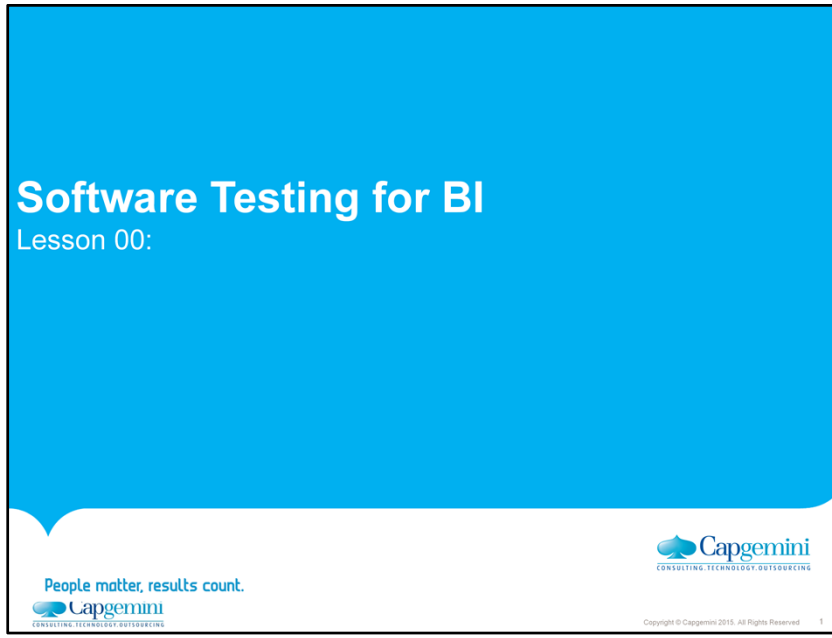


Document History

Date	Course Version No.	Software Version No.	Developer / SME	Reviewer(s)	Approver	Change Record Remarks
16-Apr-09	0.1D	NA	Priya Rane			Content Updation
08-Feb-10	0.1D	NA	CLS Team			Review
20-July-11	0.1D	NA	Karthikeyan/Vandana			iGate Patni Integration
11-Sep-12	1.0	NA	Karthikeyan			iGate Format
July-16	1.1	NA	Rajita Dhumal	Swati Rao	Mahima Sharma	Material Revamp as per Integrated ToC for I & D LoT

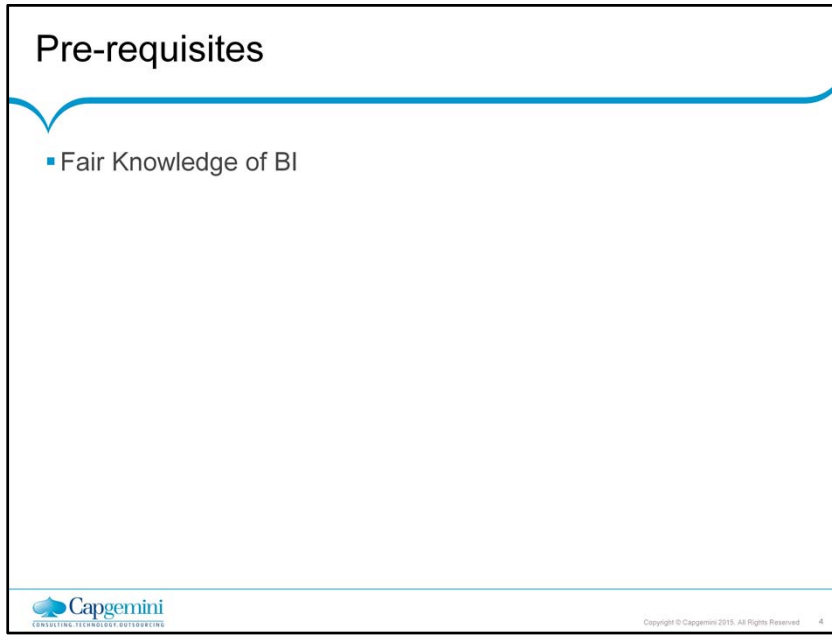
Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

Course Goals and Non Goals

- Course Goals
  - At the end of this program, participants gain an understanding of the need of separate software testing for Business Intelligence (BI)
- Course Non Goals
  - Detailed introduction of tools and implementation is not part of this course

Capgemini
CONSULTING. TECHNOLOGY. OUTSOURCING

 3


Pre-requisites
Fair Knowledge of BI
Capgemini
CONSULTING. TECHNOLOGY. OUTSOURCING
Copyright © Capgemini 2015. All Rights Reserved
4

Intended Audience

- Programmers in BI Technology

Capgemini
CONSULTING. TECHNOLOGY. OUTSOURCING

 5

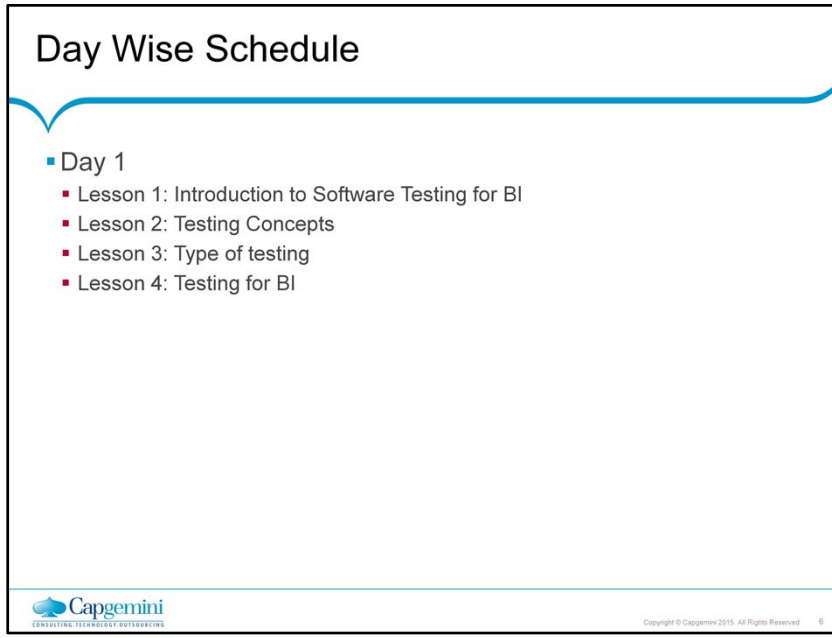

Day Wise Schedule
Day 1
Lesson 1: Introduction to Software Testing for BI
Lesson 2: Testing Concepts
Lesson 3: Type of testing
Lesson 4: Testing for BI
Capgemini
CONSULTING. TECHNOLOGY. OUTSOURCING
Copyright © Capgemini 2015. All Rights Reserved
6

Table of Contents

- Lesson 1: Introduction to Software Testing for BI
  - 1.1. Business Requirements
  - 1.2. BI Projects versus BI Program
  - 1.3. Why is BI testing required?
  - 1.4. How is BI testing different from code based testing?
- Lesson 2: Testing Concepts
  - 2.1. Testing Concepts
  - 2.2. Principles of Testing

Capgemini
CONSULTING. TECHNOLOGY. OUTSOURCING

Table of Contents

- 2.3. Test Case
- 2.4. Test Suite & Test cycle
- 2.5. Testing Scope
- 2.6. Testing Strategy
- 2.7. Verification and Validation

- Lesson 3: Types of Testing
  - 3.1. V Model for BI Testing

Table of Contents

- Lesson 4: Testing for BI
  - 4.1. Testing Document Purpose (Test Documentation)
  - 4.2. General BI Testing Principles
  - 4.3. iGate Testing Mission
  - 4.4. Production Verification Testing
  - 4.5. Possible Areas of Automation

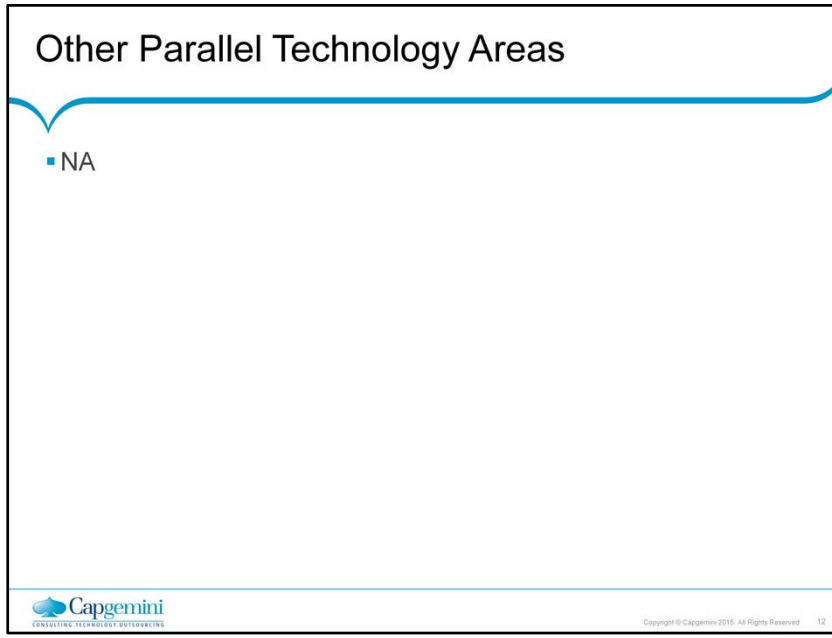
Capgemini
CONSULTING. TECHNOLOGY. OUTSOURCING

References

- Student Material:
  - Class Book (presentation slides with notes)

Next Step Courses (if applicable)

- Testing tools for BI


Other Parallel Technology Areas
NA
Capgemini
CONSULTING. TECHNOLOGY. OUTSOURCING
Copyright © Capgemini 2015. All Rights Reserved 12

Software Testing for BI

Lesson 1: Introduction to Software Testing for BI

Lesson Objectives

- To understand the following topics:
  - Business requirements
  - BI Project versus BI Program
  - How is BI testing different from traditional code based testing?

