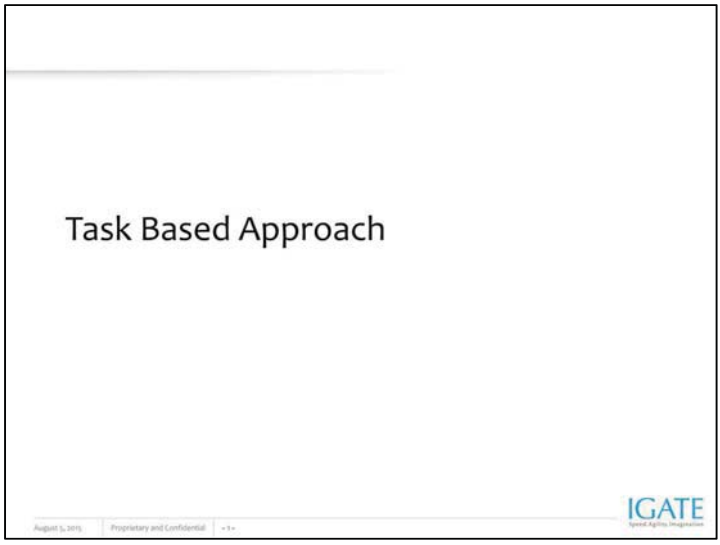


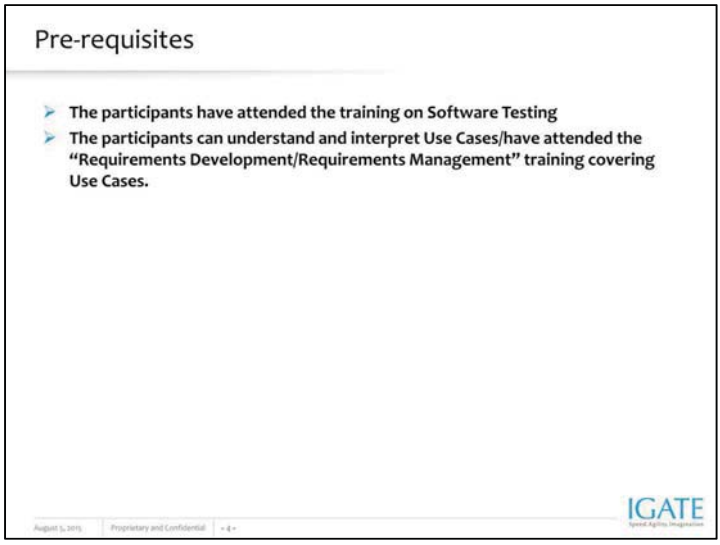
Document History

Date	Course Version No.	Software Version No.	Developer / SME	Change Record Remarks
	0.1D	NA		Content Creation
	0.1	NA		Review
May-2009	2.0	NA	Priya Rane	Material Revamp
Jun 2011	2.1	NA	Selvalaxmi	Material revamp

Course Goals and Non Goals

- **Course Goals**
  - At the end of this program, participants gain an understanding of how to describe the Task based approach to software testing and to describe the "sequence of developing test cases" using the task based approach.

- **Course Non Goals**
  - This course does not cover automation process of testing.

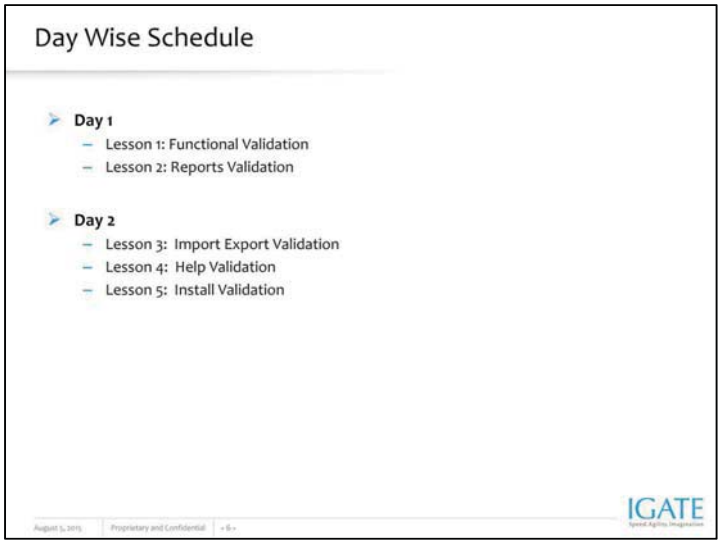

Pre-requisites
The participants have attended the training on Software Testing
The participants can understand and interpret Use Cases/have attended the "Requirements Development/Requirements Management" training covering Use Cases.
IGATE

