

The Software Quality Advisor Online



January 2002—The Software Quality Advisor
©2001, Rice Consulting Services, Inc.

- Order today at <http://www.riceconsulting.com/SQAdvisornew.htm>

Top Ten Challenges of Software Test Automation

- #10 - Lack of tool availability
- #9 - Lack of tool compatibility and interoperability
- #8 - Lack of configuration management processes
- #7 - Lack of a basic test process or understanding of what to test
- #6 - Lack of tool ownership and acceptance
- #5 - Inadequate tool training
- #4 - Incomplete coverage of test types
- #3 - Lack of management support
- #2 - Inadequate test team organization
- #1 - Buying the wrong tool

Surviving the Top Ten Challenges of Software Test Automation

by **Randall W. Rice, CQA, CSTE**



Abstract

For the past six years I have been surveying training and conference audiences with the question, “How many of your organizations own some type

of automated capture/playback test tool?” Capture/playback tools capture or record the actions performed during a test session into software-like scripts that can be replayed against the same or updated version of software. The idea is to be able to repeat two or more tests identically and compare the results. A difference in test results may indicate the presence of a regression defect. Capture/playback tools own the largest market share of any test tool category and it is that category of tools that is the focus of this article.

a hand. The next question I ask is “How many of you who just raised your hand would consider automated test tools an integral part of your testing effort?” For that question, the typical response is about 10 to 20% of the first group. The main conclusion is that there is a large gap between the people who own automated test tools and the people who actually realize the benefits from test automation. Another observation is that the findings of this survey have not changed significantly over the past six years.

In answering the question I pose to them, typically 80 to 90% of the audience will raise

(continued on pg. 6)

Inside this issue:

Book Review—The Accidental Project Manager	2
Skills of a Test Team Leader	3
Testing on a Budget	4
Links	5
Notable Quotes	5
Frequently Asked Questions	5
Rice Consulting Services' Offerings	17

Book Review—The Accidental Project Manager by Patricia Ensworth



Format: Paperback, 288pp.
ISBN: 047141011X
Publisher: Wiley, John & Sons, Incorporated
Pub. Date: January 2000

Reviewer: Suzanne Chandler

IT background.

Overview

This book would be great for anyone who has been placed in the role of an IT project manager with either a “techie” background or a non-

It has been nicely broken in to four sections that will introduce a new project manager to the four major phases of a software development project.

(continued on page 2)

Book Review—The Accidental Project Manager by Patricia Ensworth, continued...

As a project manager you wear many hats and Ms. Ensworth breaks out each section and chapter with the role you will be playing and what the topic covers:

Section I Before Coding Begins (Research and Analysis)

Chapter 1 Exploring the Elephant

Role: Entrepreneur

Topics: Learning about your user community, understanding the political forces affecting your project, investigating your organization's technology environment.

Chapter 2 On Blueprints and Leaps of Faith

Role: Technology Partner

Topics: Gathering requirements, creating a project plan, negotiating an agreement with your client.

Chapter 3 Who's on First

Role: Team Captain

Topics: Recruiting staff, providing training, building a strong team.

Section II During Development (Design and Construction)

Chapter 4 A View of the Forest

Role: Entrepreneur

Topics: Collaborating with users on product design and testing, communicating with management, designing user manuals and training materials.

Chapter 5 Governor and Legislator

Role: Technology Partner

Topics: Developing test plans, organizing documentation, tracking program changes, monitoring the technology environment, planning the deployment.

Chapter 6 Artisans in Their Workshop

Role: Team Captain

Topics: Maintaining standards throughout the project, promoting effective communication among your staff, resolving interpersonal conflicts.

Section III At Rollout (Deployment)

Chapter 7 Step Right Up, Ladies and Gentlemen...

Role: Entrepreneur

Topics: Finalizing the product features, arranging publicity, training the users.

Chapter 8 The Conductor Taps the Baton

Role: Technology Partner

Topics: Arranging logistics, testing the production system, monitoring users' problems and questions.

Chapter 9 All Hands on Deck!

Role: Team Captain

Topics: Establishing a command-and-control center, dealing with your team members' behavior under pressure, managing your own stress.

Section IV After the Release

Chapter 10 Battlefield Reconnaissance

Role: Entrepreneur

Topics: Evaluating user satisfaction, creating a grass-roots network of "power users," writing reports to management.

Chapter 11 At Cruising Altitude

Role: Technology Partner

Topics: Investigating unexpected consequences, planning maintenance and enhancements, compiling statistics about the project, finishing the documentation.

Chapter 12 Back in Base Camp

Role: Team Captain

Topics: Recognizing accomplishments, evaluating team members' performance and training needs, planning your celebration.

Ms. Ensworth does an excellent job of bringing any new project manager up to speed with issues at each phase of the project. She provides terms you need to know, project templates (visit her web site at <http://www.wiley.com/legacy/compbooks/ensworth/> for downloadable files), tips on how to deal with the people issues within your group, the politics of the project and how to professionally play the game at each level, and much more.

This is a great play book for novice and seasoned managers. I found myself wishing I had this book when I was given my first project manager position!

I highly recommend this book.

Scoring

Readability - 5
Breadth of coverage - 5
Depth of discussion - 4.5
Accuracy - 5
Credibility - 5
Organization - 5
Overall Score - 5



buy it now



Skills of a Test Team Leader

By Carl J. Chandler



The most important skill in leadership is more of a quality that comes with the skill of self control. It was pounded into us over and over again in the United States Air Force—you must have *INTEGRITY*.

The common definition of integrity is that you say what you do and you do what you say. This is a must in any leadership position. Leadership kicks in at the point of disagree-

ment between you and an employee because up until that time the people under your command are doing exactly what they want. At the point of disagreement you, as a leader, must sell or motivate that employee to get in line with your position. If you are a person of integrity then most of the time your employees will be happy to comply.

Being an effective leader starts with integrity because without integrity there is no trust. Without trust the subordinate will not be sold or motivated to follow.

The second important skill is *communication*. Randy teaches in his book “Surviving the Top Ten Challenges of Software Testing” that the people issues of testing are the most challenging. People issues are best dealt with through good communication skills. Most people fall short on this by not listening. If you quietly listen to them people will tell you what the problem is. Some times they are not quite so forward in telling you’re the problem (that can be because they themselves do not know the real problem) but if your listening skills are honed you can pick out their true meaning and feelings by listening to the tone of their voice and watching their non-verbal gestures.

Good communication skills are also important when dealing with superiors since you are the buffer between your employees and upper management. Keep upper management informed of your status and any obstacles they may cause delays or problems. Listen to what they are telling you they want and pay attention to what is going on around them. This will build trust between you and upper management and you will also be able to convey their needs to your team.

The third skill required for a good leader is that of possessing good *interpersonal relations* skills. This is one area that is ex-

tremely difficult to train. It has more to do with your natural personality. If a person has 5 out of 10 people who meet them dislike them then they will most likely will not make a good leader. Your employees do not have to be your friend but they should consider you personable and approachable.

Another skill that is helpful to a leader is that of a *salesman*. After all, to be a great leader you must be able to motivate people. To do that you must “sell” them on your agenda. Sales skills are also important when selling your people/product to upper management and other stakeholders.

