Software Testing for BI

Lesson 2: Testing concepts

Lesson Objectives

- To understand the following topics:
  - What is testing?
    - Testing – Why?
    - Testing – How?
  - Principles of Testing
  - Test Case and Test Suite
  - Testing scope
  - Test strategy
  - Verification and Validation

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 2

2.1: Testing concepts

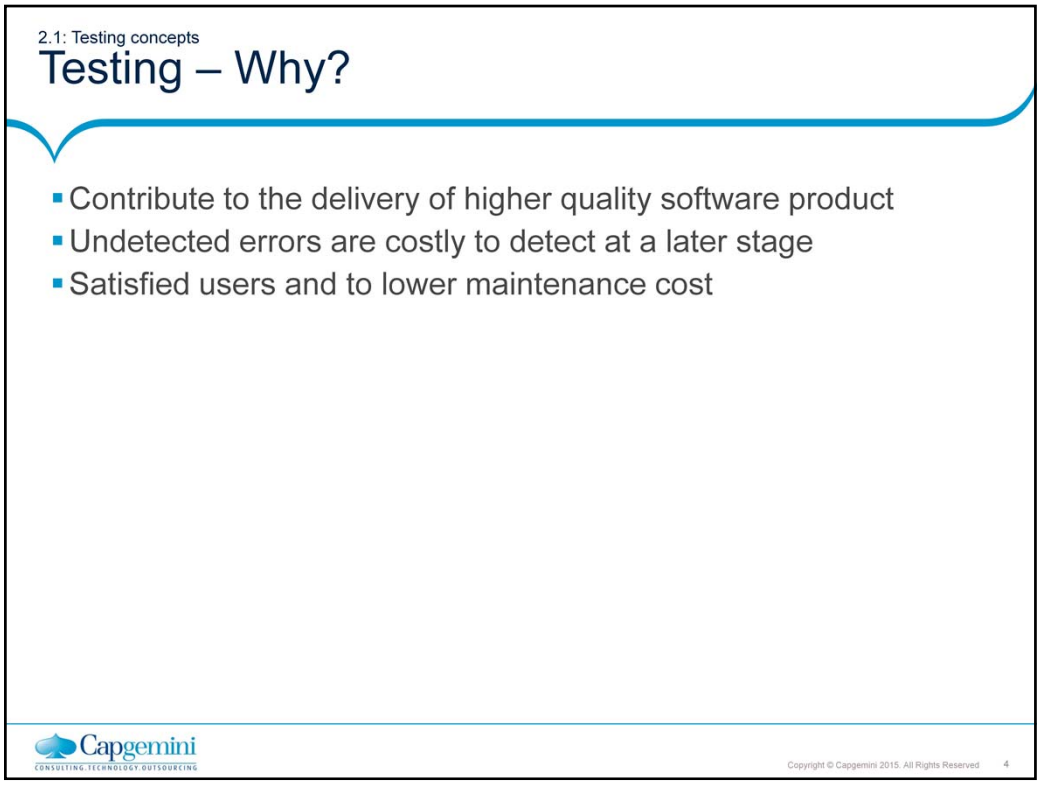
What is testing

- Testing is the process of executing a program with the intent of finding errors
- To find greatest possible number of errors with manageable amount of efforts applied over a realistic time span with a finite number of test cases.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