Intended Audience

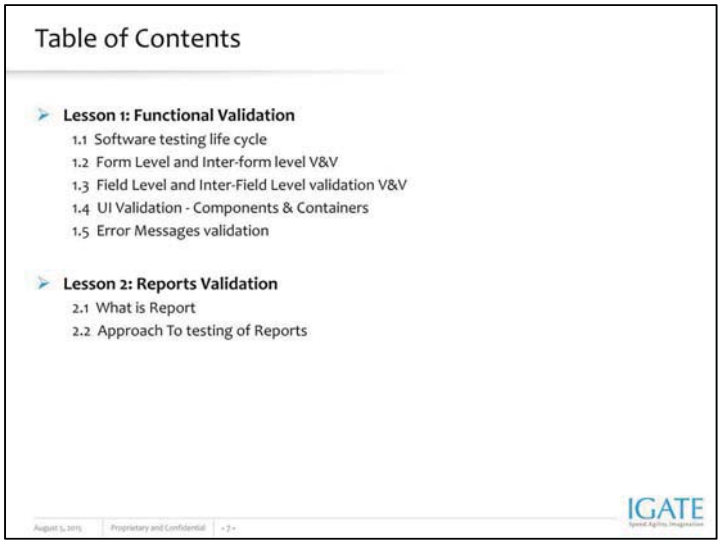
- Software Engineers and Senior Software Engineers.

IGATE
Speed. Agility. Imagination

Task Based Approach



Task Based Approach



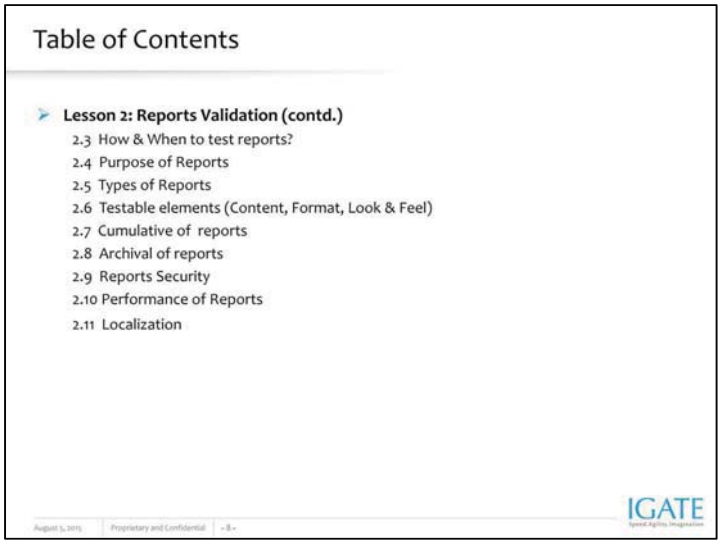


Table of Contents
Lesson 2: Reports Validation (contd.)
2.3 How & When to test reports?
2.4 Purpose of Reports
2.5 Types of Reports
2.6 Testable elements (Content, Format, Look & Feel)
2.7 Cumulative of reports
2.8 Archival of reports
2.9 Reports Security
2.10 Performance of Reports
2.11 Localization
IGATE

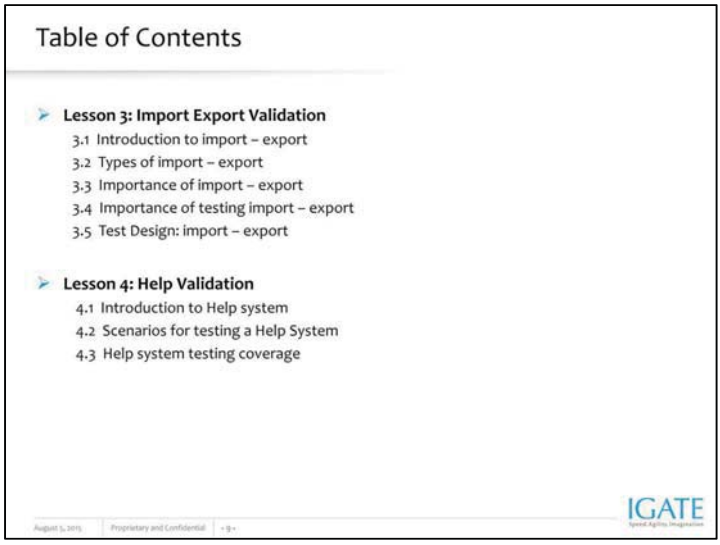


Table of Contents

- **Lesson 3: Import Export Validation**
  - 3.1 Introduction to import – export
  - 3.2 Types of import – export
  - 3.3 Importance of import – export
  - 3.4 Importance of testing import – export
  - 3.5 Test Design: import – export
- **Lesson 4: Help Validation**
  - 4.1 Introduction to Help system
  - 4.2 Scenarios for testing a Help System
  - 4.3 Help system testing coverage