Sales takes you right back to integrity. People will relate to you better and be more open with you if they trust you. People will buy what you have to sell if you have integrity.

The bottom line is that the skills you need as a test team leader have to do with relating to people and not necessarily your technical skills. After all a computer will never follow you anywhere but people that respect you will.

The three most important things to have to be a great leader is:

- 1) Integrity
- 2) Integrity
- 3) Integrity

Subscribe to *The Software Quality Advisor*

The *Software Quality Advisor* is a “how-to” newsletter of software quality assurance published online monthly by Rice Consulting Services, Inc.

Articles and features include:

- “How-to” articles on software testing, QA, metrics and measurements written by Randy Rice, Carl Chandler, and other software quality practitioners.
- Book reviews
- Conference reviews
- QA and test tool reviews
- Calendar of events
- Interviews with consultants, authors, and software quality practitioners

An individual membership to *The Software Quality Advisor* costs only \$85 (USD) per year (12 issues).

As a bonus gift, you will receive a signed copy of Randy's book, *Surviving The Top Ten Challenges of Software Testing!*

Testing on a Budget

By Suzanne Chandler



There are two aspects of the project budget for testing in which we are going to look at, the first is setting up and approving the budget, the second is living to that budget.

Before you know how much money you are going to need you first need to examine the following documents.

- Processes for the development team
- Product definition document
- Project requirements document

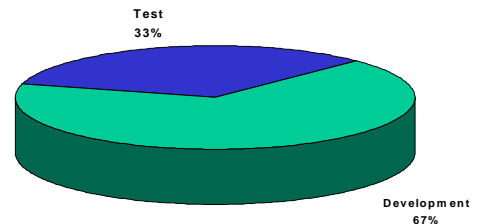
The *processes for the development team* will show the development methodology the development team will be using (waterfall, rapid application development (RAD), or extreme programming (XP)) to understand when you will receive deliverables, the change management process for regression testing, and their processes for version control. If this is an established development team their history is very important. Were they successful at following their processes? Are these new processes to them? How did they handle scope creep? This will help you determine if you need to budget higher for a team that has a shaky record, new processes, or a brand new team with no history together. You can budget more accurately with a team that has a successful history with their processes and who handled scope creep successfully.

The *product definition document* will provide you with guidelines for the test plan. It will also show you who the user community is so that you can begin to plan your user acceptance test. The project manager should have, as backup documentation for the product definition document, a list of users that were interviewed. Use that information to help identify potential candidates for testing that show good organizational skills, a good knowledge of their processes, willingness to learn and improve, or even a fear of computers and automation. These individuals, with the right attitude, can help you test the product as well as sell it to their peers. Review the amount of money and time spent by the development team. A good rule of thumb for planning your testing hours is to assume that one third to one half of the projects time will be allocated to some form of testing. Unit testing will be completed by the developers so you will need to work closely with the project manager of the development team to ensure they have an understanding of unit testing requirements. Test plan-

ning should be approximately 33% of your testing hours as should time spent on test reporting. Be sure you add this time to the plan so you do not find yourself short on money at the end of the project.

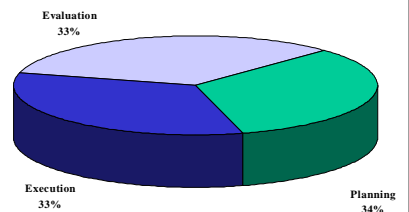
How Much Time Should be Spent on Test Planning?

- Many organizations report spending one-third to one-half of a project's time in test-related activities.



Planning Time Guidelines

- Of the total test time, roughly one-third of the time can be allocated each to:
 - Test planning
 - Test execution
 - Test evaluation



The *project requirements document* will provide the design of the project. For example, if it is an off-the shelf package you will not need unit testing. It should also provide a list of customers and the expected number of users. This is critical when performing risk assessment.

(continued on pg. 13)

Links to Recommended Web Sites

Project Management Templates:

<http://www.method123.com/home.htm>

<http://www.wiley.com/legacy/compbooks/ensworth/>

Test Tools:

<http://www.kratronic.com/recorder/> (Mouse and key recorder lets you automate windows functions)

http://www.netmechanic.com/cobrand/zd_dev/ (NetMechanic's HTML Toolbox is a free online tool that can find errors in your HTML

code. HTML Toolbox will validate your Web pages against the HTML standards and point out browser compatibility issues that affect how visitors see your site. It will also find coding errors and offer design tips on how to improve your site.)

<http://www.netmechanic.com/> (Offers several tools and allows you to try before you buy.)

Glossary:

<http://www.riceconsulting.com/library/gloss.htm>

Articles of Interest:

<http://www.riceconsulting.com/articles.htm> (a group of choice articles written by Randy Rice)

<http://www.ondaweb.com/sti/salary.htm> (software tester salaries)

The Software Quality Advisor:

<http://www.riceconsulting.com/SQAdvisornew.htm>

Published online monthly by Rice Consulting Services, Inc.



Check out our web site picks for this month...

Notable Quotes

"These are the times that try men's souls. The summer soldier and the sunshine patriot will, in this crisis, shrink from the service of their country; but he that stands it now, deserves the love and thanks of man and woman. Tyranny, like hell, is not easily conquered; yet we have this consolation with us, that the harder the conflict, the more glorious the triumph. What we obtain too cheap, we esteem too lightly: it is dearness only that gives everything its value. Heaven

knows how to put a proper price upon its goods; and it would be strange indeed, if so celestial an article as FREEDOM should not be highly rated." -Thomas Paine

"Remember that great love and great achievements involve great risk." - Author Unknown

"Open your arms to change, but don't let go of your values." -Author Unknown

"Dictators ride to and fro upon tigers which they dare not dismount. And the tigers are getting hungry." -Sir Winston Churchill

"Honest criticism is hard to take, particularly from a relative, a friend, an acquaintance or a stranger." -Franklin Jones

"This you know, my beloved brethren. But let everyone be quick to hear, slow to speak and slow to anger." - The Holy Bible

"Those who do not remember the past are condemned to repeat it."
-George Santayana

Frequently Asked Questions

Q: Please define both terms and note any difference if possible. The CSTE guide is confusing. There is a section on Versioning but the guide goes on to talk about Configuration management.

R: I would first of all say that version control is a subset of configuration management. CM makes sure that all components of the system are controlled and implemented together in a packaged way. As part

of this, it is essential that people understand which version of the items they are working with. For example, you would want to know if two people were working on the same product. The two products would need to be merged and tested as a whole before implementation.

Secondly, CM also seeks to maintain a correctness in existing environments. For example, to ensure that the test environment is a representa-

tion of production, etc.

The two activities are often lumped together, sometimes erroneously, kind of like QA and QC! However, there are distinctly different activities performed in each function.

CM: manages environments, packages releases, migrates releases to production, maintains existing configurations.

Version Control: coordinates the distribution of work products between people and identifies which product is in a particular stage of completion.

Surviving the Top Ten Challenges of Software Test Automation, cont.

The observation that automated test tools account for a lot of “shelfware” is not new or all that insightful. In fact, the shelfware problem is shared with many types of software tools. What is useful is to examine the trouble spots in test automation, deal with them proactively, and perhaps mitigate the risks of tool abandonment.

