

# FACTFILE: GCE SOFTWARE SYSTEMS DEVELOPMENT

## (iv) TESTING

### </> Testing

#### Learning outcomes

Students should be able to:

- assess the significance of testing throughout the development cycle, particularly at the design stage;
- discuss the different types of testing that can be applied throughout the development cycle including programme, system, white box, black box, integration, user acceptance, alpha and beta;
- explain the role of the personnel involved in testing procedures;
- examine the concept of a test plan and its likely content;
- develop and apply test plans for specific scenarios.

#### Content

**Testing is:**

- Used to detect the *presence* of **errors/bugs/defects** by **executing** a software system and observing its behaviour; it does not show the complete *absence* of errors!
- One type of quality control activity the software development team undertakes to validate and verify the correctness, consistency and completeness of the software product. They may ask: Is it 'fit for purpose'? Does it do what it is expected to do?
- Monitored by project managers to ensure the software development process will produce a high quality product.

The emphasis in modern software development processes is on preventative testing; that is running tests early in the software development process so that quality can be monitored and processes optimised. Fixing errors/defects at an early stage takes less time and costs less money.

## The Testing Process

The testing process does not just involve running tests; it includes other activities such as:

- **Planning** the tests, Consider:
  - What needs to be tested?;
  - Who will be involved in the testing activities?;
  - When will testing take place and how long will it last?;
  - How the testing will be performed?
  - What resources will are needed?
- **Designing** the tests: Someone will need to design the tests or test cases and predict the expected results for each test.
- **Setting up** the test environment: The environment for testing needs to be prepared. Hardware and software will need to set up and test data and databases created.
- **Executing** the Tests: Someone will need to run the tests.
- **Recording and reporting** test results: The testers should store the results in a common format and in a specified location; various stakeholders in the project will be interested in the results of the tests.

## Planning the Tests

In the traditional waterfall model testing was left to the end of the project after implementation, just before release to the customer. Errors were often not discovered until very late in the development process and correcting these errors usually consumed a lot of time and cost a lot of money. It was recognised that the testing process could actually begin much earlier in the software development lifecycle. The 'V-model' was suggested which showed that different levels of testing could start as soon as each project phase was completed, see Figure 4.1.

For example:

- When the user requirements or needs are being discussed with the customer/client, acceptance tests could also be discussed and designed.
- When the system requirements specification is created, outlining the functional and non-functional requirements, the system tests could start to be outlined.
- When the design specification for each subsystem is created the integration tests for these could be designed.
- When the programmer has the detailed module specification, the unit tests can be designed (even before coding starts).

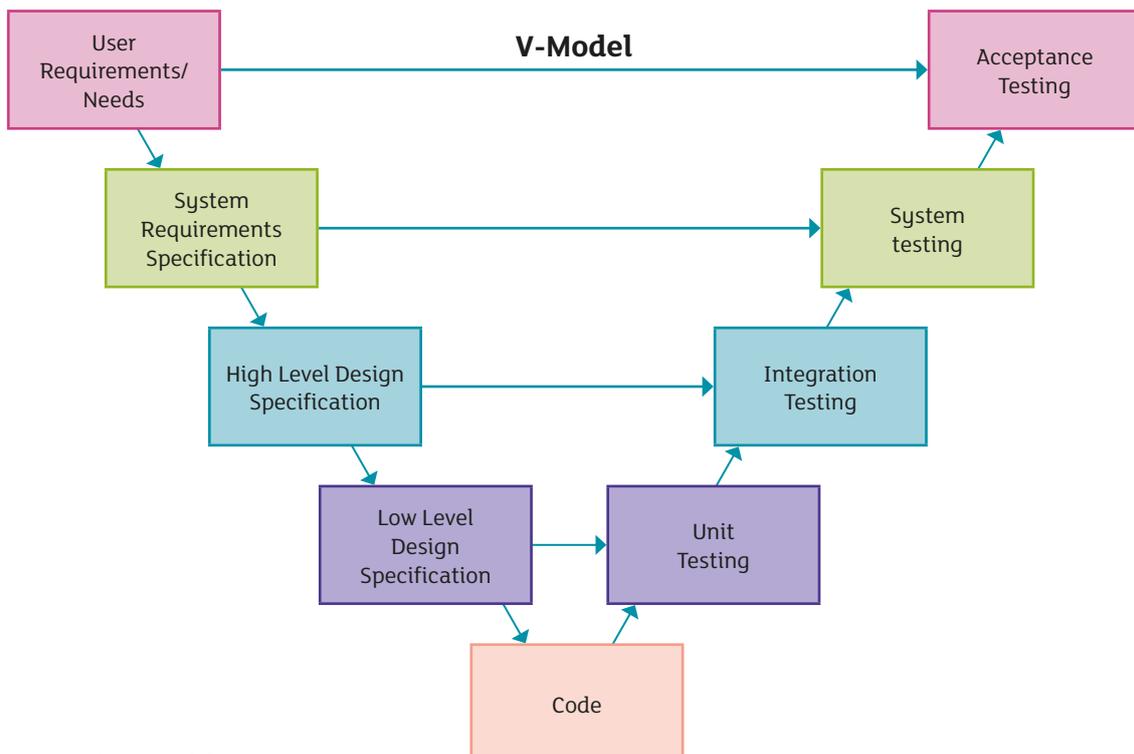


Figure 4.1 - The 'V' Model



## The Test Plan

The analyst or another person with a similar role, who understands the analysis and design of the system, should be involved in the creation of the test plan. It should contain various sections, which cover all possible testing activities, including for example:

