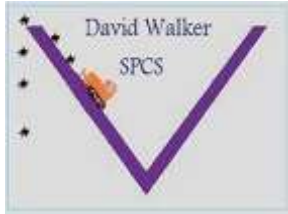




## SPONSORS:



CMMI Training  
20% ASQ Discount  
Introduction to CMMI-DEV  
SEI Licensed  
[More Information](#)

Advertise Here, email  
[swd@asq.org](mailto:swd@asq.org)

## In This Issue:

### ARTICLES & NEWS

- [Chair's Message - Bill Trest](#)
- [Membership News – Hank Sobah](#)
- [Opportunity to Publish - SQP Journal Articles](#)
- [Management Commitment to Quality Requires Measures - John Balza](#)
- [Discover Your True Project Management Style - By Sabrina Jackson](#)

### ANNOUNCEMENTS

- [Mark your Calendars - GLSEC 2009 in Grand Rapids](#)

### CONFERENCES & WEBINARS

- [International Conference on Software Quality 2009](#)

### COLUMNS

#### GET CERTIFIED

- o [Try the CSQE Quiz – Linda Westfall](#)

#### MEDICAL DEVICE SOFTWARE

- o [New Guidance for Networked Medical Devices IEC 80001 – David Walker](#)

### FROM THE REGIONS

- [Region 1 New England – Howie Dow](#)
- [Region 6 Pacific Northwest – Tom Gilchrist](#)
- [Region 8 Ohio Valley – Greg Zimmerman](#)
- [Region 10 Michigan – Louise Tamres](#)
- [Region 14 Southwest & Mexico – David Peercy](#)

## Questions or comments?

### See:

[Software Division Web Site](#)

[Software Division Management](#)

### Or Contact:

[Bill Trest](#), Chair

[Nicole Radziwill](#), Chair-Elect

[Tom Ehrhorn](#), Secretary

[Brenda Fisk](#), Treasurer

[Tom Ehrhorn](#), Newsletter Editor

How can we improve Software Quality Live? Does this issue provide helpful information? [Let us know!](#)

## ARTICLES & NEWS

### Chair's Message

By W.L. 'Bill' Trest, ASQ CSQE, Chair

Since the first of 2009, software division may well have started its busiest year ever providing face-to-face (F2F) conferences and member value. This year's fun began in Santa Ana, CA, where Software Division (SWD) personnel supported the annual ASQ Quality Management Division (QMD) meeting. Software division participation in the ASQ QMD conference was well received and a last minute slide show presentation was provided on Software Process Sampling by Attributes. Several United States West Coast SWD members attended and were very complimentary about the joint QMD/ SWD conference.

Our ASQ 2009 World Conference on Quality and Improvement (WCQI 2009) will be held in Minneapolis, MN, in May, where the SWD's Institute for Software Excellence (ISE) will provide several world class speakers and presenters from the software industry.

Join us at WCQI 2009 as we explore the timely and sometimes controversial topics of importance to both purchasers and builders of software. Presentations and speakers planning to attend the Institute for the Software Excellence of the ASQ SWD include:

- **CMMI High Maturity Made Practical**— by Bob Stoddard, senior member of the technical team at the Software Engineering Institute at Carnegie Mellon University
- **Software Is a Risky Business**— by Linda Westfall, President of The Westfall Team
- **How to Set up IT Dashboards Using the CTQ (Critical to Quality) Process** — by Kandy Senthilmaran, Six Sigma Master Black Belt at CSC/Covansys
- **Impact of EU (European Union) Medical Device Directive Revision on Software (with ASQ Biomedical Division)**—by Glen E. Emelock, Senior Partner with the CRO Group
- **Are you smarter than a CSQE?** — by Carol Dekkers, President of Quality Plus Technologies and member of the US ISO team since 1994
- **"How to" Lean Processes and Procedures Using Best Practices** — by Tim Olson, Partner at Lean Solutions International and 2008 National Baldrige Quality Examiner
- **Data-Driven Software Management** — by Taz Daughtrey, Associate Director for Software Development for the Institute for Infrastructure and Information Assurance at James Madison University
- **Reliability-Centered Lean Software Testing** - by Dr. Zigmund Bluvband, President of Advanced Logistics Developments, and Dr. Sergey Porotsky, CTO of FavoWeb Ltd.
- **Selecting and Implementing a Best Practice Framework** by Dr. Mark Paulk, Senior Systems Scientist at the IT Services Qualification Center at Carnegie Mellon

*(Bill Trest is Chair, ASQ Software Division, as well as a Senior Staff Specialist Software Quality Engineer with Lockheed Aeronautics Company, Fort Worth, Texas. Bill is also an ASQ Certified Software Quality Engineer (CSQE) as well as a Senior Member of ASQ, with Greater Fort Worth, Section 1416.)*

[Back to top](#)

### ASQ Software Division Membership News

By Hank Sobah, Chair Membership Committee

Please be sure to join the Software Division Social at the Division's Hospitality Suite on Monday Night, May 18 during the **World Conference on Quality and Improvement** in Minneapolis, MN, USA. There will be food and drink, an opportunity to discuss the division with the Division Management Committee and an opportunity to network with other Software Quality Professionals and Experts.

As of April's data, there are 3,297 Members of the Software Division. This number is down from 3,570 last July, the beginning of our fiscal year. These numbers seem to fluctuate throughout the year, sometimes up and sometimes down. As our membership fluctuates, the division is trying to understand what is driving the changes that we see in our membership numbers. If you are interested in helping, the division could use your assistance in trying to contact departing members to find out why they leave the division, and to contact new members to welcome them and understand what they need from their division membership. If you are interested, please contact me at [hsobah@yahoo.com](mailto:hsobah@yahoo.com) with your contact information, your availability and

If you are interested in any other Software Division opportunities, please feel free to contact me, or any other Software Division Management Committee member, or contact ASQ directly at 1-800-248-1946.

[Back to top](#)

## Opportunity to Publish - SQP Journal Articles

By Sue Carroll

Software Quality Professional (SQP) Journal is an ASQ journal supported by the ASQ Software Division. In each quarterly issue there is an article that is available to non-subscribers. The abstract for this article is included below.

If you would like to write an article for the journal – [see the author guidelines](#). If you would like to volunteer to review articles or books, or write a letter to the editor contact [sue\\_carroll@bellsouth.net](mailto:sue_carroll@bellsouth.net).

ProPAM: Discussion for a New SPI Approach by Paula Ventura Martins, INESC-ID Universidade do Algarve and Alberto Rodrigues da Silva, INESC-ID /Instituto Superior Técnico.

Software process improvement (SPI) is one of the main challenges of software development. Unfortunately, process descriptions generally do not correspond to the process executed during software development projects. They represent high-level plans, but do not contain the information necessary for the concrete software projects. This deficient alignment between process and project is caused by processes that are unrelated to project activities and the failure to detect project changes to improve the process. Process and project alignment is essential to determine how process management is important in achieving an organization's strategic objectives. Considering this approach, this article presents a comparative study of some of the most recognized SPI approaches and proposes a new SPI methodology, Process and Project Alignment Methodology (ProPAM). The authors' aim is to show problems observed in existing SPI approaches and recognize that further research in process and project alignment based on person oriented approaches is required. To read more go to <http://www.asq.org/software-quality/2009/03/process-management/propam-discussion-for-a-new-spi-approach.pdf>

[Back to top](#)

## Management Commitment to Quality Requires Measures

By John Balza

### Abstract:

In 1998, Hewlett-Packard began a program for its UNIX software to reduce the number of customer-found defects by a factor of 10 over 5 years. True to the literature, we found that strong management commitment to quality was the key to any improvement. Engineers and managers, however, were only being measured on delivering functionality on schedule. In order to change the culture, we had to demonstrate to our software development teams that management valued quality over functionality while maintaining schedule. A key enabler was providing management with ways to measure quality as easily as schedule.