Introduction

When faced with the dynamic world of automated test tools, many organizations make the best choice they can, try to make the tool work in their environment, and hope the rest of the organization will embrace the tool as well.

The purpose of this article is to outline the major challenges that I see most often in organizations struggling to make effective test automation a reality. The earlier these challenges are understood, the better prepared an organization will be to deal with them. Plus, the more an organization understands the issues of test automation, the less risk is seen in acquiring a tool in terms of time and money.

These challenges will be presented from the least to highest impact on the overall test automation effort. It is also important to understand that even organizations who have developed core competencies in working with test automation struggle at times with these challenges.

Challenge #10 - Lack of tool availability

This can be seen from two perspectives:

- 1) The tool is available, but you can't get the funding for it, or
- 2) There doesn't seem to be a tool on the market that does what you need or fits in your environment.

The first issue is based in your own management's understanding of testing and the priority given to testing in terms of funding. While the cost of automated test tools is high

as compared to more common software packages, such as office automation products and software development suites, the anticipated value of the tools is seen in reduced time and greater precision in testing. The promise of test automation is increased productivity and accuracy, which is where the business case must be made. The cost of a single defect in most organizations can offset the price of one or more tool licenses. It's all a matter of where management chooses to spend the money – in defect detection or post-production rework, which is many times more costly than early defect detection.

The issue of not having tools available in a particular environment is more troublesome. Although there is now automated tool support in most environments, this does not mean the support in every environment is great. In older environments, tool support is very limited.

Solution Strategies

If funding is the issue:

- Measure the current cost of defects, especially in post-implementation rework. Use this information to help build a case for faster and more reliable testing using tools.
- Show the value of automated test tools for other groups besides testers, such as the value of developers using the tools.

If getting a good technical fit is the issue:

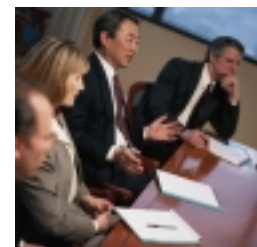
- Network with other testers to find information about lesser-known test tools. Online Quality Assurance (QA) forums are also good places to query people about tools in lesser-supported environments.
- Try to find tools that will work between platforms. This will likely require the use of PC-based emulators as opposed to host-based tools.

(continued...)



Lack of tool availability

“Network with other testers to find out about lesser known test tools.”



Find ways to show the value of automated test tools for other groups besides testers.

Surviving the Top Ten Challenges of Software Test Automation, cont.

- Investigate the possibility of building your own tools, or at least achieving a level of test automation using common scripting commands and comparison programs.

Challenge #9 - Lack of tool compatibility and interoperability

The issue of tool incompatibility and lack of interoperability is seen in organizations that have diverse technologies and applications. The desire is to be able to automate tests that bridge applications and platforms. This is a big challenge because most automated test tools have proprietary scripting languages and approaches, and for the most part, competing vendor tools do not interoperate.

A related challenge is to conduct identical tests on a number of different platforms. This speaks to the issue of tool compatibility to various computing platforms and the ability to share scripts between tools and platforms.

Solution Strategies

- Select tools that have cross-platform capability to the greatest extent possible.
- Consider writing shell scripts and bridging scripts, perhaps in non-proprietary scripting languages, such as Tcl.
- Evaluate critically whether the ability to perform cross-platform testing is a firm requirement.

Challenge #8 - Lack of configuration management processes

Test automation is software testing software. This means that items created using the automated test tools should be subject to the same level of control as any other software asset.

When software configuration management (SCM) is not in place for automated testing, the discipline is missing to work with the tools. Without SCM for automated test tools:

- Effort is duplicated because different people may each be building similar test scripts
- Reuse is not realized because people are all creating test scripts for single-use purposes.
- Existing automated test scripts are at risk of corruption if they are modified without the knowledge of the original author.

What's required for effective SCM for test automation is:

- A workable process that everyone using the tool can understand and follow,
- A tool to manage the ownership, versions and organization of the automated test scripts, and
- A person to own the SCM process and ensure that people are following it.

Many of the popular automated test tools have integrated test case and test script management applications. However, you still need the process and the people to make the SCM effort work. You can also build your own test management tool using a database and basic file organization to group related tests into suites.

A related issue in this challenge is keeping up with changes to applications that are under test. This has been one of the biggest challenges in test automation since its inception. The degree of difficulty in dealing with application changes in automated testware depends on the tool and the technologies involved. In the object-based world, the more robust tools can be configured to ignore user interface changes as long as the objects still behave the same. However, if the tool uses row and column positioning, then each application change will require a change (or many changes) to the automated test scripts.

In character-based applications, such as mainframe CICS applications, all of the changes that impact the user interface will most likely require maintenance to the automated test scripts that test those interfaces.

Solution Strategies

- During your tool search, consider the people and processes that will be required to manage the automated test cases and test scripts.

If you are in the object-based environment (such as Graphical User Interfaces), look for tools that accommodate changes to the user interface gracefully. These tools cost more than those tools that do not offer such flexibility. However, many people have found that the added cost is small compared to the cost on continued manual maintenance of the testware.

(continued...)

Surviving the Top Ten Challenges of Software Test Automation, cont.

- Consider automated test scripts as part of an application's configuration set.
- Involve the prospective test automation SCM person in evaluating test tools and their respective test management offerings.
- Investigate the use of existing SCM tools currently owned by your organization.
- Trace your automated test scripts to functional requirements and defects.
- Ease of automation
- Level of documentation of the function (requirements, etc.)
- Examine your existing set of test cases and test scripts to see which ones are most applicable for test automation.
- Examine your current testing process and determine where it needs to be adjusted for using automated test tools.
- Be prepared to make changes in the current ways you perform testing.

Challenge #7 - Lack of a basic test process or understanding of what to test

Most automated test tools do not tell you what to test. Even the tools that have test case generation features do so at a user interface level and not at the functional requirement level.

If you don't know which tests are the most important and which tests are the most applicable for automation, the tool will only help perform a bad test faster. This is a universal principle that I often illustrate with the example of a power tool.

Let's say that I want to build a bookcase. I try cutting the wood with a handsaw, but it's far too slow and laborious. So, I decide to go to the hardware store and buy a power saw. Upon the purchase of whatever tool I can afford, looks good, or the salesperson convinces me of, I go home and start cutting wood. However, if I don't have bookcase plans or a very good understanding of how to build a bookcase, the saw will just help me make my mistakes faster! To be successful, I will need to first learn enough woodworking skills to understand not only the "what" and "when", but the "why" and "how" of building the bookcase. Then, I'm ready to use the tool effectively.

The tool vendors can train you in how to use the tool with all of its functionality, but the burden is on you to examine your own applications and determine which functions should be tested and to what extent.

Solution Strategies

- Create a set of evaluation criteria for functions that you will want to consider when using the automated test tool. These criteria may include:
 - Repeatability of tests
 - Criticality/Risk of applications
 - Simplicity of operation

- Involve people that will be using the tool to help design the automated testing process.
- Train people in basic test planning skills.

Challenge #6 - Lack of tool ownership and acceptance

In this challenge, the tool is not applied or is ignored. This is often the result of someone's good intention of buying a tool to make life easier, but the rest of the people don't use it.