IGATE
Speed. Agility. Imagination

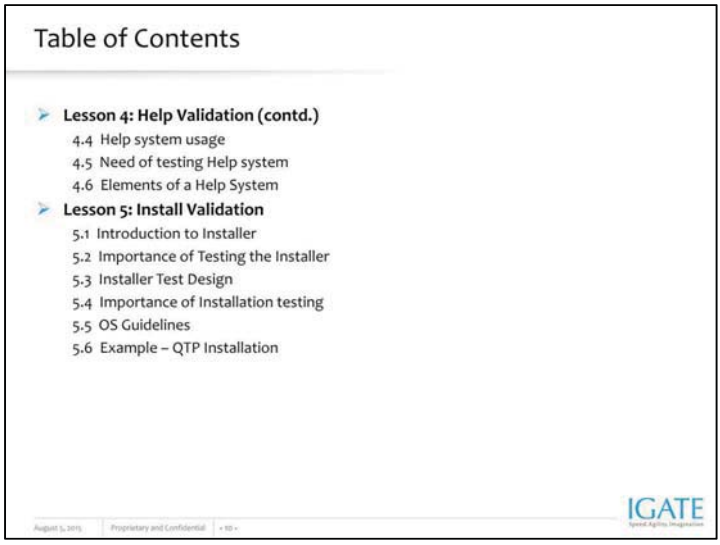


Table of Contents
Lesson 4: Help Validation (contd.)
4.4 Help system usage
4.5 Need of testing Help system
4.6 Elements of a Help System
Lesson 5: Install Validation
5.1 Introduction to Installer
5.2 Importance of Testing the Installer
5.3 Installer Test Design
5.4 Importance of Installation testing
5.5 OS Guidelines
5.6 Example – QTP Installation
IGATE

References

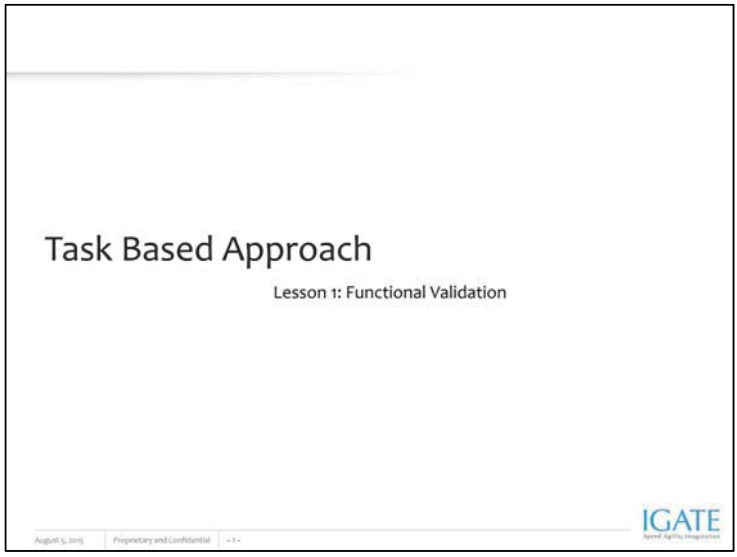
- **Book:**
  - Techniques for Functional Testing of Software and Systems - Boris Beiz
  - The Software Test Engineer's Handbook - Graham Bath and Judy McKay