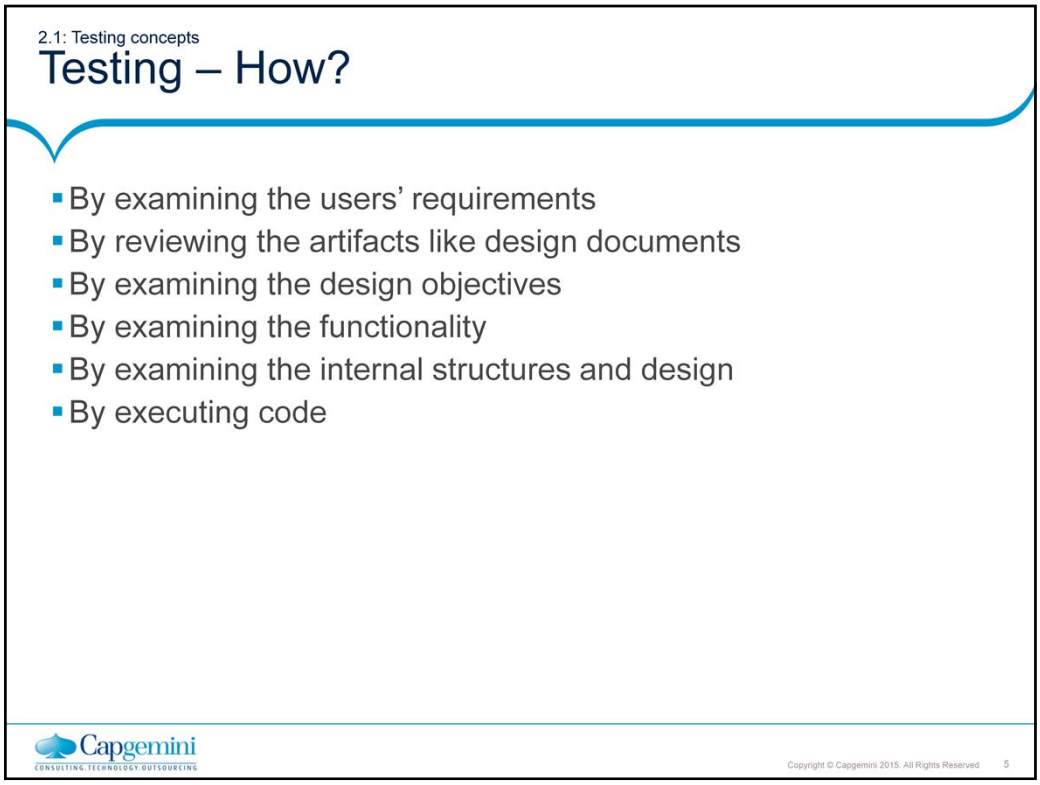
 3

Testing is successful if you can prove that the product does what it should not do and does not do what it should do.

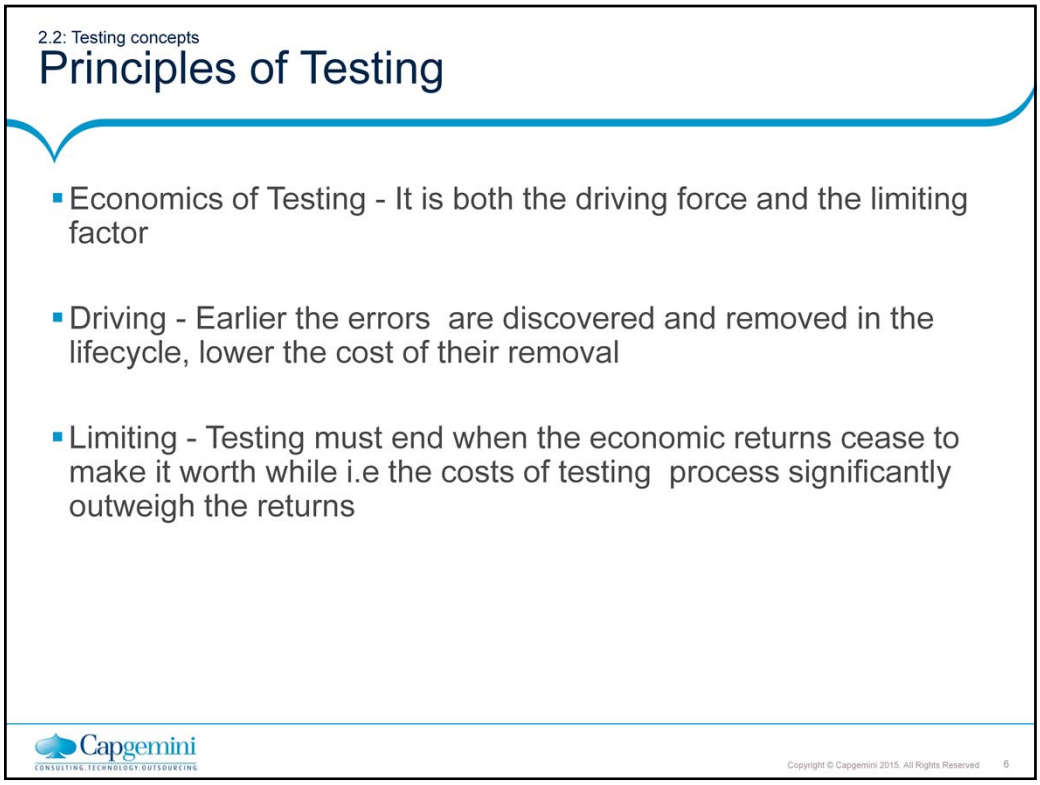


Why is testing becoming such a crucial activity? – Because applications are becoming very complex with n-tiers in an application.
When one tests a program one adds value to it through improved quality and reliability.
If not tested it can cause an unpleasant navigational error in case of a browsing applications or death or injury in case of safety critical applications.

