Volume 5, Issue 6
July 2002—The Software Quality Advisor

The Software Quality Advisor Online

Standards Are Your Friend

Randall W. Rice, CSQA, CSTE

Standards are Your Friend

This article discusses how to view standards in a new light. Instead of seeing standards as confining and overwhelming, standards can be a great help in describing what a product should look like.

- Standards are just as important for software development as they are for any other discipline.
- Standards are a key element of a process.
- Standards can reduce the time it takes to produce something by providing a repeatable pattern.

Standards have an unfortunate bad reputation. Standards for anything are seen as confining, a limitation on creativity, a hoop to be jumped through, and a number of similar perceptions.

However, standards exist for very good reasons. While it is true that standards can be applied in ways that actually make them counterproductive, we shouldn't be quick to throw out the good with the bad. People can also abuse measurements, processes, and many other disciplines that make software stable.

Why Standards are Important

I often like to relate software development to other disciplines or industries, mainly to show how software gets by with an incredible number of transgressions that no other field could bear. For example, let's look at the building trade.

How many people would start a building project with blueprints? Today, no one would. There are several reasons. First, it's against the law for safety and zoning concerns. Second, blueprints convey from the customer to the contractor what is to be built. Third, you can't get funding from the bank without blueprints.

I know people who would build buildings without plans if they could. They don't because it's against the law in most places. I'm not advocating that software should be regulated (at least at this point in time), but I am saying there are risk takers in both software and construction.

Blueprints are based on standards. Codebooks for the building trade are huge volumes that address everything needed to build a building – electrical, plumbing, heating, air, metalwork, concrete, wood – you name it. The codes are in place to standardize construction and are in place because of hard lessons learned in the past. Many of these concerns have to do with the structural integrity of a building in regard to fires, earthquakes, hurricanes, etc. In other words, the standards are there for a reason. To fail to follow them is not only against the law, but it places the occupants at risk of injury or death.

Standards play an important role

(Continued on page 2)

Inside this issue:

Book Review—*Communication Gaps and How to Close Them* by Naomi Karten

This book goes a long way to help address perhaps the most fundamental need of any project – good communication. Karten does a good job in explaining the source of many common communication gaps and adds context by providing real-life examples of missed communication. As I read the book, I thought of the times I had experienced many of the same communication gaps that Naomi describes.

What I Liked About This Book

I liked the way that Naomi discusses communication gaps that draw on her own experiences, both good and bad. It's not uncommon for an author to tell you how to do something right, but it's less common for an author to be transparent enough to tell you where they have made mistakes.

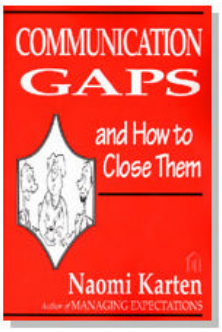
This book has a very wide scope of applicability. Although I'm sure the book will find a home with IT people, just about anyone could read it and apply the techniques to their jobs. I could see this as a great book for a study group.

I expect some people will read this book and gloss over the significance of how communication gaps occur and more importantly, the impact these gaps have on everyday life. That's too bad, because it's the things we take for granted in conversation that Naomi addresses in the book.

Topics

CHAPTER 1: Mind the Gap

(Continued on page 2)



★★★★★

buy it now

Book Review—*Communication Gaps and How to Close Them* by Naomi Karten

(Continued from page 1)

Section 1: Gaps in Everyday Interactions

CHAPTER 2: Getting Through: Responsibilities of the Sender
CHAPTER 3: Misinterpretations: How Messages Cause Confusion
CHAPTER 4: Untangling Tangled Interactions: Reaction of the Recipient

Section 2: Gaps in Relationship Building

CHAPTER 5: Building a Strong Foundation
CHAPTER 6: Appreciating and Benefiting from Communication Differences
CHAPTER 7: Understanding the Other Party's Perspective
CHAPTER 8: The Care and Feeding of Relationships