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 2

1.1: Business requirements

Introduction to STBI

- BI solutions development uses a distinctly different lifecycle to conventional software development
- It is iterative in nature where a single project develops subset of a more extensive BI environment.
- Thus each deliverable must go beyond meeting just the stated requirements and provide a solid foundation upon which future solutions or iterations are built.
- Beyond the differences imposed by the different lifecycle, there are several other differences that add to the complexity

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 3

What is testing?

- Testing is the practice of making objective judgments regarding the extent to which the system (device) meets, exceeds or fails to meet stated objectives

- The testing program is used to identify when the work has been "completed" so that the contract can be closed, the vendor paid, and the system
- shifted by the agency into the warranty and maintenance phase of the project.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 4

1.1: Business requirements

Introduction to STBI (Cont…)

- Analysis for software development produces a relatively stable set of requirements that serve as the basis of system design.
- BI requirements are less clear and change more rapidly.
- This circumstance makes testing of requirements difficult and creates downstream challenges when testing designs, software components, and other deliverables.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 5

1.2: BI Project verses BI program

What is BI project

- A typical software development project is self-contained.
- Dependencies are few and testing is finished when the project is completed.
- The BI program is an ongoing program without a finite end.
- Development is iterative and the system will evolve over time.
- With the continuous nature of BI development and operation there is a demand for testing.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 6

1.3: Why BI ST

Why is BI testing required?

- DW/BI pose a significant threat to organizations or applications with huge volume of data. For Eg: Banks/Financial Institutes, Insurance, Telecom etc.
- In such Institutes, the Information accuracy, consistency and reliability, information validation, data loss, reporting inefficiencies and data security are some of the important challenges that needs to be tested.
- To address these issues, the primary focus of DW/BI testing is to ensure competent and perfect database structures, ETL processes, front-end access and BI reports generations processes.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 7

1.4: How is BI testing different from traditional code based testing?

Objective of BI testing

- Software testing is predominantly testing of program code
- Verifying that it works correctly and does not fail.
- BI testing is directed at data, information and reports.
- Code failure is more readily defined and isolated than failure of data and information.
- Testing Strategy plays critical role in ensuring a common best practice testing process for the implementation of the BI Solutions.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 8

1.4: How is BI testing different from traditional code based testing?

Objective of BI testing (Cont…)

- The strategy brings in a broad understanding of Business Intelligence architecture
- It highlights the testing challenges and the value addition to the SDLC process
- It identifies generic validation approaches and the best practices for testing BI Applications.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 9

Summary

- In this lesson, you have learnt:
  - Business intelligence software testing uses a distinctly different lifecycle to conventional software development
  - It identifies generic validation approaches and the best practices for testing BI Applications.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 10

Review Questions

- Question 1: Data Warehouse Development Life Cycle has ___ phases.
  - Option 1: 5
  - Option 2: 6
  - Option 3: 7

- Question 2: Analysis phase identifies the data sources
  - Option: True / False
- Question 3: A data warehouse is really a completed project.
  - Option: True / False