IGATE
Speed. Agility. Imagination

Next Step Courses

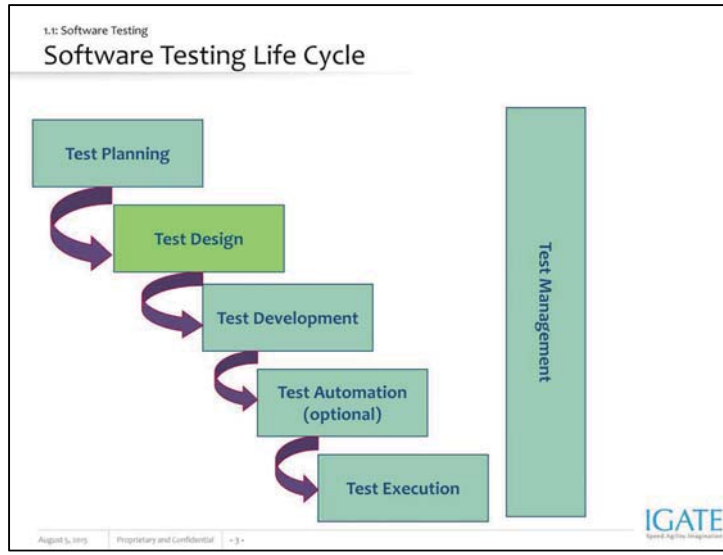
- **DFDR**

IGATE
Speed. Agility. Imagination


Task Based Approach
Lesson 1: Functional Validation
IGATE

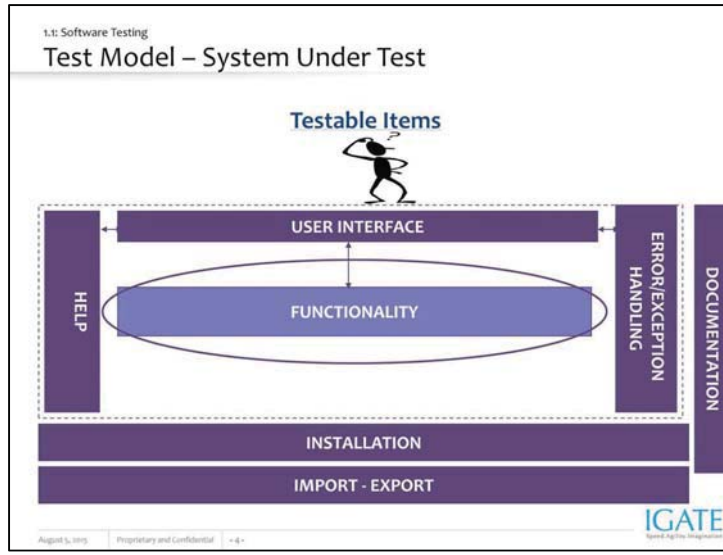
Lesson Objectives

- **To understand the following topics:**
  - Software testing life cycle
  - Form Level and Inter-form level V&V
  - Field Level and Inter-Field Level validation V&V
  - UI Validation - Components & Containers
  - Error Messages validation



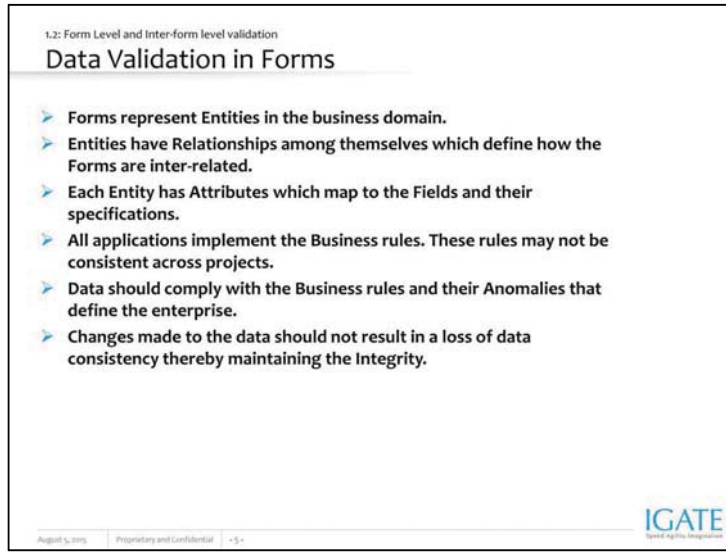
Test Planning – Plan is a strategic document which describes how to perform a task in an effective efficient and optimized way.
Test Scope, Test Environment ,Different Test phase and Test Methodologies, Manual and Automation Testing, Defect Mgmt, Configuration Mgmt, Risk Mgmt. Etc
Evaluation & identification – Test, Defect tracking tools
Test Design - Test Design is done based on the requirements of the project. Test has to be designed based on whether manual or automated testing is done. For automation testing, the different paths for testing are to be identified first. An end to end checklist has to be prepared covering all the features of the project.
Test Traceability Matrix and Test coverage, Test Scenarios Identification & Test Case preparation, Test data and Test scripts preparation, Test case reviews and Approval, Base lining under Configuration Management
Test Development - This is the most important stage of the testing life cycle, in this section or phase of the testing part; the testers will develop the test cases based against the requirements of the customer. There are usually three levels of requirements, to be understood by the testers before they can proceed to write the test cases for the product
- HLI (High level Information)
- LLI / Use Cases (Low level Information)
- Snapshots (Prototype or images of a similar product or framework.)

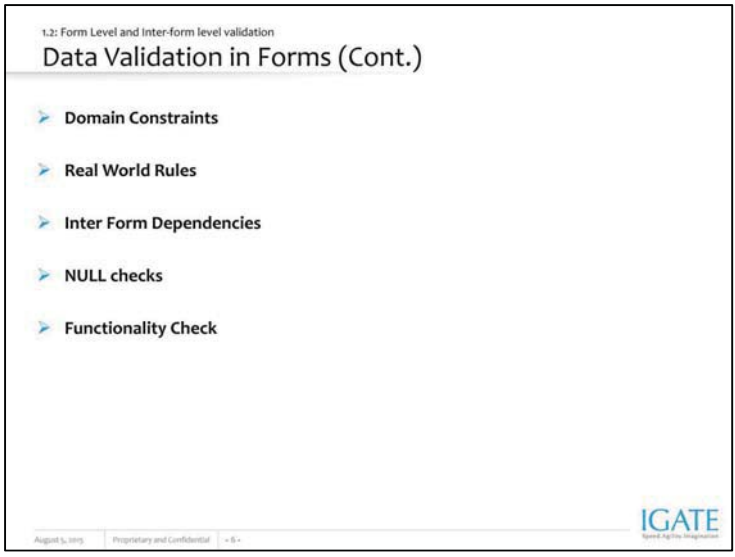
Test Automation - Software testing using an automatic test program will generally avoid the errors that humans make when they get tired after multiple repetitions. The test program won't skip any tests by mistake. The test program can also record the results of the test accurately. The results can be automatically fed into a database that may provide useful statistics on how well the software development process is going.
Test Execution - This is the phase where the test engineers will prepare and execute the test cases. One of the best techniques is to refer the use cases, pick up the standard test case templates and prepare test cases based on different categories. While do remember to maintain Traceability matrix alongside test execution to avoid rework and confusions at a later stage.