### Background:

The literature is full of lessons on how important it is to have management commitment in order to drive change. In our case, we wanted to improve our product quality. Management quickly declared this to be the number one objective. But in order to make an objective truly number one, it needs to be measurable. How can you provide management real-time measures of quality, so that it can be measured as easily as schedule? This paper discusses our real world experience in making quality number one, measuring that quality, and in the end seeing an 83% reduction in customer-reported defects.

The HP-UX product is Hewlett-Packard's version of the Unix operating system. It consists of approximately 18 million lines of code produced in a distributed organization of 1500 people in 7 worldwide locations. These 1500 people are organized into 13 different R&D labs reporting into several division managers. The HP-UX 11.00 version of our product was released with significant functionality additions to the previous release; in particular it was our first 64-bit operating system. Many customers began to use the operating system in mission critical applications where availability and reliability were very important. In these new environments, customers were demanding higher levels of quality than in our previous releases. In fact, the next year, in a survey conducted by Interex, the HP User Group,

developing higher quality software, was the 2<sup>nd</sup> highest issue in the poll.

### **Typical Tradeoffs**

Quality has always been a top priority within Hewlett-Packard, in fact it was usually described as "Priority 0" among the traditional tradeoffs of functionality, resources, and schedule.

We had always had rigorous quality standards, but most of these standards could only be measured at the end of a release. Resources, functionality, and schedule are much easier to measure throughout the lifecycle. The most typical measure of functionality and schedule was "did the functionality get checked-in to the release on the scheduled date". Managers did detailed plans for meeting the functionality complete milestone for each feature. They were measured primarily on meeting these check-in targets. Without an adequate measure of quality at check-in, however, the quality of the code varied widely. While they may have met their targeted date, they may actually have caused the entire products shipment to slip because of checking-in low quality code.

One way of thinking is that each new check-in introduced new defects, so as we progressed the number of defects to be found was constantly increasing until we reached functional complete. We then spent the testing cycle of the release, removing all these introduced defects. Typically, it would take us 9 months from the last kernel check-in until the release met our release criteria. 2 months were spent stabilizing the kernel, 2 more months adding the higher-level layers of functionality that depended upon this kernel functionality, and 5 months to stabilize the entire system.

This behavior was further aggravated by the marketing and management behavior of selecting a target date for a release and asking for as much functionality that they felt could be reasonably accomplished by that date. The expectation was unless you could 'prove' that you couldn't make the date with the functionality, you weren't allowed to remove it from the requirements. This naturally led to slips in schedule. Every time the schedule slipped marketing would insist upon more functionality to meet the 'competitive environment'.

### **Management Commitment**

Given the new markets we were entering and the customer concern about our software quality, management knew that we needed to improve our software quality. The general managers agreed that for our next release quality was going to be the number one objective. But how was this any different than quality being priority 0?

First, we decided to set a tough goal. The key metric we chose to measure our quality was the number of customer-reported defects in the product's first year of shipments. This metric was chosen because it was easy to measure, it reflected customer experience, we had historical data from previous releases, and HP had used a similar measure in the late 1980s in a corporate program to improve quality. We initially proposed a 2X reduction in our next release. Based on our reliability models, a 2X reduction was what the software needed to contribute to meet an important customer availability goal that our systems would be available 99.999% of the time. But after further thought, we decided that quality required a longer-term commitment, one that would change the culture and the behaviors in our organization. We decided upon a 10X improvement over a 5-year time frame compared to our 11.00 release. That is in 2004 (5 years), customers would report only 10% as many defects on that versions of HP-UX as they reported on the 11.00 version in 1999.

Next, we realized that if we set quality as the number one goal, we needed to allow the project teams flexibility in some other dimension of the resources, schedule, and functionality tradeoff. Management decided functionality would be the primary trade-off.

But this would require a cultural shift, a change in everyone's behaviors. In order to make these real, top managers had to start demonstrating a change in their behavior before any of the lower level managers or the engineers would begin to change their behaviors.

The first management change was to significantly reduce the 'must' functionality for the next release to just a handful of key features. Historically, each marketing member responsible for a key functionality area would label quite a few features, as 'must' priorities for the next release. The program team would typically be dealing with hundred's of 'must' features. Instead, we decided upon a couple key business needs that the next release should satisfy. What would be the marketing message for this release? Looked at this way, the 'must' features were those that supported the marketing message. Instead of hundreds of features that were musts, we could reduce it to 10s.

The second change was a consistent message sent in all employee communications about quality being our number one objective, and functionality being the lowest objective. This was reinforced during checkpoint meetings whenever a lab said they couldn't meet the schedule. The first question asked was whether some functionality could be cut to hold the schedule. Budgets were examined for significant increases in quality resources - such things as training on quality and project management methods, new tools and process improvements; and investments in tests and the machine resources to run those tests. Schedules were examined to ensure that they included time for peer reviews and development of new tests. By subordinating these other dimensions to quality in several ways, people began to understand that upper management really was calling for a change to make "quality the number one objective". The third key change was that management established what became known as the "hour of power", an hour each week where the program management team came into the division manager's staff and they examined the progress on key milestones and the quality of the release. This meeting tended to be very action oriented - the division manager's direct staff would be given action items or asked to become involved in a tough problem area (usually leading daily meetings). Since action items flowed down the management chain rather quickly, the whole organization soon found out when their manager had an action item.

Of course, this required that we be able to measure our quality at least weekly, so that managers had action items related to improving quality as often as they had action items related to milestones.

### **Measuring Quality Real-Time**

The first quality metric we put in place was a defect backlog metric. From our defect tracking system, we could graph for the system as a whole, for every lab, and every project within a lab, the number of incoming defects, the number of defects fixed and the current number of open defects. We set a goal that every lab would have at most a two-week backlog of defects. This allowed us to set a consistent goal across the entire organization; no matter what the size of their code base or how many defects were being found.

For example, if for subsystem x, they've had 60 incoming defects in the last 6 weeks, that averages out to 10 defects/week. Their two-week backlog goal would be 20 defects. Their team was asked never to have more than 20 open defects outstanding. Another team getting only 12 defects every 6 weeks would be asked to keep the number of outstanding defects below 4.

Why two weeks? That was our practical experience of how long it typically took to get a defect fixed. We would spend approximately a calendar week to root cause the defect and another week to get the repair in the source management system.

Everyday we published the defect backlog for every project in every lab. We now had our first metric where we could clearly decide quality versus functionality. In the Hour of Power, the division manager asked his staff to stop submitting functionality to the release, if they did not have their defects under control. This wasn't cut and dry, occasionally exceptions were made when critical functionality was needed to meet other dependencies, but usually staffing was moved to fix defects whenever a lab was out of limits on this metric.

If a lab's defect backlog grew too big, the lab manager would often send voicemails or emails to the offending lower level managers asking what they were doing to get this under control, whether they had all the resources they needed, and if he or she could help in any way. At certain points of the release, engineers were lent to other organizations to help triage and fix defects, because everyone knew it would increase the overall productivity of the release, if these defects were fixed. This began to strongly reinforce the 'Quality is the number one objective' because it is what your boss talked about as often as they asked about whether you were on schedule.

