CHICAGO QUALITY ASSURANCE ASSOCIATION
Member of the QAI Global Community
CQAA

The Quality Journal

A publication of the Chicago Quality Assurance Association
113 McHenry Road, Suite 218 Buffalo Grove, IL 60089 • www.cqaa.org

In This Issue

CQAA
Thanks Our 2011
Sponsors

Diamond Sponsor
Microsoft

Lunch and Learn Series
SPR Companies

Message from the President

We have done it! We are delighted to have the first publication of our journal. It has been a goal of the CQAA Board to reach out to our members in as many different ways as possible. Everyone has their own favorite way that they like to keep informed.

Last year we added a member portal, made some significant changes to ongoing emails regarding events and information, and are active in LinkedIn.

This year, we thought we would go with a "classic" and make a newsletter available. The table of contents may change from time to time but our overall goal is to bring you information that is relevant to your career and professional growth and enable a channel to share information.

We'll try a few things and make adjustments along the way. After all isn't it important to adapt over time? We hope you enjoy this offering and we look forward to your candid feedback and, hopefully, your contribution to help us make this a viable and useful communications tool.

We have a history lesson on our industry in this issue which we think you will enjoy. We've gathered some topics that people want to know more about. And, you get a preview of upcoming events that should spark your interest.

Thank you to those who brought this to life, especially our editor, Mike Lawler. Now, get to a comfortable place and Read On…

Nancy Kastl, President CQAA

CQAA News

Save the Date

CQAA Spring Dinner

The Future of Software Projects: How Quality Assurance and Development Must Change

Pete DuPre, Chief Solutions Architect of Micro Focus

May 24, 2011

Meadows Club, Rolling Meadows

I recently heard the CEO of a financial institution state “We are a software company, masquerading as a bank.” Business is built on software. Today, if we aren’t building precisely what the users need, we’re on the road to irrelevance. As technologists, we have to start hitting the mark every time for our users, and stop settling for mediocrity.

Quality and Development Professionals are under ever-increasing pressure to perform better and deliver faster. In the face of rapid technology changes, vastly more complex systems, globalization, and new regulatory burdens, changes in the way we test and develop software are critical. In spite of all the methodologies and tools that have been thrown at the problem, the statistics tell a dismal story:

- Only 32% of software projects are considered successful
- 40% of a development team’s time is wasted on unproductive rework
- 70% of defects in production are introduced in the requirements stage

Consider; 32% software project success? Inexcusable - we have to fix this.

Trends such as Agile are gaining popularity, but are they delivering business value? Analysts encourage organizations to drive quality from the start of the software development lifecycle to streamline and reduce cost. This sounds logical, but is it really achievable? Also, we have been talking about getting better at requirements for 25 years – what’s new now?

Join us as we discuss:

1. Current state of software project success - industry trends
2. What's wrong with the traditional QA approach?
3. Latest innovation and concepts in modernized "Continuous Quality Assurance"

CQAA News (cont.)

CQAA Sponsor Program Highlights

CQAA offers a number of sponsorship opportunities which are outlined below. Sponsorships help fund a variety of events and activities and serve to minimize participation costs for CQAA members.

CQAA Annual Sponsorships

Service Provider Sponsorship

Visibility through your company logo, description, and website URL incorporated on the CQAA website as a featured sponsor and recognition of your company as a CQAA sponsor in a variety of ways including recognition at events and CQAA media resources.

Enterprise Sponsorship

Recognition of your company as a CQAA sponsor in a variety of ways including recognition at events and CQAA media resources along with discounts for all employees at a variety of QAI and CQAA events

CQAA Dinner Events Sponsorships

CQAA has two dinner events each year, one in the spring and one in the fall. The dinner event features an industry renowned keynote speaker. Sponsorships include **Speaker**, **Information Table**, and **Dinner Table.**

CQAA Contributors Program

Donations of funds of any amount are accepted as well the opportunity to provide a facility and refreshments for CQAA program events.