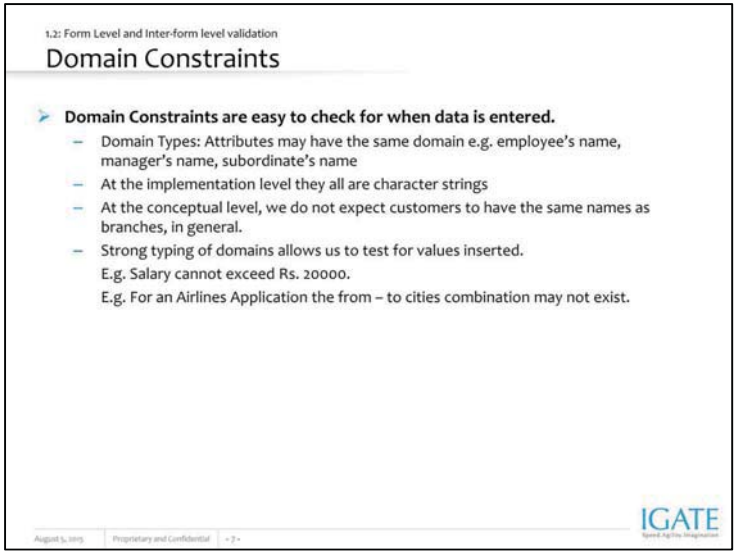


The system under test is considered for different validations during testing. They are as follows

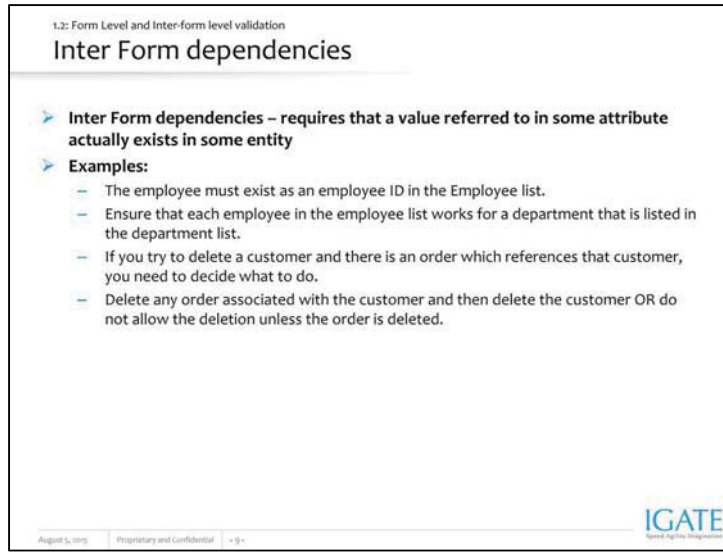
1. Functional Validation
2. User Interface Validation
3. Import Export Validation
4. Help Validation
5. Install Validation
6. Reports Validation


1.2: Form Level and Inter-form level validation
Data Validation in Forms
Forms represent Entities in the business domain.
Entities have Relationships among themselves which define how the Forms are inter-related.
Each Entity has Attributes which map to the Fields and their specifications.
All applications implement the Business rules. These rules may not be consistent across projects.
Data should comply with the Business rules and their Anomalies that define the enterprise.
Changes made to the data should not result in a loss of data consistency thereby maintaining the Integrity.
IGATE


1.2: Form Level and Inter-form level validation
Data Validation in Forms (Cont.)
Domain Constraints
Real World Rules
Inter Form Dependencies
NULL checks
Functionality Check
IGATE