### **The Critical Resource**

The second quality measure that was put in place was the result of an analysis of our development process to find the critical resource. Eliyahu Goldratt in his book, Critical Chain[1], discusses the concept of the critical resource. This is the resource that controls the throughput of the entire process. While you could make process improvements in several areas of the process, they may have no effect on the process throughput. Only by improving the throughput of the critical resource, can you improve the overall throughput of the entire process.

As we examined our process, we identified that the critical resource in our process was the top-of-branch of our software configuration management system. This is the point where an engineer publishes his

latest software changes and makes them available to the rest of the development community. We realized that if we kept the quality of this point at high quality, we would improve everybody's productivity. This top of branch, became known as the "mainline". Productivity was improved because we recognized that engineers stumble over each other's defects all the time. Sometimes they ended up tracing a defect in their code to find that it actually was a defect in someone else's code, but it wasn't known yet. You're unable to make progress on your project until this defect is fixed. Other times, while you're triaging your defects, others are continuing to submit new code, and may mask the defect that you've been trying to work. So it was very easy to be in a situation where the quality of 'mainline' continued to deteriorate as more functionality was added.

We put in place what we called the 'mainline stability test', which we ran everyday. This suite of tests consisted of our critical functionality and reliability tests. We required that the reliability tests run 48 continuous hours of operation and that there were no regressions in the pass rate on the functionality tests. (Note: we actually have 3 test set-ups and start a new one every day.) These tests were run in a wide variety of machine and networking configurations and represented a significant step toward the quality requirements to release the product. Anytime, the tests failed, we would stop any new check-ins and triage the problem until we found the guilty check-in. Either we removed that check-in or we put in a quick fix to remove the problem, and then re-ran the mainline stability test. Here was another real-time measure of reliability, everyday we could report that our mainline was at a known good quality level. This mainline stability test complimented our backlog metric. By running these rigorous tests we increased the number of defects found early in the product lifecycle. These defects were added to the backlog, so that they were quickly fixed before the next submittal to the mainline.

We now had a second measure that was brought into the Hour of Power. The number of days the mainline was 'open' versus 'closed'. If mainline had been closed for a while, a lab manager was assigned to ensure adequate resources were applied to solve the problem. The division manager also held his lab managers accountable for the number of times, their submission shut down the line. Each defect was root caused, and if the engineering team determined that the problem should have been found earlier, it was credited to that lab. It became well known which lab had caused the line the most and affected that lab manager's pay.

Now we could really measure 'quality submissions on time'. This became the new key metric for all of our managers. It was no longer acceptable to just get the functionality checked in, it also had to be at this know good quality rate. By changing our process to shut down the line whenever the release wasn't stable, we also changed the behavior of the entire organization because of peer pressure. When the line was shut down, no engineers could submit their functionality, no one wanted to cause this shut-down. Thus not only was this a measure for upper management, but project managers and engineers starting being driven by this key measurement point. As shown below, now instead of the defects increasing through functional complete we keep our defects under control throughout the release process. Since the release must always be at least at the mainline stability criteria.

Mainline stability test introduced a new problem - it often found problems that the engineers hadn't been able to find themselves, because they only had 1 or 2 systems to test their code on. The engineers demanded more test resources at their disposal, particularly more varieties of machine and networking configurations, in order to have a higher probability of finding the defect before mainline testing. This, of course, would be to everyone's benefit because the defect would be found before it was published on the mainline.

We couldn't afford each engineer to have a rigorous test environment, so to satisfy this need we set up a distributed test environment available to everyone. Each geographic location had one or more test centers with a variety of machine and network configurations usually a set of our lower cost machines and storage devices. For the large systems and more complex configurations, we centralized those resources in one location that we shared across the entire organization. From a web interface at the engineer's desk, they could schedule tests to be run in their local test center, as well as the complex tests and configurations in the central test center. Now engineers routinely run their new code through a series of 'pre-submittal' testing which runs the critical functional test and at least 10 hours of reliability in a small configuration. Today, we run over 100,000 tests a day in this distributed environment.

### **Testing**

Both of our real-time quality metrics depend heavily upon our ability to test correctly. So a key part of this improvement program also required us to improve our testing capability. Three separate efforts were undertaken to improve our testing. All 3 are driven by metrics, but at a lower level of the organization.

First, we found that many of our functionality tests were 'flaky', that is, they would pass in some circumstances and fail in others. So we began what we called a project to remove these flaky tests. Any time a test failed if we determined that the test was the problem, not the product, we removed the test from our regular test runs, calling it a 'known test failure'. We set a goal to have our known test failures track close to zero. This encouraged folks to either permanently remove the test or fix the test. Of course, the short-term expedient might be to remove the test, but that was precluded by our second metric: code coverage.

Measuring the percentage of functions called and branches taken during our functional test runs was the second test metric. For purposes of setting goals for this metric, we divided our code into "old" code, which had existed on previous versions of the release and was being modified for this release; and "new" code, modules either being added or significantly changed as part of this release. For "new" code, we asked teams to reach 100% functional coverage (all functions called) and 85% branch coverage (85% of all branches taken). In our environment, the development team was responsible for both the new product code and new test code. By having this requirement defined early in the release, teams actually designed their product code to both be functional and testable. Many of our teams actually achieved over 95% branch coverage because of how the code was designed. "Old code" was more problematic. Here we asked teams to use the code coverage tool to identify what they saw as high risk areas in their product code that were not being covered by their tests. For the release, we set a want requirement to increase branch code coverage by 15 percentage points over the previous release or to reach 85% coverage. The must goal was that code coverage not decrease over the base measurement made at the end of the last release. All teams met the must goal and over 35% met the want goal.

The third metric we employed was the defect escape rate. This measured the percent of defects that were found outside of the development team – by other development teams, by system test, and, of course, by customers. Our long term goal is that 90% of all defects are found by the development team through peer review, and testing. This metric was accompanied by a regular review of the defects escaping the team. By examining these defects and performing a root cause on the most common types, the team learned what particular improvements they needed to make in their development process. Often, this led to bringing in additional tests from teams that used their modules or directed the team to hot modules that might need to be redesigned or where the code coverage needed to be improved. Just as often, however, it led to other improvements – improvements in the peer review checklists, training in areas like HP-UX internals or locking mechanisms, or to do joint design reviews with other teams.

## **Results**

Our test metrics encouraged significant improvements in our tests. By incorporating these improved tests into our mainline testing, we improved our two key real-time quality metrics: defect backlog and mainline stability. We have added other real-time quality metrics over time, but these two were the key measures that allowed us to measure quality as easily as schedule and resources. They caused a significant change in the culture that truly made 'quality the number one priority'. In particular changing the key measure of results to 'quality submissions on time' resulted in a significant change in behavior. The 11i release of HP-UX surpassed our initial 2x improvement in quality goals. This release had 83% fewer defects found by customers in the first year of life compared to any of our previous releases, even though we sold it to more customers. Another measure of our success is defect density. The defect density in delivered defects/function point show that this product is in the top 1% of products as tracked by Capers-Jones [2]. The second release had reduced the defect density by another third.

The quality improvements were sufficient that we significantly improved our productivity. D. H. Brown in 2002 rated our version of UNIX #1 in all 5 categories of functionality. To quote their report [3]: "Clearly reflecting HP's increased investment in its Unix product line, HP-UX moves to the head of the class for UNIX operating systems functions. HP-UX occupies the top spot in every studied category, with a particularly strong lead in internet and web application services, and an impressive surge forward in the intensely competitive RAS (reliability, availability, supportability) category"

Just as important are the cultural changes that have occurred which will serve us well in the future as the organization changes. Trading off functionality to achieve quality and schedule are now the norms. The defect backlog metric and mainline stability metric continues to reinforce that we submit quality functionality on time. We're continuing in our efforts to improve our quality and productivity for our current release under development.