Some of the reasons for lack of tool ownership and acceptance include:

- Difficulty in using the tool
- Not enough time to learn the tool and perform their normal level of work
- Lack of tool training
- Lack of management support for using the tool
- Lack of tool support, either internally or from the vendor
- Tool obsolescence

Solution Strategies

- Don't cut the tool training. Training doesn't guarantee success, but without it you are at risk of tool abandonment.
- Have someone in your organization in the role of a "tool smith." This person's job is to be the resident expert on the tools used for testing.

(continued on page 10...)

Rice Consulting Services—2002—Schedule of Events

February 11 - 13, 2002—Atlanta, Georgia

<http://www.riceconsulting.com/atlanta2002.htm>

June 10-12, 2002—Kansas City, Missouri

August 14-16, 2002 — Chicago, Illinois

http://www.riceconsulting.com/chicagoq3_2002.htm

A Three-day course in User-Oriented Practices for Delivering Quality Software—Now, more than ever before, more responsibility is being placed on software users to define and validate the systems they acquire. This three-day course presents two important sides of user involvement in software projects: gathering and documenting user requirements, and testing from the user perspective.



April 10-11, 2002—White Plains, New York

<http://www.riceconsulting.com/newyork2002.htm>

April 16-17, 2002 — Phoenix, Arizona

<http://www.riceconsulting.com/phoenix2002.htm>

A Two-day course in Becoming an Effective Test Team Leader—Sometimes people feel intimidated by the technical aspects of software testing and lack the confidence they need to be credible test leaders in their organization. Learn the issues and processes for effectively testing software by attending this hands-on course.



May 8-10, 2002 — Chicago, Illinois

http://www.riceconsulting.com/chicagoq2_2002.htm

November 13-15, 2002 — Chicago, Illinois

http://www.riceconsulting.com/chicagoq4_2002.htm

A Three-day course in Web Testing Techniques

Because of recent increased demand for these courses on web-based testing, we are offering this training in Chicago to accommodate those that are looking for a complete and reasonable priced course.

This offering is actually a combination of three one-day sessions, which focus on many aspects of web testing.

Day 1 - A Web Testing Overview presents the overall process and tools for testing web applications in a variety of environment, not just e-commerce.

Day 2 - E-Commerce and Security Testing presents a complete process for testing e-commerce sites, along with a bonus module for testing web security.

Day 3 - Testing Web Technology digs deeper into how to test the technologies that power many web applications.



Process Management Group, Ltd.

(www.pmg ltd.com)

Founded in 1992, **Process Management Group, Ltd. (PMG)**, has established itself as a premier provider of information technology software quality and software testing services. **PMG** has effectively implemented quality and test disciplines in information technology organizations throughout the United States. **PMG** is dedicated to services within the areas of Software Quality and Software Testing.

Process Management Group, Ltd.

1500 W. Shure Dr.

Suite 250

Arlington Heights, IL 60004-1465

Phone 847-253-3773

Fax 847-253-2760

Surviving the Top Ten Challenges of Software Test Automation, cont.

- Management needs to emphasize that the tool effort is important to them and tool usage is a required part of the testing process.

Challenge #5 - Inadequate tool training

We discussed the training issue in relation to Challenge #6, but this challenge carries its own set of concerns.

Some of the key issues for this challenge include:

- Skipping the vendor's training. The main motivation for this is lack of time and/or money. You'll spend more of both without the training!
- Not getting the right training, due to the incorrect selection of topics. For example, some tool users will need to learn in detail the tool's test scripting language, while other users will need to learn only the basic tool functionality.
- Inability to apply the training to your environment. This is where you learn to use the tool on the vendor's canned example, but have difficulty getting the tool to work on your own applications.
- Trying to learn by self-study. Yes, it can be done, but it takes time and dedication. More often than not, people tend to spend time with only the basic functions and not have the benefit of learning the lesser-known and perhaps more powerful features of the tool.
- Not enough time for training. This goes along with the "dive right in" approach often seen in Information Technology groups. When time is scarce, people tend to gravitate toward the easy and basic functions at the expense of not learning the more difficult but more powerful ones.

Solution Strategies

- Include money in the tool proposal for training at least a core group of people.
- Match people to the most applicable training topics.
- Have tool training performed by the vendor at your location using some of your own applications as exercises.
- Find a skilled local consultant experienced with the tool to sit with your team for about 3 to 4 weeks to help get you started in creating automated tests. It is very important that your team does most of the work to accomplish the trans-

fer of knowledge!

Challenge #4 - Incomplete coverage of test types

As you profile your tests and defect types, you will often find a wide variety of types of tests that need to be performed. These include tests for:

- Correctness
- Reliability
- Security
- Performance
- Usability
- Interoperability
- Compatibility
- Data Conversion

Although the tool may be very adept at automating many of these tests, there may be test types that the tool simply can't support. In fact, most organizations are very happy with a coverage level of 80% of their existing test case libraries.

Solution Strategies

- During tool evaluation, prioritize which test types are the most critical to your success and judge the candidate tools on those criteria.
- Understand the tools and their tradeoffs. You may need to use a multi-tool solution to get higher levels of test type coverage. For example, you will need to combine the capture/playback tool with a load test tool to cover your performance test cases.
- Manage expectations by reminding people that 100% test type coverage is not likely. However, by automating 80% of the tests, you have time to deal with the rest manually.

Challenge #3 - Lack of management support

Management support is needed in designing and deploying test processes that will support the effective use of test tools, reinforcing the role and use of automated test tools in the organization, and allowing time for tools to be integrated in the testing process.

(continued...)

Surviving the Top Ten Challenges of Software Test Automation, cont.

Without management support, the entire test automation effort is at risk. If management doesn't clearly and consistently show their support of test automation, people will be less inclined to show interest in using the tools. This is a major concern, especially considering that the learning curve of some tools requires dedication to overcome.

Perhaps the greatest challenge seen in management support is balancing high expectations of the tool benefits against the time, effort and discipline it takes to implement the tool. Management may become impatient about the lack of tool progress and shift their support to other initiatives.

The pressure is on the people who made the business case for the tools to show progress in a given time-frame. The problem is there are many unforeseen things that can delay or derail a tool initiative. In reality, if people fully knew all of the future problems with any given effort, they would be very reluctant to proceed. There is a place for optimism in acquiring tools. However, a heavy dose of realism is also needed to keep expectations in line with what is achievable.

Solution Strategies

- Communicate that it takes time and planning to build a firm foundation of people, processes and the right tools.
- When making the case to management for acquiring test tools, present the challenges as well as the benefits.
- Reinforce to management that they carry a great deal of influence in how people will accept automated test tools.
- Keep management informed of tool progress and issues that arise.

Challenge #2 - Inadequate test team organization

Most test organizations learn that automated testing is a new world in terms of how tests are designed and maintained. Most tests require more than just capture/playback. The tool user must also be able to work with the tool's scripting language to accurately replay the test session. It helps if the tool user is comfortable working with coding languages. However, if the tool user is not suited to coding, there is a risk that the tool will not be used.

Solution Strategies

- Add a person to the test team who is a "test scriptor." This

person should be comfortable in working with code and be able to take the basic test that has been designed by a test analyst and convert it into an automated script.

