Testing Wireless Applications – Part 2
by Randy Rice, CQA, CSTE

Primary Concerns and Risks of Wireless Applications

A very useful starting point for a test strategy is to examine the risks. For wireless applications, the risks are:

Correctness – Like any other application, if the information is not correct, people will abandon the application. Here are two interesting findings from Price Waterhouse Coopers survey of 600 mid-sized and dot-com companies in the U.S., U.K. and Australia, as reported in The Industry Standard, April 18, 2001:

1. "Only 1 in 3 traditional companies and only half of the e-businesses, which should have state-of-the-art technology, reported being "very confident" about the quality of the data they collect."

2. "a third of e-businesses and a quarter of traditional companies surveyed said they had lost sales because of poor information about what inventory they had in stock."

These are shocking findings from both the customer and business perspective!

Usability – Wireless applications shrink the user interface to the extent that two concerns emerge greatly when dealing with wireless usability.

1. Interface – The display size of wireless devices is often very small. This affects readability and feature sets in a big way.
2. Ergonomics – Wireless devices are compact and have smaller keys for input. If you have small fingers, the issue is lessened, but if you have fingers that resemble small sausages, you need another way to input data beside your fingers.

Item	Av Value
Blank CD	4.25
Modem	55.25
Music Box	15.75
Socks	5.125
Select Again	

What does the research show on wireless usability factors?

1. Most WAP users rate usability very low.
2. Over 50% of wireless users in the U.S, U.K and Finland surveyed by Accenture saw "no compelling reason" to use wireless web services. They cited factors such as poor interface design and slow performance as key reasons for their lack of interest.
3. Even small usability flaws loom large in wireless apps.

Security

Even though WAP is designed to be a secure technology when used correctly, there is a big difference between knowing and doing when it comes to security. Hackers have become adept at "cell phone cloning" and can intercept phone transmissions out of the air. This happened to me personally while on a trip in New York City a year or so back. Hackers intercepted one of my phone calls and proceeded to charge over $4,000 of phone calls to my account. Fortunately, the people at my phone service provider were helpful and understanding, so I didn't have to pay for the fraudulent calls. However, it makes me think twice about what I would do via wireless technology.

In addition to the street threats, there are also the security issues of what happens to the transaction at the gateways, which are controlled by nameless, faceless third parties. Peer-to-peer computing also holds risks for the privacy of data sent between devices.

Performance

Performance issues can result from a variety of sources including:

Slow infrastructure

One of the most pressing challenges in wireless applications is the growing demand on bandwidth. This demand grows daily as new users are coming onto the web and as businesses increase the quantity and quality of media content, such as streaming media. There is considerable discussion as to whether or not third generation (3G) networks will ever become a reality in the U.S. In addition to slow networks, the reliability of wireless connections leave much to be desired. It's one thing to be able to decipher a garbled voice message, but another to transmit a stream of data across the same faulty connection.

Client bottlenecks

Performance problems can also be traced back to client bottlenecks in WAP applications as well as hardware limitations with the wireless device.

Integration

As in any robust application, wireless or otherwise, integration often plays a major role in the ability of the application to deliver correct and efficient results. Wireless integration concerns include:

- **Online wallets** for fast and secure online payments
- **Business systems** which must handle the transactions generated by wireless devices
- Other devices which receive data from wireless devices:
    - **Notebooks**, which may sync with PDAs
    - **Vending machines**, which my provide inventory control data to an central business system
    - **Handheld units**, which may transmit inventory or shipping data to central systems
    - **Web-enabled phones**, which my send data to other cell phones such as with instant messaging

Flexibility

People want to make the most of their investment in wireless technology and are often cautious about hardware and applications that may become obsolete in a year or less. Therefore, flexibility is a very important issue to wireless users. Flexibility in this context can include:

- Ease of upgrades to software and hardware
- Application flexibility with various hardware and wireless operating systems.

Profitability

This is an opportunity to apply some of the lessons learned from the recent dot-com boom and bust experience. Wireless providers must be able to prove that the envisioned users will actually buy the applications and services under development. In addition, the wireless development companies must be able to survive until the wireless market can support the new technology and pray that a new technology does not catch on and render their product or service obsolete.