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 11

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

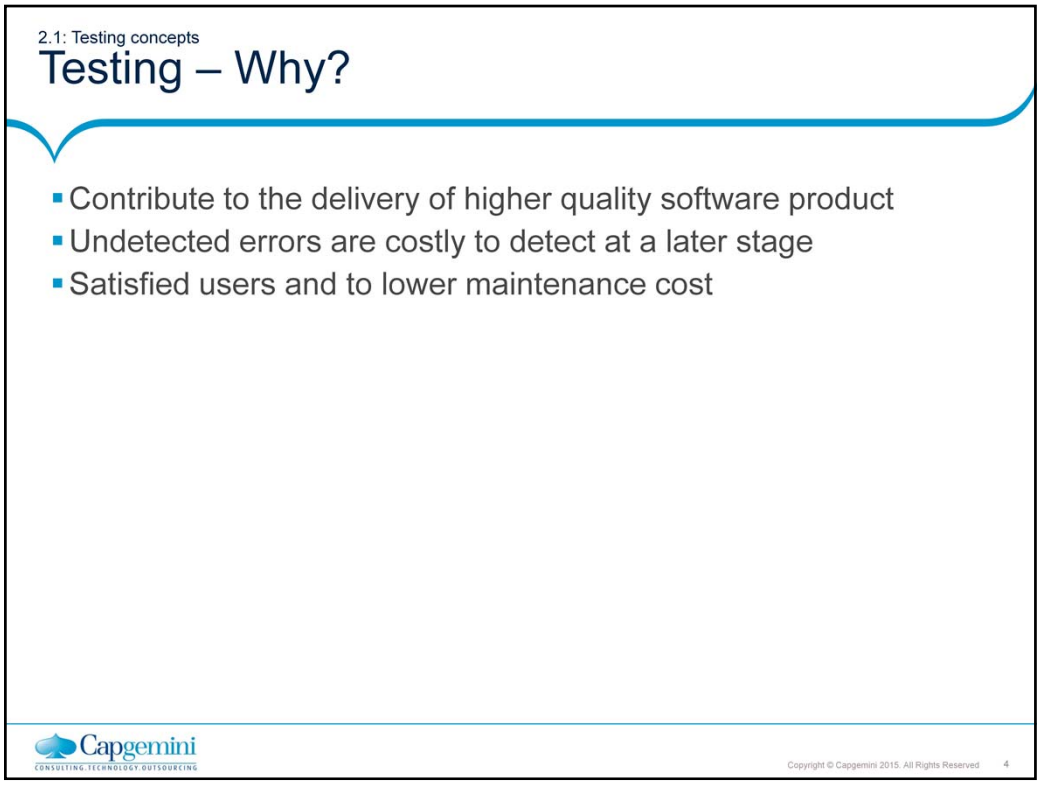
What is testing

- 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

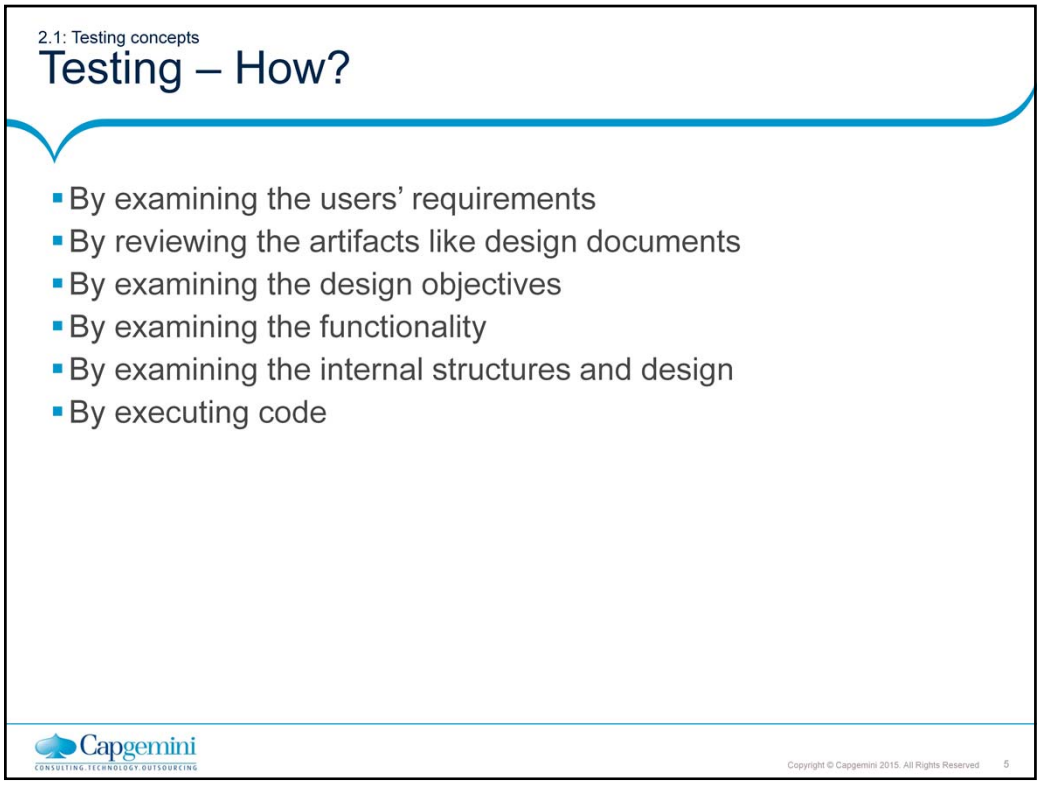
 3

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

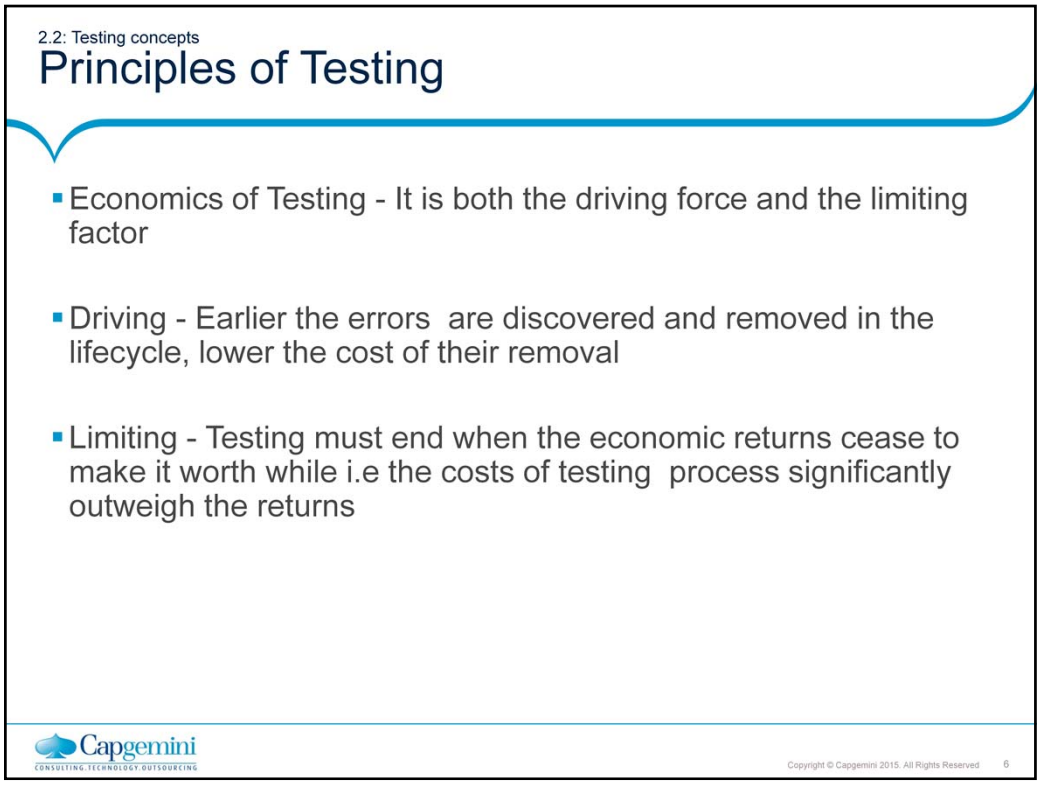


Why is testing becoming such a crucial activity? – Because applications are becoming very complex with n-tiers in an application.
When one tests a program one adds value to it through improved quality and reliability.
If not tested it can cause an unpleasant navigational error in case of a browsing applications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

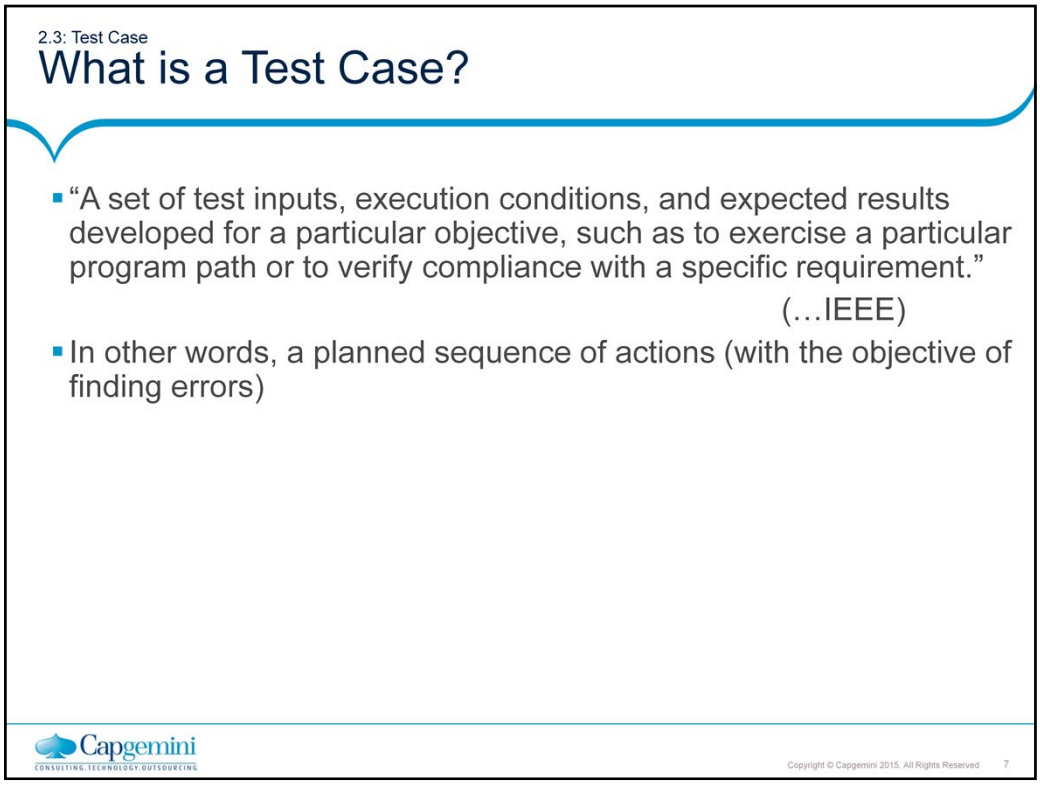
End customers are becoming more demanding & conscious about quality (defects)?
Why are company's outsourcing the testing phase? – It is being realized that testing is an extremely important phase, customers today are conscious of quality as they need to be more competitive in the market.
It is being 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.



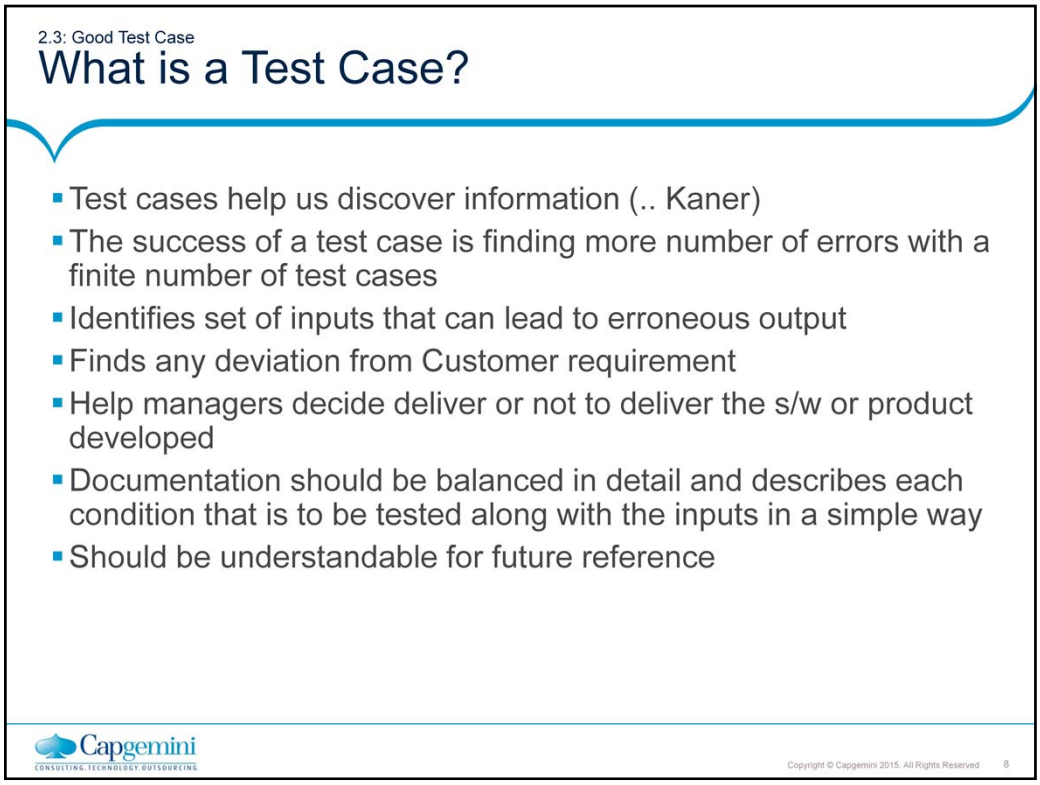
Trainer Notes



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.