- Start simple with basic scripting concepts and add complexity later.

Challenge #1 - Buying the wrong tool

This is listed as the number one challenge in test automation because no matter what kind of process or organization you have, if the tool is not a good technical or business fit, people will not be able to apply it.

We know that a good process and organization are also essential for test automation. However, if the tool won't function at a basic level, people using the tool will simply give up trying to use it.

Unfortunately, too few people do adequate research before buying a test tool. Adequate research includes defining a set of tool requirements based on what the intended users of the tool need to accomplish, a set of evaluation criteria by which candidate tools will be judged, and the experience of other people who have used the tools under consideration.

Solution Strategies

- Take time to define the tool requirements in terms of technology, process, applications, people skills and organization.
- Involve potential users in the definition of tool requirements and evaluation criteria.
- Build an evaluation scorecard to compare the performance of the tools against a common set of criteria. Rank the criteria in terms of relative importance to the organization.
- Perform a "proof of concept" (POC) as opposed to an evaluation. In a POC the vendor often sends their technical team to your site to automate tests using your applications in your environment. Usually, a POC takes about a day to perform. The planning of the POC should be based on the evaluation scorecard. Testers should identify and define the most critical and most common tests they currently perform manually. These tests are often the ones that consume the most time and are the ones that offer the highest payback in test automation. (continued...)

"However, a heavy dose of realism is also needed to keep expectations in line with what is achievable."

Surviving the Top Ten Challenges of Software Test Automation, cont.

Summary

These ten challenges are certainly not the only ones that are seen in test automation, but they are very common and have been the cause for many test automation projects to fail.

Successful software test automation is possible if fundamental issues are addressed and managed.

Success depends on multiple factors that require the coordination of efforts between various groups in an organization. Automated software testing is truly a different way of testing and requires adjustments to current test methods and organizational structures. However, the pay-back from test automation can far outweigh the costs.

Contact Information

Randall W. Rice
Rice Consulting Services,
Inc.
P.O. Box 891284
Oklahoma City, OK 73189
Voice 405-793-7449
Fax 405-793-7454
E-mail:
rrice@riceconsulting.com
Internet:
www.riceconsulting.com

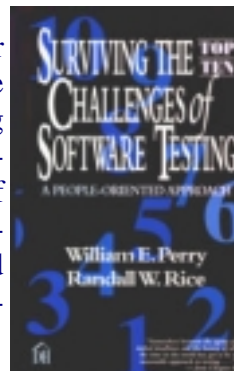


Randall W. Rice is a leading author, speaker and consultant in the field of software testing and software quality.

Randy Rice's Bio

Randall W. Rice is a leading author, speaker and consultant in the field of software testing and software quality. Rice, a Certified Quality Analyst (CQA) and Certified Software Test Engineer (CSTE) has worked with organizations worldwide to improve the quality of their information systems and automate their testing processes.

Mr. Rice has over 25 years experience building and testing mission-critical projects in a variety of environments, including defense and private sector projects.



“Successful software test automation is possible if fundamental issues are addressed and managed.”

Additional Reading

Surviving The Top Ten Challenges of Software Testing by William E. Perry and Randall W. Rice, published March 1998 by Dorset House Publishing.

Software Test Automation by Mark Fewster and Dorothy Graham, pub-

lished May 2000 by Addison Wesley Longman

Automated Software Testing by Elfriede Dustin, John Paul and Jeff Rashka, published June 1999 by Addison Wesley Longman



Rice Consulting Services, Inc. is located in the State of Oklahoma.

Testing on a Budget, cont.

A risk assessment should be completed first in order to determine what your critical success factors are. From this you will know what to test and to what extent. Knowing what and how much to test is always a trick. If you do not have experience in this area you may want to budget in a consultant to assist you with this phase. At this point you should also begin looking at tools that can help you in the testing process. Remember, do not introduce a new tool once testing has started if all of your testers are not trained on it. Even if it is a tool you are familiar with. Follow your plan! Select your tools at the beginning during the test planning stage. Make sure all of your testers are trained on any selected tools so that there are no delays once testing begins.

Test Tools

When searching for tools to assist you in testing look at licensing and trial periods. Some test tools can be cost effective if only a few licenses are required while others can eat up your budget quickly. Most test tools have a trial period that will allow you to use the product for a period of time to see if it will work in your environment. Even if tools are free, or cheap, be sure to test them before implementing the tool into your test environment. Check to see if the company you are working with to purchase tools will come by and give your team a proof of concept demonstration. See if they will use your environment. If so, be sure to have examples ready.

Here are a few links to test tools that are free or inexpensive:

PushToTest offers test automation solutions that test Web services for scalability and performance and also monitor your Web service to proof a service level agreement. Open source, free.

<http://www.pushtotest.com/ptt>

OpenSTA is designed to be used by Performance Testing Consultants or other technically proficient individuals. Using OpenSTA a user can generate realistic heavy loads simulating the activity of hundreds to thousands of virtual users. This capability is fully realized through OpenSTA's distributed testing architecture. Open source, free.

<http://www.opensta.org/>

VeloMeter is an open-source Java-based load testing tool for Web sites. It's designed to simulate the requests generated by thousands of typical users. Open source, free.

<http://www.velometer.com/>

Microsoft Web Application Stress (WAS) is a simulation tool that is designed to realistically reproduce multiple browsers re-

questing pages from a web application. Free download – no eval period.

<http://webtool.rte.microsoft.com/>

Astra Site Manager is a link checker for web sites that can be downloaded for free from Mercury Interactive. No eval period.

<http://www.astratryandbuy.com>

Selling Your Budget

You should now have enough information to begin your budget for this project. When budgeting time and money use the worse case scenario, knowing that Murphy's Law is still in effect.

Probably the most difficult part of the budget for any project manager is the approval process. Upper management or the customer wants all they can get for less than what it costs to actually do the work. So how do you win? You don't want to inflate the budget so that they feel as if they are being taken advantage of. This only causes you to appear untrustworthy. You also don't want to undershoot the real cost of doing the job or you will sell out the farm from under you. At this point you, as a manager, must put on your salesman hat. Help those who are approving the budget understand the cost of not testing correctly and thoroughly.