Some questions that should be asked about any project include:

Does the application solve a problem effectively and easily?
Is there a customer base?
Can development costs be recouped quickly?

A Testing Strategy for Testing Wireless Applications

Now that we have examined the basis of wireless applications and the risks of wireless technology, let's look at a strategy for testing wireless applications, regardless of the method of application.

Rice Consulting Services, Inc.
P.O. Box 891284
Oklahoma City, OK 73189
405-793-7449
405-793-7454 FAX
Chat with Randy

- Type of Wireless Application
- Audience Demographics
- Technologies Used
- Tools
- Test Execution Strategies

Type of Wireless Application

Some possibilities include:
- WAP
- M-Commerce
- Workgroup
- Personal productivity
- Peer-to-peer
- Business system integration
- Entertainment

The type of application will determine the type of tests you will need to perform.

Audience Demographics

This will dictate the types of testers you will need to get a representative assessment of usability.

- Age (The most likely acceptance group for wireless apps are teens!)
- Geography
  - This applies to issues of localization as mobile applications are deployed in a variety of countries.
- Web comfort level
  - This will determine the important of usability factors.

Technologies Used

This will determine the type of test environment and tools you will need.
- WAP
- Bluetooth, etc.

- XML, WML
- Security protocols

Tools

- WAP Simulators by hardware vendor: Nokia, Ericsson, Motorola
- Capture/Playback – such as TestPartner by Compuware
- Performance Testing – such as QA Load Compuware

Test Execution Strategies

Pilot tests

- Other countries - With early deployment in other countries, it is possible to introduce a product to a smaller and more remote audience. You can practice in a small room!
- Limited user base – This type of test would include usability testing with a small number of users to get an assessment of initial usability.

What Does This Mean for Testing Wireless Apps?

It's a small world after all!

User interfaces tend to be much smaller than other applications, which translates into smaller test cases.

There's a large part of testing wireless apps that is like other testing.

This includes tests for validating the integration of transactions, platform portability and integration testing, and having a basis for testing such as use cases and/or requirements.

Wireless test strategies can take many forms

Just a few of the many flavors include M-commerce, data collection and delivery, entertainment and workgroup collaboration.

Tools can help

This is especially true in regression and load testing.

The test lab is greatly different than the real world.

Simulators are great, but you need to test also with real people in real-world conditions.

Rice Consulting Services, Inc.
P.O. Box 891284
Oklahoma City, OK 73189
405-793-7449
405-793-7454 FAX
Chat with Randy

Summary – Dos and Don'ts of Testing Applications

The world of testing wireless applications is growing and changing daily. I'm sure that within a few weeks, I will learn even more considerations for this rapidly changing technology. In the meantime, here are my closing thoughts.

Do:

- Ask questions early about application feasibility
- Remember that wireless technology is still very young
- Perform early usability testing
- Be prepared for an onslaught of new development when people and the technology are ready.

Don't:

- Forget that you still need something to base tests upon, such as use cases and requirements.
- Forget that much of M-commerce is based on integration between organizations.
- Forget your audience.

I hope this article helps in your search for information on testing wireless applications. I would like to hear about your challenges and success stories. You can contact me at rrice@riceconsulting.com.

Book Review
by Randy Rice, CQA, CSTE

Software Testing by Ron Patton
★★★★★

Format: Hardcover, 389pp.
ISBN: 0672319837
Publisher: Sams
Pub. Date: November 2000
Price: $39.99

Overview

Software Testing is a book oriented toward people just entering or considering the testing field, although there are nuggets of information that even seasoned professionals will find helpful. Perhaps the greatest value of this book would be a resource for test team leaders to give to their new testers or test interns. To date, I haven't seen a book that gives a better introduction to software testing with this amount of coverage. Ron Patton has written this book at a very understandable level and gives practical examples of every test type he discusses in the book. Plus, Patton uses examples that are accessible to most people, such as basic Windows utilities.