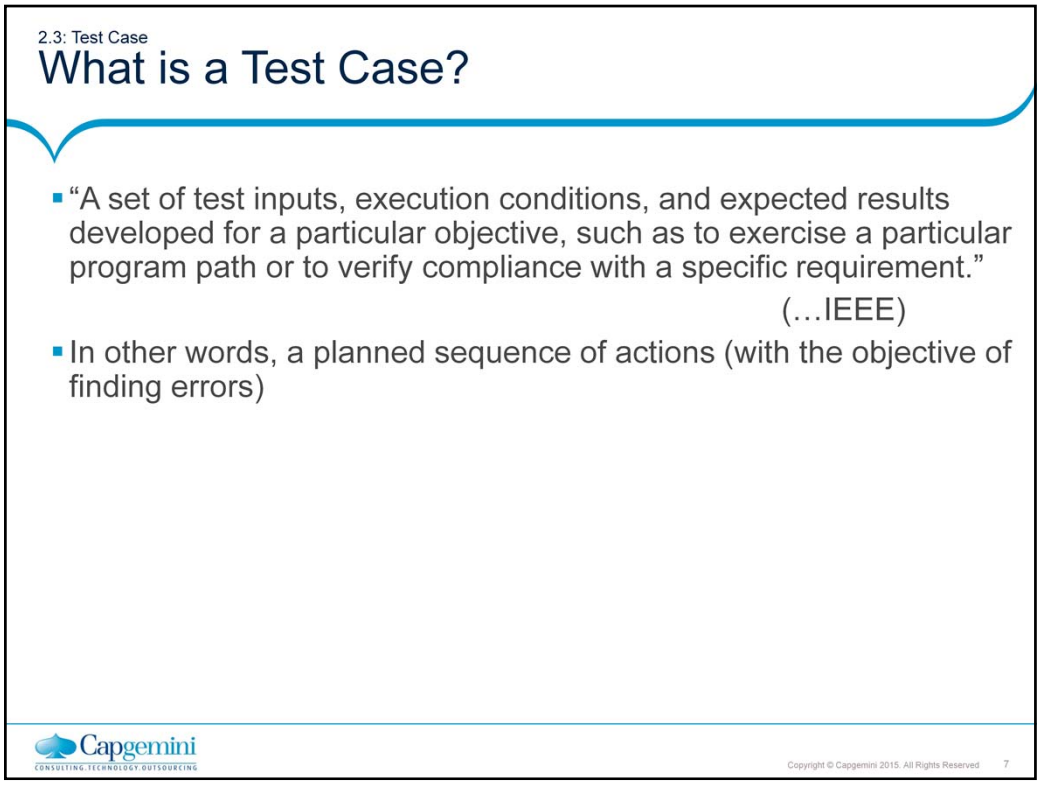
End customers are becoming more demanding & conscious about quality (defects)?
Why are company's outsourcing the testing phase? – It is being realized that testing is an extremely important phase, customers today are conscious of quality as they need to be more competitive in the market.
It is being realized that the best people to test an application are the ones who have not developed the application. The testers would have the approach of a user and have an unbiased mind.