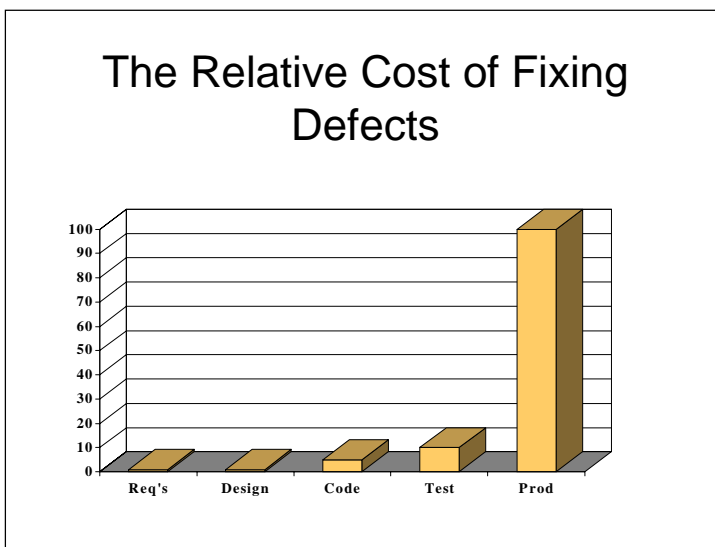
First, the approving party needs to understand the difference between price and cost. All aspects in testing have a price; man-hours, tools, time—just to name a few. When looking at cost we have to measure money over a period of time to measure the return on investment. For example, if a project was developed at the price of \$100,000 and was not sufficiently tested, or tested at all, was launched and subsequently failed the usability testing completed by the customer it would cost approximately 100 times that – or \$10,000,000 – to recall, re-design and re-launch the application. Along with that more customer relation problems then anyone would care to deal with as well as potential loss of customers, current and future. Now, on the other hand if this application had been successfully tested for an additional \$50,000 and then launched the total price of development and launching would be \$150,000. Compared with the first scenario you are looking at a savings of \$9,850,000. This, of course, should be followed up with a few examples. These are not hard to find!

(continued...)

Testing on a Budget, cont.

Most applications complete testing at the end of the project. This approach is called the “big bang” and will cost the company more money than if testing was completed throughout the life of the project. You save money by doing it right the first time! Your mother was right when she asked you “Why do you always have time to do it over and you never have time to do it right the first time?” The same with money spent on a project.

There is a rule called the 1:10:100 which tells us that the later the defect is found, the more expensive it is to fix. The chart below shows you how this works.



Rice Consulting Services teaches that if a defect costs one unit (hour, dollar, etc.) to fix in requirements and design, it costs 10 units to fix in testing (system/acceptance) and over 100 times to fix in production. Sometimes the cost to fix a defect in production costs much more than 100 times the cost of fixing it in the requirements phase.

This cost of defects doesn't even take into account the impact cost of defects. These costs could be attributed to lost revenue, reimbursements, fraud, lost customers, bad public relations, and litigation. In the case of safety critical systems, how can one put a cost value on a human life?

During the meeting to review your budget request you should present this information in a manner that is non-threatening but shows deep concern on your part for the money that is being invested in this project. Be specific and have examples to help make your points. Be ready to defend your position on all areas of the budget but expect flack when you begin talking about tools and administrative support. If you believe in the tools you have selected and have done your research you should be able to de-

fend this easily. Remember, you were sold on this product so use those same techniques when you are selling it. The administrative support is easy because you can use this as a way to get support from other parts of the organization that will not impact your budget. Most managers are willing to share administrative support.

Once your budget is approved, don't be disappointed or discouraged if your requested budget is cut, you need to be sure you have a plan in place and work your plan to bring about the most effective testing you can for the money you were approved.

Developing Your Test Plan

Having a good workable plan that is flexible enough to deal with any changes that come your way is the key to your success. Here are a few tips Rice Consulting Services gives for developing your test plan:

- **Start early**

Even though you might not have all of the details at hand, you can perform a great deal of the planning effort by starting at the general and working toward the specific. By starting early, you can also identify resource needs and plan for them before they are taken by other areas in the project.

- **Keep the test plan flexible to deal with change**

Build in easy ways to add test cases, test data, etc. The test plan itself should be changeable, but subject to change control.

- **Frequently have the test team review the test plan**

Other people's observations and input greatly help arriving at a comprehensive test plan. The test plan should be subject to quality control just like any other project deliverable.

- **Keep the test plan concise and readable**

The test plan does not need to be large and involved. In fact, the more concise and readable the plan, the more useful it will be. The test plan is intended to be a communication document. The details should be kept in a reference location for use when needed.

Even a great test plan is useless without *management support*. Be sure you put on your political hat – keep your integrity in place - and work with upper management to gain their support for your project. Without this support people will not take the project seriously and your work will be in vain. (continued...)

Testing on a Budget, cont.

Be sure to have *defined and repeatable process* in place. Look at past projects in your area to see if successful processes are already in place. If not, you should define these processes immediately. Anything you define should be able to be repeated with consistent results to decrease the risk of defects in production. If you have limited experience in this area you may want to consider bringing in a respected test engineer to assist you through this phase. Well-defined and repeatable processes will go a long way in helping you gain management support.

Throwing warm bodies at testing will not get the job done. *Train your testers* from the beginning on the processes as well as on any tool you plan to use. Have a playbook in place for your testers that includes processes, reporting requirements, etc. Include contact names, responsibilities, telephone numbers, and email addresses for anyone that you expect your testers to work with, report to, or request help from. Let your testers know that it is their responsibility to find defects and to report them so that as a team you can produce the best possible product. Teach them that the developers are part of that team and should be treated with respect when pointing out defects. This certainly does not mean to pamper the developers and allow them to convince you that a defect is a feature but to show respect to them as they would expect to be shown respect in the same situation. Testers should understand that they are the ambassadors for the end user. Have your testers take time to work with the end users. This is an easy sell to the manager of the end users because it will provide your testers with a better understanding of their work environment, any special data that may pass through the system, etc. while allowing your testers the opportunity to work hands-on with the legacy system to see first-hand what the users go through on a daily basis.

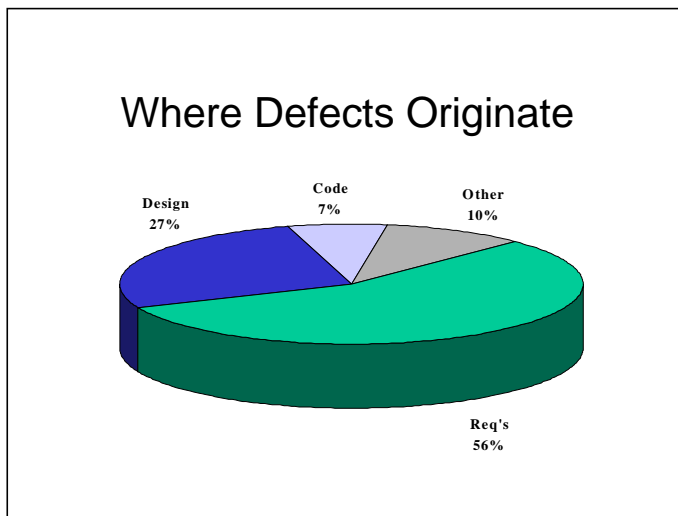
Following these simple tips from the beginning will save money throughout the entire project.

Begin at the Beginning

Most applications fail because sufficient testing was not completed during requirements gathering and application design. These two areas are the most important areas testing should be completed on during the project so use well trained testers in these areas.

The three key problems to watch for are in not allowing for a change in process (which inevitably happens), using ambiguous words (i.e. large, many, soon, etc.), and not involving the end user. So watch for these during the testing of requirements and design. One money saving tip is to convince the manager of the end user how important it is to have a person(s) on their staff in on the development of the requirements and the application design since it is being developed for them and they have a vested

interest. They should pick up the bill for the users time instead of applying it to your budget.



