear.com/newsletter/understanding_quality_cost.asp>.
13. Krasner, Herb, and Dan Houston. "Using the Cost of Quality Approach for Software." CrossTalk Nov. 1987.
14. DeFeo, Joseph A. "The Tip of the Iceberg." Quality Progress Vol. 34, No. 5, May 2001: 29-37.
15. Black, Rex. "Investing in Software Testing: The Cost of Software Quality." 2000 <www.compaid.com/caiinternet/ezine/cost_of_quality_1.pdf>.

About the Authors

LTC Nanette Patton is currently serving as the Army Medical Department Deputy Chief Information Officer for Army Systems Integration in the Office of the Surgeon General (OTSG). LTC Patton has over 10 years experience as a health services systems manager and is Level III DoD-Acquisition-certified in Information Technology and Level II-certified in Program Management. She has master's degrees in human resources management from Chapman University and in business administration (with a concentration in computer information systems) from Colorado State University.

OTSG
Information Management Directorate
5109 Leesburg Pike
STE 594
Falls Church, VA 22041-3258
Phone: (703) 681-0187
E-mail: nanette.patton@us.army.mil

George Webb is currently serving as lead software quality assurance specialist for the 45th Space Wing Range Management Squadron, Patrick Air Force Base, Florida. Webb is Level III Defense Acquisition Workforce Improvement Act-certified in Production, Quality, and Manufacturing and Level I-certified in Logistics and Program Management. He also has a Business Management Certification for the State of Florida. Webb has worked as a quality assurance specialist for the DoD for 22 years and has focused his efforts on software quality for the past decade.

45th Space Wing Range Management Squadron
14640 Hangar RD
Canaveral Air Station, FL 32925-2206
Phone: (321) 494-0755
E-mail: george.webb@patrick. af.mil