



An e-newsletter published by
Software Quality Consulting, Inc.

Jan 2011
Vol. 8 No. 1 [Text-only version]

Dear <name>,

Welcome to **Food for Thought**TM, an e-newsletter from [Software Quality Consulting](#). I've created free subscriptions for my valued business contacts. If you find this newsletter informative, I encourage you to continue reading. Feel free to pass this newsletter along to colleagues by using this [Forward Email](#) link. If you've received this newsletter from a colleague and would like to subscribe, please use this [Enter New Subscription](#) link. If you don't wish to receive this newsletter, use the [SafeUnSubscribe](#) link at the bottom of this newsletter, and you won't be bothered again.

Note: If you have trouble reading this e-newsletter or accessing any of the links, [click here](#) to open the newsletter in a web browser.

Your continued feedback on this newsletter is most welcome. Please send your comments to steve@swqual.com

In This Issue

In [This Month's Topic](#) I discuss the basics of SQA...

Regular features to look for each month are:

- [Monthly Morsels](#)
Hints, tips, techniques, and references related to this month's topic

This Month's Topic

SQA in 2011 Back to Basics

In past e-newsletters, we discussed the evolution of Software Quality Assurance over the past 50 years. Today, as we begin the second decade of the 21st century, we re-examine what we call "SQA" and review the basic set of SQA practices...

What is SQA?

As we saw in my **June 2009 e-newsletter**, Internal IV&V evolved into what we commonly refer to as "SQA". However, if you talk to people who work in "SQA" at software development companies, you will see significant differences in the roles and responsibilities of the SQA function.

New software engineering standards, such as **IEEE-12207:2008 Software Life Cycle Processes** have helped clarify the role of SQA in a software development organization. IEEE-12207 defines **quality assurance** as:

"all the planned and systematic activities implemented within the quality system, and demonstrated as needed, to provide adequate confidence that an entity will fulfill requirements for quality." [1]

Currently, several IEEE Software Engineering standards are being harmonized with IEEE-12007, including **IEEE-730 Software Quality Assurance**. IEEE-12207 further identifies two very distinct roles that an SQA Group can perform – one is a **product assurance role** and the other is a **process assurance role**.

The **product assurance role** is defined as providing assurance that:

- all the plans required by the contract are documented, comply with the contract, are mutually consistent, and are being executed as required.
- software products and related documentation comply with the contract and adhere to the plans.
- the software products have fully satisfied their contractual requirements and are acceptable to the acquirer

As you can see from the definitions, the language used in IEEE-12207 is based on a supplier-acquirer business model where there is a contract that defines the business relationship between supplier and acquirer as well as the product being developed. Many organizations use a different business model, one where there isn't a contract per se nor a one-to-one relationship between supplier and acquirer. The IEEE-12207 standard still applies in those situations. "Contracts" become procedures and requirements and "acquirers" become users and customers.

Some key points from these definitions are that in the **product assurance role**, SQA is responsible to provide assurance that the:

- required plans (such as a Software Development Plan, a Software Test Plan, etc.) are in place and either comply with a contract or perhaps with your organization's internal **Software Development Life Cycle** (SDLC) Model.
- software product and its documentation comply with the contract (or procedures and requirements) and is developed according to those plans.
- software meets all requirements and is accepted by the acquirer (or user and customer).

In the **product assurance role**, SQA resources are responsible for performing the tasks listed above. SQA would have a hands-on role in performing one or more levels of testing that traditionally has been used to demonstrate whether software meets all requirements and is accepted by the acquirer or user as the case may be...

The **process assurance role** is defined as providing assurance that the:

- software life cycle processes (supply, development, operation, maintenance, and support processes including quality assurance) employed for the project comply with the contract and adhere to the plans.
- internal software engineering practices, development environment, test environment, and libraries comply with the contract (or procedures).
- applicable prime-contract requirements are passed down to the subcontractor, and that the subcontractor's software products satisfy prime-contract requirements.
- Acquirer (user or customer) and other parties are provided the required support and cooperation in accordance with the contract, negotiations, and plans.
- software product and process measurements are in accordance with established standards and procedures.
- staff assigned have the skill and knowledge needed to meet the requirements of the project and receive any necessary training.

