1.4.2_4 Test Design

Test design has the following major tasks, in approximately the following order:

• Review the test basis (such as the product risk analysis, requirements, architecture, design specifications, and interfaces), examining the specifications for the software we are testing. We use the test basis to help us build our tests. We can start designing certain kinds of tests (called black-box tests) before the code exists, as we can use the test basis documents to understand what the system should do once built. As we study the test basis, we often identify gaps and ambiguities in the specifications, because we are trying to identify precisely what happens at each point in the system, and this also prevents defects appearing in the code.

• Identify test conditions based on analysis of test items, their specifications, and what we know about their behavior and structure. This gives us a high-level list of what we are interested in testing. If we return to our driving example, the examiner might have a list of test conditions including 'behav ior at road junctions', 'use of indicators', 'ability to maneuver the car' and so on. In testing, we use the test techniques to help us define the test condi tions. From this we can start to identify the type of generic test data we might need.

• Design the tests (you'll see how to do this in Chapter 4), using techniques to help select representative tests that relate to particular aspects of the soft ware which carry risks or which are of particular interest, based on the test conditions and going into more detail. For example, the driving examiner might look at a list of test conditions and decide that junctions need to include T-junctions, cross roads and so on. In testing, we'll define the test case and test procedures.

• Evaluate testability of the requirements and system. The requirements may be written in a way that allows a tester to design tests; for example, if the per formance of the software is important, that should be specified in a testable way. If the requirements just say 'the software needs to respond quickly enough' that is not testable, because 'quick enough' may mean different things to different people. A more testable requirement would be 'the soft ware needs to respond in 5 seconds with 20 people logged on'. The testabil ity of the

system depends on aspects such as whether it is possible to set up the system in an environment that matches the operational environment and whether all the ways the system can be configured or used can be understood and tested. For example, if we test a website, it may not be possible to iden tify and recreate all the configurations of hardware, operating system, browser, connection, firewall and other factors that the website might encounter.

• Design the test environment set-up and identify any required infrastructure and tools. This includes testing tools (see Chapter 6) and support tools such as spreadsheets, word processors, project planning tools, and non-IT tools and equipment - everything we need to carry out our work.