Certification Highlights

Professional certification does make a difference for many organizations. In addition to the actual accomplishment, those achieving this status gain valuable knowledge in the profession. The QAI Global Institute provides the certification and local chapters provide some level of support to prepare for certification.

CQAA actively supports and hosts study groups and Prep Courses in the Chicago area for the professional certifications listed below. **Study Groups** are provided at no cost to the attendee and are formed based upon the number of people interested and typically meet weekly for a period of several months. **Prep Courses** are delivered over a two day period. Location and cost of Prep Courses is shown below. Go to www.cqaa.org/certifications for more information and to register.

Certified Software Tester (CSTE)
June 6-7, 2011

Certified Software Quality Analyst (CSQA)
June 8-9, 2011

DePaul University O'Hare Campus - Chicago, IL
8770 W. Bryn Mawr Avenue Room 205, Chicago, Il 60631
Cost for each Certification Prep Course: CQAA Members: $450, Non Members: $500

Sponsor Showcase

Microsoft is a Diamond Sponsor of the Chicago Quality Assurance Association. In addition to financial support, Microsoft also provides facilities at the Downers Grove and Chicago offices for CQAA meetings and events. We sincerely appreciate their ongoing support.

In this issue, they are providing an overview of their Visual Studio 2010 software.

Take quality assurance to the next level with the Microsoft Visual Studio 2010. This breakthrough tester-developer collaboration experience supports complete plan-test-track workflows and full traceability of requirements and user stories.

Visual Studio 2010 Speeds Development with Tools for:

- Real-time visibility into project quality and status
- Increased collaboration through common interface
- Concurrent coding and debugging by incorporating test early
- Eradicating “no repro” issues with rich actionable bugs
- Empowering manual testing and automating rote tasks
- Provisioning virtual labs for efficient test and build
- Eliminating waste with agile project management

Help eliminate waste and accelerate collaboration in your development and test processes. Visual Studio 2010 integrates Test Tools into the development environment, unites team workflow, and can help save time and money.

Get Proof at http://bit.ly/VSTestPro

CQAA February 10th Lunch and Learn featured:

Meeting New Business Challenges With Dynamic Quality Assurance
Angela Dugan, Microsoft

For a copy of Angela’s presentation slides go to http://www.cqaa.org/default.asp?contentID=22

The Quality Journal
March 2011

Industry Perspective

Industry Perspectives are offered by recognized professionals for the purpose of sharing information with our members on approaches, techniques, or information that may be useful to professional growth and development. The content is the property of the author and any reproduction of this content outside of this publication is not permitted. Anyone interested in this content or the information conveyed in this article should contact the author directly.

This article is the first of three that detail the history of Software Quality. It is not often that professionals are allowed to understand the history of the career path they have chosen. We feel that Mr. Rakitin has captured and presented the history of our industry rather well.

While the article is somewhat lengthy, we chose to publish it in its entirety to allow the reader to benefit from a cohesive message.

About the author…
Steve has over 35 years experience as a software engineer and software quality manager. He frequently speaks on topics related to software development and software quality at conferences worldwide. He's published several papers on the subject of software quality and a written a book titled Software Verification & Validation for Practitioners and Managers. As President of Software Quality Consulting, Inc., he works with clients who are interested in improving the predictability of their development process and the quality of their products. His company's website is www.swqual.com E-mail him at steve@swqual.com

Software Quality Assurance Turns 50

A critical look at the state of the profession

Part 1 of 3 - History and Evolution

Steven R. Rakitin, President Software Quality Consulting, Inc.

Software Quality Assurance (SQA) was used for the first time on a software development project about 50 years ago. This article is the first of three where I plan to take a critical look at the state of the SQA profession as a way to recognize the significance of this milestone. In this month's installment, I discuss the history and evolution of SQA. In subsequent articles, I will discuss successes and failures, and the future of SQA.

During the past half-century, the software industry has gone through dramatic changes. Today, software is an integral part of daily life. Many software-based products that were beyond one's wildest imagination five decades ago are now commonplace. The explosion of the Internet, digital gadgetry, and cheap hardware has resulted in software finding its way into millions of products and services, many of which are **safety-critical or mission-critical.** Today software plays an integral role in most every major segment of the global economy – including energy, transportation, healthcare, banking and finance, defense, and space exploration.