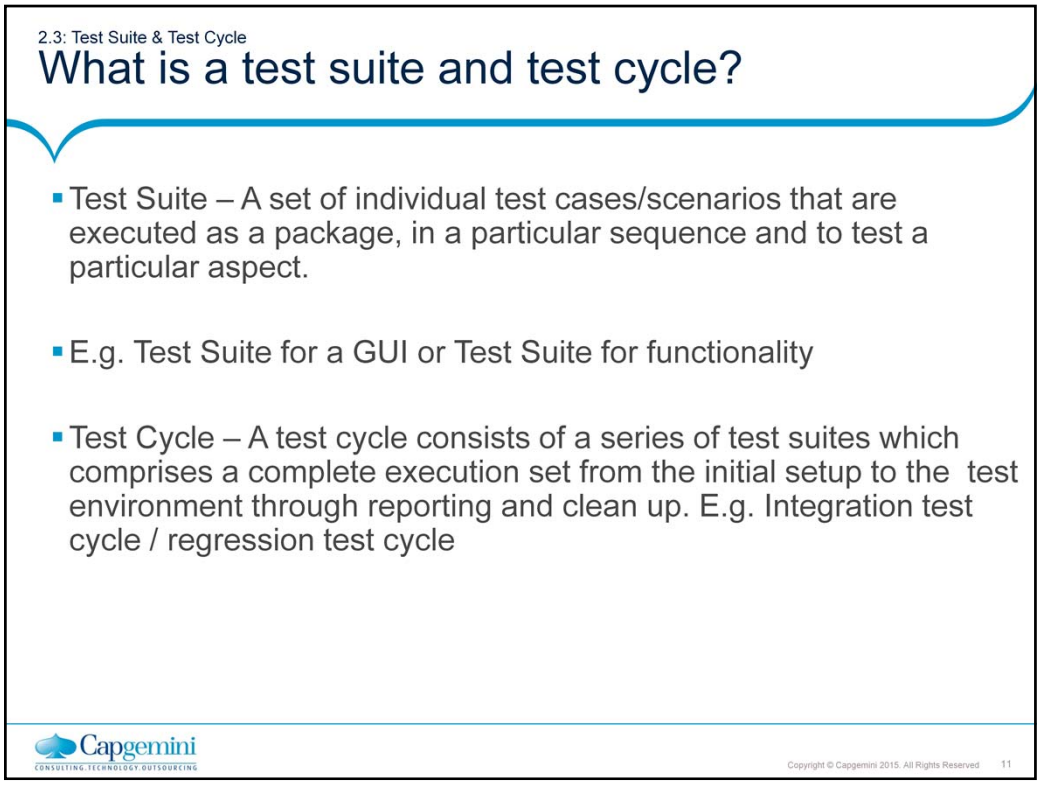
Testing at the boundary values has a higher probability of revealing errors and in case the test case detects an error it turns out to be a successful test case as well. The tester must understand the software and attempt to develop a mental picture of how the software might fail.

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

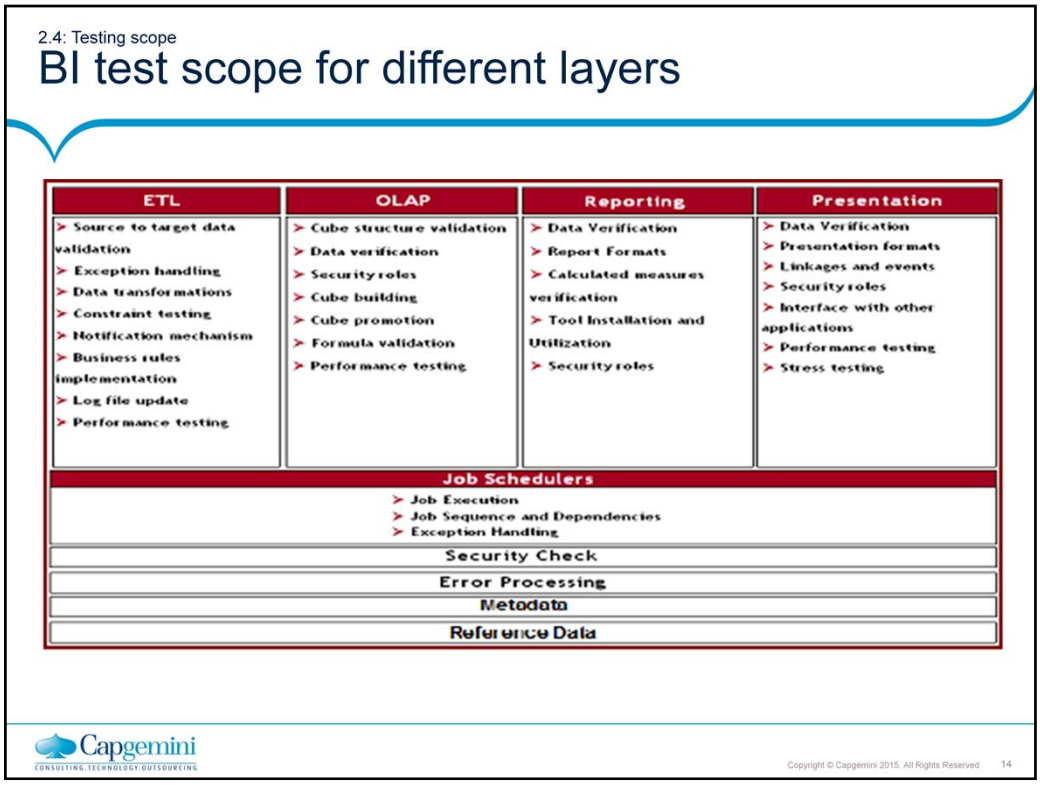
2.4: Testing scope

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.

