

The Software Quality Advisor Online



Rice Consulting Services, Inc.
P.O. Box 891284
Oklahoma City, OK 73189
405-793-7449
405-793-7454 FAX
April 2001 Newsletter



Solving the Right Problem – Step One in Software Development by Randy Rice, CQA, CSTE

Note: The following article has been adapted from Rice Consulting Services' new course, *Defining, Documenting and Testing Software Requirements* by Randall W. Rice. To find out more about this course, just go to www.riceconsulting.com/requirements.htm.

When you boil it all down, the first and primary task of software development is problem analysis. Although many people commonly view software development as the process of defining requirements and specifications, writing code and testing it, if the problem to be solved has not been correctly defined, the wrong solution will be delivered. Perhaps this helps to explain the high rate of software project failures.

I was thinking about comparisons of the software industry to the building industry and the question came to mind, "How many times do you see the wrong building built?" At first, I thought "Never. Builders build what they are told to build." Then I thought about a

building I drive past about once a week. It's been finished for over a year, a nice looking industrial-type building in an industrial part of the city, but still for sale and standing vacant. I have verbalized on more than one occasion, "Why would anyone build such a large building without someone to buy it?" It appears that someone thought there was a problem that needed to be solved by building another building. However, the market seems to be indicating that the problem or need didn't really exist. So I guess that is an example that software developers are not alone in building systems that meet a need that doesn't exist.

I think it's important to understand that needs and problems are transitory. Today's problem may not be so next week! Therefore, we must consider taking the longer view when we think about what really is the problem at hand. Problems are also opportunities in disguise. Just consider that most innovations were born out of the necessity of solving a problem.

If all of this is not confusing enough, there are times when trying to address a problem too quickly is worse than taking no action at all. And even more vexing is failing to react to problems quickly enough to prevent further damage! This leads us to ask questions such as, "When does putting out fires shift from necessity to habit?" and "When is a problem truly an emergency instead of a 'perceived crisis'?"

Within this context, we can view the high-level software development process as:

1. Problem analysis
2. Requirements definition and refinement
3. Software specifications and detailed design
4. Software creation and refinement
5. Software validation

These are often not performed in sequential order, but rather in cycles where things are learned in the process of performing the steps. However, the problem analysis step is where much of the learning needs to occur and learning more about the problem later often results in painful changes and project problems.

To embark upon requirements definition too early is human nature, but it also causes problems later in the project that are difficult and costly to resolve. We software people are often guilty of “just trying to get the job done” at the expense of first understanding what the job is.

What is Problem Analysis?

Problem analysis is the process of understanding and defining the problem to be solved. Problem analysis is not problem solving! Problem solving identifies solutions that conform to the needs and constraints of the problem. It is common to propose a solution too early that does not consider the restrictions and possible shortcuts associated with the problem. When these things are overlooked, we either have an incomplete or excessive solution.

Five Step Process For Problem Analysis

Much of what is done in designing and building information systems is to solve problems, even though the objective of the system may be seen as improving existing systems or taking advantage of market opportunities. The basic motivation of people to buy or build something is to meet a need. Needs arise most often from problems.

Therefore, a big part of defining requirements is to understand the problem so that the right solution can be delivered. The following process is very helpful in defining, understanding and solving problems. We will go through each of the following steps in detail in the rest of this module.

Step 1 - Define the Problem

This is the first and foremost concern. If we can't state the problem, we can't even know where to begin in solving it. This can be one of the most difficult parts of any project and the most important. An amusing, yet sad, example of this was when a group of diplomats spent three days at the outset of an international summit debating over the size and shape of the meeting table.

Likewise, when we can't get a clear view of the problem, we can get easily distracted by all sorts of side issues. Defining the problem is easier said than done because people have different views of what the problem really is.

Here are some tips to help define the problem:

- **Put the problem in writing**

Stating the problem in writing firmly documents it for a common and firm understanding among a group of people. When defining a problem in writing, standardized processes help greatly to achieve consistency and completeness.

- **Get multiple perspectives**

In viewing a particular problem, people will have differing views of the problem depending on the individual person's background and perspective. If you miss a perspective, you may miss an important angle of understanding an important aspect of the problem.

- **Look for deeper problems**

In some cases, the problem that is most obvious is only the tip of the iceberg. The deeper problems may be the contributing causes to the effect that the main problem is manifesting.