[1] The Critical Chain, Eliyahu M. Goldratt

[2] Applied Software Measurement, 2<sup>nd</sup> Edition, Capers Jones

[3] 2002 Unix Function Review, D.H. Brown Associates

[Back to top](#)

## Discover Your True Project Management Style

By Sabrina Jackson

### **Abstract**

Project management is vital in the world of IT; however, many placed in the role may not have a clear understanding of their management style. As important is the necessity of understanding the styles of those who the Project Manager is working with. This model not only helps with creating effective communication within a project team but assist with any relationships both professional and personal.

### **Introduction**

The project manager within the IT industry continues to face numerous challenges as he or she attempts to create a positive customer experience. High turn over continues to create challenges, impacting on cost and ultimately reduced customer satisfaction. There is also a huge disconnect between the perceptions of service provided, depending on whom you ask. Fifty six percent of in-house (IT) call centers believe their customers are “satisfied” while only 22% of customers agree<sup>1</sup>. Given that seventy five percent of (IT) contact centers use customer satisfaction as a key metric...well the dilemma is obvious.

Many project managers seek out and implement new technologies and strategies to address these challenges. Technology cannot modify attitudes, nor can it provide the agents, your front line access to customers, the tools necessary to effectively deal with your customers.

Project managers may have access to a lot of performance statistics on specific agent activity and performance, but again, they have not traditionally been provided with the tools necessary to evaluate agent attitudes and personalities during the hiring and evaluation process. Nor are they able to identify easily, areas for improvement with a clear path for change.

The people principle is not being fully nor effectively addressed which results in the following:

- High staff turnover
- Stressful work environment
- Compromised customer services
- Poor staff morale
- High training cost
- Poor staff communication
- Inadequate conflict management

More than just technology is needed to create an environment that fosters both customer and agent satisfaction. What is the “people principle”? The people principal recognizes that some issues are about human behavior and interaction. Simply stated it is an understanding that we are all “wired” differently and as a result we each perceive the world and respond to it and what it throws at us, in different ways.

If the “people principle” is addressed, the organization can create a seamless system that embraces differences in agents styles, understands the different styles of the customers, provides the project manager with a skill set and understanding of a staff member’s behavior and ultimately provides stellar customer service while empowering all staff to have some control of how they deal with various customers.

### **The People Principle**

Too often business leaders, managers, or supervisors want help in how to manage “people”. That is the wrong approach because you do not manage people, you manage a process. The problem most organizations have is they have not used the right information to develop a process all people can benefit from.

Most Project Managers have not developed an effective process because they have not considered the different types of people who may be working within the organization and/or the different types of people

who are the customer.

### **A Bit of History**

The history of character typing dates back to 460 BC when the Greek physician Hippocrates developed a model with four basic types of people; the Sanguine (buoyant, cheerful, hopeful, optimistic, sunny), the Choleric (angry, cantankerous, peevish, irate, testy), the Phlegmatic (languid, lethargic, listless, indifferent, passive), and the Melancholic (dejected, despondent, gloomy, morose). While these definitions are derived from Webster's Thesaurus rather than from Hippocrates, you can see that each refers to very different personality or temperament characteristics.

In 1920 the Swiss physician Carl Jung developed a model, which outlined 16 different types in his book Psychological Type. These 16 different types break down into 4 basic types of people. His work was a very interesting read but until the mother-daughter team of Myers-Briggs began to study the topic did we have an assessment to get us to our type. In 1956, the Myers-Briggs Type Indicator (MBTI) assessment was presented to the Educational Testing Service and is used worldwide.

Well, one of the major criticisms of the MBTI assessment is that it is difficult for people to remember their type or the type of others. In the 1960, David Kersey began to refine the MBTI and returned to classifying personalities into four different types. He uses the names of Greek gods to classify the 4 types within his publication Please Understand Me. The four names provided Apollo, Prometheus, Dionysus, and Epimetheus. These names are a bit difficult to pronounce and difficult to remember. Therefore, in 1978 Don Lowry who quickly recognized the potential to improve people's lives, careers and relationships. So he set about developing a fundamental and universal way to package the information into practical guidelines that could be understood and easily applied by both children and adults alike. The result is True Colors, which expands upon Kersey's four temperament types, and translates complicated personality and learning theory into "one of the easiest, most convenient ways of understanding and appreciating human behavior."

True Colors is a personality assessment tool developed in 1978 by Don Lowry. This model is based off of years of research on temperament typing dating back as far as 460 BC. The model uses 4 colors to describe the four basic types of personalities.

The True Colors program asks participants to identify their "color spectrum" using four cards that represent key personality types: Blue, Gold, Green or Orange. Each color has particular strengths and each analyzes, conceptualizes, understands, interacts and learns differently. But these differences, if not acknowledged and understood, can become barriers to interpersonal communication, making understanding between people of different types difficult.

It is imperative to understand the 4 basic types of people and design processes, structures, and trainings that meet them at their core. We should always work from the perspective that we must meet the 'esteem' needs of every customer, staff, supervisor, etc. Utilizing a model that assesses each type and provide strategies for creating the most harmonious work place for each is what True Colors was designed to do.

Blue- relationship driven people  
Gold-order & structure driven people  
Green- knowledge and information driven people  
Orange- excitement and hands-on now driven people

As you change your thinking about addressing these contact center issues. Begin to think about how to meet the esteem needs of the staff based on their True Color:

Blue- This staff person needs positive contact with other people. The Blue person must feel that they are helping others in their job. Create opportunities for them to talk about their challenges but more importantly their successes.

Gold- This staff person needs clear expectations of the job. They need to plan their work then work their plan Create opportunities for them to read manuals, procedures, and maintain clear lines of communication.

Green This staff person needs information. The work assigned must make logical sense to them. They must know where the information originated from and that is valid. Create opportunities for them to solve problem, research, and develop new ways of doing things.

Oranges- This staff person needs flexibility to just "DO". They are master negotiators and thrive in

the moment. Create opportunities for their skill to be showcased and turn to them to handle your crisis.

### **Benefits of implementing True Colors:**

The benefits of True Colors to the workplace can be enormous by positively affecting each of the areas that they are currently plagued with:

- **High Staff Turnover-** By implementing a True Colors program with the staff, they will learn the strengths they bring to the work place. If the managers, utilize the TC model when an employee first begins with the organization both will have a better understanding of how to improve job performance. Also, most staff leave a job when they feel their needs are not being met. The True Colors Model will help the management develop communication strategies that meet the needs of the staff specifically.
- **Stressful Work Environment-**By implementing a True Colors program with the staff, all employees will gain knowledge of how to identify those things that cause them stress. The model will provide each staff with ways to handle their individual responses to stress as well as how to reduce stress for their co-workers.
- **Compromised Customer Service-** By implementing a True Colors program with the staff, they will learn specific ways to identify the type of customer based on the style of communication. Once the type is identified the needs can quickly be addressed in a manner that is key for that specific color type. Examples:

Blue customers may become emotional while discussing their issue and may address how the issue is affecting “people”.

Green customers will be very concerned with facts and how to logically resolve the issue. They will have a suggestion of how to handle it before every calling.