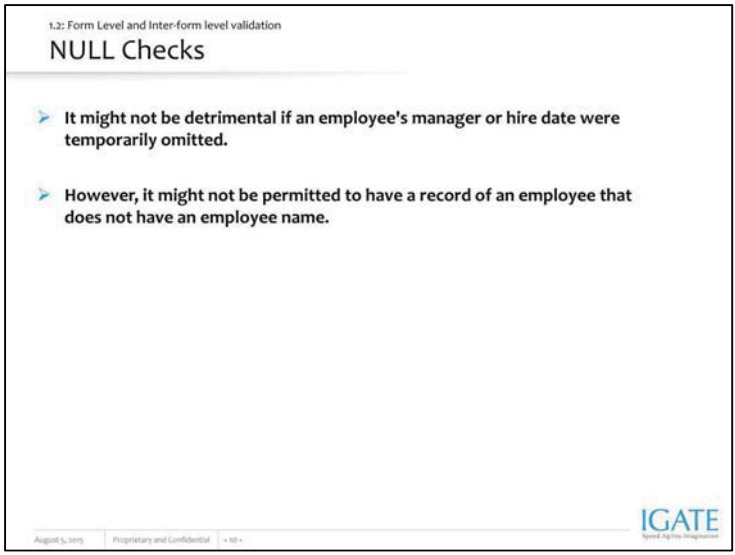

1.2: Form Level and Inter-form level validation
Domain Constraints
Domain Constraints are easy to check for when data is entered.
Domain Types: Attributes may have the same domain e.g. employee's name, manager's name, subordinate's name
At the implementation level they all are character strings
At the conceptual level, we do not expect customers to have the same names as branches, in general.
Strong typing of domains allows us to test for values inserted.
E.g. Salary cannot exceed Rs. 20000.
E.g. For an Airlines Application the from – to cities combination may not exist.
IGATE
August 5, 2015
Proprietary and Confidential
- 7 -

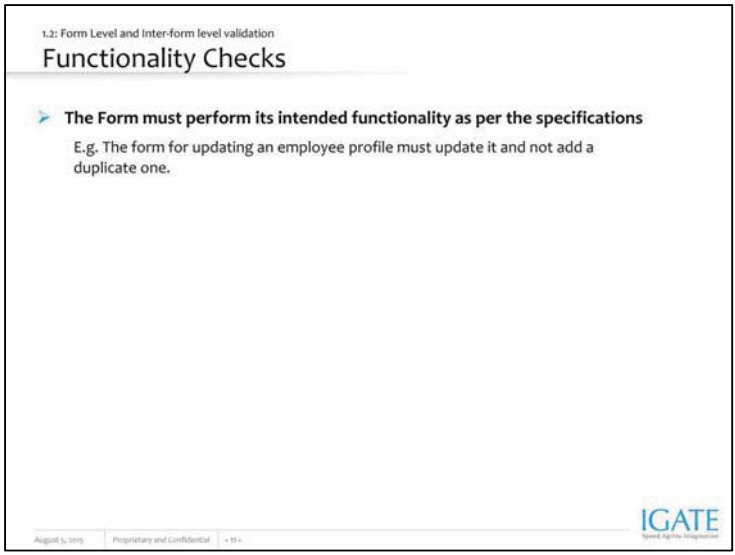

1.2 : Form Level and Inter-form level validation
Real World Rules
Real World Rules would not be domain/project specific.
Examples:
An employee has only one name, is in one department.
An employee's salary should not be greater than that of the manager of the department.
A check on the salary and commission values to prevent the commission value from being greater than the salary value.
IGATE

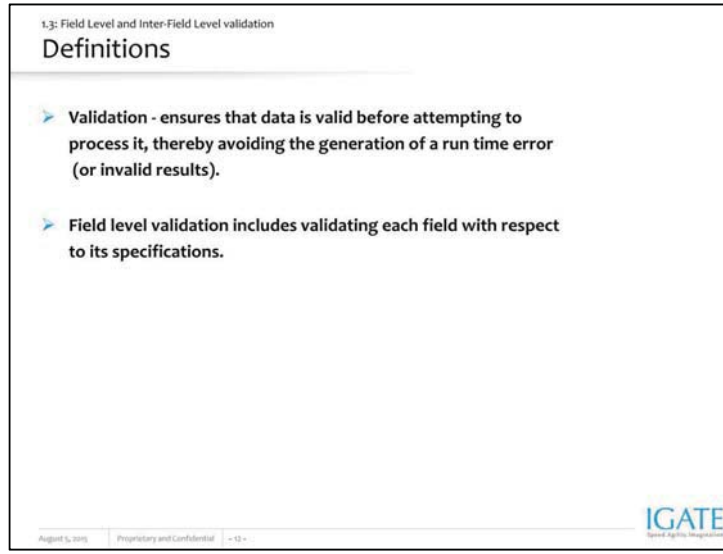


Note : For E.g.1 applicable when giving a ref. Of Emp Id. for say logging a call & not when profiling the employee.

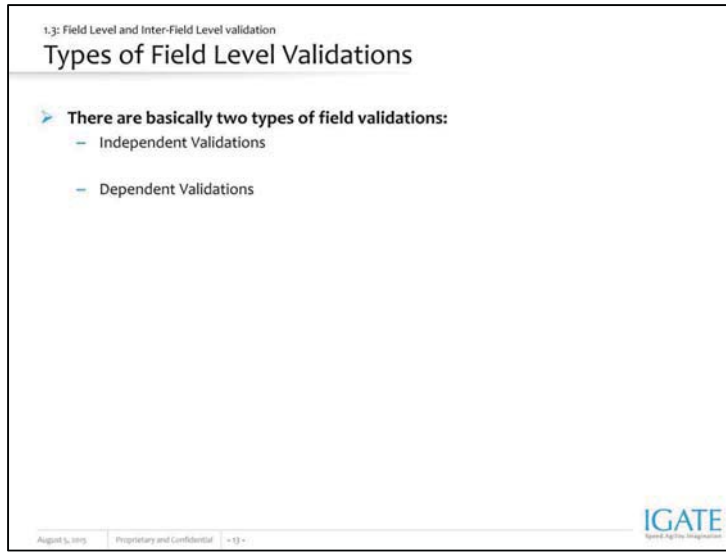
Some More Examples
When an order is entered, the entered customer id must exist in Customers list.
If the customer associated with an order is changed, the new value must exist in Customers list.

