

The Software Quality Advisor Online



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



Questions & Answers by Randy Rice, CQA, CSTE

This month instead of an article, I would like to take this opportunity to answer some questions. If you have a software quality question, e-mail me at rrice@riceconsulting.com.

Question:

“Could you please define what type of testing gray, black, and white box testing is?”

Answer:

These terms get thrown around quite a bit by people in different ways, but the most common sage is that white box testing is based on the internals of what is being tested (aka structural testing), black box testing has no knowledge of the internals (such as testing a software package), and gray box testing is a situation where you have a limited insight to the internals, but still have to rely a lot on external functional testing.

Question:

“Do you know of any other websites which can help with formal specifications?? Its an area I'm interested in and would like to further my knowledge.”

Answer:

You might check the following links:

<http://www.sei.cmu.edu/publications/documents/91.reports/91.tr.021.html>

http://satc.gsfc.nasa.gov/support/STC_APR97/write/writert.html

<http://fit.edu/~jskim/se.html>

<http://stsc.hill.af.mil/> (Click on Library and look for "Requirements Analysis and Design")

<http://fast.faa.gov/flowcharts/testflow/sys60D.htm>

<http://www.epri.com/eprisoftware/roadmap/swspec.html>

<http://www.keycomputerconsultants.com/kcc/kccspec.html>

<http://www.construx.com/survivalguide/desspec.htm>

<http://cohesion.it.swin.edu.au/teamb/documentation/srs/index.shtml>

Question:

“What is Integration Test?.”

Answer:

The standard definition of integration testing is the testing of interfaces between software units. However, the scope can also be defined as the testing of points of integration between systems as well.

Question:

“I read your article about UAT testing. I am myself a software engineer. At this point I am not very clear about the difference between the UAT & system testing. In system testing we test as per the functional requirement & In UAT the same? Can you please give example & explain?”

(Editor’s note: The article referred to is called “User Acceptance Testing: Is it or Isn’t It?” and be found at

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

Answer:

System testing is a producer view of building software. The system as a whole is validated against the system requirements. System testing can include functional testing, performance testing, security testing, etc. For example, a system test of a banking application would focus on the correctness of the transactions, correctness of calculations, correctness of the data conversion, testing the performance of the systems, etc. System developers and testers would look to the requirements to design the test cases and to evaluate the correctness of results.

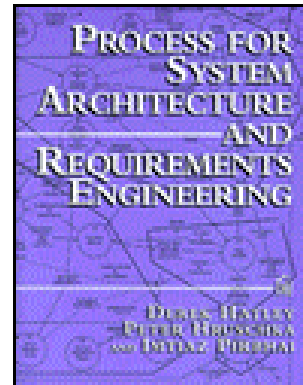
UAT is a user/customer view of validating that a system or application actually meets their needs in the real world. The users don't care what the specs or requirements indicate - they want to know if the application will work in their day-to-day work. Therefore, the basis of testing is not requirements, but business processes and business cases. UAT is more concerned with testing business processes as opposed to field edits and other low-level tests. How many times have developers built the wrong system in terms of meeting user needs? The UAT would test transactions designed to simulate real-world transactions in the business.

Here's one more example outside of the computer realm: When a car is built, workers inspect each component as it is assembled. This is like verification. When the car rolls off the line, there is an overall inspection to ensure the car works according to design and specs. This is like system testing. Finally, when the customer buys the car, they take a test drive to make sure the car handles

the way they like, make sure there is ample headroom, etc. However, the user will not make sure all of the belts and hoses are installed correctly and the engine tolerances are correct - they don't know how to check them and they can't check them. All of the above activities must be done to deliver a quality product, but they each have a different focus and objective.

Book Review

Process for System Architecture and Requirements Engineering



Format: Paperback, 456pp.

ISBN: 0932633412

Publisher: Dorset House Publishing

Pub. Date: September 2000 ★★★★★

Overview

Process for System Architecture and Requirements Engineering presents a broad approach to developing systems which involve multiple disciplines. The techniques in this book build on those published by Derek Hatley in his book, *Strategies for Real-time System Specification*. As technologies have changed since 1988, this book brings a new context to the original methods, known as the Hatley/Pirbhajani methods.