Gold customers will be concerned with the “right thing to do”. They will quote things from the contract, etc.

Orange customers may become rude and want to complete the discussion RIGHT AWAY!!!

- **Poor Staff Morale-**By implementing a True Colors program with staff, they become immediately excited because the training session is fun, highly interactive, and immediately transferable to both their professional and personal worlds. Many employees feel like the organization values them after the session and praise them for the insight in selecting such an awesome model.
- **High Training Cost-**By implementing a True Colors program with the staff, the managers will gain key information that will assist with maintaining that staff person in their job. If employees are identified, trained, and supported utilizing the True Colors model several style issues will be avoided. The cost of training new staff is lowered if staff remains in their positions longer.
- **Poor Staff Communication-**By implementing a True Colors program with the staff, they will identify their personal communication style as well as the unique communication style of each “color”. The training session with the staff has an activity that immediately assist with improving communication among staff. This model increases the teambuilding skills of each employee.
- **Inadequate Conflict Management-**By implementing a True Colors program with the staff, they will understand their individual conflict management style and how to reduce conflict with the other color types. So, often people are in conflict and they think it is personal. After completing a True Colors session people understand that it is “type” difference and that it is not personal.

In order to address major challenged in the contact center we need to recognize the impact of the “people principle” and introduce soft skills training specifically geared towards providing supervisors and agents with the tools necessary to ensure an optimal customer experience. Technology always plays a key role as an enabler; however, until we address the human relationship obstacles we will continue to focus on the results rather than the cause.

### **References**

- [1] International True Colors Association [www.truecolorsassociation.org](http://www.truecolorsassociation.org)
- [2] True Colors, Inc [www.true-colors.com](http://www.true-colors.com)
- [3] Miscisin, M Showing Our True Colors CA 2001

## ANNOUNCEMENTS

### Mark Your Calendars - GLSEC 2009 in Grand Rapids

#### The Great Lakes Software Excellence Conference in Grand Rapids Has Set the Date for 2009!

**Conference Dates:** November 16-17

**Theme:** Compete Globally, Act Locally

[www.glsec.org](http://www.glsec.org)

#### Call for participation

If you are interested in participating through making a presentation or providing a tutorial, please email [Patrick Bailey](#) or call him (Phone: 616-526-7543).

[Back to top](#)

## CONFERENCES & WEBINARS

### [International Conference on Software Quality 2009](#)

**November 10-11, 2009**

**Pre-conference Tutorials on November 9, 2009**

**Hilton Northbrook**

**Northbrook (Chicago), Illinois**

#### **Theme: Controlling Software Before Software Controls You!**

Proactively addressing software quality through organizational design, product design, traditional approaches, and novel ideas.

#### **Conference Highlights**

- ASQ certification exams (held on Sunday, Nov. 8)
- Social event and networking
- Job information exchange
- On-site recruiters and resume assistance
- Keynote speakers – confirmed speaker: Bill Curtis
- Full-day and half-day tutorials (listed on reverse)
- Concurrent track sessions (partial list below)

Budget your travel and block your calendars now.

On-line registration opens June 15, 2009.

<http://www.asq-icsq.org>

#### **Tutorials (November 9, 2009)**

##### **Full Day Tutorials (8:00 a.m. – 5:00 p.m.)**

Managing Testing Projects, **Robin Goldsmith**

Requirements-Based Estimating and Scheduling Best Practices, **Steve Rakitin**

Unleash the Power of Inspections, **Rebecca Staton-Reinstein**

Using Root Cause Analysis Effectively, **Ed Weller**

Software Metrics and Analysis – A CSQE Mini-Refresher, **Linda Westfall**

### **Morning Tutorials (8:00 a.m. – 12:00 noon)**

Controlling the Internal Quality of IT Applications Before They Embarrass You, **Bill Curtis**

Agile Methods and Process Discipline, **Mark Paulk**

### **Afternoon Tutorials (1:00 p.m. – 5:00 p.m.)**

Economics of Software Quality Engineering, **Taz Daughtrey**

Best-In-Class Early Defect Detection and Defect Prevention to Achieve Measurable ROI, **Timothy G. Olson**

### **Conference Program (November 10-11, 2009)**

#### **Track Session Topics and Speakers** (confirmed as of May 1)

Your Quality Tools for Software, **Taz Daughtrey**

I Went to a Testing Conference and All They Talked About was Requirements, **Robin Goldsmith**

How to Write 'Lean and Mean' Requirements, **Timothy G. Olson**

Using Lean Principles and Process Models to Achieve Measurable Results, **Timothy G. Olson**

Best-In-Class Early Defect Detection and Defect Prevention to Achieve Measurable ROI, **Timothy G. Olson**

The Impact of Process Discipline on Personal Productivity and Quality, **Mark C. Paulk**

Management's Role in Achieving Predictable Software Development, **Steve Rakitin**

A Practical Approach for Building CMMI Process Performance Models, **Bob Stoddard**

Test Case Design through Requirements Modeling, **Louise Tamres**

Eight Steps to Effective Use Cases, **Linda Westfall**

Additional Topics: ITIL and Software Process Improvement

Establishing a Measurement Capability

Managing the Human Performance Side of Software and Change

[On-line registration opens on June 15, 2009](#)

For additional and ongoing information updates, please visit our web site at <http://www.asq-icsq.org>

[Back to top](#)

## **COLUMNS**

### **Get Certified**

#### **CSQE Quiz for 2<sup>nd</sup> Quarter 2009 By Linda Westfall**

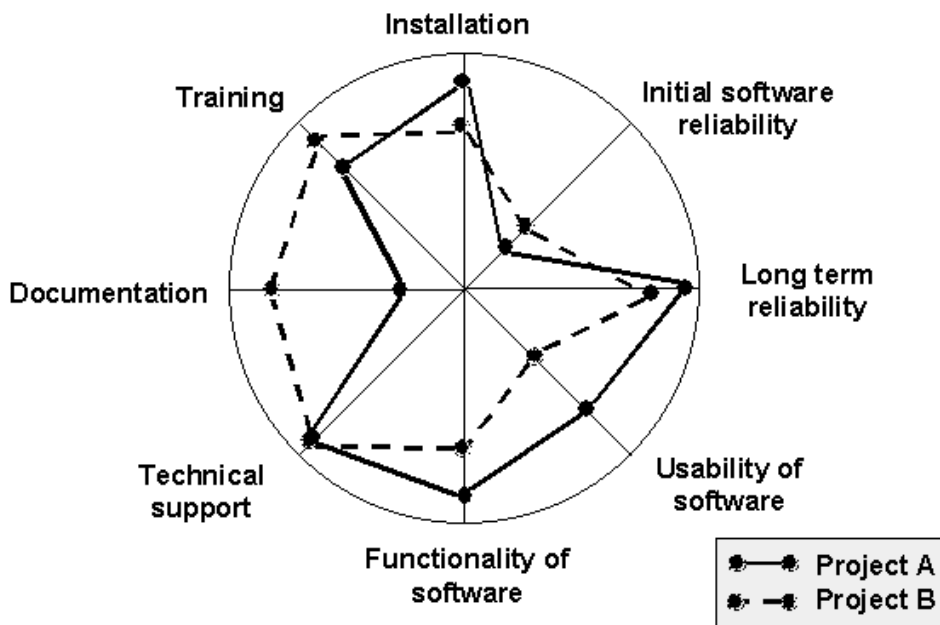
Note: The items in these quizzes are NOT from the past CSQE examinations NOR were they created as part of the CSQE exam development process.