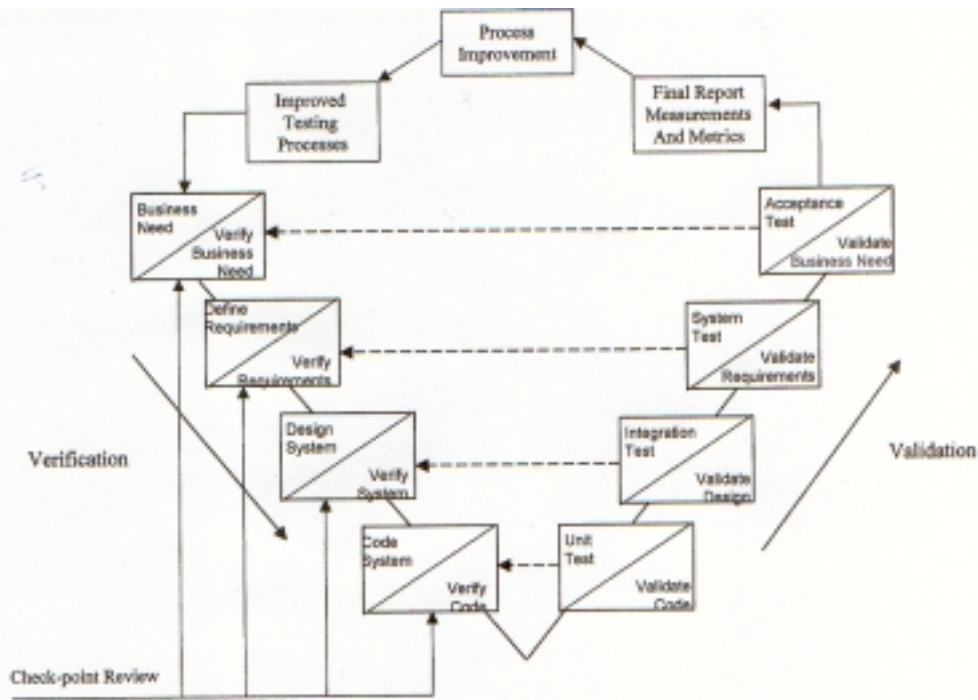
Checkpoints

Completing checkpoint reviews during each step of the process will help you find the defects before moving on to the next stage of development. In the example below each block on the left of the V is a stage in the development process. At the end of each stage a review should be held with all stakeholders (especially users – how else will you know if you are building the right system) involved to ensure the deliverable is correct and ready to move to the next stage. This must be a mandatory meeting. The reason everyone should attend is because you need people with the big picture as well as users with a funneled view of the project. If someone cancels then you reschedule. Upper management is notorious for not showing up. If they consistently reschedule you may need to go back and sell the importance of these meetings to that person. Remember, money talks. Use the economics of testing as discussed earlier to bring them back around. Don't take too long because time is money!

It is important that these checkpoints are pre-determined before development ever begins. If you don't do a checkpoint review you can count on living by the 1:10:100 rule. These checkpoints will also eliminate the expectation gap experienced by users who are dumped in the project at the very end and find that the project is nothing like they expected it to be.

(continued...)

Testing on a Budget, cont.



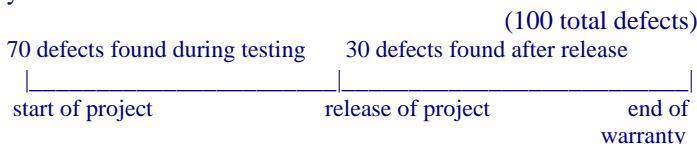
Process Improvement

When one project is finished and before the next project begins the role of process improvement takes place. Process improvement as shown at the top of the chart greatly improves your efficiency of testing. This is done through measurements and metrics and root cause analysis.

- A measure is a single standard (i.e. 10 miles is a measure of distance, 10 hours is a measure of time)
- A metric is two measurements combined (i.e. miles per hour)

One *metric* that is important in improving your process is your defect removal rate. That is the number of defects found by your testing process in the product over the number of defects found during the life-cycle of a product.

Example: If your software is covered with a one-year warranty following release then the life-cycle of that software would be from the start of development until the end of the warranty. The number of defects found by your testing process up until the time of release over the number found during the entire life-cycle is your defect removal rate:



Your defect removal rate would be 70 divided by 100 or 70%.

This is a good way to determine how you did on this project and your process improvement over your last project.

To improve your process you take the defects found, including those found after the release of the project and do a *root-cause analysis* to find out where the defects were input into the project (i.e. requirements, design, development, etc.). Once each defect is traced back to the root cause you will need to determine why it occurred (i.e. ambiguous words in requirements, etc.). Use this information to refine your process to guard against these same mistakes in the next project.

Conclusion

In order to save money in your testing processes you need to do it right the first time. Plan your work and work your plan. Focus on the process and not the product. Work smarter, not harder. All of these clichés are true. The 1:10:100 rule shows that if you take short cuts and do it wrong you will have to re-work it later at a higher rate with less management support. Follow-up your project with root cause analysis and process improvement techniques that will help you to be more efficient on your next project.

Consulting and Course Offerings by Rice Consulting Services, Inc.

Rice Consulting Services' Consulting Offerings:

Testing Assessments

Rice Consulting Services' testing assessment is a quick and effective way for an organization to determine where they are in terms of software testing maturity. The assessment looks at three areas that are critical to testing:

Test organization - Who performs testing, what levels of experience are present, and when testing is performed in the development/maintenance life cycle.

Test process maturity - How well-defined, well-deployed, and repeatable the test process is, and whether it incorporates good testing management, practices, tools, and techniques.

Readiness - An assessment of the organization's readiness to improve the testing process. This involves an assessment of the staff's testing awareness, testing skills, and motivation to change current practices.

The deliverable is a report detailing the assessment's findings, a recommended quality improvement strategy, and a plan for addressing the improvement needs identified. If the assessment uncovers the need for in-house skills training and consulting, we will include proposed training and consulting plans in the report. The report is typically about 15 pages in length.

In-House Software Testing Certification Programs

There is a lot of interest in certification programs for software development and software quality. There is also value to both the individuals being certified as well as the organizations that employ them. As you examine the various certification programs that are available, you need to ask:

- How recognized is the certification?
- What is the basis of the certification (i.e., what does it cover)?
- How is the certifying organization accountable and responsive to its members?
- How closely do the certification criteria reflect the items important to your career and organization?
- What is the required investment to get and maintain the certification?
- What is the future for the certification program?
- What is the initial cost to your company to get certified?
- What is the annual cost to your company to retain certification?

After examining the above questions, some organizations have

determined that the best certification program may be their own. One of the greatest advantages of an in-house certification program is that you can control the criteria, future and investment of the certification. As for objectivity, there are options that allow you to administer the in-house program while an independent organization verifies certification criteria.

Rice Consulting Services, Inc. has been working with several organizations recently to develop this kind of program. **We have extended certification training programs of 10, 15 and 20 days in length. These programs are tailored to your people, business, technologies and tools.** Participants range from experienced testers and QA personnel to people just entering the field.

This type of program makes the training effort more than a "one shot" event. People are tested at the end of each major topic area and are also evaluated by direct observation during exercises. The certification is normally determined by a combination of demonstrated proficiency during training as well as actual work experience. The certification criteria are defined by you, but we can help you with templates and examples.