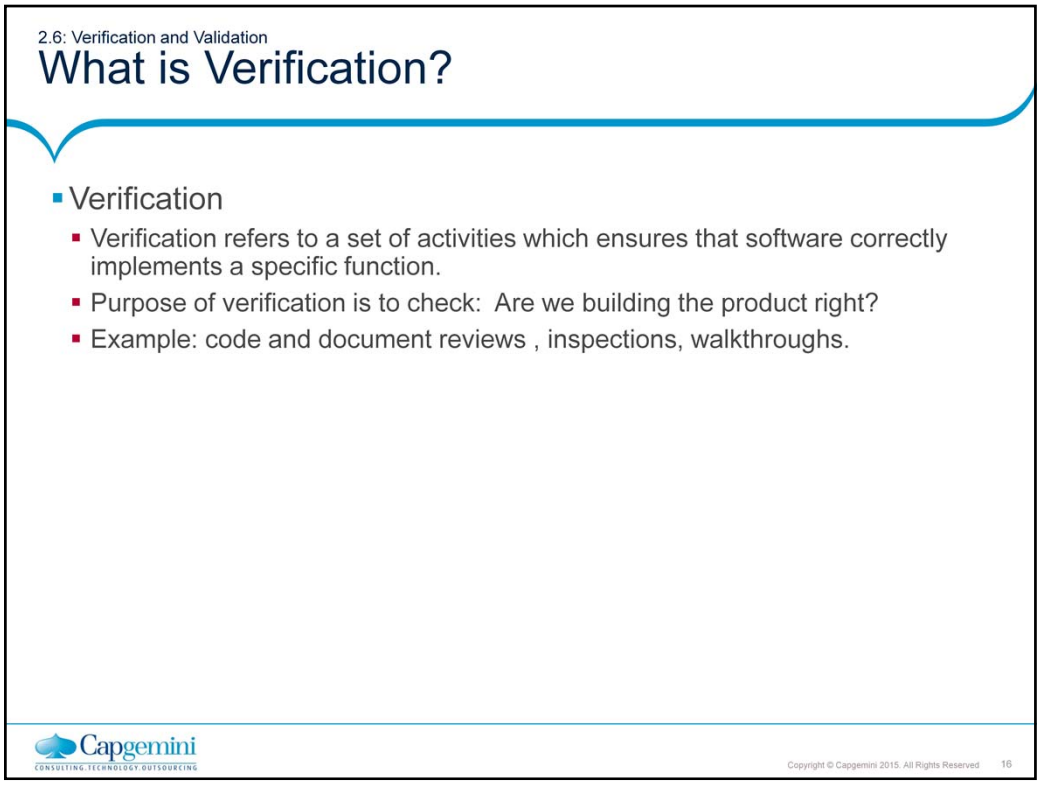
2.5: Test strategy

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

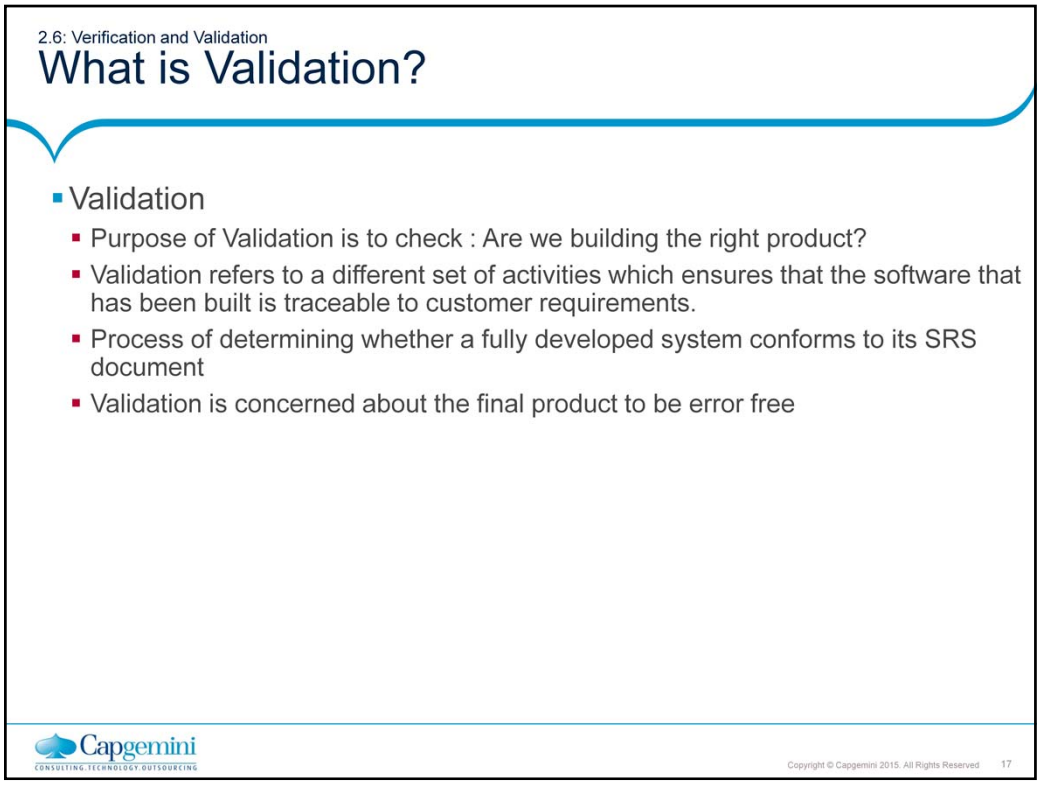
 15



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

Software Testing for BI

Lesson 3: Types of Testing

Lesson Objectives

- To understand the following topics:
  - V Model for BI Testing
    - Unit Testing
    - Integration Testing
    - System Testing
    - User Acceptance Testing
    - Operational Acceptance Testing

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 2

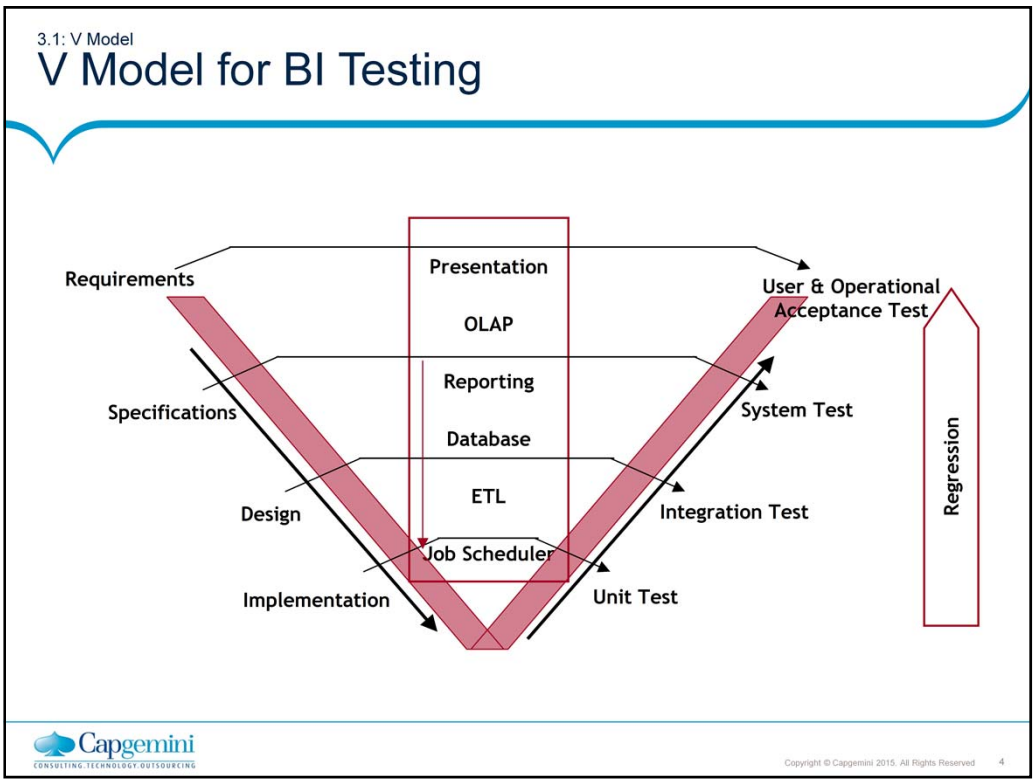
3.1: V Model

V Model for BI Testing

- There are some distinct test phases that take place in the implementation of BI project
- It is easier to visualize through the famous Waterfall model of development and V- model of testing
- The V proceeds from left to right, depicting the basic sequence of development and testing activities

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 3



V model represents various phases of software development life cycle. Each phase is been tested with different testing types. Each testing type here is concerned for the separate phase of the life cycle.

There are various phases as follows:

Requirements

Specifications

Design

Implementation

Each of these phases has corresponding testing associated with it.

Unit testing – It is done at Implementation phase.

Integration testing – It is carried out in Design phase.

System testing – It is done at Specification phase

User & Operational acceptance test – This is done at Requirement phase.

Regression Testing is the testing of software after a modification has been made to ensure the reliability of each software release. Testing after changes have been made to ensure that changes did not introduce any new errors into the system. It applies to systems in production undergoing change as well as to systems under development Re-execution of some subset of test that have already been conducted.

Types of testing

- Manual Testing
  - This type includes the testing of the Software manually i.e. without using any automated tool or any script.
  - In this type the tester takes over the role of an end user and test the Software to identify any un-expected behaviour or bug.
  - There are different stages for manual testing like unit testing, Integration testing, System testing and User Acceptance testing.
  - Testers use test plan, test cases or test scenarios to test the Software to ensure the completeness of testing.
  - Manual testing also includes exploratory testing as testers explore the software to identify errors in it.

Unit testing of software applications is done during the development (coding) of an application.
The objective of unit testing is to isolate a section of code and verify its correctness. In procedural programming a unit may be an individual function or procedure
The goal of unit testing is to isolate each part of the program and show that the individual parts are correct. Unit testing is usually performed by the developer.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 5

Manual testing

- Manual software testing is the process of manually testing software (having the possible forms for example, user interfaces navigation, information submission, or attempt to hack the software or database etc.), carried out by an individual or individuals.
- Manual software testing is labor-intensive and slow.
- It requires a tester to play the role of an end user, and use most of all features of the application to ensure correct behavior.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 6

Automation testing

- Automation testing which is also known as Test Automation, is when the tester writes scripts and uses another software to test the software. This process involves automation of a manual process. Automation Testing is used to re-run the test scenarios that were performed manually, quickly and repeatedly.
- Apart from regression testing, Automation testing is also used to test the application from load, performance and stress point of view. It increases the test coverage; improve accuracy, saves time and money in comparison to manual testing.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 7

3.1: V Model for BI Testing

Test phases for BI

- Unit Testing
- System Testing
- System Integration Testing (SIT)
- User Acceptance Testing (UAT)
- Operational Acceptance Testing (OAT)

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 8

3.1: V Model for BI Testing

Unit Testing

- It ensures that each component within the system successfully performs its individual responsibility when executed individually.
- The BI Solution development team should perform this testing on the BI development environment.
- There are some guidelines as to the general tasks that need to be executed during this testing type.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 9

3.1: V Model for BI Testing

Unit Testing (Cont…)

Layer/Area	Typical Tasks
ETL Testing	▪ Checking extraction rules ▪ Transformation validation ▪ Target system data integrity ▪ Checking input data validation ▪ Test the error-handling logic ▪ Test slowly changing dimension implementation by checking ▪ the integrity of surrogate keys ▪ Test Notifications/Warnings/Error messages

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 10

3.1: V Model for BI Testing

Unit Testing (Cont…)

Layer/Area	Typical Tasks
Report Testing	▪ Fields appearing in the reports, proper layout ▪ Drills down features, Report printed with proper format ▪ Column heading for business sense and functionality ▪ Column headers for 'Units' displayed, wherever applicable and conversion of data elements as per unit specified. ▪ Report Type, colour, labels and legends ▪ Header, footer, page header, page footer ▪ Charts and graphs, Archival of reports in different mode ▪ Data elements and labels to in the charts and the graphs

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 11

3.1: V Model for BI Testing

Unit Testing (Cont…)

Layer/Area	Typical Tasks
Universe Testing	▪ Connection parameter and control and SQL parameters. ▪ Refresh universe structure. ▪ Check parsing. ▪ Perform context and loop check. ▪ Joins and cardinalities ▪ Naming conventions ▪ Hierarchy of class, sub-class, objects ▪ Import/Export universe

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 12

3.1: V Model for BI Testing

System Testing

- It ensures that the system successfully supports the business requirements
- It also confirms that all functional requirements have been met.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 13