Step 2 - Understand the Root Causes

Each problem has one or more root causes. Many times, the problem being seen is the result of the root causes, and is therefore a symptom, but not the core problem.

In gaining an understanding of the root causes, you are asking “What's the problem behind the problem?” To identify the problems behind the problem, a Root Cause Analysis is performed.

In the Root Cause Analysis, questions are asked, such as:

- What is the primary problem?

This goes back to the problem that should have been stated in writing from the previous step. This question verifies that the primary problem can be described accurately and completely.

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

- What are the contributing factors?

The contributing factors are things below the surface that have a role in the problem. The impact of the contributing factors will vary by item, but once the factors are identified, a greater understanding of the problem is gained.

- Who knows about the problem?

This identifies the people who may be available to provide additional insight to the problem.

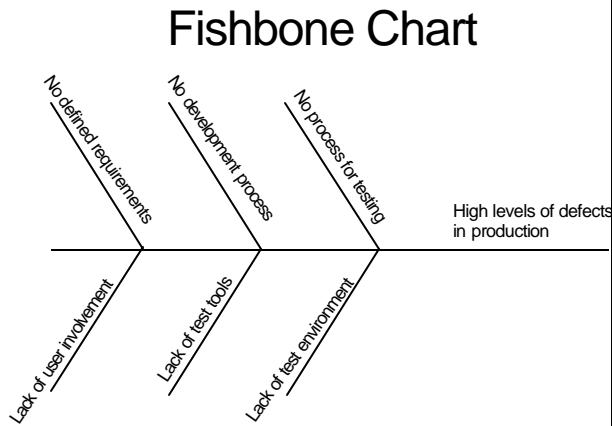


Figure 1 – A Fishbone Chart

A tried and true way to describe root causes is to use what is known as a “fishbone chart.” Each contributing factor is shown as a branch off of the main problem.

Fishbone charts are easy to draw and work well as ways to drive a brainstorming session.

Step 3 - Identify the Affected People

Each problem will impact a given group of people. Each of these people will likely have different needs and concerns, which need to be considered in the solution.

These people are known as a “stakeholders.” The stake could be positive or negative. For example, the person could have a positive stake in seeing a problem resolved because their work would be made easier. Another person could have a negative stake in seeing a problem resolved because it gives them job security in constantly fixing the effects of the problem.

The possibilities of affected people include:

- **The project sponsor**

This is the person that initiates and pays for the solution.

- **Customer**

This is the person that is served by the solution.

- **User**

This is the person that has to apply or use the solution.

- **Management**

This is the group that controls the solution.

Step 4 - Define the Scope of the Solution

The scope of the solution forms the boundary you will need to work within to solve the problem. The scope of the solution addresses the things, which are within your control to address. It is important to define the scope of the solution so that you do not overstep the boundaries of what you can reasonably fix.

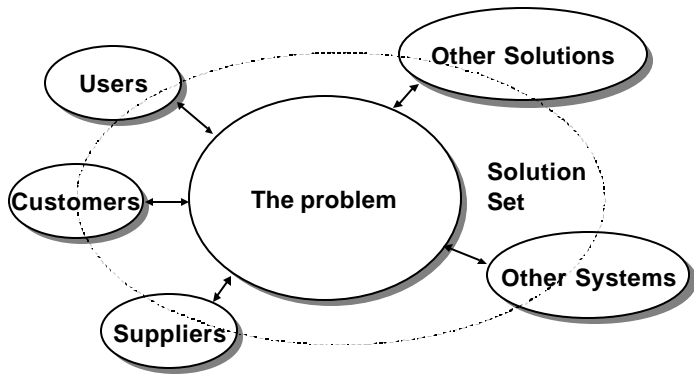
There are two perspectives of a solution:

Internal, which focuses on the problem to be solved from our perspective as the product or service provider.

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

Figure 2 – The Scope of the Solution

The Scope of the Solution



External, which focuses on the people or things that interact from the outside, such as:

- Customers
- Suppliers
- Users

Each of these perspectives will have unique considerations and needs to be seen in the solution of the problem.

There is only so much you have control over to affect a solution. These boundaries will form the scope of the solution. To solve the problem, to have to work within this scope and perhaps get cooperation from people outside of this scope to address the issues that are out of your control.

You define the scope of the problem by considering the affected areas and specifying which things you can reasonably and effectively address.

Prioritize, Prioritize, Prioritize!