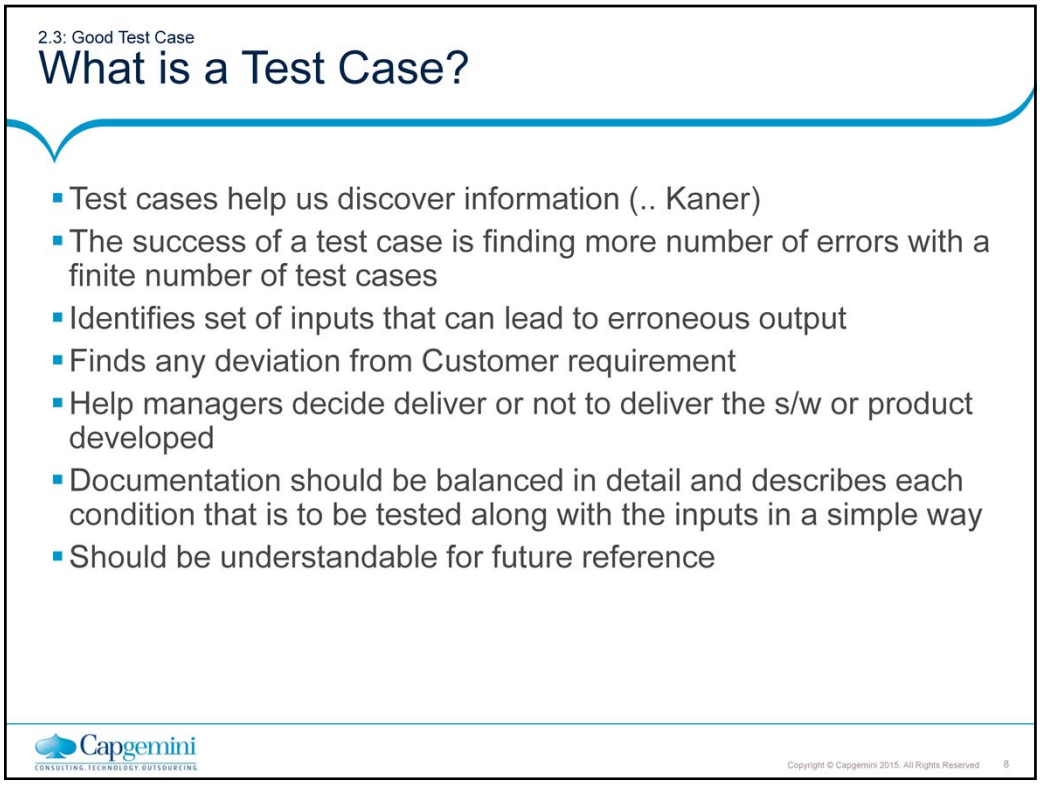
Trainer Notes



Although testing is itself an expensive activity, the cost of not testing is potentially much higher.
The most damaging errors are those which are not discovered during the testing process and therefore remain when the system goes live.
Limiting - It is infeasible to test exhaustively all but the most simple or the most vital of s/w.
Here we discuss how exhaustive testing is impossible to achieve, start with a small example and then show how for it is impossible, uneconomical and unfruitful to test all the test cases.
Following example for the Exhaustive input testing (Black box)
Example of a function say ax2 +bx+ c=0.
Assume 16 bit numbers.
So each input is 2power16.
And so total test cases is 2 power16*2power16*2power16=2power48 test cases which is impractical.
If cost of testing increases and no. of defects come down and the need to strike a balance to achieve optimum testing.



A Test case is a planned sequence of actions.



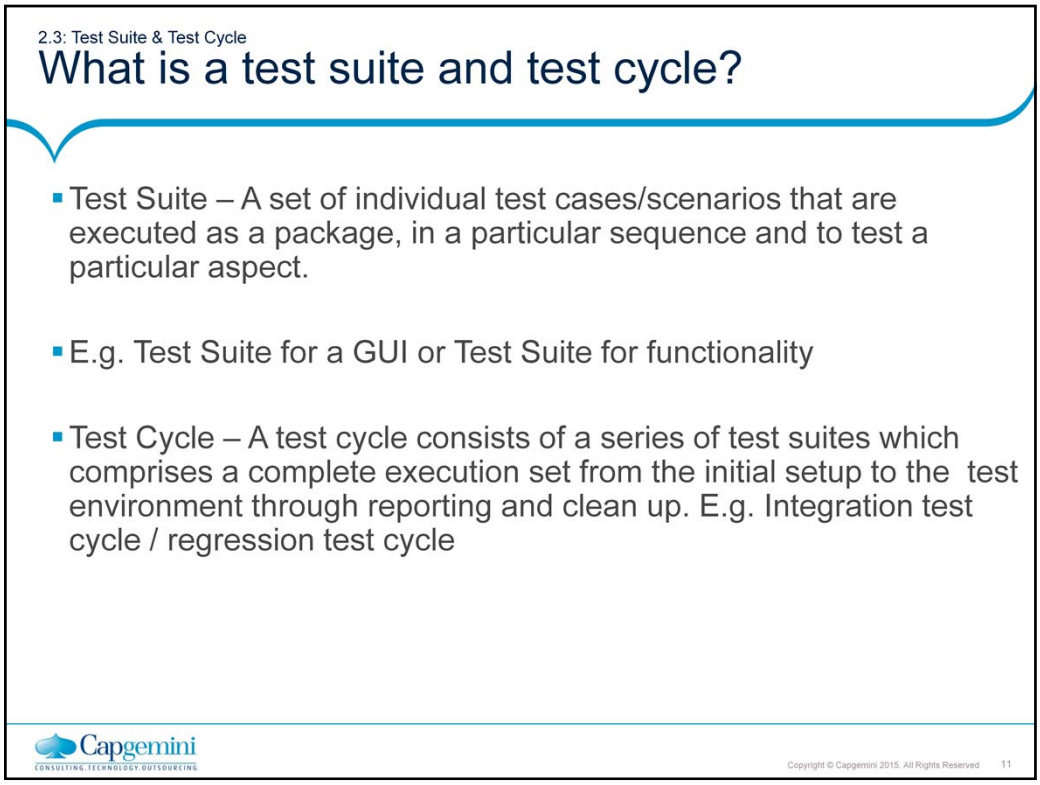
Testing at the boundary values has a higher probability of revealing errors and in case the test case detects an error it turns out to be a successful test case as well. The tester must understand the software and attempt to develop a mental picture of how the software might fail.