Section 3: Service Gaps

CHAPTER 9: The Communication of Caring
CHAPTER 10: Gathering Customer Feedback
CHAPTER 11: Service Level Agreement: A Powerful Communication Tool

Section 4: Change Gaps

CHAPTER 12: The Experience of Change
CHAPTER 13: Changing How You Communicate During Change

Other Information About the Book

For a detailed Table of Contents, visit http://www.dorsethouse.com/books/cgapscontents.html

To read an excerpt from the book, visit http://www.dorsethouse.com/news/excerpts/excgaps.html

To read an interview with Naomi Karten, visit http://www.dorsethouse.com/news/interviews/intkartenv12n1.html

Summary

Communication Gaps and How to Close Them is a great resource for project teams that want to improve their communication and ability to work together. This book is also applicable to people in just about any work setting. It's great to have such a collection of techniques to improve communication in one place! I highly recommend this book for anyone.

Reviewed by Randall W. Rice

Standards Are Your Friend

(Continued from page 1)

in many industries. If software development ever makes the transition to a true software engineering discipline, it will need to be based on a set on commonly held standards. There are plenty of people in software development and other software-related fields, even software quality, that reject the importance of standards. They are certainly entitled to that opinion, but I just don't want to use the software they produce – especially if that software is used to control the plane I'm flying in!

We are living in a day where commercial software seems to be getting worse instead of better. We are also seeing a pushback from the use of process-based methods. No matter what we believe, the ultimate test will be the quality of the software products we use. This is a serious concern since many organizations have given up trying to run their own projects and build their own software. They have placed that burden on software vendors who take on that risk and expense. This places the customer in a precarious position at the mercy of the software vendor's management as to what is ready to release. Software development methods need to be getting more rigorous instead of less – and this implies a greater embracing of standards.

On the positive side, standards:

- *Form an underpinning for processes*

In the original Deming Workbench Model, the "Plan-do-check-act" cycle is influenced by customer needs and supported by standards and tools (Figure 1).

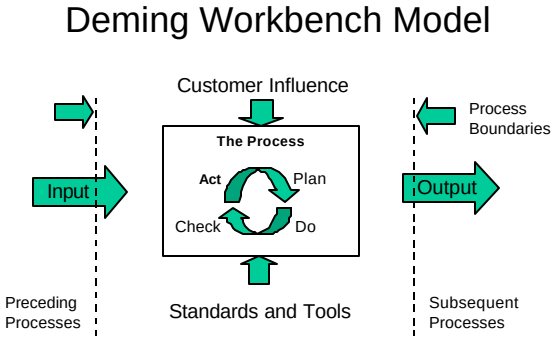
- *Provide a common way of dealing with things*

Aren't you glad that a common user interface emerged for graphical user interfaces? Remember when GUIs first came out and it seemed like every GUI menu bar was different?

Common usage and understanding can be seen at several levels – development, testing, usage by customers and users. Also, without common usage, rework is minimized.

This common usage is perhaps the single greatest benefit of standards. A developer in the U.S. and a developer in India can communicate and understand each other when discussing a point of software architecture when they base

(Continued on page 3)



Adapted from: "Implementing TQM" by Joseph R. Jablonski

Figure 1—The Role of Standards in Processes

"When defects are seen only as problems, people can get defensive and become more concerned with avoiding guilt than taking positive action. "

Standards Are Your Friend

(Continued from page 2)

their conversation on terms and standards they understand the same. It's bad when people use the same terms to mean different things, only to find out later they were on different meanings all along.

A tester in one organization can pick up a test plan from another organization and understand the structure and content more readily because the plan was written to an industry standard.

- *Reduce the time to develop something by having a pre-established framework*

Once I have a framework established for something, I don't have to create a new solution each time I have to do it. Take test plans, for example. A good test plan standard becomes a template for each project I'm called to test. I can make adjustments as needed, but after I've written two or three test plans based on the template, I can write a test plan very quickly.

