Writing Testable Requirements

The Writing Testable Requirements course is a three-day techniques and process-oriented class that focuses on problem avoidance by teaching how to write requirements correctly the first time. This course furnishes writing style guidelines for describing all types of processes and data. It also will ensure that requirements are written to a level of detail required to ensure that a sufficient set of test cases to validate the system's functionality are designed from the requirements.

For those using Agile methodologies the course will interweave how the various topics apply in getting from User Stories to detailed Functional Specifications by the end of the release. This ensures that no technical debt is accrued from release to release. It also addresses creating an end state vision of the application's scope giving you a portfolio view. This makes it easier to manage the master backlog.

Writing Testable Requirements is student-paced and participants are encouraged to bring sample from their own projects to evaluate during class. The third day of the class is a working session spent on an expanded set of class exercises and/or working on requirements from the students' projects.

Objectives

- Provide a set of practical guidelines for writing requirements which ensure that the requirements document is testable and that sufficient test cases can be created from the requirements
- Understand the limitations of the English language and how to quantitatively measure the quality of a specification via Ambiguity Reviews, including a detailed ambiguity review checklist.
- Understand the importance of clear objectives, identify good objectives, differentiate objectives from solutions and describe how an objective can be subverted
- Understand the functions/features of Requirement Management tools and how they support the requirements definition and maintenance effort

Prerequisites
None.

Materials Provided
Student manual containing the course slides and exercises.

Intended Audience

Anyone writing, reviewing, or contributing to the content of the requirements and design specifications.

Duration: Three days

Class Limit: 25 students

Exercises: Over 60% of class

Course Outline

Section 1 - Introduction
- Definition of requirements
- Why good requirements are critical
- Impact on costs of development
- Impact on schedules

Ambiguities of English
- Section 2 - Part I: Confusing Constructs
  - Dangling Else's
  - Ambiguity of Reference
  - Scope of Action
  - Omissions
  - Ambiguous Logical Operators
  - Negation
  - Ambiguous Verbs, Variables, Adjectives, Adverbs
  - Built in Assumptions
  - Ambiguous Precedence Relationships
  - Implicit Cases
  - Temporal Ambiguity
  - Boundary Ambiguity
  - Conducting and Tracking Ambiguity Reviews
- Section 3 - Part II: Jargon and Unnecessary Complexity
  - The Language Barrier
  - Carelessness
  - Assumed Functional Knowledge
  - Jargon
  - Unnecessary Complexity

Section 4 - Defining Clear Objectives
- What Are Objectives
- System/Product Objectives Versus Project Objectives
- Types of Objectives
- Measuring Project Success: Time, Function, Resources, Quality = Return on Investment
- The Objectives Document Template

Section 5 - Requirements Specification Definition
- Classes of Requirements
- The Components of a Requirement
- Physical Organization of Requirements
- Applying an Iterative Approach to Requirements
- The Requirements Document Template

Section 6 - Naming Conventions
- The Importance of Good Naming Conventions
- Names Show Class Membership
- Names Show Data States
- Structuring Good Names
- Dealing With Acronyms and Aliases
- Glossaries

Section 7 - Documenting Data
- Documenting Entities and Entity Relationships
- Documenting Data Structures
- Normalized Data Model
- Documenting Command Line Parameters
- Documenting UI Standards
- Documenting Reporting Standards
- Detailed Attributes of Data Stores
- Detailed Attributes of Data Flows
- Detailed Attributes of Data Elements
- Templates for Data Flows, Data Stores, and Data Elements

Documenting Processes
- Section 8 - Part I: Process Modeling
  - Creating the Initial Process Model
  - Use Cases and Functions
  - Benefits of Good Process Models
  - Process Packaging Rules
  - Process Description Templates
- Section 9 - Part II: Process Descriptions
  - Style suggestions for readability
  - Describing decision logic
  - Describing transforms
  - Documenting state machines
  - Documenting asynchronous processes
  - Impact of physical design on the external specification
  - Structured English versus pseudo code
  - Review of eliminating ambiguities
- Section 10 – Additional Considerations
  - Documenting State Machines
  - Handling Asynchronous Processes
  - The Impact of Design on Requirements

Section 11 - Tuning the Process by Project Type
- Rapid Prototyping, Rapid Application Development, Agile Methodologies
- New Development
- Maintenance
- Third Party Packages
- Technology Ports
- Rewrites, Re-Engineering, and ERP

Section 12 - Introduction To Requirements Based Testing
- The Test Case Design Challenge
- Overview of the Validation and Verification Steps
- Test Considerations and the Order of Writing Requirements
- Test Design Strategies
- Using Test Cases to Validate the Requirements

Section 13 - Requirement Management Tool Support
- Collaboration
- Version Control
- Traceability
- Planning, Estimating, Change Control
- Data Base versus Document Centric
- Added Value Features

Section 14 - Summary

- Review of the Key Characteristics of Testable Requirements
- Review of Course/Class Objectives
- Defining the Path Forward

Bender RBT Inc.
17 Cardinale Lane
Queensbury, NY 12804
518-743-8755
rbender@BenderRBT.com