Every Test case should be Atomic, Repeatable, Self contained.

Information Objectives
So what are we trying to learn or achieve when we run tests? Here are some examples:
•Find defects. This is the classic objective of testing. A test is run in order to trigger failures that expose defects. Generally, we look for defects in all interesting parts of the product.
•Maximize bug count. The distinction between this and “find defects” is that total number of bugs is more important than coverage. We might focus narrowly, on only a few high-risk features, if this is the way to find the most bugs in the time available.
•Help managers make ship / no-ship decisions. Managers are typically concerned with risk in the field. They want to know about coverage (maybe not the simplistic code coverage statistics, but some indicators of how much of the product has been addressed and how much is left), and how important the known problems are. Problems that appear significant on paper but will not lead to customer dissatisfaction are probably not relevant to the ship decision.

•Minimize technical support costs. Working in conjunction with a technical support or help desk group, the test team identifies the issues that lead to calls for support. These are often peripherally related to the product under test. For example, getting the product to work with a specific printer or to import data successfully from a third party database might prevent more calls than a low-frequency, data-corrupting crash.
•Assess conformance to specification. Any claim made in the specification is checked. Program characteristics not addressed in the specification are not (as part of this objective) checked.
•Conform to regulations. If a regulation specifies a certain type of coverage (such as, at least one test for every claim made about the product), the test group creates the appropriate tests. If the regulation specifies a style for the specifications or other documentation, the test group probably checks the style. In general, the test group is focusing on anything covered by regulation and (in the context of this objective) nothing that is not covered by regulation.
•Minimize safety-related lawsuit risk. Any error that could lead to an accident or injury is of primary interest. Errors that lead to loss of time or data or corrupt data, but that don't carry a risk of injury or damage to physical things are out of scope.
•Find safe scenarios for use of the product (find ways to get it to work, in spite of the bugs). Sometimes, all that you're looking for is one way to do a task that will consistently work--one set of instructions that someone else can follow that will reliably deliver the benefit they are supposed to lead to. In this case, the tester is not looking for bugs. He is trying out, empirically refining and documenting, a way to do a task.
•Assess quality. This is a tricky objective because quality is multi-dimensional. The nature of quality depends on the nature of the product. For example, a computer game that is rock solid but not entertaining is useless a game. To assess quality -- to measure and report back on the level of quality -- you probably need a clear definition of the most important quality criteria for this product, and then you need a theory that relates test results to the definition.

•For example, reliability is not just about the number of bugs in the product. It is (or is often defined as being) about the number of reliability-related failures that can be expected in a period of time or a period of use. (Reliability-related? In measuring reliability, an organization might not care, for example, about misspellings in error messages.) To make this prediction, you need a mathematically and empirically sound model that links test results to reliability. Testing involves gathering the data needed by the model. This might involve extensive work in areas of the product believed to be stable as well as some work in weaker areas. Imagine a reliability model based on counting bugs found (perhaps weighted by some type of severity) per N lines of code or per K hours of testing. Finding the bugs is important. Eliminating duplicates is important.
Troubleshooting to make the bug report easier to understand and more likely to fix is (in the context of assessment) out of scope.
•Verify correctness of the product. It is impossible to do this by testing. You can prove that the product is not correct or you can demonstrate that you didn't find any errors in a given period of time using a given testing strategy. However, you can't test exhaustively, and the product might fail under conditions that you did not test. The best you can do (if you have a solid, credible model) is assessment--test-based estimation of the probability of errors. (See the discussion of reliability, above).
•Assure quality. Despite the common title, quality assurance, you can't assure quality by testing. You can't assure quality by gathering metrics. You can't assure quality by setting standards. Quality assurance involves building a high quality product and for that, you need skilled people throughout development who have time and motivation and an appropriate balance of direction and creative freedom. This is out of scope for a test organization. It is within scope for the project manager and associated executives. The test organization can certainly help in this process by performing a wide range of technical investigations, but those investigations are not quality assurance.
Given a testing objective, the good test series provides information directly relevant to that objective. Different types of tests are more effective for different classes of information.