What I Like About This Book

I like the simplicity and practicality of this book. There are no complex formulas or processes to confuse the reader that may be getting into testing for the first time. However, the important of process is discussed. I also have to say a big THANK YOU to Ron Patton for drawing the distinction between QA and testing! Finally, the breadth of coverage in Software Testing is super. Patton covers not only the most important topics, such as basic functional testing, but also attribute testing, such as usability and compatibility. He also covers web-based testing and test automation – and as in all topics covered in the book, Patton knew when to stop. If you want to drill deeper on any of the topics in this book, there are other fine books that can take you there!

Who Will Benefit From This Book

- Beginning Testers
- Trainers
- QA Analysts
- Business Analysts
- IT Management
- Test Team Leaders
- Software Developers
- Students
- Career Changers

Chat with Randy

Scoring

Readability - 5
Breadth of coverage – 5
Depth of discussion - 4
Accuracy - 5
Credibility - 5
Organization - 5
Overall Score – 5

Topics Covered

Rice Consulting Services, Inc.
P.O. Box 891284
Oklahoma City, OK 73189
405-793-7449
405-793-7454 FAX
Chat with Randy

Summary

I love this book because it is practical, gives a good introduction to software testing, and has some things that even experienced testers will find of interest. This book is also a tool to communicate what testing and QA are all about. This is something that test organizations need as they make the message to management, developers and users. No test library should be without a copy of *Software Testing* by Ron Patton!

Reviewer

Randy Rice

Frequently Asked Questions
by Randy Rice, CQA, CSTE

Q: I have questions regarding Volume and Stress testing. How do I prepare the Volume and stress Testing test cases? How is Volume and Stress testing done manually?

A: First, we need to clarify terms,

Load testing seeks to validate that an application can process a specified amount of transactions or users in a given period of time. These times can be measured at average load, peak load (the high points in a day), and maximum load (the highest load ever).

Stress testing seeks to validate where the breaking point of a system is by continuing to pile on load.

The type of cases depends on the environment, but think of how an engineer designs a structural test of a bridge. There are two things they need to know: 1) The expected load and when it will occur, and 2) the stress points in the structure. Likewise, you will need to know your traffic patterns and the transactions they will be performing. You will not need to test every function for load and stress, just the ones that are critical and bear high load. Be careful, though, and don't forget that many transactions may extend to other platforms. Don't stop your tests at just a web server, for example.

As for tools, I can't even imagine how to perform load and stress testing without a tool, unless the performance is already so bad that 10 people can crash the system (I have seen that before!). The reason is that you need high load and sustained load. Something else to keep in mind is that although you may have 500 users at one time on your application, you may only need to have a tool licensed for a fraction of that number, since all the people will not be actively hitting the app at the same instant. If you don't want to invest in the tools, there are load testing services, such as Compuware's Point Forward service that will handle this for you.

Q: Quite a simple term but I need to define "install". To many, installation is a process of updating our systems and running a new install, etc. I just don't know how to define this...can you help?

A: The term "install" would definitely depend on the context in which it is used. For example, we have people ask that we test the install portion of PC-based software, which is basically a compatibility test of the install routine. Installation could also mean the first time an entire system in placed into operation. The hardware platform could be anything from a mainframe to a PC. In the context of ongoing operation, what some might call an installation may also well be called a release, version, patch or upgrade.

In all cases, you will have to install the new version, so we are back to defining this term in the context of your own systems and technology. Here's how I might define this in a set of standards:

Install - the process of loading a new or modified application into an operational environment, either production, test, or staging.

Release - a set of new or modified functionality contained in an application.

Version - an identifier to set apart one operational release from another.

Links to Check Out…

Page of Automated Test Tool resources by Keith Zambelich - http://www.sqa-test.com/toolpage.html

Lots of QA and Testing Links - www.qalinks.com

Managing Attitudes

by Carl Chandler

Managing attitudes is not an easy part of testing but can be one of the most critical. To address attitudes, it is very important to keep stressing the importance and the positive aspects of testing. In our Test Team Leadership course we teach some of the people and cultural issues often encountered in software testing and how to address those issues.