Industry Perspective (cont.)

Every day, we become more and more dependent on software. As I have stated many times, **all software is inherently defective**. Unless you live in a third world country, a typical day involves using a significant amount of software - either implicitly (as in software embedded in a product) or explicitly (as in software applications). For people living in third world countries, there are on-going efforts (some controversial) aimed at providing children with inexpensive laptops so they can connect to global virtual communities.

We begin this critical review by starting at the beginning…

In the beginning…

In the late 1950's, software first began to find its way into systems procured by US government agencies such as the Census Bureau and the Dept. of Defense (DoD). Not surprisingly, these projects were always behind schedule, over budget, and suffered from both technical and management problems. Frequently, software did not work as intended and many projects were cancelled before anything was delivered.

Software development contractors often gave overly optimistic assessments of the software development status to managers - the origin of "We're 90% done!" Managers were frequently unaware of schedule, budget, and technical problems until very late into the program – when they were often unable to understand them, assess their impact or do anything to change the situation.

The Atlas Missile was the first operational intercontinental ballistic missile in America's nuclear arsenal and marked the beginning of the US space program. The **Atlas Missile Program** was one of the first software projects to try to address these problems. The program manager hired an "independent software tester" to "perform additional, unbiased testing of the software". [1] The program manager hoped to get a timelier, accurate and objective technical assessment of the project's status by employing someone **independent** of the software development contractor.

Around the same time, the first independent test team on a large software project was formed and led by **Jerry Weinberg** on **Project Mercury** - the first US manned space flight program.

During the 1960's, the role of the independent software test team evolved from just focusing on testing to focusing on the entire software development life cycle. This role became known as **Independent Verification & Validation (IV&V).**

Today, IV&V is a critical function contractually required on most large, mission-critical projects for US government agencies including DoD, NASA, FAA, HUD, EPA and DEA. The set of tasks performed by IV&V contractors is comprehensive and spans procurement, development and deployment.

Much data has been collected to support the assertion that projects with IV&V perform much better than similar projects without IV&V. [2], [3] As a result of this data, NASA now requires IV&V to be applied on applicable NASA projects. [4]

Industry Perspective (cont.)

Much of the success of IV&V is attributable to the fact that IV&V contractors are completely **independent** of the software development organization. Working for and reporting to the procuring entity, IV&V contractors **provide an unbiased, objective technical and managerial assessment** of a project. As a result, the procuring entity is in a much better position to identify and resolve issues that could otherwise easily be overlooked (intentionally or unintentionally) by the software development contractor. Raising these issues in a timely manner ensures that they are more likely to be resolved and not affect the project.

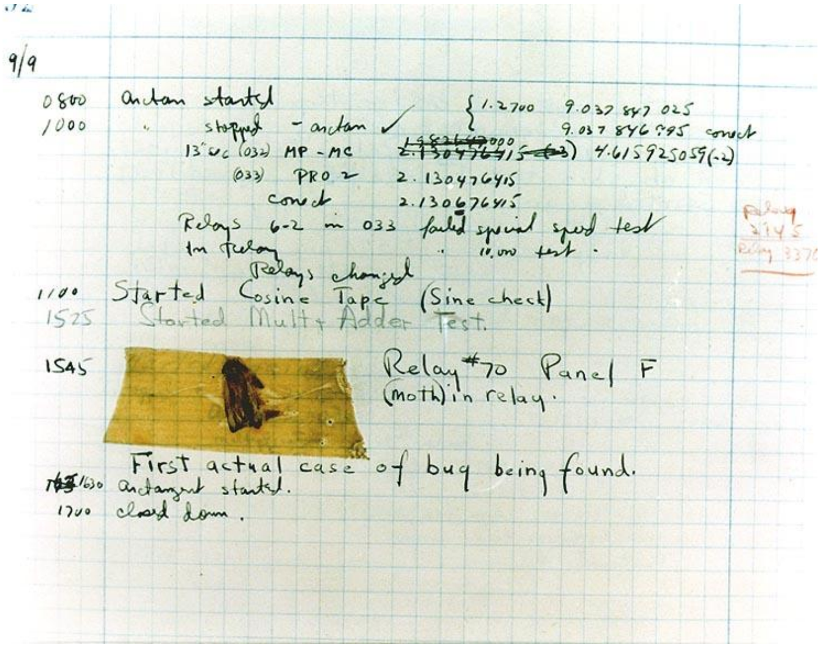
When did it become Software Engineering?