Test case is a triplet [I, S, O] where

- I is input data
- S is state of system at which data will be input
- O is the **expected** output

Test Suite – Test suite is set of all test cases. Suites are usually related by the area of the application they exercise or by their priority or content.
For E.g. When you Login to the screen, some functionalities like validating user name, password with different invalid inputs can act as Test suites.
Test Cycle – It's a combination of series of test suites.
For E.g. The Test Cycle for the same application is combination of multiple Test Suites like, Functional validations, Database validations, GUI validations, etc. form the Test Cycle.
Test Scenario -– A set of test cases that ensure that the business process flows are tested from end to end. They may be independent tests or a series of tests that follow each other, each dependent on the output of the previous one. The terms "test scenario" and "test case" are often used synonymously.

Test cases are not randomly selected. Instead even they need to be designed.

2.4: Testing scope

What is the scope of BI testing?

- Business Intelligence testing needs to be done at various levels to ensure that all requirements, both business and technical have been met.
- Typically Business Intelligence architecture adheres to a target BI reference architecture that is used as a framework to build BI solutions.
- Testing of any BI solution will cover all the layers utilized by the BI Solution within this architecture and also the complete end-to-end process

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 12

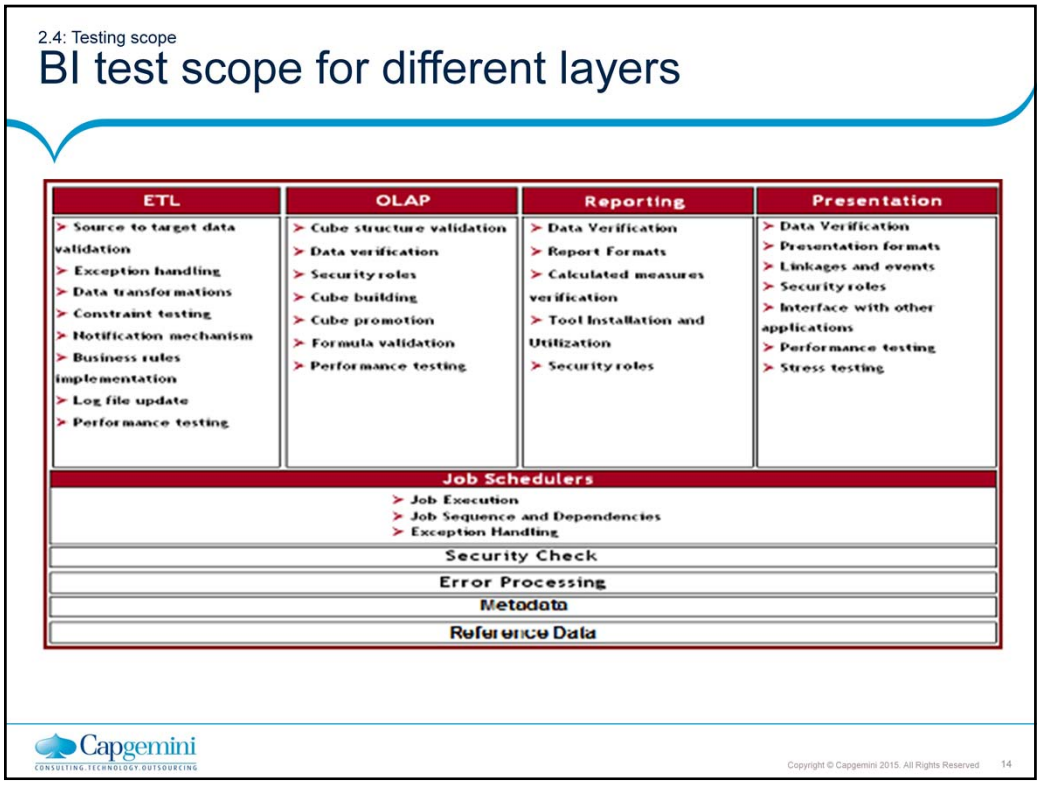
2.4: Testing scope

What is the scope of BI testing? (Cont…)

- This testing process includes
  - Sourcing and transformation of Business Data
  - Reporting and Presentation of Business Data
  - Configuration of the BI Tools
  - Scalability and Performance
  - Operational characteristics including scheduling, security, metadata

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 13



The table gives an overview of the typical BI specific test coverage involving the different BI specific layers.