- **Test Items:** the items that will be tested (for example databases, functions, forms, menus, classes); if something is not tested then explain why it has been left out.
- **Approach**<sup>1</sup> which includes:
  - **Levels and type of testing:** module, integration, system, acceptance, white box, black box, alpha, beta.
  - **Software Tools** used to create, execute and store the test cases. Who is the vendor? What version is used? What is their help desk number?
  - **Records:** A format for recording the testcases may be specified, for example:

| Test Case ID | Test Case Description | Input Data | Expected Result | Actual Result | Pass/Fail | Remarks |
|--------------|-----------------------|------------|-----------------|---------------|-----------|---------|
|              |                       |            |                 |               |           |         |

Table 4.1 – Sample Test Plan

- **Test Data:** How much test data is required for each level of testing? In unit testing the programmers may create their own test data but for system testing a large database of test data may need to be specially created.
- **Environment:** The physical location/space for testing needs to be identified and equipped with hardware and software. Unit testing may use the developer's computer but acceptance testing may involve setting up a large computer lab with many computers simulating the user's real work environment.
- **Testing Schedule:** A testing schedule/timetable should be created which lists all the testing activities (planning, designing, setting up, executing, and reporting), who will perform each activity and when resources (hardware and software) are required. The cost of testing should be accounted for in the budget.

<sup>1</sup> Some organisations may have a separate Test Strategy which details the approach to be used across all projects in their software development consultancy. The project manager may simply reference this strategy in their test plan and state how it applies to their particular project.



## Program Testing<sup>2</sup>

| <b>What is tested?</b> | The item that is tested is small for example a C# or Java class, a web page or menu.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |          |                   |                |                  |                    |                       |
|------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|-------------------|----------------|------------------|--------------------|-----------------------|
| <b>Focus/Purpose</b>   | <ul style="list-style-type: none"> <li>• Test each item in <b>isolation</b> from the rest of the system so that testers can easily determine where the errors are occurring. Focus on the internal behaviour of the class/unit rather than how it interacts with other classes/units.</li> <li>• Does the <b>code</b> function correctly and completely (all features included) as specified in the low level design specification?</li> <li>• Minimise defects/errors that could be hard to find and correct at a later stage.</li> </ul>                                                                                                                                                                                                                                |          |                   |                |                  |                    |                       |
| <b>How?</b>            | <p>As the class is tested by itself, and not as part of the overall system, code may have to be created by the tester to stand in for the missing parts of the system, often termed a <b>test harness</b>. The test harness may pass test data to the object and/or receive results from the object. For example it may create an object from the class using the constructor, change/set the values of the attributes and read/get these values.</p> <p>Both <b>black box</b> and <b>white box</b> testing are used. Black box testing concentrates solely on inputs and outputs whereas white box testing is concerned with the internal code. More detail is provided in the table in the following section.</p>                                                       |          |                   |                |                  |                    |                       |
| <b>When and Who?</b>   | Writing code to test the units and examining the code requires programming experience therefore the programmers often perform the unit tests as they write the code. It is preferable that someone other than the programmer (an independent person who is objective and impartial) tests the code but in practice this does not often happen. In <b>eXtreme Programming (XP)</b> the programmers/software developers design the module tests before they write the code, using the design specification as a guide. This is called <b>test driven development</b> . As the programmers work in pairs they can also check each other's code.                                                                                                                              |          |                   |                |                  |                    |                       |
| <b>Example</b>         | <p>As a programmer the following class has been provided for implementation:</p> <table border="1" data-bbox="443 1335 746 1570"> <thead> <tr> <th>Customer</th> </tr> </thead> <tbody> <tr> <td><b>customerNo</b></td> </tr> <tr> <td><b>surname</b></td> </tr> <tr> <td><b>firstname</b></td> </tr> <tr> <td><b>dateOfBirth</b></td> </tr> <tr> <td><b>calculateAge()</b></td> </tr> </tbody> </table> <p>An example of a typical black box test that may be planned and executed is:</p> <ul style="list-style-type: none"> <li>• Set the value of the customer's <b>dateOfBirth</b> attribute to 28/10/73.</li> <li>• Display the result of the <b>calculateAge</b> operation.</li> <li>• Assuming today's date is the 29/10/13 the expected result is 40.</li> </ul> | Customer | <b>customerNo</b> | <b>surname</b> | <b>firstname</b> | <b>dateOfBirth</b> | <b>calculateAge()</b> |
| Customer               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |          |                   |                |                  |                    |                       |
| <b>customerNo</b>      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |          |                   |                |                  |                    |                       |
| <b>surname</b>         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |          |                   |                |                  |                    |                       |
| <b>firstname</b>       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |          |                   |                |                  |                    |                       |
| <b>dateOfBirth</b>     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |          |                   |                |                  |                    |                       |
| <b>calculateAge()</b>  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |          |                   |                |                  |                    |                       |

Table 4.2a - Schedule for **Program** Testing

<sup>2</sup> Sometimes this is called unit, class, component or module testing.