- *Convey how to do something by describing how the end result should look*

This is important, especially when asking someone else to do something for the first time. The question in their minds is, "OK, just tell me what it should look like." A standard conveys that view. Think of asking developers to write unit test cases and perform them. The developers may not rebel at the task itself, but may become resentful of being asked to do something without a clear idea of what it should look like. A simple test case template would solve the problem and add consistency.

- *Reduce the complexity level of trying to accommodate multiple types of interfaces*

One of the reasons systems are hard to test is because they were not designed with testing in mind and they often lack standardized ways of interfacing with other systems. For whatever reason, standards may not be used in building a system, which often results in a set of complex and unique ways the system interfaces. From the systems maintenance and testing perspectives, this presents a huge challenge to understand the complexity of how interfaces are realized. Once standards are in place, "plug and play" can be more of a reality, as long as the standards are stable and people follow them.

- *Reduce the level of effort required to maintain software*

One of the biggest challenges in software maintenance is gaining an understanding of something you did not develop. This becomes an even greater challenge when everything look at has a different look and feel.

On the negative side, standards:

- *Can give the illusion of achieving quality*

It is possible to follow a standard or process incorrectly, or to follow an incorrect standard or process, and deliver a poor quality product. The magic is not in the standard or the process. People have used this fact as an argument against standards and processes. However, the point that is not well recognized by this viewpoint is that to develop without standards and processes will result ensure missing the quality mark at some point.

Let's use test case development as an example. If I have a test case standard that describes the components of the test case (test condition, expected results, and procedure) and even the types of test cases to be performed, I have a good framework. Now, the job is up to me to work within the framework to develop test cases that will find defects and provide a good return on investment in terms of the value of the defects found. I also want to design test cases that are achievable.

The important point to understand is that the goal is to do quality work to create a quality product and let the standard be your guide.

- *Can be ambiguous on purpose*

Ambiguous standards are often an indicator of unresolved conflict or non-commitment. Sometimes its easier to write a fuzzy standard that to resolve the conflict or to establish a hard line that will be enforced. The classic case is when an organization establishes guidelines instead of standards. Guidelines give the framework, but miss the rigor of providing consistency. Guidelines are optional and give people latitude in what they do. An ambiguous standard missed the goal of a standard.

- *Can be seen as the only standard for evaluation of a product*

There are many other points of evaluation for a product and standardization is just one point of quality. When I was looking for some American food in Hong Kong, I discovered that a supreme pizza at Pizza Hut in Hong Kong tastes exactly like a supreme pizza at the Pizza Hut a mile from my home in the USA. At that place and time, consistency was my main quality criteria. Recently on a trip to Chicago, my quality criteria was uniqueness, so I went to Giordono's for their unique stuffed pizza, and it was fantastic – like none other. Likewise, some software needs to be consistent and some can have uniqueness as a quality factor.

Examples of Test Standards

One of the simplest and most useful standard for testing is a test case standard. You can download a sample test standard overview and a sample unit test standard and unit test workbench at www.riceconsulting.com.

Sample Test Standard Overview – www.riceconsulting.com/library/test_std_overview.doc
Sample Unit Test Standard – www.riceconsulting.com/library/unit_test_std.doc
Sample Unit Test Workbench – www.riceconsulting.com/library/unit_test_wb.doc

Where to Find Standards

The Institute of Electrical and Electronic Engineers (IEEE) has standards for test planning. You can get the standards at www.ieee.org. A highly recommended companion book is *Implementing the IEEE Software Engineering Standards* by Michael Schmidt, which tells how to make sense of the IEEE standards for software development.

William E. Perry's book, *Effective Methods for Software Testing* is also a source of test templates and standards.