3.1: V Model for BI Testing

System Testing (Cont…)

Layer/Area	Typical Tasks
Data Acquisition	▪ Successful source access ▪ Application and validation of cleansing algorithm/logic ▪ Proper delivery of the files to staging area/ODS – ▪ formatting, layouts, file naming convention, order of ▪ precedence ▪ File Loads, Dependencies, Completeness ▪ Formats and layouts ▪ Naming conventions ▪ Scripts and Triggers

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 14

3.1: V Model for BI Testing

System Testing (Cont…)

Layer/Area	Typical Tasks
ETL Testing	▪ Successful staging area access, Order of Extraction ▪ Successful extraction of data ▪ Application and validation of transformation logic ▪ Rejects based on applied algorithms ▪ Recovery and Restart, Data Auditing/Logs ▪ Proper generation of the code ▪ Scheduling, Job triggers, Job decencies ▪ Alerts and notification, Warnings and check point validations ▪ Metadata recording/deliver to internal/external repositories

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 15

3.1: V Model for BI Testing

System Testing (Cont…)

Layer/Area	Typical Tasks
OLAP/ Reporting	▪ Correct reflection of the schema in the tool architecture ▪ Source to object mappings ▪ Correct formatting of the object, Report schedules ▪ Correct definitions, constructions of derived measure/metrics ▪ Population of universe, triggers, data and structures ▪ Hierarchical Organization of Dimension, user friendliness ▪ Online analytical computation, e.g. drill downs, running sums ▪ Time computations, Report layouts, formatting

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 16

3.1: V Model for BI Testing

System Integration Testing

- It ensure seamless run of the entire process within an application or a specific stage.
- It focuses on the details of each of the steps/modules.
- It helps in capturing the responses as the data moves across the system.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 17

3.1: V Model for BI Testing

System Integration Testing (Cont…)

Layer/Area	Typical Tasks
Data Acquisition	▪ Successful runs of each job ▪ Successful execution of each script ▪ Proper delivery of the files to staging area in the order ▪ Capture of errors ▪ Capture of rejects ▪ Processing of statistics/audits trails
OLAP	▪ Report Schedules ▪ Correct reflection of data from data warehouse onto reports, Query generation to test table joins

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 18

3.1: V Model for BI Testing

System Integration Testing (Cont…)

Layer/Area	Typical Tasks
ETL	▪ Triggered start of the ETL process ▪ Successful extraction in order of precedence ▪ Count of rejects based on applied algorithm ▪ Successful run of dependent jobs, Failure of jib runs ▪ Restarts, Alerts and notification deliveries
Operational/Data Quality	▪ Audit trails, Triggered start of the Data Feedback process ▪ Successful audit trail in order ▪ Count of rejects based on applied algorithm, Restarts ▪ Successful run of dependent jobs, Failure of job runs

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 19

3.1: V Model for BI Testing

System Integration Testing (Cont…)

Layer/Area	Typical Tasks
End-to-End	▪ Black-box Testing, Random Testing, Time computations ▪ Source (File load) to object (OLAP) mappings ▪ Report layouts, formatting, user friendliness ▪ Hierarchical Organization of Dimension ▪ Correct reflection of data onto reports ▪ Ad-hoc features/builders, Online analytical computation
Performance	▪ Benchmarking, Gauge the impact of configuration, Tuning ▪ Load/Volume Test, Availability, Log into Application

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 20

3.1: V Model for BI Testing

User Acceptance Testing

- Business users of the system, Business Analysts and Test Analysts execute predefined Test Cases emulating production business scenarios.
- The test shall be designed to ensure that the users can interact with the system and accomplish daily business operations.
- Specific test cases need to be established, and expected results identified.
- This testing should be for specific BI functions, including data transformation rules, and data correctness.

3.1: V Model for BI Testing

User Acceptance Testing (Cont…)

- User acceptance testing should be completed in the BI testing environment, during the construction stage and then in production at the implementation.
- Business Stakeholders should have a nominated Testing manager who will ensure that all UAT testing, is planed, prepared and executed, to the satisfaction of the BI Program Manager.
- UAT Testing covers
- Functional testing
- Business BI information testing

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 22

3.1: V Model for BI Testing

User Acceptance Testing (Cont…)

Layer/Area	Typical Tasks
Business Testing	▪ Information accuracy ▪ Source data rejections ▪ Data transformation/aggregation rules ▪ Key performance metrics/reports ▪ Information presentation ▪ User access points to OLAP tools, query tools

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 23

3.1: V Model for BI Testing

Operational Acceptance Testing

- It verifies that a solution is able to operate at the expected volumes in production and is able to support the response times as specified .
- It mainly focuses on system performance, deployments, backup and recovery.
- Operation Acceptance Testing (OAT) should be carried out by the IT team to ensure the technical operation of the EDW solution is acceptable to both the business users and to the IT's operational staff.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 24

3.1: V Model for BI Testing

Operational Acceptance Testing (Cont…)

Layer/Area	Typical Tasks
Database	▪ Multiple users logging in the system ▪ Queries retrieving massive amount of data, e.g., across ▪ multiple time units ▪ Simultaneous queries lunched to the system ▪ Multiple users running large queries
Enhancement	▪ Ensuring all task change are carried out from previous baseline
Availability	▪ Availability of various components of BI solution

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 25

3.1: V Model for BI Testing

Operational Acceptance Testing (Cont…)

Layer/Area	Typical Tasks
Deployment	▪ Completeness
Disaster Recovery	▪ Disaster Recovery procedures ▪ Recovery using the customer daily back-ups procedure
Documentation	▪ Various documents as specified in OAT certification template
Monitoring	▪ Checking Automated Alerting
Middleware	▪ Checking Performance and Load of the system
Network	▪ Checking Performance and Load of the system

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 26

Summary

- In this lesson, you have learnt:
  - There are various test phases of V model for BI testing like
    - Unit Testing
    - System Testing
    - System Integration Testing (SIT)
    - User Acceptance Testing (UAT)
    - Operational Acceptance Testing (OAT)

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 27

Review Questions

- Question 1: Testing a software with execution on a computer
  - Option 1: Static testing
  - Option 2: Dynamic testing
  - Option 3: Automated testing

- Question 2: Operational acceptance testing ensures that the system successfully supports the business requirements
  - True / False

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 28

Software Testing for BI

Lesson 4: Testing for BI

Lesson Objectives

- To understand the following topics:
  - Testing document purpose (Test documentation)
  - General BI Testing Principles
  - BI Testing Mission
  - Production Verification Testing
  - Possible Areas of Automation

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 2

4.1: Testing document purpose

Test documentation

- The purpose of this document is to define a template for the Testing Strategy of Business Intelligence Solutions.
- The document will:
  - Identify and describe each testing phase
  - Identify and describe each test type
  - Define entry and exit criteria of each test phase including test activities and validation tasks

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 3

4.1: Testing document purpose

Test documentation (Cont…)

- Each phase will have its own test plan prepared and accepted that will cover all aspects in detail appropriate to that phase.
- It will reference this document where necessary.
- The document also sets some high-level guidelines for BI testing.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 4

4.2: BI testing principles

General BI testing strategy

- For each Reporting Solutions, all canned reports specified with the business should each be tested individually.
- To demonstrate the add-hoc capabilities of the system, the solution development vendor should demonstrate that all canned reports within the scope of the application can be produced using the ad-hoc toolset.
- Following release to production there should be process to support retro fitting of minor changes done by the application-support team and with major changes done by the development team.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 5

4.2: BI testing principles

General BI testing strategy (Cont…)

- BICC should record defects in a tool (such as excel or better) with the intention to display bugs per modules and bug discovery rate
- As data quality is a major consideration in these type of projects the following should apply:
  - Data quality issues should be fixed in the upstream system, rather in the data warehouse.
  - A review of the data quality should be performed at each project decision point.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 6

4.2: BI testing principles

General BI testing strategy (Cont…)

- The usability test should be conducted as follows.
  - Step 1 - Untrained users should be trained on the usage of the system.
  - Step 2 - These users should then be given a list of tasks to perform within the environment. The users performance should be monitored.
  - Step 3 – Based on the users feedback and performance either the training data set, or the training material should be improved.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 7