Back in the day (1940s-50s), people who worked with computers were mathematicians and were called **programmers** or **data processing specialists**. Computers of the time, such as the ENIAC, were very clunky and were usually “programmed” by re-wiring patch panels that changed the sequence of operations the computer’s electromechanical relays performed.

Back then, designing hardware was much more prestigious and as a result, hardware engineers were mostly male. Women, most of whom were mathematicians like Admiral Grace Hopper, often performed the “lowly” job of programming the computers.

Grace Hopper was a brilliant mathematician who worked at Harvard on the Mark II Aiken Relay Calculator – an early analog computer built from hundreds of electromechanical relays. She liked to tell a story about an event that occurred in late summer of 1947. It was before the advent of air conditioning so the windows in the computer lab were open most of the time.

A technician solved a problem with the Mark II machine by pulling an actual insect (a moth) out from between the contacts of one of its relays. Admiral Hopper taped the moth to her lab notebook on September 9, 1947 and made the entry shown to the right:



9/9

0800 Arctan started

1000 " stopped - arctan ✓ {1.2700 9.037 847 025
9.037 846 995 correct

13" uc (032) MP - MC ~~1.982147000~~ ~~2.130476415~~(-3)) 4.615925059(-2)

(033) PRO 2 2.130476415

correct 2.130676415

Relays 6-2 in 033 failed special speed test
in relay " 11,000 test.

Relays changed

1100 Started Cosine Tape (Sine check)

1525 Started Mult + Adder Test.

1545 Relay #70 Panel F
(moth) in relay.

First actual case of bug being found.

1630 Antangent started.

1700 closed down.

Relay 2145
Relay 3370

In the fall of 1968 and again in 1969, the NATO Science Committee sponsored technical conferences to bring together experts to discuss problems with the development of software. The conference organizers used the term **software engineering** as a way to provoke interest and discussion. The term wasn’t used prior to the conference and the birth of the software engineering profession is generally recognized to coincide with these conferences.

Industry Perspective (cont.)

As observed by one of the conference attendees:

> "Although the term was not in general use at that time, its adoption for the titles of these conferences was deliberately provocative. As a result, the conferences played a major role in gaining general acceptance, perhaps even premature, for the term. The motivation for these conferences was that the computer industry at large was having a great deal of trouble in producing large and complex software systems. (Does that sound like dêja vu all over again?)" [5]

The participants at the conference represented computer hardware manufacturers, computer users, representatives from a few small software development companies, and academia. Most of the people attending the NATO Conference acknowledged that there were many problems associated with producing large, complex software systems. (Note that systems called "large" and "complex" in 1968 terms would be considered "small" and "trivial" by today's standards). In the summary of the NATO Science Committee report [6], it states:

> "... the report also contains sections reporting on discussions, which will be of interest to a much wider audience. This holds for subjects like the:
>
> - problems of achieving sufficient reliability in the data systems which are becoming increasingly integrated into the central activities of modern society
> - difficulties of meeting schedules and specifications on large software projects
> - education of software (or data systems) engineers"

The problems observed in 1968 are striking in how similar they are to problems we have today.