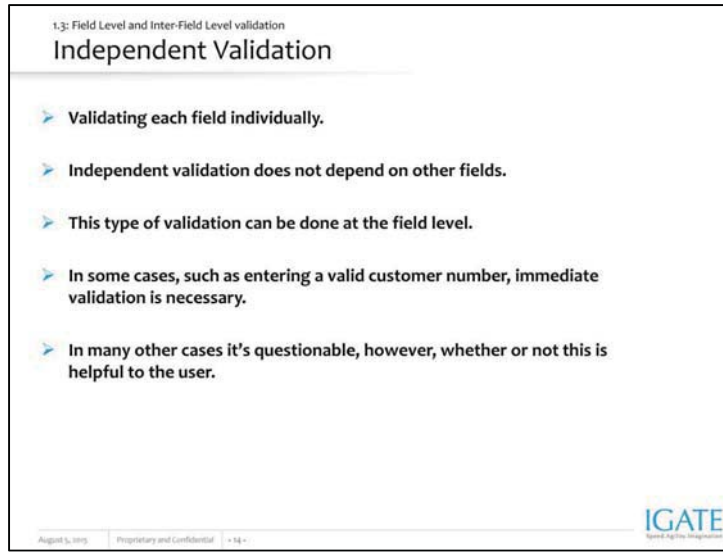

1.2: Form Level and Inter-form level validation
NULL Checks
It might not be detrimental if an employee's manager or hire date were temporarily omitted.
However, it might not be permitted to have a record of an employee that does not have an employee name.
IGATE


1.2: Form Level and Inter-form level validation
Functionality Checks
The Form must perform its intended functionality as per the specifications
E.g. The form for updating an employee profile must update it and not add a duplicate one.
IGATE



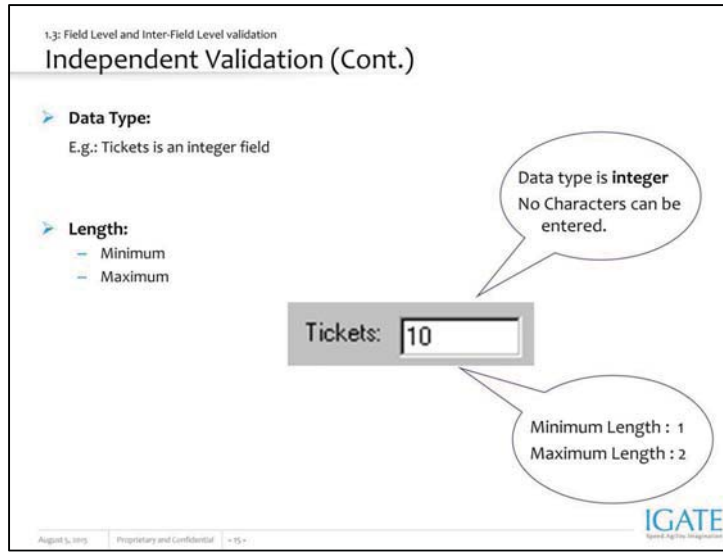
For instance in a date field characters are entered, and the form is submitted then it will not get recorded in the database, because of incompatible types in the data provided and the data expected.
The database operation will return an error, which will be conveyed to the user. The user will then rectify the error and re submit the data.
This approach is time consuming, increases the response time. This can be avoided by ensuring that the data provided is in the same format / range as expected before submitting it. This is accomplished through field level validations. The tester has to ensure that all fields are validated before submitting the data.
If the user is not allowed to work with certain Fields on the Window, then those Fields should not be enabled for that user, thereby, reducing the possibility of invalid information from entering into the Database.
This approach can be divided into two sections, the Setup/Infrastructure of Field Level Validation/Security and the actual Implementation/Usage of it.
With the use of an example – Maintaining Customer Information. Lets say we need to maintain a few Fields in the Customer Table
Customer Number, Name, Sales Rep., Country, State, Credit Limit and Comments.
We have also determined that Customer Number, Sales Rep., Country and Terms are Required Fields (have to be filled in). In addition, we need to validate (information has to conform to allowed values) the Sales Rep., Country, State, and Credit Limit Fields.
Based on this example, we will develop a dynamic Field Level Validation/ Security mechanism for the Customer Maintenance. Not only will this mechanism allow us to validate the information entered on the UI, but furthermore, allow us to validate it based on the User, or the User's Group properties.