The **process assurance role** has a different focus. Here, SQA is responsible for assuring that defined SDLC processes are followed. This is most often done through audits and reviews of documents and records. In this role, SQA assures that the process is followed. SQA is not responsible for testing, but rather, for assuring that testing is performed as defined in the SDLC, test plans and, if applicable, the contract.

Typically, the **process assurance role** is required for CMMi organizations and in cases where software development is regulated.

What is your SQA Role?

Now that we have discussed the two roles for SQA, my question to you is – which of these two roles does the SQA function at your company perform – **product assurance** or **process assurance** or **both**?

Having worked with over 150 software development organizations, I can tell you that there are still many organizations where the SQA role is limited to only testing. It's hard to believe that after 50 years of experience with SQA, there are still organizations that have yet to realize the benefits of having a true SQA function – one that is fully involved in the entire software development lifecycle – from requirements through product retirement.

The Basics of SQA

My definition of SQA is independent of the product or process assurance role discussed above and applies to both roles. I believe SQA is:

- A planned set of tasks, activities, and actions performed independently of the software development organization that provides Management with objective, timely, and factual information that can be used to make appropriate business decisions regarding a software product.

Given this definition, I've identified several key areas that SQA must be involved in to improve effectiveness. These key areas represent the bare minimum that any SQA function should perform:

- **Involvement in the entire Software Development Life Cycle (SDLC)**

To be effective, SQA must be involved throughout the entire SDLC process since, at each phase of the SDLC, there are opportunities for defects to be introduced into the product (documentation and software). The SDLC must be written down and SQA's role at each phase must be clearly defined and supported by Management. SQA must be very familiar with the product (domain knowledge) and the process to ensure maximum benefit and highest quality.

- **Requirements**

Requirements are the foundation of every software development project. Without clearly defined requirements, developers improvise and meaningful testing is not possible. SQA must be involved in reviewing and analyzing requirements from the earliest stages of a project as well as assessing changes to requirements throughout a project. In this role, SQA's focus should be on assessing testability – is every requirement testable as written?

In addition, SQA should be involved in creating a Requirements Trace Matrix (RTM). Tracing requirements to tests is a key tool for ensuring that all requirements are adequately covered by test cases.

- **Reviews and Audits**

Reviews and audits are key tools for finding defects and assessing compliance. SQA needs to participate in technical reviews – especially in the early stages of projects when requirements are evolving. SQA needs to be advocates for effective peer reviews and where appropriate, formal inspections.

As stated by David Parnas,

“Early inspection of a document that states system requirements can help insure that the correct system is

built." [4]

Read more about the role of inspections from an SQA perspective.

SQA also needs to perform meaningful audits as appropriate. Audits can help identify weaknesses in the SDLC or areas where additional training may be required.

- **Measurements and Metrics**

Measurements and metrics are essential if organizations are to improve since you can't improve what you can't measure. Measurements and metrics can be applied to the product as well as the process used to develop the product. SQA needs to help define appropriate measurements and help present metrics in a coherent and unbiased manner so Management can take appropriate corrective actions.

- **Testing**

In a **process assurance role**, SQA needs to be able to review testing performed by others to assess completeness and conformance to approved plans. This role clearly requires familiarity with the testing process.

In the **product assurance role**, SQA performs most of the testing. For many organizations, the SQA function is limited to testing even though we know that:

testing is necessary but not sufficient to produce quality software.