In this step, the problem can be prioritized based on severity and potential impact. Potential impact underscore the fact that a problem may not be seen as urgent or important today, yet may have the risk of manifesting as a major problem in the future. Wisdom is required at this point to distinguish between borrowing trouble needlessly and planning wisely.

Practical guidelines, please!!!

Okay, I know this all gets very philosophical at times, so here are some questions I have found helpful in the past to help distinguish between the urgent and the important:

The important:

Does it affect the safety and well-being of people?
Will the company lose money because of it?
Will we be breaking the law because of it?
Will the organization be at risk of legal action because of it?
Will we lose an important window of opportunity because of it?
Will solving this problem make a positive difference in the long-term?
Is the problem adversely affecting the welfare and attitude of the organization?

If any of the answers to the above questions are “yes”, then you likely have an important problem to solve and should devote immediate attention to it?

The urgent:

Is there a dictate from senior management to address this by a particular date?
Are people unsure why a particular target completion date was set?
Is this a concern to just a few people, even one?
Have we dealt with a similar problem recently without a satisfactory solution?
Is this a recurring problem?
Is management unwilling to enact a permanent solution?
Are members of management in conflict over the problem and/or the solution?

If any of the answers to the above questions are “yes”, then you likely have an urgent problem which may be overshadowing other more important issues. To attempt a solution may please management, but will likely not address the core issues. You will likely have the same problem again in a slightly different form at

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

some point in the near future. In fact, these is a strong likelihood that if you ignore this problem it will either go away or mutate into another one with the same false sense of urgency.

Step 5 - Identify Solution Constraints

There will often be barriers to solving the problem. Once identified, the barriers can be addressed instead of ignored.

A constraint is a know lack of something, an inhibitor. Constraints can be seen as:

- A lack of time to complete the solution
- The lack of money to perform the solution
- A lack of people to work on the solution
- The lack of a technology to give you leverage in solving the problem
- Political problems which inhibit people from cooperating in solving the problem
- Environmental problems, which inhibit a solution due to the nature of the business, technical environment, geographic environment, etc.

Once the solution constraints are known, they can be addressed. It is common to see only a partial relief of constraints. In fact, sometimes no matter what you do, the constraints are out of your hands and you just have to figure out a way to work around them. As I have been known to tell my test teams, "When the going gets tough, the lazy get creative."

Conclusion

This brief article only scratches the surface in what it takes to *understand* the problem to be solved. There have many books written on the topic, but experience is often

the best teacher. If you don't already have a mentor, find someone who has some gray hair and is good at solving problems and watch them in action as they work to identify the real problem and set priorities. Be warned, however, these people are in a very small minority. Based on what I've seen in many companies, I would look outside of the IT organization and perhaps, even outside of a corporation.

I know that in some ways this article may be frustrating to those who are looking for ways to solve problems right away, but I hope this article has given you a new or greater appreciation of analysis before solution. I would like to hear about your experiences and ideas in problem analysis. Feel free to write me at rice@riceconsulting.com.

Checklist for Problem Analysis

1. Is there a standard for problem definition in your organization?
2. Has the problem been described in writing?
3. Has the problem been reviewed by people who are knowledgeable about it?
4. Has a root cause analysis been performed?
5. If so, were the appropriate people involved in the analysis?
6. Has a standard set of context-free questions been developed for your organization?
7. Have context-free questions been asked about this particular problem?
8. Have all the appropriate people been queried?
9. Have the context -free questions been answered to your satisfaction?
10. Have false assumptions been identified?
11. Has the scope of the problem been determined?
12. Have the affected people been identified?
 - Internal
 - o Sponsors
 - o Users
 - o Technical people
 - o Project managers
 - o User management

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

- External
 - o Customers
 - o Users
 - o Suppliers

13. Has the scope of the solution been defined?

- Internal
- External

14. Have constraints to the solution been defined?

- Time
- People
- Money
- Politics
- Technology
- Environment

15. Have modeling techniques been used to graphically understand the process or product described?

- UML
- E-R models
- Data Models
- Process flows

Resources for Problem Analysis

Are Your Lights On? By Gerald Weinberg

Problem Frames and Methods: Structuring and Analyzing Software Development Problems By Michael Jackson

Root Cause Analysis : Simplified Tools and Techniques

By Bjorn Andersen (Editor), Tom Fagerhaug, Bjorn Anderson, Birn Andersen

Root Cause Analysis: Improving Performance for Bottom Line Results By Robert J. Latino, Kenneth C. Latino

Apollo Root Cause Analysis - A New Way Of Thinking

By Dean L. Gano

The Root Cause Analysis Handbook -A Simplified Approach to Identifying, Correcting, and Reporting Workplace Errors

By Max Ammerman

Tool News

by Randy Rice, CQA, CSTE

Starbase Corporation (www.starbase.com) signs definitive agreement to acquire TBI, Inc. TBI, Inc. develops and markets Caliber-RM®, a leading collaborative requirements management system for eBusiness and enterprise application development.