Process for System Architecture and Requirements Engineering does a very good job in presenting a cohesive approach to using models and

requirements to build systems. One of my big complaints back in the popular days of CASE technology was that all of the CASE methodologies minimized or ignored the role of requirements. "From cartoons to code." Was the mantra in those days. After studying the debris from enough failed CASE projects, people finally conceded that more than models were needed to build a system. This book builds the bridge between a model-based design approach and requirements engineering/management.

The book is presented in two parts: concepts and a case study. I thought this was an effective way to illustrate the concepts.

Who Will Benefit From this Book

- System designers and architects
- Testers wanting to understand a sound development process
- Requirements engineers

What I Liked About This Book

At the outset of the book, in big bold print appears the following statement:

"Methods and automated tools are of no use without properly qualified people who are using a well-defined development process, and who are dedicated to satisfying customer needs."

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

I liked that and agree wholeheartedly.

I also liked the generous use of diagrams and case studies to reinforce the concepts in the book.

Would Have Been Nice to Seen

Realizing that space is often a limiting factor in writing a book, I would liked to have seen more on the human side of requirements definition and management process. Also, I could only find one reference to testing in the book, besides the definition in the glossary. It would have been nice to read the authors' thoughts on the verification and validation of the design models and requirements. On the plus side, there are plenty of books on the verification and validation topics, so this does not keep me from recommending the book.

Topics

Part I: Concepts

1: Introduction

2: What Is a System?

3: A Framework for Modeling Systems
Exploiting System Hierarchies

4: System Development Models
Requirements/Architecture Relationships
A Note on Object Orientation

5: The System Development Process
Process, Methods, and Tools

6: Applying the Models to Development
The Generic Development Structure
Hospital Monitoring System
Completing the Architecture
Numerous Hardware Technologies

7: System Development Overview
A Requirements Model for System Development
A Metamodel for a Development Project

Part II: Case Study -- Groundwater Analysis System

8: Initial Problem Statement
Required Capabilities

9: Fitting In the Known Pieces
System Entity/Relationship/Attribute Model

10: Building Upon the Known Pieces
Enhancing the Essential Model

11: Filling In the Blanks
Adding the Architecture Flows and Interconnects
Merging the Top-Down and Bottom-Up Pieces

12: Completing the Models
Requirements and Architecture Dictionaries

13: Groundwater Analysis System Summary

Appendix: Changes, Improvements, and Misconceptions Since the Methods' Introduction

Scoring

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

Summary

I found *Process for System Architecture and Requirements Engineering* to be an excellent resource for my clients that need an understanding of system models and how they relate to the requirements process. The concepts are presented in a clear and organized manner, with complete examples to aid in applying the concepts.

Reviewer: Randy Rice

Quotes

"Microsoft isn't the answer. Microsoft is the question, and the answer is NO."
-Anonymous

"Anything that is too stupid to be spoken is sung."
-Voltaire

"If you want a guarantee, buy a toaster."
-Clint Eastwood

"If you think that something small cannot make a difference - try going to sleep with a mosquito in the room."

-Unknown

Calendar

[QAI's International IT Quality Conference](#)

April 23 - 27, 2001

Caribe Royale Resort Suites & Villas - Orlando FL

(Randy will be speaking on Tuesday, April 24th on the topic of Validating Web Usability)

[QAI's Seattle IT E-Commerce Applications Testing Conference](#)

June 11-13, 2001

Seattle, WA

(Randy will be speaking on the topic of testing wireless applications)

For information on all of these conferences, contact the Quality Assurance Institute at 407-363-1111 or visit their web site at www.qaiusa.com.

February 2001 Issue:

- **Validating Web Usability**
by Randy Rice, CQA, CSTE
- **Stages of Process Maturity - Moving Up in Capability Maturity Model (CMM) Levels**
by Carl Chandler
- **Book Review: Improving Data Warehouse and Business Information Quality**