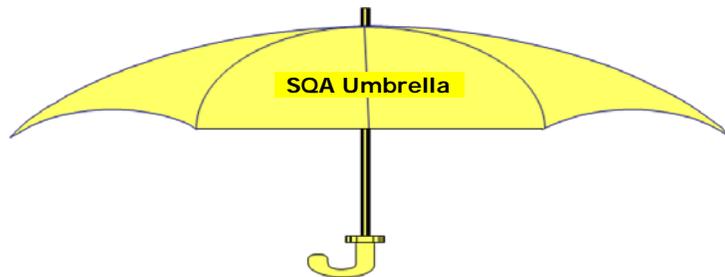
Data from Capers Jones [2] has indicated that the more levels of testing that are performed, the higher the **defect removal efficiency**. Defect removal efficiency is one example of a metric that measures the percentage of known defects found by your users or customers.

Number of Testing Stages	Percent of Effort Devoted to Testing	Cumulative Defect Removal Efficiency
1 testing stage	10%	50%
2 testing stages	15%	60%
3 testing stages	20%	70%
4 testing stages	25%	75%
5 testing stages	30%	80%
6 testing stages	33%	85%
7 testing stages	36%	87%
8 testing stages	39%	90%
9 testing stages	42%	92%
10 testing stages	45%	94%
11 testing stages	48%	96%
12 testing stages	52%	98%
13 testing stages	55%	99%
14 testing stages	58%	99.9%
15 testing stages	61%	99.99%
16 testing stages	64%	99.999%
17 testing stages	67%	99.9999%
18 testing stages	70%	99.99999%

The average for all software development organizations is 6 testing stages that results in a defect removal efficiency of 85%. This means that of all known defects, customers encounter 15% of them.

The SQA Umbrella

Dr. Roger Pressman [5] coined the term **SQA Umbrella** to convey the concept that SQA is not one-dimensional but rather a collection of complimentary tasks and activities that provide significant benefits to organizations that embrace quality. Here is my view of the SQA Umbrella:



Audits	Product Assessment	Risk Assessment
Triage	Process Assessment	Testing
Peer Reviews	Domain Knowledge	Test Management
Formal Inspections	Quality Management	Test Automation
Verification	Configuration Management	Defect Tracking
Validation	Project Retrospectives	Measurements and Metrics
Project Tracking	Requirements Traceability	Release Process
Root Cause Analysis	Estimating and Scheduling	Documentation

In Summary...

Overall software quality has to improve in order for the software industry to continue to grow and thrive. We are approaching a tipping point - we have become dependent on software that is riddled with defects. Software has huge implications for safety and for the continued growth of the economy. As observed by Parnas [4]:

“Despite more than 30 years’ effort to improve software quality, companies still release programs containing numerous errors. Many major products have thousands of bugs. It’s not for lack of trying; all major software developers stress software quality assurance and try to remove bugs before release.”

We need software organizations to expand the role of SQA beyond that of testing so that we can produce software that is not harmful to people or the economy.

‘till next time...

Monthly Morsels

Every month in this space, you'll find additional information related to this month's topic.

1. IEEE Std 12207: 2008 Systems and Software Engineering – Software Life Cycle Processes
2. Capers Jones, Presentation on "Software Defect Removal, The State of the Art in 2010."
3. Jones, Capers, "Software Defect Removal Efficiency", IEEE Computer, April 1996. P. 94-95.
4. Parnas, D. L. and Lawford, M., "Inspection's Role in Software Quality Assurance", IEEE Software, July-August 2003, p. 16-20.
5. Pressman, R., Software Engineering – A Practitioner's Approach, McGraw-Hill, 1997, 4th edition, p. 179.

About SQC

Software Quality Consulting provides a full-range of software engineering services for safety-critical industries and mission-critical projects. Our goal is to help create safety-critical and mission-critical software that meets our client's needs, complies with all applicable standards and regulations, with the highest level of quality possible, and in the most cost-effective and timely manner possible.

To learn more about how we can help your organization, [visit our web site](#) or [send us an email](#).

Food for Thought, Predictable Software Development, Act Like a Customer, and ALAC are trademarks of Software Quality Consulting, Inc.

Copyright © 2011, Software Quality Consulting, Inc. All rights reserved.

Graphic design by [Sarah Cole Design](#)

[Forward E-mail link](#)
[Safe UnSubscribe Link](#)

[Constant Contact logo](#)