Each in-house certification program is different. For details about how we can help you design and conduct an in-house certification program that is right for you and your organization, just call Carl Chandler at 405-414-6759 or email us at carlchandler@riceconsulting.com.

Rice Consulting Services' Course Offerings:

For those who hold professional certifications each hour of instruction should qualify for one CPE credit. We recommend that you check with the certifying organization to verify CPE credit applicability.

Visit our web site at:

http://www.riceconsulting.com/course_listings.htm

Building an Effective QA and Testing Process for Ongoing Validation 2 days — This course is designed to teach participants how to design and implement processes for quality assurance and quality control.

Web-based Testing Overview 1 day — A practical computer-based interactive seminar designed to provide a quick start in testing web-based applications.

E-Commerce and Security Testing 1 day — A practical hands-on seminar to explore the deeper issues of testing e-commerce applications.

Testing Web Technology 1 day — A practical hands-on seminar to explore the deeper issues of testing web-based applications.

Consulting and Course Offerings by Rice Consulting Services, Inc., cont.

Build Your Own Course

— 2 – 20 days

Because all of our courses are designed to be modular, we can easily customize a course for you for presentation at your facility! A typical course day is 6 to 7 hours of instruction.

We provide a listing of all of our course modules at http://www.riceconsulting.com/build_your_own_course.htm. Simply select the modules you would like to have presented to your team. We provide a brief description of each module but if you would like to see more details, just click on the Module ID link. Upon submitting your course design, we will get a copy of your selections and will contact you by e-mail and phone.

If you would like to learn more about the information covered in Carl's article we at Rice Consulting Services, Inc. offer an excellent course that will enhance your company's software quality process.

Becoming an Effective Test Team Leader

- 2 days

This two-day session is designed for test leaders and test managers, people who expect to be in a test leadership role, or people who lead other test managers and test leaders. The main objective of this session is to teach you how to be the very best test manager and leader. This course also answers the question, "What does it mean to be the best?" There are many people functioning as test managers, but how many are really leading the team? In leading a test team, you must not only understand the basics of software testing, but you must also understand your own organizational culture. Once you understand your organizational culture, you might find that testers have a less than positive image. This session will discuss how to transform the image of testers from one of police to one of team members.

You will learn the terminology, process, and challenges of testing in the real world. Team-based exercises reinforce the concepts of facilitating team activities and performing leadership activities. As a result of attending this seminar, you should have a good working knowledge of software testing and what it takes to design and conduct an effective test of software, regardless of the technology.

Becoming an Effective Test Team Leader will help you become more comfortable and confident in leading the testing effort in your organization. You will emerge from this two-day session knowing how to develop test cases and test plans. You will also leave with a knowledge of how tools can help you perform testing.

Sometimes people feel intimidated by the technical aspects of

software testing and lack the confidence they need to be credible test leaders in their organization. Learn the issues and processes for effectively testing software by attending this hands-on course.

Return on Investment

- Learn how to find costly and embarrassing problems before your customers find them.
- Understand the key issues in testing software applications.
- Learn how to deal with people issues that can easily derail your project.
- Get the most out of your existing investment in testing and how to leverage that investment.
- Advance your career by reinforcing your testing expertise.

Who Will Benefit

- QA and Test Managers
- Aspiring QA and Test Managers

Other courses offered by Rice Consulting Services, Inc.:

Integration and Interoperability Testing 3 days — This is an intermediate level course to build skills in testing systems and applications in diverse integrated environments, especially where compliance to interoperability standards must be valid.

Gathering, Documenting and Testing User Requirements 2 days — This is a basic course in understanding the process of gathering, defining, testing and managing user requirements. You will learn the requirements process from start to ongoing maintenance.

Managing and Controlling Testware 2 days—Presents processes and tools to manage test plans, test cases, test scripts, and other items needed to perform tests through the project lifecycle.

Basic Training in Software Testing 2 days — A quick start course in software testing for people just getting into the field, or for people who just need a refresher course or validation for their current testing techniques.

Introduction to QA and Testing 2 days — A quick start course in software quality, quality assurance and testing.

Structured User Acceptance Testing 3 days — This is one of the few courses available that teaches a non-technical and easily learned process for testing computer systems from a business process perspective.

Unit Testing 2 days — This course teaches how to test at the unit or component level.

Walkthroughs, Reviews and Inspections 1 day — Teaches practical processes for verification of project deliverables.



January 2002—The Software Quality Advisor
- Order today at <http://www.riceconsulting.com/SQAAdvisornew.htm>
©2001, Rice Consulting Services, Inc.

Rice Consulting Services, Inc.
P.O. Box 891284
Oklahoma City, OK 73189
Phone: 405-793-7449
Fax: 405-793-7454
Email: racs@telepath.com

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

- **How Are We Doing?**
by Randy Rice, CQA, CSTE
- **Core Tester Skills**
by Carl Chandler
- **The Politics of a Project**
by Suzanne Chandler

Coming to Atlanta! February 11-13, 2001

Rice Consulting Services, Inc., a world recognized leader in Quality and Testing Training.

To present to you a Three-day course in User-Oriented Practices for Delivering Quality Software

Log on to <http://www.riceconsulting.com/atlanta2002.htm> to learn more about the course and to register.

WE'RE ON THE WEB!
WWW.RICECONSULTING.COM

Meet Your Rice Consulting Services Team—Ready to Serve You

Janet's Experience and Credentials

- Over 12 years experience as a business owner
- 11 years experience as financial director
- 2 years as Chief Executive Officer of Rice Consulting Services, Inc.

Randy's Experience and Credentials

- Over 25 years experience in the development and testing of information systems in a variety of environments, including web-based, traditional, client/server, and Object-oriented approaches,
- Certified Software Test Engineer (CSTE)
- Certified Quality Analyst (CQA),
- Fellow of the Life Management Institute (FLMI),
- Chairperson of the Quality Assurance Institute's (QAI) annual

International Software Testing Conference, 1995-2000

- Co-author with William E. Perry of the book, *Surviving the Top Ten Challenges of Software Testing.*

Randy has been published on the topic of software testing in:

- *The Journal of the Quality Assurance Institute,*
- *Client/Server Computing,*
- *Powersoft Applications Developer,*
- *Enterprise Systems Journal,*
- *Crosstalk*

He is also publisher of the *Software Quality Advisor*, the "how-to" newsletter of Software Quality Assurance.

Carl's Experience and Credentials

- Over 18 years experience in quality assurance and product improvement
- 20 years experience troubleshooting mission-critical systems

in a variety of environments

- Over 8 years experience as an instructor of quality assurance, trouble-shooting, and the theory of electronics
- 6 years experience in sales and marketing

Carl has been received numerous awards and decorations through the United States Air Force to include the John Levitow Award for Outstanding Leadership Abilities and Top of Class for Air Force Leadership School

Suzanne's Experience and Credentials

- Over 17 years experience in software development and the IT field
- 4 years experience as a Project Manager in and out of the IT field
- 2 years experience as Project Coordinator over Project Managers
- 1.5 years experience as Director of Operations



Left to right: Top row—Randy Rice, Vice President Research and Development, Instructor—Carl Chandler, Director of Sales and Marketing, Instructor. Bottom row—Janet Rice, Owner—Suzanne Chandler, Director of Operations