The Emergence of SQA

The 1968 NATO report also used the term **Software Quality Assurance**. During the conference, the participants discussed the issue of SQA and raised several very interesting questions: [6]

- Is software quality assurance done by an independently reporting agency representing the interests of the eventual user?
- Is the product tested to ensure that it is the most useful for the customer in addition to matching functional specifications?
- Do software quality assurance test programs undergo the same production cycle and method (except Q/A) as the software they test? Are they defined and constructed concurrently with the software?
- Is at least one person engaged in software quality assurance for every ten engaged in its fabrication?
- Are there tests for overall system performance as well as for components?
- Are software quality assurance tests a part of the general hardware acceptance test on the customer's machine before it leaves the factory'?
- Can software field release be held up if these tests are not passed?
- Do the tests include a system logic exerciser?

Industry Perspective (cont.)

- Are tests provided to ensure matching of computational results with those of other equipment?
- Is this test library applied upon issuance of each modification of the software system?
- Is each customer's system tape tested on the software production machine for a sufficient period of time, where feasible?

These issues, raised more than 40 years ago, still resonate today.

During the 1970's, software development activity expanded to commercial companies. These companies experienced the same poor results that US government agencies had seen a decade earlier. These companies had difficulty delivering software within the constraints of schedule, budget, and quality. Many projects undertaken in the 1980's and 90's were disasters. Several projects failed to deliver anything. The few projects that did deliver something were significantly over budget and years behind original schedules and delivered software of such poor quality that it was often unusable.

In the 1980's, the software industry experienced what became known as the "software crisis" – the point in time when spending on software maintenance exceeded spending on creating new software products. The advent of the "software crisis" brought with it a host of changes - not the least of which was the emergence of SQA as a critical function to be performed on software development projects. Initially, SQA was viewed as sort of an internal IV&V function.

Drawing on its roots in IV&V, SQA evolved into an effective tool that software development companies have used to help identify quality problems earlier in the development process. While SQA was viewed as the "poor stepchild" of software development, many enlightened managers of the day saw measurable benefit from integrating SQA into the software development process.

By the 1990's, many software companies had SQA functions within their organizations. Yet, high profile software failures continued to occur. (see [7, 8, 9]) Was SQA not living up to expectations? Hard to say. But there were several differences in the nature of software being developed during this time that are worth noting:

- Complexity of software developed during the 90's increased significantly.
- Competitive business pressures also increased significantly.
- Software was being used in many new areas – especially areas that were life threatening.
- Many people working in SQA received little formal training in SQA. SQA engineers were expected to learn their craft primarily from on-the-job training.
- Universities failed to recognize that SQA is a legitimate discipline unto itself and that it requires specialized training.

Industry Perspective (cont.)

The Bottom Line...

Edsger W. Dijkstra was one of the distinguished participants at the 1968 NATO Conference. He made an incredibly insightful remark at this conference that, while important in 1968, is even more important today:

> "The dissemination of knowledge is of obvious value — the massive dissemination of error-loaded software is frightening." [6]

'Til next time...

References

1. Nelson, J. G., "Software Testing in Computer-Driven Systems", in *Software Quality Management*, ed. Fisher, Matthew J., and Cooper, John D., Petrocelli Books, 1979
2. Arthur, J. D. and Nance, R. E., "Verification and Validation Without Independence: A Recipe for Failure", Proc. 2000 Winter Simulation Conference, Orlando FL, 2000.
3. Wallace, D. R., and Fuji, R. U., "Software Verification and Validation: Its Role in Computer Assurance and Its Relationship with Software Project Management Standards", National Institute of Standards and Technology, Special Publication 500-165, May 1989
4. NASA Policy Directive NPD 8730.4A, Effective August 1, 2001.
5. Robert M. McClure, Introduction to the 1968 NATO Software Engineering Conference
6. Software Engineering, Report on a conference sponsored by the NATO Science Committee Garmisch, Germany, October 7-11, 1968
7. Glass, R., Software Runaways: Lessons Learned from Massive Software Project Failures, Prentice-Hall PTR, 1997
8. Weiner, L., Digital Woes: Why We Should Not Depend On Software, Addison-Wesley, 1993.
9. Johnson, J., "Chaos: The Dollar Drain of IT Project Failures", *Application Development Trends*, January 1995, p. 41-47.