## Integration Testing

|                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>What is tested?</b> | Several associated, interdependent, modules or classes are tested together in a group or subsystem.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| <b>Focus/Purpose</b>   | <ul style="list-style-type: none"> <li>• Check if the unit of code/classes <b>work together</b> as intended without errors.</li> <li>• Focus on flow of data/information/messages/control from one class or module to another.</li> <li>• Verify or check that the code matches the high level design specification or architectural model.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| <b>How?</b>            | <p>As parts of the system may be missing at this stage code may have to be created to stand in for the missing parts.</p> <p>The testing will be mainly black box at this stage as the focus is on the interfaces between the modules/classes and the message/data passing between objects/subsystems rather than the internal code.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| <b>When and Who?</b>   | <p>Test cases can be created as soon as the design specification has been created, possibly from, for example, sequence and class diagrams.<sup>3</sup></p> <p>Who is selected to design and run the tests probably depends on what level of integration is involved. Integration at a high level (also called system integration or integration in the large) is when several subsystems are interconnected (such as the GUI interface, the Business Classes and the Data Access Classes). In this situation the analyst who has designed the system may be the person responsible for integration test design as they have the required knowledge of how the classes link together.</p> <p>Integration could also be regarded as a low level process (sometimes called component integration) when a small number of classes are tested together. If integration is defined as this then it may be that the programmer will design and run the integration tests. Integration testing is more important when using object-orientated programming languages such as C# and Java. It is hard to test a single object without linking it to at least one other object. There are still differences in opinion as to whether this is program or integration testing as it depends on what is classed as a unit (a single class or a group of classes?).</p> <p>It is better to perform the integration tests incrementally as the software is developed rather than leaving the integration of all the classes to the end in a big bang. Agile methodologies promote the idea of <b>continuous integration</b> in which a class is integrated into the system as soon as it has been unit or module tested.</p> |

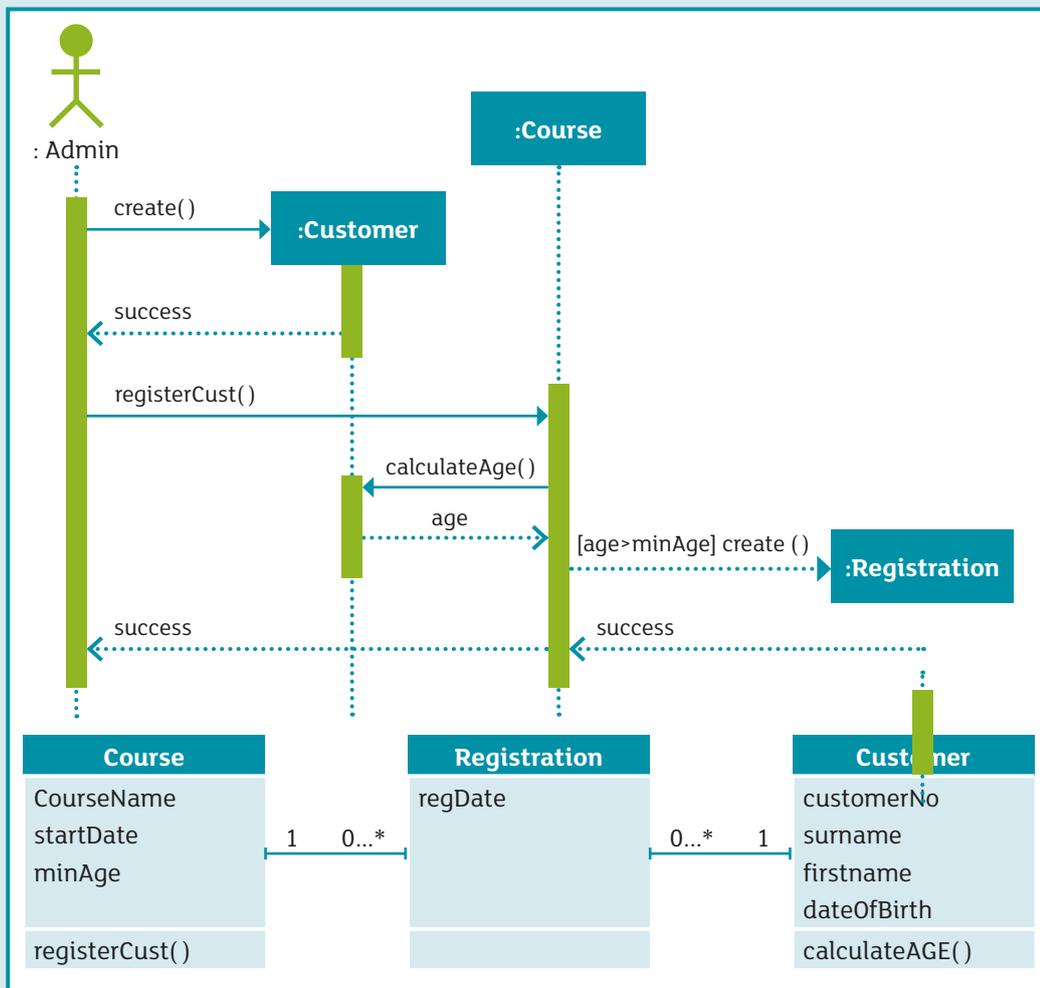
Table 4.2b - Schedule for **Integration** Testing

<sup>3</sup> **Class Model:** a static model which visualises the system's structure showing its **classes** (possibly **attributes** and **operations**) and the **associations** between them (with **multiplicities** at the association ends). The level of detail depends on the stage of the lifecycle. During analysis classes refer to places, events and things about which the system will capture information. Later during implementation classes can refer to software classes, windows, forms and other objects used to build the system.