How to Develop Your Own Standards

Many organizations choose to develop their own test standards that are customized for their own needs. It is not uncommon to start with standards such as the IEEE standards

(Continued on page 5)

Links

Bobby WorldWide tests Web pages using the guidelines established by the World Wide Web Consortium's Web Access Initiative as well as Section 508 guidelines from the Architectural and Transportation Barriers Compliance Board (Access Board) of the U.S. Federal Government.

http://bobby.cast.org/html/en/index.jsp

Automatic Test Generation from Formal Specifications

http://hissa.nist.gov/~black/FTG/autotest.html

Interworking Labs, Inc. —SNMP testing software

http://www.iwl.com/

A Roadmap for Software Test Engineering

http://www.sdmagazine.com/articles/2001/0102/

Top Ten Use Case Mistakes

http://www.sdmagazine.com/articles/2001/0102/

Software Standards —Their Evolution and Current State

http://www.stsc.hill.af.mil/index.asp

Standard for Software Component Testing

http://www.stsc.hill.af.mil/index.asp

Common Vulnerabilities and Exposures

The Common Vulnerabilities and Exposures (CVE) site is a list of standardized names for vulnerabilities and other information security exposures.

www.cve.mitre.org

Organization for the Advancement of Structured Information Standards

OASIS, the Organization for the Advancement of Structured Information Standards, is a non-profit, international consortium that creates interoperable industry specifications based on public standards such as Extensible Markup Language (XML) and the Standard Generalized Markup Language (SGML), and others related to structured information processing. OASIS members include organizations and individuals who provide, use, and specialize in implementing the technologies that make these standards work in practice.

www.oasis-open.org

Quotes

“I can't understand why people are frightened of new ideas. I'm frightened of the old ones.”
-John Cage

"Live out of your imagination, not your history."
-Stephen Covey

"Read, every day, something no one else is reading. Think, every day, something no one else is thinking. Do, every day, something no one else would be silly enough to do. It is bad for the mind to continually be part of unanimity."
-Christopher Morley

"Only two things are infinite, the universe and human stupidity, and I'm not sure about the former."
-Albert Einstein

"Good leaders make people feel that they're at the very heart of things, not at the periphery. Everyone feels that he or she makes a difference to the success of the organization. When that happens people feel centered and that gives their work meaning."
-Warren Bennis

"Effective leadership is putting first things first. Effective management is discipline, carrying it out."
-Stephen Covey

"The first responsibility of a leader is to define reality. The last is to say thank you. In between, the leader is a servant."
-Max De Pree, "Leadership Is an Art"

“Zeal without knowledge is not good; a person who moves too quickly may go the wrong way.”

Proverbs 19:2, The Bible

Questions From the e-Mail Bag

Q: What is a good way to start performing reviews in an organization that does not currently place a priority on quality control.

A: First, you must get strong management support for reviews. Management will be the primary force behind reviews. If people do not see reviews as important to management, they will not get behind the effort.

Second, you need to get project management to allow time in the project schedule to perform reviews.

Third, there must be a high level of trust that management will not use the information from reviews against the people who create the products. Once information from reviews is used for things such as performance reviews, trust is lost and people will find all kinds of creative ways to manipulate the data.

Fourth, conduct initial training both on the process you plan to use and how to communicate in an open, non-threatening way.

Finally, don’t try to do too much too quickly. You may want to start with walkthroughs, which are very informal.

Q: How extensive should a test plan be?

I often tell people that test plans are tools for communication and should be used to facilitate plan-

Questions from the Mail Bag (Continued from Page 4)

ning, not to create huge volumes of paper.

I have written test plans for major system tests that are less than 20 pages in length.

You should cross-reference detailed items such as test scripts and test cases instead on including then with the test plan.

Q: How many people does it take to perform usability testing?

A: Fewer than you might think! Jakob Neilsen has written an interesting article on his web site at www.useit.com that suggests that you can do a decent usability test with five users. The reason is that the first user will find 50% or so of the usability problems, the second user may only find an additional 20%, and so on. At about five users you have reached the point of diminishing returns.

Q: In Unit Testing, what is a unit?

A: That's a question that even the IEEE unit test standard doesn't exactly define. To me, a unit is the smallest scope you can test independently. A unit can be a program, a module, an object, a component, a web page, etc.