TBI's Caliber-RM is a collaborative, Web-based requirements management system that enables teams to work together efficiently on project requirements and ensures that applications will meet end-user needs. Caliber-RM stores requirements and their attributes in a central repository that every project team member can access to view, add, modify and delete the dynamic requirements that impact the software development life cycle, whether they are connecting from a Windows or a Web-based client.

Starbase provides end-to-end collaborative products for both the development and management of integrated content and code for e-business applications. Collaboration is the simultaneous coordination, management and communication of geographically dispersed contributors of both code and content for e-business applications.

With more than 2,500 Starbase customers worldwide, the company's technology supports the continuous cycle of creating, linking and managing digital assets, which comprise complex e-business applications. Starbase products enable users with differing technical and functional backgrounds to collaborate on the production and management of Web sites and e-business initiatives from multiple locations. Current Starbase products include StarTeam, StarEstimator and Roundtable, among others.

Starbase has headquarters at 4 Hutton Centre Drive, Suite 900, Santa Ana, Calif. 92707.

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

Frequently Asked Questions

by *Randy Rice, CQA, CSTE*

Q: How would I justify the use of the following tests for a database application?

- Functional testing
- Stress testing (which includes testing concurrent users)
- Volume testing
- Configuration testing (for database configuration)
- Regression testing (both during the project and during maintenance)
- Security testing
- Recovery testing
- Documentation testing
- Human factors testing (aka Usability testing)
- Installation testing
- Unit testing
- Integration testing
- System testing
- User acceptance testing

A:

- **Functional testing** - This is the most critical to make sure calculations and data updating is correctly performed.
- **Stress testing** (which includes testing concurrent users) - Since I don't know how many users will be on the system, it's hard to say how important this will be for you. However, if performance is bad, it only takes 2 or 3 people to bring down the entire application!

- **Volume testing** - Once again, I don't know how many employees you are dealing with, but this is an important test to make sure the batch routines (such as paychecks and backups) can run efficiently.
- **Configuration testing** (for database configuration) - The importance of this depends on the different types of hardware and operating systems that the system will be deployed upon. Also, it validates that the application logic works correctly with the DBMS.
- **Regression testing** (both during the project and during maintenance) - This is essential to make sure any changes to software, equipment, data indexes, etc do not cause an adverse effect on the overall system.
- **Security testing** - For a payroll system, this protects the confidentiality of employee data and validates that unauthorized people cannot access functions such as pay rate.
- **Recovery testing** - This is very important for database applications to validate that the system can be recovered with minimal or no data loss in the event of a system failure.
- **Documentation testing** - This validates that the user documentation accurately tells users how to use the application. People can make big mistakes when they don't know how to use the application correctly.
- **Human factors testing** (aka Usability testing) - This is also important to tell if people are struggling with the user interface of the system. Struggling users = inefficiency of operation.
- **Installation testing** - This depends on how many locations you will be installing the system. If only one, then you need to have some way to test that the installation was performed correctly at the one location before people start using it.

Links to Check Out...

PROBLEM SOLVING WITH PAST Problem Analysis and Solving Technique

<http://itaxle.virtualave.net/isdp/fundamentals/past/past.html>

Problem Analysis Chapter from Intro to Software Engineering at Middle East Technical University

<http://www.ii.metu.edu.tr/~is507/lectures/chapter5/ch5.htm#5.2>

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



Automated Testing

by Carl Chandler

One of the most frequent asked questions we are asked here at Rice Consulting Services is "What are the key things that need to be done to automate testing?"

Although the list could be longer, here are five tips that will serve you well as you travel along the automated testing trail:

1. *Select a tool that fits your technical and usage needs.* This might seem obvious, but I have seen more than one situation where the tool was bought on other criteria, such as price. The technical fit is one level of evaluation criteria and the tool usage is another. For example, the key distinction between the tool you eventually purchase may be ease of use, script maintainability, the number of test activities supported (e.g., test planning, test execution, defect tracking, etc.).
2. *Have a testing process in place to automate.* After the novelty of an automated tool wears off, the next challenge is to increase your productivity and effectiveness. Productivity increases come from knowing when and how to test. Effectiveness comes from knowing what to test. No matter what kind of automated tool you have, if you only cover a

fraction of the functionality, your test will be incomplete and lead to a false sense of confidence.