**Sequence Diagram:** Illustrates the objects that participate in a single use case scenario and the messages that pass between them over time. An interaction diagram/dynamic model. Emphasis on time order.

**Example**

The analyst may create a sequence diagram, for one successful scenario of a use case (Register Customer), during a workshop with the client at the analysis stage<sup>4</sup> along with an outline of a possible class diagram. This will be elaborated in the design phase to show more detail such as classes for the user interface and the boundary with the database. A simplistic version from analysis is shown below.



Analysts can use this sequence diagram to design tests for the collaboration of the 3 objects called Customer, Course and Registration. Before the test begins the objects are populated with test data such as a course called 'Sailing the Sea' starting on 21/05/15 with a minimum age of 18.

**Test:** Register the customer Fiona Woods, with a Customer No of 45 and date of birth 28/10/73 (above the minimum age for the course that she will be registered for), on the 'Sailing the Sea' course. View all customers registered on the 'Sailing the Sea' course.<sup>5</sup>

**Expected Result:** A success message will be returned. The user will see on the display that the course 'Sailing the Sea' has a customer registered called Fiona Woods (showing that a registration object was created and linked to course).

<sup>4</sup> The sequence diagram at this stage does not show a lot of detail; no user interface or database access is specified yet; and the parameters for the method calls to the objects have not been defined. These will be added later.

<sup>5</sup> For simplicity, this is not included this on the sequence diagram.



## System Testing

| <b>What is tested?</b>                                                         | A whole complete software system (all subsystems are integrated together) including documentation.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |                    |  |             |  |                                                                                |  |             |                 |                                                |                                    |                                          |                                                            |                                              |                                    |
|--------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------|--|-------------|--|--------------------------------------------------------------------------------|--|-------------|-----------------|------------------------------------------------|------------------------------------|------------------------------------------|------------------------------------------------------------|----------------------------------------------|------------------------------------|
| <b>Focus/Purpose</b>                                                           | Does the software product meet all the functional (these may be expressed as a set of use case stories with their various scenarios) and non-functional requirements as detailed in the system requirements specification? Was the system developed correctly?                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |                    |  |             |  |                                                                                |  |             |                 |                                                |                                    |                                          |                                                            |                                              |                                    |
| <b>How?</b>                                                                    | <p>The whole system is treated as one large black box for testing; the internal working of the code is not analysed.</p> <p>Functional tests will examine the input/output behaviour.</p> <p>Non-functional tests will examine areas like performance, reliability, usability, security, and recovery.</p> <p>The tests should run on a system that simulates the real production/operational/working environment in the customer's business.</p> <p>As the whole system has been created no code is required to replace missing parts.</p>                                                                                                                                                                                                                                                                                                           |                    |  |             |  |                                                                                |  |             |                 |                                                |                                    |                                          |                                                            |                                              |                                    |
| <b>When and Who?</b>                                                           | <p>Test cases can be created as soon as the system requirements specification is ready.</p> <p>Tests are performed by members of the team who did not write the code, probably the analyst or a team of independent testers.</p> <p>Tests are executed after integration testing of all subsystems and before acceptance testing.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |                    |  |             |  |                                                                                |  |             |                 |                                                |                                    |                                          |                                                            |                                              |                                    |
| <b>Example</b>                                                                 | <p>The analyst may design the tests as soon as the user requirements are specified. For example he may derive a test case scenario from a use case.</p> <table border="1"> <thead> <tr> <th colspan="2">TEST CASE Scenario</th> </tr> <tr> <th>Description</th> <td></td> </tr> </thead> <tbody> <tr> <td>The administrator creates a new customer and adds them to an available course.</td> <td></td> </tr> <tr> <th>User Action</th> <th>System Response</th> </tr> <tr> <td>1. Create a new Customer called 'Fiona Woods'.</td> <td>2. A success message is displayed.</td> </tr> <tr> <td>3. View all currently available courses.</td> <td>4. A list of all currently available courses is displayed.</td> </tr> <tr> <td>5. Select a course called 'Sailing the Sea'.</td> <td>6. A success message is displayed.</td> </tr> </tbody> </table> | TEST CASE Scenario |  | Description |  | The administrator creates a new customer and adds them to an available course. |  | User Action | System Response | 1. Create a new Customer called 'Fiona Woods'. | 2. A success message is displayed. | 3. View all currently available courses. | 4. A list of all currently available courses is displayed. | 5. Select a course called 'Sailing the Sea'. | 6. A success message is displayed. |
| TEST CASE Scenario                                                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |                    |  |             |  |                                                                                |  |             |                 |                                                |                                    |                                          |                                                            |                                              |                                    |
| Description                                                                    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |                    |  |             |  |                                                                                |  |             |                 |                                                |                                    |                                          |                                                            |                                              |                                    |
| The administrator creates a new customer and adds them to an available course. |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |                    |  |             |  |                                                                                |  |             |                 |                                                |                                    |                                          |                                                            |                                              |                                    |
| User Action                                                                    | System Response                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |                    |  |             |  |                                                                                |  |             |                 |                                                |                                    |                                          |                                                            |                                              |                                    |
| 1. Create a new Customer called 'Fiona Woods'.                                 | 2. A success message is displayed.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |                    |  |             |  |                                                                                |  |             |                 |                                                |                                    |                                          |                                                            |                                              |                                    |
| 3. View all currently available courses.                                       | 4. A list of all currently available courses is displayed.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |                    |  |             |  |                                                                                |  |             |                 |                                                |                                    |                                          |                                                            |                                              |                                    |
| 5. Select a course called 'Sailing the Sea'.                                   | 6. A success message is displayed.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |                    |  |             |  |                                                                                |  |             |                 |                                                |                                    |                                          |                                                            |                                              |                                    |