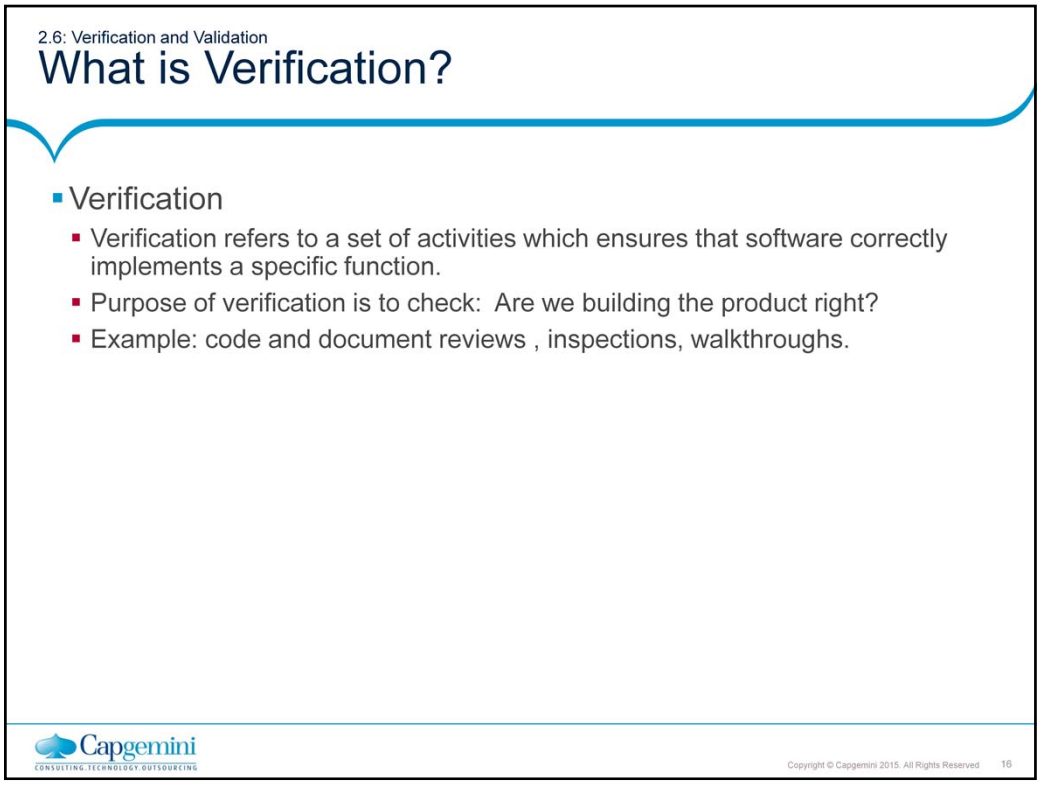
2.5: Test strategy

What is testing strategy?

- It provides a road map that describes the steps to be conducted as part of testing, when these steps are planned and then undertaken, and how much effort, time and resources will be required.
- It must incorporate test planning, test case design, test execution and resultant data collection and evaluation.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

