

Knowledge Base for Writing Test Cases (Part 1)

Name: Charles Gundlach

Date: July 19, 2008

Version #: 2.0

Course Description:

- **The purpose of this training is to present a knowledge base regarding Writing Requirements Based Test Cases**

Audience:	BAE Systems SW Test Engineers
Instruction Method:	Instructor
Length of Course:	1 hour
Prerequisites:	Prior testing experience
Follow-on Courses:	Knowledge Base for Writing Test Cases (Part 2) Writing Requirement Based Software Test Cases
Resources for Questions	Charles Gundlach

Learning Objectives

➤ **After completing this course you should be able to:**

- ✓ **Have a Basic Understanding of Requirements and Their Evolution**
- ✓ **Have a Basic knowledge of the Software testing Process**
- ✓ **Have a Basic Understanding of how to determine steps needed to fully verify a Requirement (considering boundaries, risk assessment, etc.)**
- ✓ **Have a Basic Understanding of Writing Good Requirement Based Test Cases**

**A REQUIREMENT is a
CONTRACT
Between a
Customer
And
A
Supplier/Developer**

Requirements

➤ Caveat:

Software Test Engineers do not typically write requirements, but they are (and definitely should be) involved in reviewing them! This Section is to provide information regarding Requirement Writers and the characteristics of, and the rules regarding Good Requirements.

➤ About Requirement Writers:

- ✓ Requirement writers are normally either “Business Analysts” or “System Engineers” (SE).
- ✓ An SE may be thought of as an information gatherer. He or she works as a liaison among stakeholders in order to elicit, analyze, communicate and validate requirements.
- ✓ An effective SE has and maintains:
 - a **clear understanding** of the needs of the users, customers and stakeholders
 - a **collaborative relationship** between the users, customers and stakeholders and the technical team

Requirements (continued)

Good Requirements Reference Card

A requirement must state: "Who is responsible" and "What shall be done"



A PRODUCT requirement is: "something the product must do or a quality that the product must have."

The system shall operate at a power level of...
The software shall acquire data from the...
The structure shall withstand loads of...



A SOW requirement is: "the definition of a task to be done or what the provider will deliver."

The provider shall perform a trade study to ...
The provider shall deliver the following documents...
The provider shall certify that...

Know where you are going before you write requirements

- Define need, goals and objective
- Identify all stakeholders
- Identify drivers
- Define all external interfaces
- Develop operational concepts
- Know risk before proceeding to requirement definition
- Share the vision
- Set expectations

Write Good Requirements

- With these characteristics:
 - One thought • Simple • Stated Positively
 - Needed • Grammatically Correct • Concise
 - Attainable (Technical, Cost, Schedule) • Verifiable
- Use correct terms:
 - Shall = requirement
 - Will = facts or declaration of purpose
 - Should = goal
- Avoid ambiguous terms
- Capture Rationale
- Define Verification method
- Ensure Traceability

**Ask us about our
Intact Team Training
for your project!**

Compliance Automation, Inc. specializes in helping our customers get their requirements right — the first time. We provide training, mentoring, consultation and facilitation to empower our customers to define the scope of their project and capture the requirements up-front where it is most cost effective.

Compliance



Automation, Inc.

(830) 249-0308

www.complianceautomation.com



Requirements (continued)

- **A requirement must be:**
 - ✓ **Attainable**
 - ✓ **Concise**
 - ✓ **Complete (standalone)**
 - ✓ **Unambiguous and Understandable**
 - ✓ **Implementation Free**
 - ✓ **Verifiable**

Requirements (continued)

➤ Requirement Sets

- ✓ When taken as a whole “the set” completely satisfies the need.
- ✓ Each individual requirement statement must be consistent with all other requirements in the “set”.
- ✓ Each requirement statement must not conflict with any other requirement statement in the “set”.
- ✓ No requirement statements are duplicated in the “set”.

Requirements (continued)

- The format of a requirement shall be:

[optional qualifier] the [product/process name] **SHALL** <direct action verb function statement> [optional qualifier]

example:

[From a standing start,] **the [vehicle] SHALL** <achieve a speed of no less than 48kph> [within 8 seconds]

Requirements (continued)

➤ Rules that Good Requirement Writers Should Follow

- ✓ Use the term **SHALL** for a requirement
- ✓ Do **NOT** use the statement **SHALL NOT** or any negatives.
- ✓ Do **NOT** start the requirement with the term **SHALL**. Specify **WHAT shall**.
- ✓ The requirement must be confirmable by analysis, inspection, test or demonstration.
- ✓ Do **NOT** use the statements **SHALL BE ABLE TO, SHALL BE CAPABLE OF** or **SHALL HAVE THE ABILITY TO** (use just plain old **SHALL**)
- ✓ Do **NOT** use the term **WHICH**. What you mean is **THAT**
- ✓ Do **NOT** use the terms/phrases **PERIODICALLY, EXCESSIVE, MINIMIZE, MAXIMIZE, QUICK, APPROPRIATE, AVERAGE, USER FRIENDLY, NOT LIMITED TO, NORMAL, SUFFICIENT, COST EFFECTIVE**. Be specific.