User's attitudes during Acceptance Testing

Users generally come with a preconceived notion of what they expect a system to do and can be discouraged if during testing the two do not match. Finding many defects, a lack of recognition, or if they feel insulted by responses to their defect reports can also cause users to become discouraged.

Chat with Randy

- **Keep stressing the purpose of validation is to test the fitness of use of the system.**

If a defect is found, the effort was successful.

- **Encourage upper management to keep praising their people.**

Point out they will know the system better than anyone else and will be valuable to the organization.

- **Remind them developers are also under pressure to deliver a quality system.**

Developers and users should be working toward the same goal.

- **Keep in mind that acceptance testing is not the time to redesign the system.**

Make sure users understand *why* the system is designed as it is before logging a perceived design flaw as a defect. Involving users in the entire process of development reduces this problem.

- **Stay positive and constructive !**

Remember, this too, will pass.

Developer's attitudes

Developers take ownership in the applications they develop and can take user defect findings personally or see them as roadblocks if not reminded that the application is a team effort with joint ownership.

- **Keep stressing the point of testing is not to humiliate the developers**

People take pride in their work and the way criticism is given is very important. Comments about defects should be made in a way as not to insult the person that worked on the software.

Rice Consulting Services, Inc.
P.O. Box 891284
Oklahoma City, OK 73189
405-793-7449
405-793-7454 FAX
Chat with Randy

- **Reinforce to both users and developers that cooperation is essential for a successful testing effort**

Explain to the developers that users are using a new system and might not be aware of all aspects of its operation. Again, involving users in all phases of the development process will reduce this effect.

- **Watch the backlog of defects**

If a significant backlog of defect reports accumulates, it might be wise for the users to take a testing break to allow developers to catch up

Communicating with Management

Management support is the most critical relationship to maintain during the development process. The application's chance of success increases or diminishes in proportion to the amount of support received by management.

- **The key rule: Be "up front" at all times**

Keep management informed of testing progress by stating the facts objectively.

- **Tradeoffs might be required**

Management is notorious for wanting everything yesterday. At some point, a decision between quality or deadlines might need to be made. Tradeoffs between schedule, cost, and quality depend on the criticality of the system to the business or organization.

- **Manage expectations by keeping management informed of test progress**

If management has been kept well informed of testing progress, their expectations might be more easily met.

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.

Rice Consulting Services' Course Offerings:

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

Becoming an Effective Test Team Leader – 2 days

This session teaches you how to be the very best test manager and leader.
By attending this workshop, the attendee should leave equipped to gather, define, manage and validate requirements in his or her own organization. No special knowledge of systems or requirements is required or assumed to attend this workshop.

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.

For more information on this course or one of our many other offerings please contact Carl Chandler at (405)414-6759

Chat with Randy Thursday, June 14, 3 p.m. – 4 p.m. CDT

Randy will be in our chat room at http://www.riceconsulting.com/e_mentor/chat.html to answer your questions and converse about software quality topics. This chat room is part of our training follow-up program available to clients and customers.

Rice Consulting Services, Inc.
P.O. Box 891284
Oklahoma City, OK 73189
405-793-7449
405-793-7454 FAX

Notable Quotes…

"There is no safety in numbers, or in anything else."
James Thurber (1894 - 1961)

"After an access cover has been secured by 16 hold-down screws, it will be discovered that the gasket has been omitted."
De la Lastra's Corollary

"Outside of a dog, a book is man's best friend. Inside of a dog it's too dark to read."
Groucho Marx (1890 - 1977)

"We're not lost. We're locationally challenged."
John M. Ford

"He who scorns instruction will pay for it, but he who respects a command is rewarded."
The Bible (Proverbs 13:13)

July 2001 Issue:

- ## Building a Test Tool Evaluation Scorecard
  by Randy Rice, CQA, CSTE

- ## Structured User Acceptance Testing
  by Carl Chandler

Rice Consulting Services, Inc.
P.O. Box 891284
Oklahoma City, OK 73189
(405) 793-7449 / (405) 793-7454 Fax
e-mail rcs@telepath.com
http://www.riceconsulting.com