1. Software Engineering Institute's Capability Maturity Model Integration staged representation includes a risk management process area at level:
  - A. 2.
  - B. 3.

- C. 4.
- D. 5.

2. One of the responsibilities of the client of an audit is to:
  - A. define to scope of the audit.
  - B. ensure the independence of the audit team.
  - C. implement corrective actions.
  - D. document the audit report.
  
3. A requirement that states, "The software must have a processing throughput of a minimum of 50,000 transactions per hour" is an example of which of the following types of requirements?
  - A. External interface requirement
  - B. Data requirement
  - C. Performance requirement
  - D. Design constraint requirement
  
4. The team that is developing software for an energy management and building automation system conducts an analysis of all the potential failure modes of that software and how those failures might adversely affect the safe operation of the building. This is an example of which of the following types of analysis?
  - A. Hazard analysis
  - B. Root cause analysis
  - C. Project risk analysis
  - D. Ergonomics analysis

**Customer Satisfaction Survey Results  
Comparison of Project A & B**



5. Based on the Kiviatt chart above, which of the following customer satisfaction areas would be the best candidate for Project A to use project B's current practices as their benchmark for improvement?
  - A. Technical support
  - B. Installation

- C. Initial software reliability
- D. Documentation

6. Which of the following would be an example of an appropriate test suspension criterion to define in a software system test plan?
- A. More than 20% of the system test cases are blocked
  - B. The arrival rate of problems reported by testers is increasing
  - C. 95% of the planned test cases have been executed and passed
  - D. No more than 10 non-closed majors exist, all with work arounds
7. Which of the following baselines would most likely include the released software executable?
- A. Development baseline
  - B. Functional baseline
  - C. Product baseline
  - D. Allocated baseline

### ***Answers to the Certified Software Quality Engineering Quiz***

1. **Answer B is correct.** Risk Management is a process area at Software Engineering Institute's Capability Maturity Model Integration staged representation maturity level 3. **CSQE Body of Knowledge Area:** I.C
2. **Answer A is correct.** The client, also called the initiator or customer of the audit, is the person or organization requesting the audit. The client determines the need for the audit, determines the audit's purpose and scope and provides the authority to initiate the audit. The audit management ensures the independence of the audit team. The auditee management ensures that corrective actions are implemented in the audited organization. The lead auditor documents the audit report. **CSQE Body of Knowledge Area:** II.C.2
3. **Answer C is correct.** Performance requirements define the levels of performance (for example, capacity, throughput, and response times) required from the software. External interface requirements define the information flow across shared interfaces to hardware, humans, other software applications, the operating system and file systems outside the boundaries of the software product being developed. Data requirements define the specific data items or data structures that must be included as part of the software product. Design constraint requirements define the limitation on the choices that the developers can make when implementing the software or system. **CSQE Body of Knowledge Area:** III.C.1
4. **Answer A is correct.** Hazard analysis examines the potential failure modes of a product and determines their safety impacts including potential injury to humans, damage to property or negative impacts to the environment. **CSQE Body of Knowledge Area:** IV.C.3
5. **Answer D is correct.** In this Kiviat chart, each "spoke" represents a customer satisfaction score with its value plotted on that spoke. The outer circle on a Kiviat chart is the highest value (goal). The plotted value for documentation shows a large difference between a poor satisfaction score for project A and a high satisfaction score for project B. Therefore, the documentation satisfaction areas would be the best candidate for Project A to use project B's current practices as their benchmark for improvement. For the technical support satisfaction areas (Answer A), both project A and B reported similar high satisfaction scores. For the installation satisfaction areas (Answer B), project B reported a lower satisfaction score than project A. For the initial software reliability satisfaction areas, even though project B scored higher than project A, both scores are low so Project B might not be the best project to benchmark for that satisfaction area. **CSQE Body of Knowledge Area:** V.B.3
6. **Answer A is correct.** Suspension criteria are the specific measurable criteria that when met indicates that the testing activities should be stopped until the resumption criteria are met. For example, if a certain number of critical defects are discovered or if a certain percentage of test cases are blocked, it may no longer be considered cost effective to continue the testing effort until development corrects the software. The arrival rate of problems reported by testers would be expected to increase, especially during the early stages of system testing. Having "95% of the planned test cases executed and passed"

and having “no more than 10 non-closed majors all with work arounds” are examples of possible exit criteria from system testing not a suspension criteria. In addition, if the 10 major problems all have work arounds, then testing can probably continue without significant impact. **CSQE Body of Knowledge Area: VI.B.2**

7. **Answer C is correct.** A product baseline is established when the software has been approved for distribution to operations and would most likely include the released software executable. The other three types of baselines are created before the released software executable is created. Developmental baselines are established when the software design specification, code or other internal development work products are approved as specified in the configuration management plans. The functional baseline is established when the system requirements are approved. Allocated baselines are established when the system requirements are allocated to one or more software components and the software requirements are approved. **CSQE Body of Knowledge Area: VII.B.1**

[Back to top](#)

## Medical Device Software

### New IEC Guidance for Networked Medical Devices IEC 80001

By David Walker

How can patient safety be assured when a medical device is connected to an IT network?

Although there are regulations and standards that address the manufacture of medical devices and use by clinical staff, there are currently no standards to addresses how medical devices may be connected to general-purpose IT networks to achieve safe and secure delivery of health care.

A new proposed international standard, International Electrotechnical Commission (IEC) 80001 - Application of Risk Management for IT-Networks Incorporating Medical Devices, outlines requirements of both manufacturers of medical devices and organizations responsible for the operation of the IT network over its entire life cycle. According to the standard, the manufacturer of a medical device that is intended to be incorporated into an IT network is responsible for providing information about the medical device sufficient to allow the medical facility to manage risk when incorporating the device into an IT network.

The AAMI (Association for the Advancement of Medical Instrumentation) Medical Device Software Standards Committee is currently providing US comment on the draft of this standard. Although distribution of the draft standard is prohibited, this short report uses the outlines of IEC 80001 to provide a narration of its content.

#### Scope

##### 1.1 Purpose

##### 1.2 Field of application

Generally, the standard applies to medical device manufacturers and responsible organizations, but not to a single manufacturer that specifies a complete medical device that includes its own network. But the standard would apply to modifications to the specified configuration of a medical device that includes its own network.

#### 2 Terms and Definitions

#### 3 Roles and responsibilities

##### 3.1 RESPONSIBLE ORGANIZATION

##### 3.2 TOP MANAGEMENT

##### 3.3 MEDICAL IT-NETWORK RISK MANAGER

##### 3.4 MEDICAL IT-NETWORK MAINTAINER

##### 3.5 MEDICAL DEVICE MANUFACTURER(S)

##### 3.6 Other providers of Information Technology

##### 3.7 RISK MANAGEMENT team

The responsible organization (organization accountable for the use and maintenance of the medical IT network) is required to apply risk management to the planning, installation, connection, operation, maintenance, and disconnection of the medical device IT network. The medical device manufacturer is required to provide, as part of the device specifications, instructions for implementing such connection, including but not limited to:

- Intended use of the medical device's connection to an IT network

- Required characteristics of the IT network incorporating the medical device
- Required configuration of the IT network incorporating the medical device
- Constraints governing the extent to which the IT network incorporating the medical device may be extended or modified after installation
- Technical specifications of the network connection of the medical device such as functional security specifications
- Intended information flow between the medical device and the IT network and, if relevant to the key properties (safety, effectiveness, data and system security, interoperability), the intended routing through the IT network

Also discussed are non medical devices used on the network. The responsible organization is required to maintain agreements where possible for providers of active non medical devices or related services. The agreements specify documentation requirements.