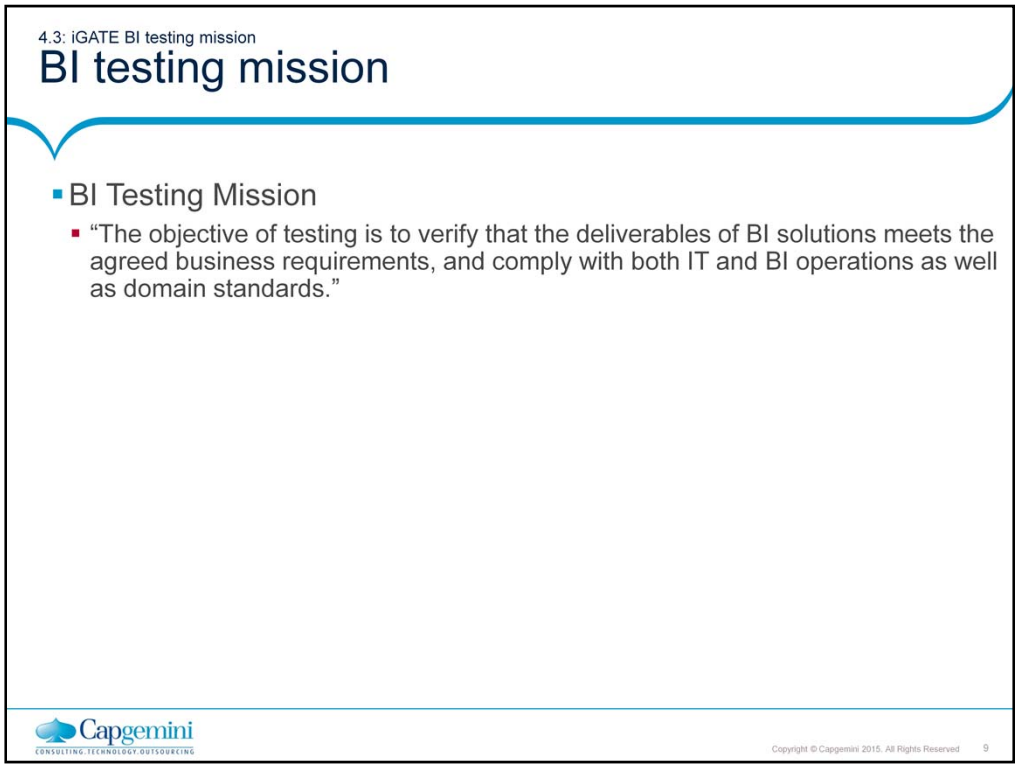
4.2: BI testing principles

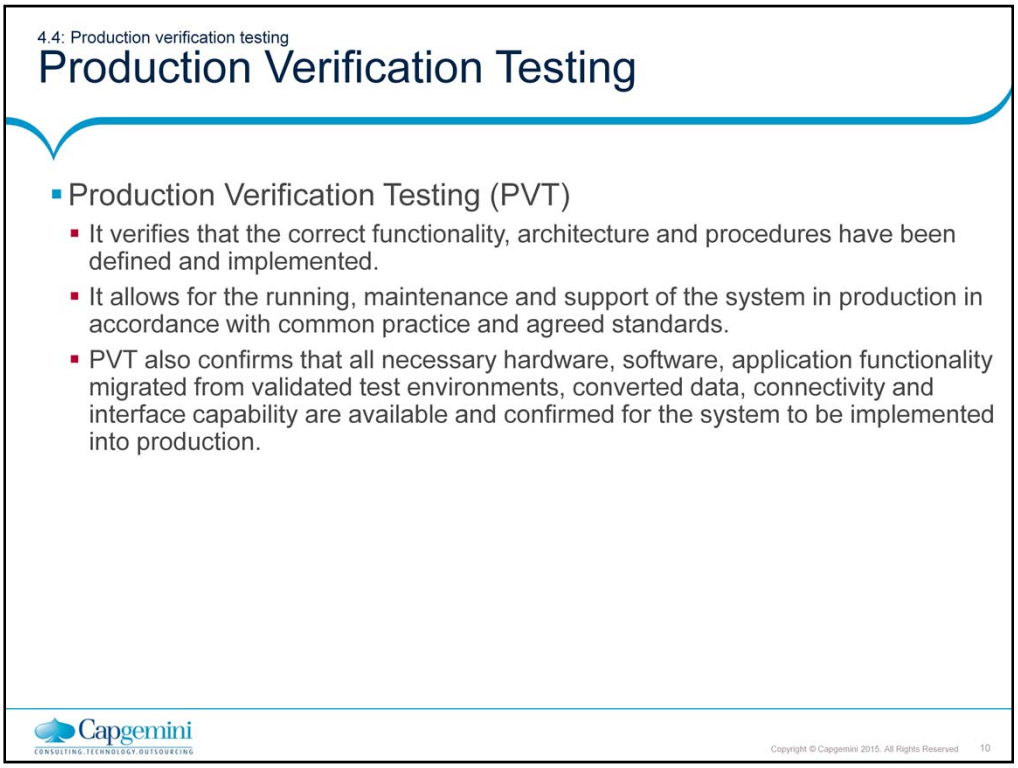
General BI testing strategy (Cont…)

- In case of unavailability of sufficient test data a test data-generating engine such as GS Data Generator or DBUnit may be used.
- For data security purposes, test data should be generated by appropriately massaging production data

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 8


4.3: iGATE BI testing mission
BI testing mission
BI Testing Mission
"The objective of testing is to verify that the deliverables of BI solutions meets the agreed business requirements, and comply with both IT and BI operations as well as domain standards."
Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING
Copyright © Capgemini 2015. All Rights Reserved
9


4.4: Production verification testing
Production Verification Testing
Production Verification Testing (PVT)
It verifies that the correct functionality, architecture and procedures have been defined and implemented.
It allows for the running, maintenance and support of the system in production in accordance with common practice and agreed standards.
PVT also confirms that all necessary hardware, software, application functionality migrated from validated test environments, converted data, connectivity and interface capability are available and confirmed for the system to be implemented into production.
Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING
Copyright © Capgemini 2015. All Rights Reserved
10

4.5: Possible areas of automation

Possible areas of automation

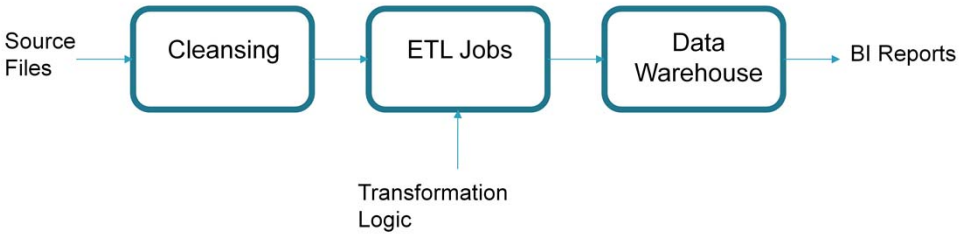
ETL Layer	▪ Generic Data Reconciliation: Programs to automate Data Reconciliation process ▪ Generic Data Auditing Routines: Programs to collect statistics related to the data being loaded from source to target
Portal/OLAP Reporting Layer	▪ Portal Look and Feel Check ▪ Reports Generation & Data Validation (CSV File Compare)
Performance /Stress Testing	▪ ETL Process Performance ▪ Report Generation Time ▪ Response times OLAP & Presentation Layer

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 11

ETL Testing

- ETL Testing :
  - Cleansed Data from the Source file is loaded into Data Warehouse after implementing the transformation logic
  - Source Files can be any of Flat Files, XML Files or Database Tables.



Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

12

Continuation……

- Validating the Transformation logic implemented in ETL Jobs while loading the Data from individual Source system to Data Warehouse.
- Compare the data in Data Warehouse with source data according to the logic given in mapping document.
- Mapping Document contains the Information regarding..
- Source System :Source File Name, Source Field(s) and its Data type
  - Transformation Logic.
  - Target System (Data Warehouse) :Target Table Name , Target Column Name and its data type.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

ETL Testing Using SQL

- Source Data Tables: Data from source files is replicated into data base tables using any SQL Loader or Simple ETL Jobs.
- Using SQL, Target(DWH) and Source tables are compared with Minus operator.
- Applying the transformation logic on source fields according to the logic given and comparing it with the field in target.
- Syntax :

```
Select   T/L (Source_Field) from Source_Table
Minus
        Select Target_Field from Target_Table
```

- No Logic to be implemented on Target.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 14

Source System Unique Key

- All Attributes in the Target Table are validated with reference to Unique Key in the Table.
- For Each Table in the Data Warehouse Unique key is implemented based on logic given in Mapping Document.

T/L(Src_Fld1,Src_Fld2)--> Unique_Key_DW

Syntax:

Select Uniq_Ky, Source_Field from source_Table

Minus

Select Uniq_Ky,Target_Field from Target Table

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

15

Smoke Test

- It confirms that total no. of records present in both Source and Target are equal.
- It's a basic test for testing a Data load/Migration.
- Syntax:

```
Select count (distinct Uniq_Ky) from Source_Table.
Select count (distinct Uniq_Ky) from Target_Table.
```

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 16

SSUK Check

- Assuring that all the records present in Source Table are loaded into the Target Table.