3. *Pay attention to the human issues.* After years of having new tools and methodologies introduced to them, developers and testers are often reluctant to change work habit. No matter how well the tool performs, it takes people to use it. You need to make sure people understand both the process and the tool, and how it can make the job of testing easier and more reliable.
4. *Reuse your testware.* This means build your test plans, test procedures, and test cases in such a way that they can be used by other people. Much time and effort is wasted when people duplicate test products. If you build modular test procedures, you can combine them in a multitude of ways to achieve many different tests with fewer 'procedures.
5. *Measure your results and use the measurement to improve the process.* Many people measure basic things such as defects and time spent on testing. However, only a minority (if people actually do anything with the measurements to improve the development or testing process. You have invested a lot of money in those defects, so you might as well learn from them!

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

The thing to remember is that a test tool will help you test more and faster, but without a testing process in place you will only be doing a bad test faster! As stated in our Basic Training in Software Testing class - the best test tool is your brain.

Rice Consulting Services' Consulting Offerings:

Testing Assessments

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

Rice Consulting Services' Course Offerings:

If you would like to learn more about the information covered in both Randy's and Carl's articles we here at Rice Consulting Services, Inc. offer two excellent courses that will enhance your companies software quality process.

A Workshop for Gathering, Documenting, Managing and Validating Requirements

— 2 days

This workshop is designed to provide a basic understanding of the requirements process from beginning to end. The sessions are designed to take the students through methods for analyzing the problem to be solved, gathering user input to determine user needs, documenting the requirements for the problem solution, managing the entire requirements lifecycle, and testing the requirements. Each module is reinforced with group exercises to allow the students immediate application of the concepts presented in each module. At the conclusion of the workshop, a question and answer session will be conducted to deal with some of the most frequently asked questions.

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.

Basic Training in Software Testing — 2 days

This is 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. This is a practical hands-on seminar to cover the critical path of testing. Your instructor will be a certified instructor in the QA and testing field. You will learn the terminology, process, and challenges of testing in the real world. 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.

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

Book Review

by Randy Rice, CQA, CSTE

Improving Data Warehouse and Business Information Quality: Methods for Reducing Costs and Increasing Profits

By Larry P. English



Paperback - 544 pages 1 edition
(March 11, 1999)

John Wiley & Sons; ISBN: 0471253839 ;
Dimensions (in inches): 1.35 x 9.17 x 7.46

I get a lot of questions about how to test data warehouses. I can understand the interest, as data warehouses add a level of size

and complexity to information systems. However, what intrigues me is why data quality is seen by some to be more important to data warehousing than other information systems technologies. This is not a criticism of this book by any means. I think it is a fine and valuable treatment of the topic of data quality. My point is that this book is valuable in any system context because all systems rely on data.

In my opinion, how the data in a system is treated in a system is the ultimate measure of IT quality. True, quality processes are very important, but if the data is wrong, the system is wrong. IT system quality is not the control of any particular component, but the result of a disciplined approach to the entire system.

This book adds a lot of understanding by relating Dr. Deming's 14 Laws of quality to IT quality. This is a very important topic because cultural and process problems can offset any tool or technique.

What I Liked About This Book

I liked this book because it builds bridges between very important concepts, such as data, process, products, measurements and assessments. I also liked how Larry gives a lot of concrete examples and templates to use in the real world. You don't have to be an experienced quality professional to immediately understand and use this book. In fact, this is a great book for people just getting into QA (I am referring to QA as quality management, not testing). Experienced quality professionals will find this book as a very helpful tool in solving all types of information quality problems.

Scoring

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

Summary

People in IT at all levels need to understand and appreciate the role of quality data in their systems. This book is a must-read to gain and apply that understanding.

May 2001 Issue:

- **Testing Wireless Applications**
by Randy Rice, CQA, CSTE
- **Test Leadership**
by Carl Chandler

Quotes of the Month

“Are you gonna get any better, or is this it?”

Baltimore Orioles Manager Earl Weaver to an umpire

“The best measure of a man's honesty isn't his income tax return. It's the zero adjust on his bathroom scale.”

Arthur C. Clarke

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