Software Testing

➤ Software Testing Overview:

- **Software testing** is the process of checking software, to verify that it satisfies its requirements and to detect errors.
- **Software testing is an empirical investigation** conducted to provide stakeholders with information about the quality of the product or service under test, with respect to the context in which it is intended to operate. This includes, but is not limited to, the process of executing a program or application with the intent of finding discrepancies and errors (bugs).
- **Testing can never completely establish** the correctness of computer software. Instead, it furnishes a *criticism* or *comparison* that compares the state and behavior of the product against a specification. *Software testing* should be distinguished from the separate discipline of Software Quality Assurance, which encompasses all business process areas, not just testing.

Software Testing

➤ Software Testing Overview (continued)

- **Complexity and size** of computer software has continued to grow. Every software product has a target audience. For example, the audience for video game software is completely different from banking software. Therefore, when an organization develops or otherwise invests in a software product, it presumably must assess whether the software product will be acceptable to its end users, its target audience, its purchasers, and other stakeholders. Software testing is the process of attempting to make this assessment.
- **The cost of software bugs the U.S. economy was \$59.5 billion** annually in a study conducted by NIST in 2002. More than a third of this cost could be avoided if better software testing was performed. **Of course, at BAE serious bugs can potentially cost lives.**

Software Testing

➤ Software Testing Overview (continued)

- **A primary purpose for testing** is to detect software failures so that defects may be uncovered and corrected. This is a non-trivial pursuit. **Testing cannot establish that a product functions properly under all conditions but can only establish that it does not function properly under specific conditions.**
- **The scope of software testing** often includes examination of code as well as execution of that code in various environments and conditions as well as examining the quality aspects of code: does it do what it is supposed to do and do what it needs to do. In the current culture of software development, a testing organization may, and is preferred to be separate from the development team. There are various roles for testing team members.

Software Testing

➤ Software Testing Overview (continued)

- **Defects and failures** - The software faults occur through the following process. A programmer makes an **error** (mistake), which results in a **defect** (fault, bug) in the software **source code**. If this defect is executed, in certain situations the system will produce wrong results, causing a **failure**. Not all defects will necessarily result in failures. A defect can turn into a failure when the environment is changed. Examples may include alterations in source data or interacting with different software. A single defect may result in a wide range of failure symptoms.
- **Input combinations and preconditions** - A problem with software testing is that testing under all combinations of inputs and preconditions (initial state) is not feasible, even with a simple product. This means that the number of **defects** in a software product can be very large and defects that occur infrequently are difficult to find in testing.

Software Testing

➤ Software Testing Overview (continued)

- **Static vs. dynamic testing** - There are many approaches to software testing. **Reviews, walkthroughs** or **inspections** are considered as **static testing**, whereas actually executing programmed code with a given set of **test cases** is referred to as **dynamic testing**.
- **Software verification and validation** - Software testing is used in association with:
 - ✓ **Verification**: Have we built the software right (i.e., does it match the specification)?
 - ✓ **Validation**: Have we built the right software (i.e., is this what the customer wants)?
- **Software Test Engineer** - **Until the 1950s the term "software tester" was used generally. Now, it is widely viewed as a separate profession, one that requires new knowledge, practices, techniques., and attitudes.**



Software Testing

➤ Communication:



- **Static Testing (Reviews, walkthroughs or inspections)** uncover problems and defects early on, and therefore **saves the most money on any project.**
- **Test Case planning and design** (to be covered later) also initially employs these same techniques **(Reviews, walkthroughs or inspections)** Inherent in the success of these techniques is the involvement of all stakeholders with a major interest in the project. **To succeed, rapport between stakeholders and the desire to cooperatively and supportively communicate is paramount. This is especially true between the SW Test Group, the Doors people, Software Development, and the CSIL!**

Software Testing (continued)

➤ Traceability:

- **Traceability is critical** – a reason (basis) for every test and where the reason comes from – the source of the need for testing. Every a test is run, the reasons for it - requirements or other source – must be know and documented. If a test fails, exactly which requirement or requirements are in danger is known. If a requirement changes, exactly which test cases are affected are known. Traceability documentation allows for reviews and early discovery of errors.
- Traceability documents decisions (reasons and choices) made during Risk Analysis. How the decision for running this test or that test is made. Use traceability to document the reasons for the choices made in risk analysis. Which test cases to run can be drawn from a prioritized list of those chosen. **Risk Analysis typically is the basis for choosing which tests should be run during Regression Testing.**
- Stakeholders, developers, CSIL, etc., all need to participate.

Software Testing

➤ Understanding Software Testing Principles

(to help facilitate decisions regarding SW Testing and the SW Testing Group)

❖ Complete Testing is not possible

- In any form of testing it is impossible to achieve total confidence
- The only exhaustive testing is when there is so much testing that the tester is exhausted.

❖ Testing Work is creative and difficult

❖ False Beliefs about testing

- Testing is easy
- Any one can do testing
- No training or prior experience is not needed

Software Testing

➤ Understanding Software Testing Principles

(to help facilitate decisions regarding SW Testing and the SW Testing Group)

❖ Why testing is not simple

- To test effectively you need to understand the system in depth
- It requires special knowledge and attitudes

❖ Essential ingredients to Good Software Testing

- **Updated Software Testing Education** is very important
- **Testing experience** is very important
- A “tester mindset” is important
- Testing methodology is important

Course Summary

- You learned about:

References