- Syntax:
- Select distinct Uniq_Ky from Source_Table
- Minus
- Select distinct Uniq_Ky from Target_Table

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

Attribute Testing

- Validating the Transformation Logic for all attributes with reference to the Unique_Key.
- Syntax:

  Select Uniq_Ky, T/L(Source_Field) from source_Table
  Minus
  Select Uniq_Ky,Target_Field from Target Table

- Some of the fields in DWH are set to a default value say '0/NULL/Space'

- Validating the Default Set Fields (0/NULL/Space) in DWH:

  Select Count(*) from Target_Table
  Where Target_Field <> '0/Null/Space'

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

18

Multiple Loads

- Data gets loaded into Data warehouse usually in different loads for various Periods, say monthly, Bi-Monthly, Quarterly, Half yearly or Yearly based on Business Needs.
- Full File Load : From Second Period Onwards Snapshot of Total Records at second period appear in Source File.
- Delta File Load: From Second Period Onwards Only Changed Records appear in Source File.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

CDC

- CDC : Change Data Capture in Data Warehouse.
- Data from Multiple Loads is controlled in DW using Two Date Fields in the Table Say EFFV_DATE and END_DATE.
- For an Active record EFFV_DATE is the Period Load Date and END_DATE is any Unused Long Date.
- For example when the first load is on 01-May-2011, the EFFV_DATE will be '01-May-2011' and END_DATE will be '31-Dec-9999'

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 20

Continuation……

- When the second load occur on '01-Jun-2011', for a given Uniq_Ky,
- 1)When the records for both Periods Matches, then no New Record is Inserted for P2 in DW and No change in Action Flag.
- 2)When the record in the Period 1 is deleted in Period 2, then END_DATE for Period 1 record gets changed to '31-May-2011' in DW.
- 3)When the record in Period 2 is a newly inserted record and not exist in Period 1, then a new record inserted with EFFV_DATE as'01-Jun-2011' and END_DATE as '31-Dec-9999'
- 4) When the records for both periods not matches, then a new record inserted with EFFV_DATE as'01-Jun-2011' and END_DATE as '31-Dec-9999' and the END_DATE for existing record with EFFV_DATE as '01'May-2011' gets changed to '31-May-2011'.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

Continuation……

- Example:

- Period 1 Load Date : '01-May-2011
- Period 2 Load Date : '01-Jun-2011

Period	Unchanged	Deleted	Inserted	Changed
Period 1	EFFV_DT:' 01-May-2011' END_DT : '31-Dec-9999'	EFFV_DT:' 01-May-2011' END_DT : '31-May-2011'	No Records	EFFV_DT:' 01-May-2011' END_DT : '31-May-2011'
Period 2	No Records	No Records	EFFV_DT:' 01-Jun-2011' END_DT : '31-Dec-9999'	EFFV_DT:' 01-Jun-2011' END_DT :'31-Dec-9999'

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 22

Modified Smoke and Attribute Test Scripts:

- In the Target Side condition for restricting the records for that specific period is included.
- SSUKT Check:

```
Select distinct Uniq_Ky from Source_Table_P1
Minus
Select distinct Uniq_Ky from Target_Table
Where Load_Date between EFFV_DATE and END_DATE
```

- Attribute Validating Script:

```
Select Uniq_Ky, T/L(Source_Field) from source_Table_P1
        Minus
        Select Uniq_Ky,Target_Field from Target Table
        Where Load_Date between EFFV_DATE and END_DATE
```

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

23

CDC Check

- For Full File:

1)Matched record Check.

Source Query:

Select Uniq_Ky, T/L(Field 1),T/L(Field 2) from Source_Table_Period_1

Intersect

Select Uniq_Ky, T/L(Field 1),T/L(Field 2) from Source_Table_Period_2

Target Query:

Select Uniq_Ky, Effv_dt,End_Dt from Target_Table

Where Uniq_Ky in(Sample records from Source Query Result)

Expected Result: There should be one record for each Unique Key with EFFV_DT as '01'May-2011'

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

Continuation……

2) Deleted Record check:

Source Query:

Select Uniq_Ky from SRC_P1
Where Uniq_Ky not in (Select Uniq_ky from SRC_P2)

Target Query:

Select Uniq_Ky,EFFV_DATE,END_DATE from Target_Table
Where Uniq_Ky in(Sample records from Source Query Result)
Only one record for each Uniq_Ky with EFFV_DATE '01-May-2011' and End Date as '30-Jun-2011'

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

Continuation……

3) Inserted Record s Check:

Source Query:

Select Uniq_Ky from SRC_P2

Where Uniq_Ky not in (Select Uniq_ky from SRC_P1)

Target Query:

Select Uniq_Ky,EFFV_DATE,END_DATE from Target_Table

Where Uniq_Ky in(Sample records from Source Query Result)

Expected Result: Only one record for each Unique Key with EFFV_DATE as '01-Jun-2011' and End Date '31-Dec-9999'.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

Continuation……

4)Un-Matched Record Check.

Source Query:

Select Uniq_Ky,T/L(Field 1),T/L(Field 2) from Source_Table_Period_1

Minus

Select Uniq_Ky,T/L(Field 1),T/L(Field 2) from Source_Table_Period_2

Target Query:

Select Uniq_Ky,EFFV_DATE,END_DATE from Target_Table

Where Uniq_Ky in(Sample records from Source Query Result)

Expected Result: There should be Two record one with EFFV_DATE as '01-May-2011' and END_DATE as '31-May-2011' and the other with EFFV_DT as '01-Jun-2011' and END_DATE as '31-Dec-9999'.

But the Source Query gives both Un-Matched and Deleted Records.

Continuation……

- Modified Source Query for Unmatched Record Check:
- Select Uniq_Ky,T/L(Field 1),T/L(Field 2) From Source_Table_Period_1
- Where Uniq_Ky in(Select Uniq_Ky From Source_Table_Period_2)
  Minus
- Select Uniq_Ky,T/L(Field 1),T/L(Field 2) From Source_Table_Period_2

- Suggested Query To Overcome Performance Issue:
  - Select Uniq_Ky From Source_Table_Period_1 P1
  - Inner join Source_Table_Period_2 P2
  - On(P1.Uniq_Ky=P2.Uniq_Ky and P1.Fld_1 <> P1.Fld_1 and P1.Fld_2<> P2.Fld_2)

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

28

Continuation……

- For Delta File:
- For Unchanged Records: No Record Present in Period 2 Load.
  - Select Uniq_Ky from SRC_P1

Where Uniq_Ky not in (Select Uniq_ky from SRC_P2)

- For Changed Records : An action field exist.
  - Select Uniq_Ky from Source_Table_Period_2 where action ='Insert/Delete/Modified'

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

Duplicate Record Check In CDC

- Select uniq_Ky,EFFV_DT_END_DT ,count(*)from target_Table group by uniq_Ky,EFFV_DT_END_DT having count(*) > 1

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

DATA COMPLETENESS AND QUALITY CHECK

- An integral part of DWH testing is verifying the quality and completeness of data. Data completeness testing ensures that all expected records from the source are loaded into the database by reconciling with error and reject records. A data quality check ascertains proper and accurate data, as per the recommended standard, is processed to the data warehouse; this includes data transformation testing.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

DATA COMPLETENESS AND QUALITY CHECK

- The following activities are recommended, to determine data completeness and quality:
- Data extraction process for both historical and incremental loads
- Data cleansing checks, based on standards; here the testing reject threshold is important
- 'Source to target' transformation validation for thoroughness and accuracy
- Historical and incremental transformation process validation
- 'Reject and error' record analysis and validation
- Scenario-based testing with specified transformation rules
- Record reconciliation testing by comparing source, error, reject and target records to prevent record leakage
- Data load process check for both historical and incremental load process
- Negative testing for all the above mentioned cases

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 32

What Are Key Performance Indicators (KPI)

- Key Performance Indicators (KPI) are used by organizations to evaluate success and when you choose KPI's you should follow the smart approach
- Specific – a well defined goal that is clearly understood by everyone.
- Measurable – can you track your progress towards the goal?
- Agreed – both employer and employee must agree on what the goals are.
- Realistic – can you achieve the goal with the resources provided?
- Time related – will there be enough time to complete the task?

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 33

So what is a “Report”?

- A report is made up of three components:
  - Data: specifies how to extract information from backend data sources and information on the structure of that data.
  - Layout: how the information is to be presented.
  - Properties: parameters, interactions, etc.
- Typically, the report is re-used at intervals.
  - It picks up the current data from the data sources.
- The report definition may be stored in XML.
  - An XML report template can be used to define a family of related reports

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 34

KPI Testing

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 35

BI REPORT DATA TESTING

- Another important aspect of DWH testing is confirmation of the accuracy and completeness of business intelligence (BI) reports. These may vary in appearance, turnaround time, report accuracy and usability, but testing this is of paramount importance as this will be reflected in the UI and is what the end users will eventually see. The following activities are key while testing BI reports:
  - Restriction of users' access to reports, with multiple layers of security
  - Validation of the accuracy and relevance of the data displayed in each report
  - Ensuring sufficient information for analyzing graphical reports
  - Relevancy of options in the drop down lists in each report
  - Testing of pop-up reports and child reports with proper data flow from parent reports
  - Functionality of additional features such as report storage into PDF formats, print options

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 36

Summary

- In this lesson, you have learnt:
  - Test document identifies and describes each testing phase.
  - Each phase has its own test plan prepared and accepted.
  - The objective of testing is to verify that the deliverables of BI solutions meets the agreed business requirements.
  - contd.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 37

Summary (Cont…)

- In this lesson, you have learnt:
  - Production verification testing confirms the system to be implemented into production.
  - There are various areas of BI where in testing can be automated.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 38

Review Questions

- Question 1: _____ is to define a template for the Testing Strategy of Business Intelligence Solutions
- Question 2: Data quality issues should be fixed in the Data warehouse
  - True / False

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 39