Testability Analysis

Introduction

Testability analysis is an early lifecycle check applied to requirements in order to ascertain if the requirements can be proven by testing (this applies equally to customer requirements and functional specifications). If a tester cannot identify how the requirements can be tested then it is likely that the designers and developers will also have problems when it comes to designing and building a system that can meet those requirements. Some processes take this further and say that if a requirement is not testable it MUST be either re-written in terms that are testable, or the requirement must be removed.

During formal review of requirements, testability analysis is one activity that should be considered for inclusion along with missing requirements analysis.

At a high level the Requirements documents will be checked for the following:

Missing requirements analysis

- Are all requirements specified? (No business areas missed or incomplete)
- Have the Non-functional requirements been documented? (Performance, Volumes, Stress, Usability, Availability and Reliability requirements, etc.)
- Is the document internally consistent and consistent with all related documents?
- Have the requirements for failure paths as well as success paths been documented? (What error messages, when and under what conditions)

Testability analysis

During Testability Analysis, an experienced test analyst will be looking at each requirement and asking the following questions:

- Have the requirements been written so that each requirement can be uniquely identified? (essential for traceability between documents and test cases back to source document – the test basis reference)
- Is each requirement uniquely specified? (no duplication of requirements throughout the document which can lead to confusion and duplicate test cases)
- Is this requirement testable? (can I see how to test it, what test inputs and system actions are required and what the expected results should be)
- Is the test achievable, how many tests will be required to prove it, what techniques can be applied, what test data will be required, do we have the knowledge, the skill set, the tools, the time and budget for this test?
- What is the business risk related to this requirement (essential to understand depth of testing and test coverage requirements)
- Have the testing needs been considered? (Software systems can be designed and built to be testable; cheats in PC Games are coded into the game to allow the testers to easily test the game without having to spend extended timeframes learning the game!). The physical implementation of the system can also heavily impact the amount of testing required free text fields v drop down lists for example.
Some organisations use the SMART Acronym when carrying out testability review:

**Specific**

Is the requirement at the correct level? Does it contain enough detailed information to understand what the requirement actually means and what the customer wants?

**Measurable**

Can the solution be measured to prove that the requirement has been met in the final product? All requirements must have objective measures that can be applied and proven, rather than subjective statements that cannot. i.e. ‘the system response time shall be fast’ is subjective and cannot be measured objectively because two individuals will have a different interpretation of what fast actually is! Alternatively ‘the screen will complete the display of the search information within 3 seconds of the user clicking the mouse to initiate the search, with 1000 users on the system’ is objective and therefore measurable.

**Achievable**

The requirements must be achievable. Asking for the impossible does not mean that the impossible can be delivered. i.e. ‘the system shall be available for use at all times’ is not achievable. There is no such thing as a fault free system and with the best design, build, maintenance and support possible the system will still be unavailable from time to time. In order to test this you would need to test forever and make sure it is never unavailable!. A more achievable target would be to define acceptable numbers of failures in live over a set time period (no more than n failures in the first 6 months of live) or maximum acceptable unavailability (target for availability is 99.9% availability).

**Realistic**

The requirements must be realistic in terms of the test teams ability to prove that they have been met. i.e. ‘the web site will work with all combinations of client hardware, software, comms links, modem types, desk top configuration and browsers.’ Although this is possible to prove it is not realistic to check all of these due to cost, timeframe and resource requirements. A specified minimum set of combinations and a site disclaimer are a more realistic approach to specifying supported versions and combinations.

**Time bound**

The test team must be able to prove all the requirements within the timeframes available. If the expectation is unachievable before we start then the project is doomed to failure. Testers should be able to present the case for what can be done and what cannot be done within the timeframes allocated. If there is not enough time to test everything then the decision on where to cut tests is made based on the risk of failure in live. This decision is made by the customer and the project manager not the tester in isolation. The more the scope of testing is reduced the greater the risk of failure in live.
Conclusion

There are many benefits to performing early testability analysis:

1. It confirms completeness and establishes the quality of the documentation under analysis
2. It allows tester input at the documentation stage so any requirements changes or updates only require documentation rework.
3. It familiarises the testers with the system requirements at an early stage
4. If a tester can see how a particular requirement should be tested then there should be sufficient information for the technical resources to understand it also (for delivery purposes)
5. It fosters stronger relationships between testers and the other project team members at the earlier stages of the project.

Testability analysis is a fairly simple activity, not very time consuming and can add real value. Every organisation that is serious about testing should encourage testability reviews as part of the standard delivery process.