Q: What should User Acceptance Testing validate?

A: User acceptance testing should validate that user needs have been met by the system. Normally, this is accomplished by validating against a set of user-defined acceptance criteria. Of course, the real challenge is how to define tests that validate that the acceptance criteria have been met. Some people teach that user acceptance tests should be based on user requirements or some other document, such as use cases. The problem is that requirements and other documents have defects. True validation is based on real-world conditions, not paper-based specifications.

If you have a question for Randy, e-mail him at rrice@riceconsulting.com.

"True validation is based on real-world conditions, not paper-based specifications."

Standards Are Your Friend

(Continued from page 3)

and customize them with input from other standards to arrive at a standard that is workable in your environment.

How to See Standards as Your Friend

The following tips can help change people's view of standards from constraints to facilitation:

- *Understand that Standards Help, Not Hinder You*

Encourage people to try simple standards to help save time. You might want to call standards "templates" or "patterns."

- *Involve the Affected People in Developing Standards*

Instead of imposing a standard on people, have them design their own standard. People support what they help create.

- *Use Standards as a Framework, Not the Objective*

People need to understand the difference between achieving a standard and actually meeting an objective. It's possible to meet the standard and totally miss the objective!

- *Standards Help Convey What the Objectives Should Look Like*

See standards as the target of what should be delivered.

- *Don't Use Standards as a Way to Stifle Creativity*

It's a good thing to be creative. In fact, creative solutions are an important part of any job. Standards just provide commonality among solutions.

- *Learn How to Work With Standards, Not Against Them*

Some people spend more time and energy avoiding standards than following them. I learned a long time ago that it's easier to go with the flow. If the standard doesn't make sense, then work within the process to challenge it and improve it.

Summary

Standards are helpful when used correctly. However, standards can also be misapplied. When people learn that standards are a facilitating framework, not a bureaucratic exercise, standards can become your friend.

"If the standard doesn't make sense, then work within the process to challenge it and improve it."

Featured Link:

No Hypoxic Heros, Please!
Crosstalk, Dec. 1998

http://www.stsc.hill.af.mil/crosstalk/1998/dec/gray.asp

Comments on the NIST Study by Randall W. Rice

Last month we discussed the new 309 page research report sponsored by the National Institute of Standards and Technology (NIST). One of the findings of this report is that software defects are costing the U.S. economy an estimated $59.5 **billion** each year, with more than half of the cost borne by end users and the remainder by developers and vendors.

"Improvements in testing could reduce this cost by about a third, or $22.5 billion, but it won't eliminate all software errors," the study said. Of the total $59.5 billion cost, users incurred 64% of the cost and developers 36%.

After reading the report, I have the following observations:

1) I believe this report will provide a good source of research for quality professionals making the message for improved software quality to their management.
2) This report is aimed primarily at commercial software vendors and the users of commercial software. People that do a lot of in-house development can still get some good from the report, but the slant is on commercial software. Let's hope the management at many software companies read the report.
3) Thank goodness for the executive overview! Much of the report is pretty straightforward, but at some points the mathematical equations and some of the readability get overwhelming.
4) Some of the sources cited and the numbers they discuss are dated, such as the references to books written in 1990. This could be used to discredit some of the findings.
5) I found it interesting that from the user perspective, user acceptance testing and the costs of such testing are not discussed or factored into the costs of software quality.
6) The report makes a very good point that we have an inadequate software testing infrastructure and one of the problems is that we don't have enough data to guide improvement efforts.

Cogent Quotes from the Report

"In actuality many factors contribute to the quality issues facing the software industry. These include marketing strategies, limited liability by software vendors, and decreasing returns to testing and debugging.

At the core of these issues is the difficulty in defining and measuring software quality. Common attributes include functionality, reliability, usability, efficiency, maintainability, and portability. But these quality metrics are largely subjective and do not support rigorous quantification that could be used to design testing methods for software developers or support information dissemination to consumers. Information problems are further complicated by the fact that even with substantial testing, software developers do not truly know how their products will perform until they encounter real scenarios."