Table 4.2c - Schedule for **System** Testing



## Acceptance Testing

|                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>What is tested?</b> | A whole complete system (similar to system testing).                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| <b>Focus /Purpose</b>  | <ul style="list-style-type: none"> <li>• Focus on customer/user perspective and judgment. Is the system complete, accepted by the users and approved for use? Confirm the right software that meets the customer's needs has been built.</li> <li>• Is it fit for use? Is it ready for operational use?</li> <li>• Does it meet the criteria as laid out in the contract/Terms of Reference?</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| <b>How?</b>            | <p>The tests will be very similar to the system tests except that they are <b>designed and performed by the users or customers</b> rather than members of the software development team (they only assist when required). They will either be executed in the software development company (which simulates the customer's working environment) or in the user/customer's actual place of work. The tests will be black box tests as the user will not view the source code. The users will be encouraged to test the entire functionality of the system and to use it in unexpected ways entering valid and invalid data.</p> <p>Two special distinct types of acceptance testing are <b>alpha</b> and <b>beta</b> testing. Alpha and Beta testing are terms that are generally used for software that is to be mass produced and sold to the general public such as Microsoft Office; in contrast to a <b>'bespoke'</b> system that is specially designed for a particular organisation. In the literature there are many various definitions but the main points are:</p> <ul style="list-style-type: none"> <li>• <b>Alpha:</b> Conducted by test engineers or selected user groups in a controlled environment, at the developer's site with access to the development team and using their test hardware and software; test data may be created by the software development team;</li> <li>• <b>Beta:</b> Conducted by interested users in a 'real world' uncontrolled environment, at user's own site, using the user's own hardware and software and the user's own choice of data. Software may be downloaded from a website, in which case users are not specifically selected.</li> </ul> |
| <b>When and Who?</b>   | <p>The test cases can be designed at the very start of the project when the business requirements are known; possibly one to test each requirement. The test cases are designed with the customer.</p> <p>They are executed by the users after system testing and before installation/ deployment. Different user groups may be used.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| <b>Example</b>         | <p>The customer/client may have expressed one of his business needs as:</p> <p>'Administrators shall be able to register customers on our courses.'</p> <p>The customer may run a test for this business need by trying to enrol several customers on various courses possibly using real data from their own organisation; if they can do this successfully and with relative ease they may then declare the software is fit for this purpose and accept the software system.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |

Table 4.2d - Schedule for **Acceptance Testing**

## Black Box Testing<sup>6</sup>

|                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|-----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Level</b>                | Usually performed at various levels. The black box can represent the whole system (Acceptance and System Testing Level); a subsystem (Integration Testing Level) or a single unit (Unit Testing level).                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| <b>Responsibility</b>       | Depends on Level: e.g. Developer (Unit Test Level), Analyst (System Test Level), and User (Acceptance Test Level).                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| <b>Purpose</b>              | Input/Output behaviour tested. Does it do what it is supposed to do? What happens inside the box is of no concern.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| <b>Required Information</b> | Base tests on the functional requirements/Use Cases at System Test level or the detailed design specification at unit testing level.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| <b>How</b>                  | <p>The tester provides inputs and inspects the outputs. Inputs that result in <b>both success and failure</b> are required. Results and error messages can be examined to see if they are appropriate and as expected.</p> <p>There are so many possible inputs that testing all of them, called exhaustive testing, is often impractical as it would take a very long time. Instead selected inputs are used.</p> <p>A common method used to select inputs is called <b>Equivalence Partitioning</b>. Say an input parameter accepts an integer for the month of a year: the valid inputs are numbers 1 to 12 representing January to December. An input will be chosen from:</p> <ul style="list-style-type: none"> <li>• The valid range, 1-12: e.g. 4</li> <li>• The invalid range, below 1: e.g. -3</li> <li>• The invalid range, above 12: e.g. 17</li> </ul> <p>Additional inputs may also be chosen at the boundaries (boundary value analysis) of the valid and invalid ranges where many errors occur. So for example the boundary inputs are (taking the value of the boundary itself and one value either side of the boundary):</p> <ul style="list-style-type: none"> <li>• At the lower boundary, 1: 0, 1 and 2</li> <li>• At the upper boundary, 12: 11, 12, and 13.</li> </ul> |
| <b>Advantages</b>           | <ul style="list-style-type: none"> <li>• Tester may not need any programmer knowledge and testing may be performed by independent testers.</li> <li>• As it tests the behaviour of the system it highlights any missing functionality.</li> <li>• Test cases can be designed as soon as the functional specifications are complete.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| <b>Disadvantages</b>        | <ul style="list-style-type: none"> <li>• Wide range of possible inputs; impractical to do exhaustive testing entering every possible input.</li> <li>• Impossible to know if every single line of code is executed as inside the 'box' is not visible. If it has not been executed it is not tested!</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |