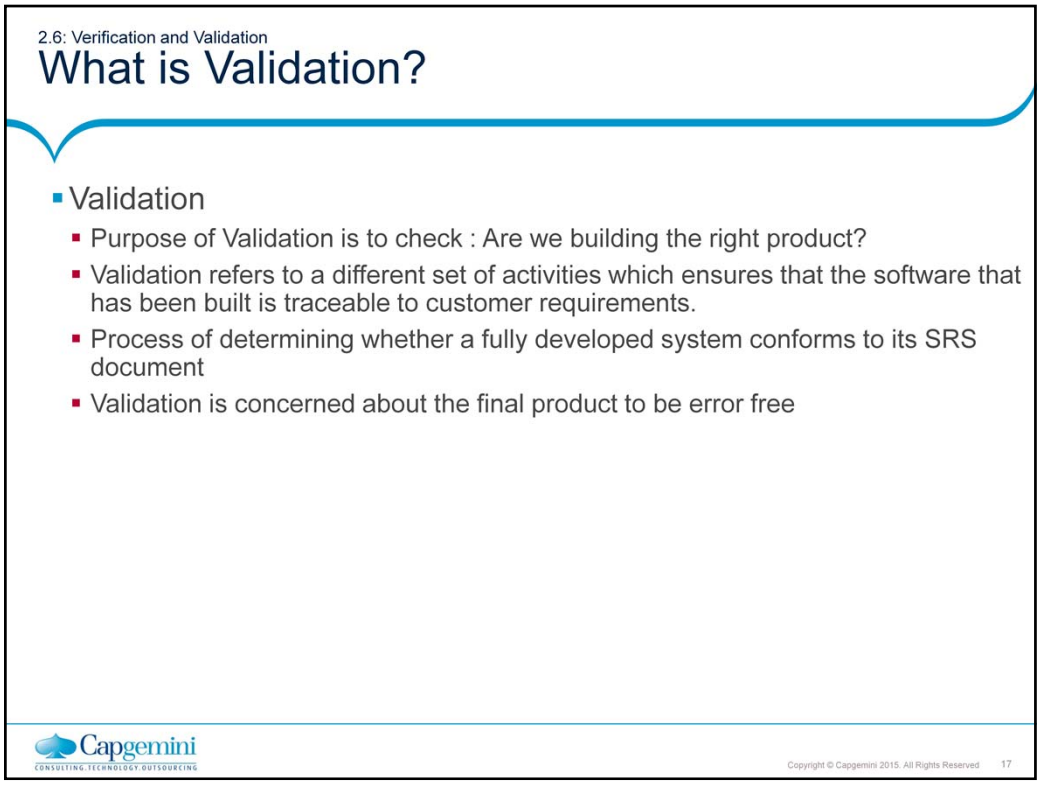
 15



V&V encompasses many of the activities that are encompassed by S/w quality assurance that include formal technical reviews, quality and configuration audits, performance monitoring, simulation, feasibility study, documentation review, database review, algorithm analysis, development testing, usability testing, installation testing. Verification is a process of determining whether output of one phase of development conforms to its previous phase. Verification is concerned with phase containment of errors For example, On Login screen if user logs in to screen, user should be navigated to user account details page. Checking for this functionality is verification.

Example of Verification : code and document inspections, walkthroughs, and other techniques.
Example of Verification : unit testing , integration testing , system testing (validation of software typically includes evidence that all software requirements have been implemented correctly and completely and are traceable to system requirements.)

If we are in a shopping centre and buy a thing with a code number 2342 and when we go to till and they check the number of that item and find it wrong then system will check all product number of the relevant number but don't find any number of this kind then we can say that the verify thing is wrong.
Verification is a process, which performs testing to ensure implemented functions meeting to designed functions.



For example, when user logs in to screen, user should provide valid data for user account. User input data will be validated for login process.

Validation is nothing but execution of Test cases, a process to check whether the Executed and Actual results are same. Validation is a process, which performs testing to ensure designed functions meeting to customer's requirements.

The word Validation resembles that all the functionalities are correctly working or not is validated for correctness.

If we are in a shopping centre and buy a thing with a code number 2342 and when we go to till and they check the number of that item and find it right then we can say that this product is valid.

2.6: Verification and Validation

What is Validation? (Cont…)

- Validation
  - After each validation test has been conducted, one of two possible conditions exist:
    - The function or performance characteristics conform to specification and are accepted, or
    - Deviation from specification and a deficiency list is created.
- Example: A series of black box tests that demonstrate conformity with requirements.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 18

Summary

- In this lesson, you have learnt:
  - Testing is to find greatest possible number of errors with manageable amount of efforts
  - Testing is carried out for the contribution to the delivery of higher quality software product
  - Driving and limiting are the basic principles of testing

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 19

Summary

- In this lesson, you have learnt:
  - Verification refers to a set of activities which ensures that software correctly implements a specific function.
  - Validation is concerned about the final product to be error free

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

Review Questions

- Question 1: Test cases help us discover_____
- Question 2: _____ is a set of test cases/scenarios that are executed as a package
- Question 3: Validation refers to a set of activities which ensures that software correctly implements a specific function.
  - True / False

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 21