#### 4 Life cycle RISK MANAGEMENT in MEDICAL IT-NETWORKS

##### 4.1 Overview

##### 4.2 POLICY FOR RISK MANAGEMENT for INCORPORATING MEDICAL DEVICES

##### 4.3 RISK MANAGEMENT PROCESS

##### 4.4 Project Planning and Documentation

##### 4.5 RISK ANALYSIS

##### 4.6 RISK EVALUATION

##### 4.7 RISK CONTROL

##### 4.8 RESIDUAL RISK EVALUATION

##### 4.9 Reporting and approval

##### 4.10 CHANGE MANAGEMENT

##### 4.11 Monitoring

Section 4 provides a valuable discussion of the application of risk management principles to medical device networks based on ISO 14971 Application of risk management to medical devices.

#### 5 Document control

##### 5.1 Document control procedure

##### 5.2 MEDICAL IT-NETWORK RISK MANAGEMENT FILE

Section 5 has no surprises. The responsible organization must maintain a secure risk management file.

#### Annex A (Informative) Overview of RISK MANAGEMENT relationships

#### Annex B (Informative) Guidance on field of application

#### Annex C (Informative) Maintaining an IEC 80001 compliant network

#### Annex D (Informative) Causes of HAZARDS associated with MEDICAL IT-NETWORKS

#### Annex E (Informative) Use of wireless technologies in medical networks

#### Annex F (Informative) MEDICAL IT-NETWORK RISK MANAGEMENT template

#### Annex G (Informative) MEDICAL IT-NETWORK security

#### Annex H (Informative) References

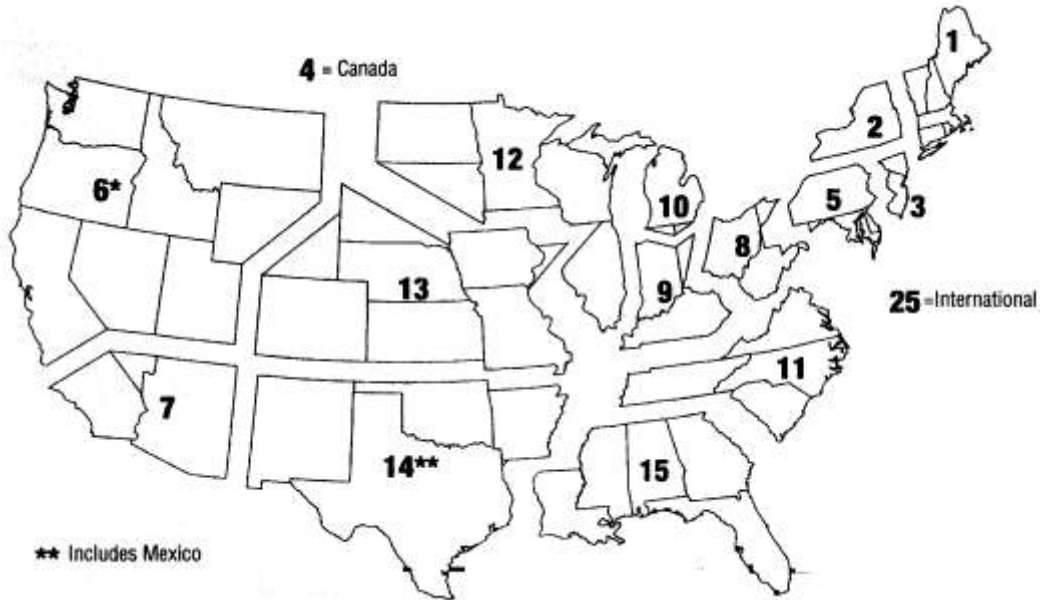
#### Bibliography

This report is intended to announce the new IEC 80001 and share some key elements of its content. More information regarding this new standard and its status will be published in future ASQ Software Division newsletter articles.

*David Walker is the immediate past chair of the ASQ Software Division. He represents ASQ's interests in medical device software standards development through membership on the AAMI Medical Device Software Standards Committee.*

[Back to top](#)

## FROM THE REGIONS



The following links will provide you with a snapshot of the latest activities in the regions.

### Region 1 (New England): Howie Dow

There are several opportunities to network, learn and meet with other ASQ members in the greater Boston area.

The Software Quality Group of New England (SQGNE) meets on the 2<sup>nd</sup> Wednesday of the month in Burlington, Massachusetts. Recent presentations covered scheduling, project management and testing. Upcoming topics include a discussion on test data management and how to decide to make or buy a test tool. The July meeting is reserved for the Annual Hot Topics Night. For details, head to the web site <http://www.swqual.com/SQGNE/main.html> Next quarter I'll provide a preview of events planned for the fall.

The presentations at the Boston Software Process Improvement Network (Boston SPIN) are varied. Topics for the past few months have included Politics of Meetings, Beautiful Teams and Building Coaching Skills. Boston SPIN meets in Bedford, Massachusetts on the 3<sup>rd</sup> Tuesday of the month and their web site is <http://boston-spin.org/>

[Back to top](#)

### Region 6 Pacific Northwest – Tom Gilchrist

If you are in the Seattle area on the third Thursday of every month (except December), The [Seattle Area Software Quality Assurance Group](#) (SASQAG) holds monthly public meetings in the Seattle area. SASQAG also supports certification and study groups. If you are in the area and want to attend, please look at [www.sasqag.org](http://www.sasqag.org) for upcoming events, directions, and meeting time.

October 26-28, 2009, the annual [Pacific Northwest Software Quality Conference](#) will be held (<http://www.pnsgc.org/>) in Portland, Oregon. The title of this year's theme, "Moving Quality Forward", is slightly ambiguous on purpose...consider these two possibilities:

- Moving the profession of software quality ahead - making it more of a recognized field, like civil engineering.
- Moving the validation and verification tasks further upstream in the software development life cycle and even moving them into other functions of the organization (e.g. finance, legal, service, sales, and marketing).

PNSQC has a new venue for PNSQC 2009 — the World Trade Center Portland.

This venue will offer an environment more conducive for intimate networking and interaction among attendees. The distinctive facility, located in the heart of downtown Portland, offers a breathtaking setting and spectacular backdrop overlooking the Willamette River and Tom McCall Waterfront Park. The facility offers convenient access to hotels, shopping, and transportation.

The Seattle SPIN (Software Process Improvement Network) is holding meetings on the first Tuesday, five months a year.. The organization is driven with a single, clear-cut goal in mind: change an organization in a way that improves that organization's ability to develop software. If you are interested in more information on SeaSpin, you can go to <http://www.seaspin.org>

**AAQG (Americas Aerospace Quality Group) News:** The Aerospace Software Quality Program Standard AS9115 has reached ballot. IAQG (International Aerospace Quality Group) Initiative #008 is sponsored by SAE to create this standard. Because of differing ballot periods for the AAQG (Americas Aerospace Quality Group), the European Aerospace Quality Group, (EAQG) and the Asia Pacific Aerospace Quality Group (APAQG), the ballot will remain open until July 23. The standard is a supplement for AS9100, the aerospace quality system standard to which organizations are registered in OASIS (On-line Aerospace Supplier Information System). There will be no separate recognition of compliance to AS9115 shown in OASIS at this time because there is no infrastructure established for the training of software qualified auditors at this time. AS9115 is an optional supplement for organizations choosing to use it. AS9115 offers greater granularity and specificity in the application of AS9100 for suppliers of software or software supported products. Acquirers of software who choose to adopt the supplement will have greater assurance that the principles of AS9100 are being applied effectively to software. Inquiries on this supplement can be submitted to Mike Kress, AAQG sector project lead, [michael.p.kress@boeing.com](mailto:michael.p.kress@boeing.com).