Table 4.2e - Schedule for **Black Box** Testing

<sup>6</sup> Sometimes called functional or behavioural testing.

## </> White Box Testing<sup>7</sup>

|                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|-----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Level</b>                | <i>Mainly</i> at Unit Test Level using programmers who have knowledge of the internal logic.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| <b>Responsibility</b>       | Developer/Programmer.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| <b>Purpose</b>              | Analysis of algorithms/logic/internal workings.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| <b>Required Information</b> | Access to actual code and algorithms.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| <b>How</b>                  | <p>Test cases should exercise all possible execution paths through the code so that all the statements are covered in the tests. White box testing should ensure that for example:</p> <ul style="list-style-type: none"> <li>• All statements/paths are executed at least once;</li> <li>• Each possible branch in a conditional statement (such as IF) is executed;</li> <li>• Loops are skipped altogether, executed at least once, and more than once;</li> </ul> <p>In this code there are two branches through the code; one for when lives_left is equal to 0 and another path for when lives_left is not zero. It is important that both paths are executed. For example when lives_left equals 0 and lives_left=4.</p> <pre> if (lives_left == 0) {     Save_current_score ();     End_game(); } else {     Decrease_score (); } </pre> |
| <b>Advantages</b>           | <ul style="list-style-type: none"> <li>• A thorough test: can check every line of code.</li> <li>• Easy to know which input values should be used to test the application.</li> <li>• Helps to determine how the code should be fixed/debugged.</li> <li>• Can tidy up the code and optimise it.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| <b>Disadvantages</b>        | <ul style="list-style-type: none"> <li>• Testers need programming expertise and their time is more expensive.</li> <li>• In large systems it is difficult to test every line of code and a defect may be missed.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |

Table 4.2f - Schedule for **White Box** Testing

<sup>7</sup> Also called Clear Box Testing, Open Box Testing, Glass Box Testing, Transparent Box Testing, Code-Based Testing or Structural Testing.



## Questions

1 Which statements are true and false about testing?

|   |                                                                                         | T | F |
|---|-----------------------------------------------------------------------------------------|---|---|
| A | Testing is used to detect bugs and defects.                                             |   |   |
| B | It is better to leave testing to the end of the software development process.           |   |   |
| C | Once the system requirements specification is defined systems tests can be designed.    |   |   |
| D | Programmers design system tests before they start coding.                               |   |   |
| E | The focus of unit testing is on how objects collaborate together in part of the system. |   |   |
| F | System tests are performed by the users.                                                |   |   |
| G | Beta testing is a form of acceptance testing.                                           |   |   |
| H | Unit testing involves testing the whole system.                                         |   |   |
| I | White box testing is performed by customers.                                            |   |   |
| J | Black box testing involves checking that all branches in the code have been executed.   |   |   |
| K | In acceptance testing the users treat the whole system like one big black box.          |   |   |
| L | Black box tests may leave some lines of code untested.                                  |   |   |

2 Place a tick beside the items that would be included in a test plan?

|   |                                                    |  |
|---|----------------------------------------------------|--|
| A | Start and end date of testing activities.          |  |
| B | The format for recording the results of the tests. |  |
| C | The actual results of the tests.                   |  |
| D | ER diagrams.                                       |  |
| E | Items that will be tested and not tested.          |  |
| F | Person responsible for system testing.             |  |
| G | Tools used to record the test cases.               |  |
| H | Sequence Diagrams.                                 |  |
| I | Hardware required for acceptance testing.          |  |



## Questions

- 3** Insert the correct term from the options below into the blank spaces of the following paragraph about testing.

|                     |                     |                |
|---------------------|---------------------|----------------|
| <b>black</b>        | <b>installation</b> | <b>system</b>  |
| <b>program</b>      | <b>programmers</b>  | <b>message</b> |
| <b>integration</b>  | <b>brown</b>        | <b>baton</b>   |
| <b>acceptance</b>   | <b>user</b>         | <b>white</b>   |
| <b>stakeholders</b> | <b>software</b>     | <b>branch</b>  |

There are four main levels of testing:

..... testing is performed by ..... who test a small unit of code, such as a C# class, in isolation from the rest of the system. They perform both ..... box (checking inputs and outputs) and ..... box (check the internal code) tests.

..... tests are concerned with checking if several associated units of code such as classes work together without error. These tests focus on the interfaces and ..... passing between the objects.

..... testing involves the whole system. Both functional and non-functional tests are performed.

..... testing of the system is finally performed by the .....

If the client is satisfied they will approve the system for release.