—

"Historically, software development focused on writing code and testing specific lines of that code. Very little effort was spent on determining its fit within a larger system. Testing was seen as a necessary evil to prove to the final consumer that the product worked. As shown in Table ES-1, Andersson and Bergstrand (1995) estimate that 80 percent of the effort put into early software development was devoted to coding and unit testing. This percentage has changed over time. Starting in the 1970s, software developers began to increase their efforts on requirements analysis and preliminary design, spending 20 percent of their effort in these phases. More recently, software developers started to invest more time and resources in integrating the different pieces of software and testing the software as a unit rather than as independent entities. The amount of effort spent on determining the developmental requirements of a particular software solution has increased in importance. Forty percent of the software developer effort is now spent in the requirements analysis phase. Testing activities are conducted throughout all the development phases shown in Table ES-1. Formal testing conducted by independent test groups accounts for about 20 percent of labor costs. However, estimates of total labor resources spent testing by all parties range from 30 to 90 percent (Beizer, 1990).

Summary

I'm glad that NIST sponsored this project and I hope the message finds a home in software organizations nationwide. The key point is how much software defects are costing both vendors and users and how those costs could be dramatically reduced.

You can download a copy of the 309 page report in PDF format at www.nist.gov/director/prog-ofc/report02-3.pdf

"I believe this report will provide a good source of research for quality professionals making the message for improved software quality to their management."

Table ES-1. Allocation of Effort

	Requirements Analysis	Preliminary Design	Detailed Design	Coding and Unit Testing	Integration and Test	System Test
1960s – 1970s	10%			80%	10%	
1980s	20%		60%		20%	
1990s	40%	30%		30%		

Source: Andersson, M., and J. Bergstrand. 1995. "Formalizing Use Cases with Message Sequence Charts." Unpublished Master's thesis. Lund Institute of Technology, Lund, Sweden.

The Software Quality Advisor Online

July, 2002

P.O. Box 891284
Oklahoma City, OK 73189

405-793-7449
405-793-7454 Fax
rrice@riceconsulting.com

"Test everything. Hold onto the good."
I Thessalonians 5:21

We're on the Web!
www.riceconsulting.com

August 2002 Issue:

Using Use Cases Effectively
by Randall W. Rice

Book Review—Introducing Software Testing by Louise Tamres

Coming to Chicago!

August 14—16, 2002

A Three-day course in User-Oriented Approaches or Delivering Quality Software

PMG

RCS
Consulting Excellence!

Presented by Process Management Group, Ltd. (www.pmgltd.com), the Midwest's Premier Provider of IT Quality and Software Testing Services, and Rice Consulting Services, LLC (www.riceconsulting.com), a world recognized leader in Quality and Testing Training.

See details and register at www.riceconsulting.com/chicagoq3_2002.htm

Calendar of Events

User-oriented Practices for Delivering Quality Software

Chicago, IL, August 14—16, 2002

Sponsored by the Process Management Group, Ltd. And Rice Consulting Solutions, LLC

www.riceconsulting.com

How to Become an Effective Test Team Leader, Madison, WI, October 3 — 4, 2002

Sponsored by the Wisconsin QA Chapter.

Register at www.riceconsulting.com/madison2002.htm

A Three-day Course in Web Testing

Chicago, IL, October 30—November 1, 2002

Sponsored by the Process Management Group, Ltd. And Rice Consulting Solutions, LLC

www.riceconsulting.com

EUROStar Conference

November 11—15, 2002

Edinburgh , Scotland

Randy will be presenting a one-day tutorial on *Surviving the Top Ten Challenges of Software Testing* and a keynote address on *Getting the Most Value from Every Person on Your Test Team.*

We hope to see you at one of these events!

If you have a group of 12 or more people in your city that would like to sponsor a training event, contact Randy Rice at rrice@riceconsulting.com to find out how to book a special presentation.