If you have information on local software quality and testing events in your area of Region 6, please send them to me for our events calendar Visit <http://www.tomgtomg.com/asq6> for information on events around Region 6.

Tom Gilchrist, Region 6 ASQ Software Division  
[tomg@tomgtomg.com](mailto:tomg@tomgtomg.com)

[Back to top](#)

### **Region 8 Ohio Valley – Greg Zimmerman**

Columbus welcomes a new SPIN! The Columbus SPIN is currently forming. The website is not up yet, but you can contact them via the Yahoo group, "columbus\_spin."

The Choices in IT conference is coming back to central Ohio. The 1-day event is being planned for June 2010. Look for the save-the-date reminder in a couple months. If you are interested in being involved with the conference planning, submitting a paper or presenting for the technical program, or volunteering at the event, please contact me directly.

Please send me any information for Region 8 members, requests for software quality related referrals, or if you would like to learn more about getting involved with the Software Division. You can reach me at: [gregz@appliedqualitysolutions.com](mailto:gregz@appliedqualitysolutions.com). A Region 8 links & info page can be found at <http://www.appliedqualitysolutions.com/asqreg8/>.

[Back to top](#)

### **Region 10 Michigan, Northeast Indiana & Northwest Ohio – Louise Tamres**

Region 10 covers Michigan and parts of northern Ohio and northwest Indiana. Most of the groups listed here are in southeastern Michigan. Please help me find out about other groups and events in your area. Send me an e-mail to the address shown at the end of this report.

Most professional societies and organizations go on hiatus for the summer months. Check the various groups' web sites to find out about upcoming events.

The Great Lakes SPIN meets the second Thursday of the month. Locations vary between Oakland University in Rochester, University of Michigan-Dearborn, and Schoolcraft College in Livonia. The GL-SPIN frequently sponsors CMMI and SPICE training programs. The May 14 meeting is on Recursive Nature of Requirements Development. Information about programs and events is available at [gl-spin.org](http://gl-spin.org).

Michigan Agile Enthusiasts is offering a joint meeting with the GL-SPIN on May 14, and Certified ScrumMaster course on June 14. More information is at [agileenthusiasts.ning.com](http://agileenthusiasts.ning.com).

The Southeastern Michigan Software Quality Assurance Association (SEMISQAA) is a local chapter of the Quality Assurance Institute (QAI). The group meets monthly on the second Tuesday, and it also sponsors QAI training sessions. As of this report spring programs have been announced. Get updated information at [semisqaa.org](http://semisqaa.org).

The Software Best Practices Conference was held in Detroit on April 30. Other cities will be hosting this conference. Information about upcoming events is at [www.itmpi.org/events](http://www.itmpi.org/events).

The 3<sup>rd</sup> Great Lakes Software Excellence Conference was held this past November in Grand Rapids, Michigan. To find out about the next conference (dates, call for papers), visit the conference web site periodically at [www.glsec.org](http://www.glsec.org).

ASQ chapters in southeastern Michigan provide programs primarily related to manufacturing. Any programs focusing on software and software quality will definitely be highlighted when available.

Are any software quality professionals in southeastern Michigan interested in CSQE exam preparation or other training courses? If either of these educational opportunities interest you (whether as an attendee or as an instructor), do let me know at [ltamres@computer.org](mailto:ltamres@computer.org) ([ltamres@computer.org](mailto:ltamres@computer.org))

[Back to top](#)

## Region 14 Southwest & Mexico – David Peercy

This column provides members with information on relevant association meetings, conferences and other events in Region 14 – Texas, New Mexico, Oklahoma, Arkansas and to some extent Mexico. If you have information that you would like to share with fellow **ASQ Software Division (ASD)** members, or you would like to get involved with the Division, contact: [depeerc@sandia.gov](mailto:depeerc@sandia.gov) or 505-844-7965. As always – if you or your section have something you'd like to include in this short quarterly report, please let me know. If you have a website that I can promote – please let me know. The 3QFY09 information is due by end-July 2009. Please check the ASQ site below for much information on multi-region activities and contacts.

<http://asqgroups.asq.org/Divisions/Soft/Newsletter/>

### TOPICS FOR DISCUSSION

#### (1) Conference Planning and Significant Actions

**a. WCSQ/Institute for Software Excellence 5/18-20/2009, Minneapolis, MN**

<http://wcqi.asq.org/>

**Theme: The Culture of Quality: Serving Customers, Organizations, and Communities**

**b. International Conference on Software Quality 2009 (ICSQ) 11/10-11/2009, Chicago, IL**

<http://www.espresso-labs.com/icsq2009/>

**Theme: Controlling Software Before Software Controls You!**

#### (2) Regional Information: Past and Upcoming

Thought some might be interested in just what ASQ Sections make up Region 14, where the Sections are generally located, and how many software division members are in each Section, as of about a year ago (March 14, 2008). See the table below.

Section Number	State	Primary Area Location	~ Number Members	Web Access (if exists)
1400	NM	Albuquerque	29	<a href="http://www.asq1400-abqnm.org">www.asq1400-abqnm.org</a>
1401	TX	El Paso	2	
1402	TX	Dallas	97	<a href="http://www.asqdallas.org/">www.asqdallas.org/</a>
1403	Mexico	Mexico City	10	
1404	TX	San Antonio	9	<a href="http://www.asqsa.org/">www.asqsa.org/</a>
1405	TX	Greater Houston	31	<a href="http://www.asghouston.org/">www.asghouston.org/</a>
1406	TX	Waco	1	<a href="http://www.asq1406.org/">www.asq1406.org/</a>
1407	AR	Little Rock	3	<a href="http://www.asq-centralark.org/">www.asq-centralark.org/</a>
1408	OK	Oklahoma City	9	<a href="http://www.asq.org/sections/mini-sites/1408">www.asq.org/sections/mini-sites/1408</a>
1409	OK	Tulsa	2	
1412	TX	Amarillo	2	<a href="http://www.asq-1412.org/">www.asq-1412.org/</a>
1414	TX	Austin	29	<a href="http://www.asqaustin.org/">www.asqaustin.org/</a>
1416	TX	Fort Worth	25	<a href="http://www.asqfortworth.org/">www.asqfortworth.org/</a>
1420	TX	Beaumont	1	<a href="http://www.asq1420.com/">www.asq1420.com/</a>
1421	AR	Warren	1	
1422	TX	Bay Area	3	<a href="http://www.asq.org/sections/mini-sites/1422">www.asq.org/sections/mini-sites/1422</a>
1425	Mexico	Monterrey	2	
1426	TX	North Central	4	
1429	Mexico	El Paso/Juarez	1	

Total Region 14 ASQ Software Division Membership = 261

### **(3) Newsletter Article – an executive summary on what I think “Quality” means**

I've been working on an article for the Software Division Newsletter on the term “Quality” and what it means. I may or may not get it done for this next newsletter, but it will eventually be done. I'd like to venture in a slightly different direction than is perhaps typical – yet a direction that I believe integrates most concepts of quality – in a conceptually simple way. Just to peak your interest – try this definition for Quality (any comments?):

**“Quality is the result of managing vulnerabilities to a targeted risk”**

[Back to top](#)