## Questions

- 4 Using the tables provided on the next page, design the white box and black box tests to test the function below. The function is required to satisfy the following:

| Result           | Grade       |
|------------------|-------------|
| 85-100 inclusive | Distinction |
| 60-84 inclusive  | Merit       |
| 40-59 inclusive  | Pass        |
| 0-39 inclusive   | Fail        |

```
static string courseGrade(int result)
{
    string grade;
    if
        (result >= 85)
        grade = "Distinction";
    else if
        (result >= 60)
        grade = "Merit";
    else if
        (result >= 40)
        grade = "Pass";
    else
        grade = "Fail";
    return grade;
}
```

Give some examples of the input data that might be used for each type of test (presuming that access to the code would only be allowed for white box testing).

What is the expected result? Imagine the test runs...what would the actual result be?  
Would the tests pass or fail?



## Questions

### BLACK BOX TESTS

| Input Data | Expected Result | Actual Result | Pass/Fail |
|------------|-----------------|---------------|-----------|
|            |                 |               |           |

### WHITE BOX TESTS

| Input Data | Expected Result | Actual Result | Pass/Fail |
|------------|-----------------|---------------|-----------|
|            |                 |               |           |

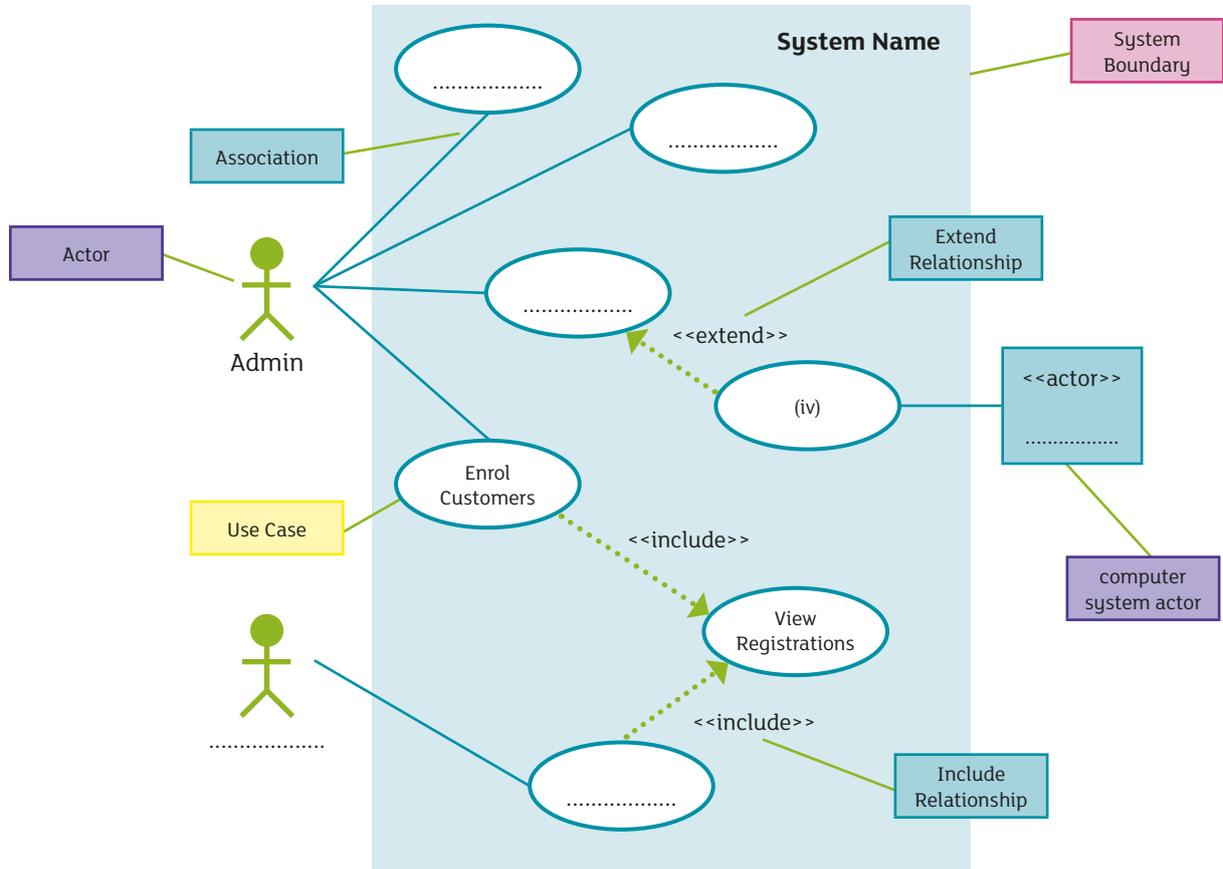
Discuss the following question:

- Did the black box tests exercise all the lines of code?
- Did the white box tests exercise all the lines of code?
- Does the code function as expected and if not how might it be fixed?
- Which tests, white box or black box, performed better?