IV&V Resources

- Lewis, Robert O., Independent verification and validation: a life cycle engineering process for quality software, Wiley-IEEE, 1992
- Schulmeyer, C. G. and Mackenzie, G. R., Verification and Validation of Modern Software-Intensive Systems, Prentice Hall-PTR, 2000.
- Rakitin, S., Software Verification and Validation for Practitioners and Managers, 2nd edition, Artech House 2001.
- Arthur, J., et. al., **"**Evaluating the Effectiveness of Independent Verification & Validation", IEEE Computer, October 1999.
- NASA IV&V Overview

Hot Topics

Software Quality and Testing Professionals face a number of issues as they attempt to provide value to their project teams and organizations. This section will present some issues shared with the CQAA Newsletter Editors. We have gathered a few questions/issues to start and will continue gathering ideas in the future. These topics will be addressed in future issues with a recap, as shown below, or perhaps a full article.

Should QA have the authority to stop a project from going live?

How do I know I am using the right methodology? Why does it matter?

Who decides "best practice?"

What is "just enough" testing?

Help us help each other. *Have you solved one of the questions above or are you facing a similar issue? Please share what worked for you.*

Send us topics you would like addressed in future issues or let us know if you wish to contribute to the article through an interview or by providing some content. Contact us at Editor@cqaa.org

CQAA LinkedIn Topic of the Month

A recent topic presented to the CQAA Linked-In community asked "**What are the benefits and challenges of offshoring testing?"** Listed below is a recap of the discussion.

Challenges

- Immediate benefits are cost-savings but those benefits may be short-sighted
- Offshore, like any outsourced service, will lack familiarity with the business. Unclear or non-existent requirements will result in poorly designed test cases and an unsatisfactory testing effort
- Typical talent pool is junior level, hence the low hourly bill rate and are migratory, moving to other firms based upon the best and newest available opportunities

Possible Solutions

- Ensure that you have a strong Offshore Team Lead who is well versed in process and is trained and capable of working with the organization's processes.
- Set up primary and backup resources for each area that needs to be covered and cross train the team to offset the impact of the personnel that leave during the project.
- Depending on budget constraints, make sure as many of the offshore team visits the team in the United States and vice versa to establish a rapport and ensure understanding of the work to be done. At a minimum, make sure the Leads and key team members achieve this.
- Ensure that the offshore team knows that they must communicate any issues or confusion as soon as known to minimize project impact.

The Quality Journal

March 2011

CQAA Mission

The Chicago Quality Assurance Association, the Chicago Chapter of the QAI Global Institute is a nonprofit professional organization that was established in 1984 to promote quality principles and practices within Chicago-area companies. CQAA provides a forum for information professionals to present and discuss quality and process management within information systems, technology, and services.

Objectives

- Provide a variety of educational opportunities in the Chicago area for quality professionals and other advocates of quality.
- Facilitate networking and the exchange of ideas among quality, process management and information technology professionals.
- Sponsor presentations on quality and related topics by experts and by members.
- Foster professional certifications through access to examinations held in the Chicago area.
- Provide an opportunity to earn recertification credits towards professional certifications through membership and attendance at educational programs.
- Maintain lines of communication with other professional associations and foster cooperative activities of common interest.

Upcoming CQAA Events

April 20	**Distributed Teams Lunch & Learn**
April 26	**Regression Testing Program**
May 24	**Spring Dinner Event on Future of Software Projects**
June 6-7	**CSTE Certification Prep Course**
June 8-9	**CSQA Certification Prep Course**

Chapter Leadership

CQAA is served by a self-perpetuating Board of Directors that meets to plan, implement and review programs and functions. To ensure that the CQAA Board remains strong, the directors have adopted a succession plan. All board positions have detailed descriptions that identify the responsibilities of the Board Members. Board positions that become vacant will be posted on the CQAA website and applications will be accepted at that time.

Board Members

Sandy Doyle, Secretary
Cindy Glaser, Membership
Nancy Kastl, President
Kim MacDonald, Certifications
Fabrizio Stortoni, Programs
Sara Thomas, Treasurer

Mike Lawler, Journal Editor