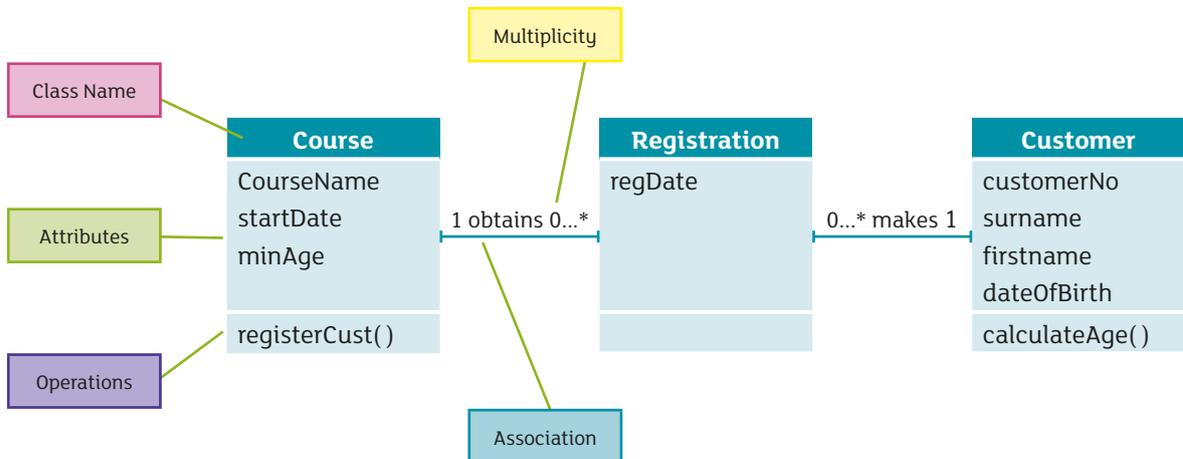
## Appendix

Some examples of the Unified Modelling Language, UML, diagrams which are referred to in this fact file.

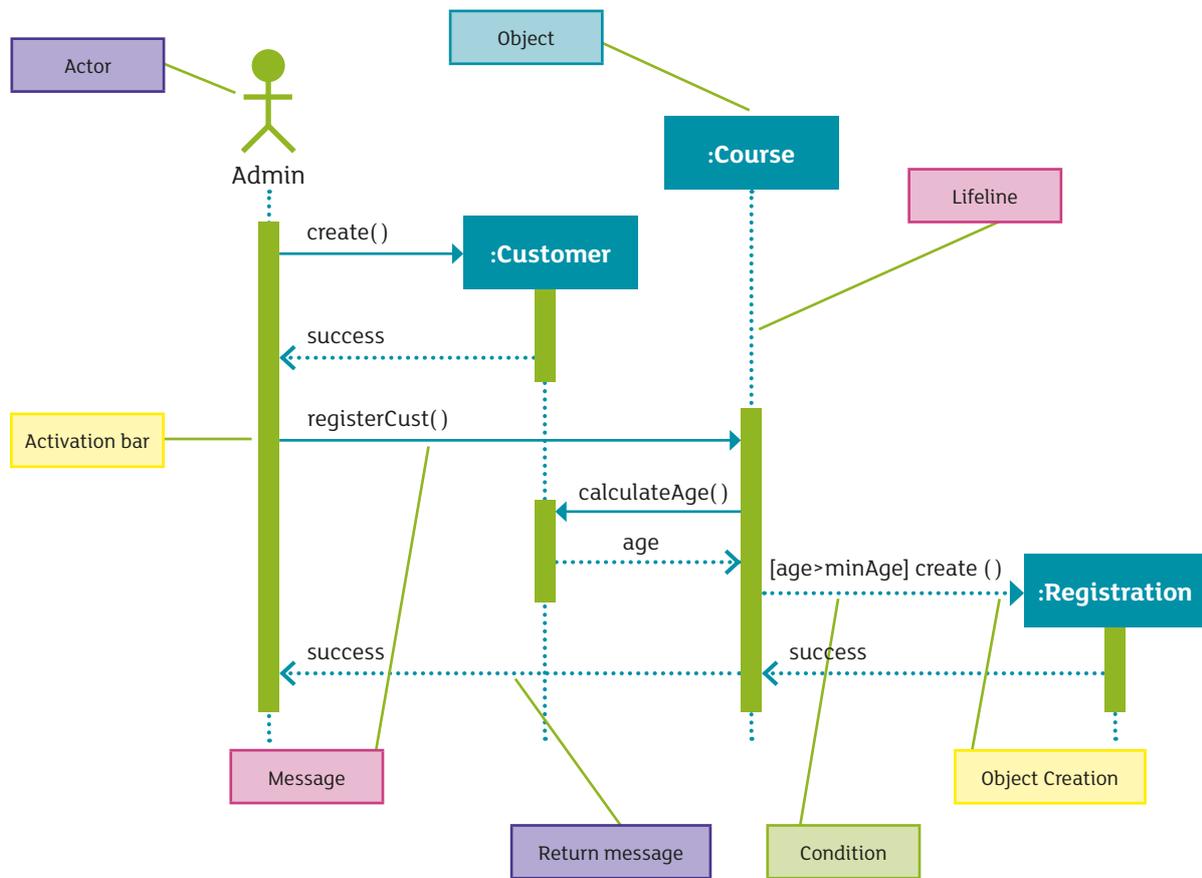
### Components of a Use Case Diagram



### Components of a Class Diagram



Components of a Sequence Diagram



**Note** - Iteration is not shown in this example.