1.3: Field Level and Inter-Field Level validation
Types of Field Level Validations
There are basically two types of field validations:
Independent Validations
Dependent Validations
IGATE

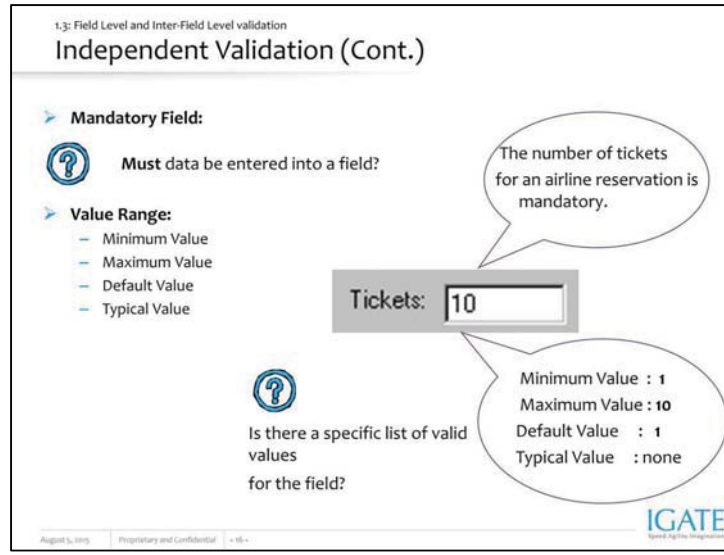


This will involve applying all the testing techniques
Valid equivalence classes
Invalid equivalence classes
Boundary value analysis

Almost every field in your database requires some type of input validation. Checklist
1. What characters are valid or invalid for the field?
2. Do you want only numeric data? Do you want uppercase letters only?
3. For numeric fields, is there a high/low range limit?



Length
The number of characters to be entered in a field.
For example, a required zip code field must have 5 or 9 numeric characters to be considered valid, and the UI may prevent the user from leaving the field while the value is not valid.

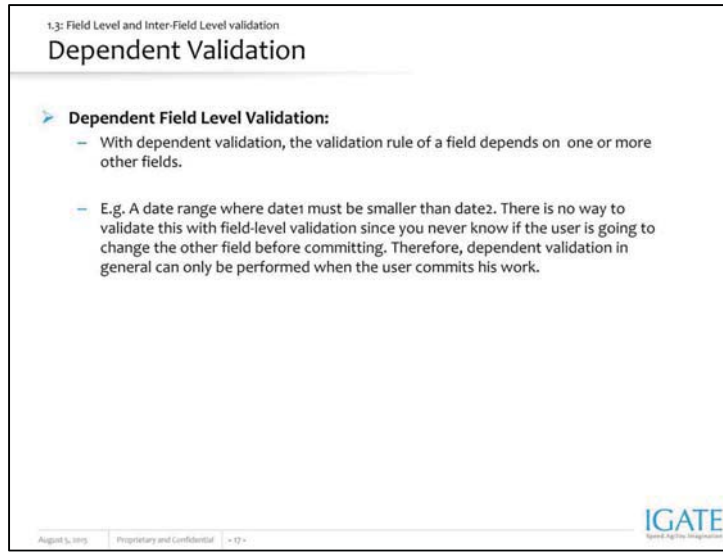


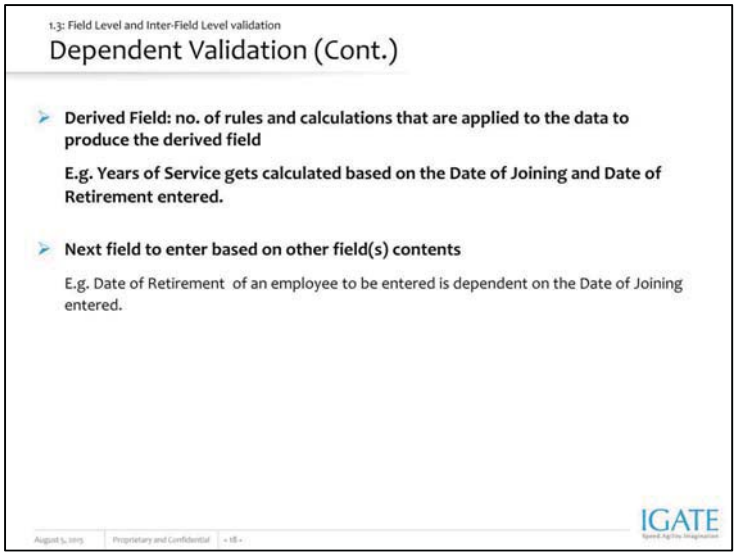
Range
Implicit constraints - 0 < age < 100
Explicit constraints - In case of an employee related form the 18 <= age <= 55 (retirement age)
Default Value – 18

Mandatory
Q. Must data be entered into a field?
Are there any required fields on the form? Almost every input form has at least one required field.

While placing an order, there should be at least 1 item in the order.
Validation stage – on submit (to avoid forcing the user to enter the mandatory fields in a specific order)

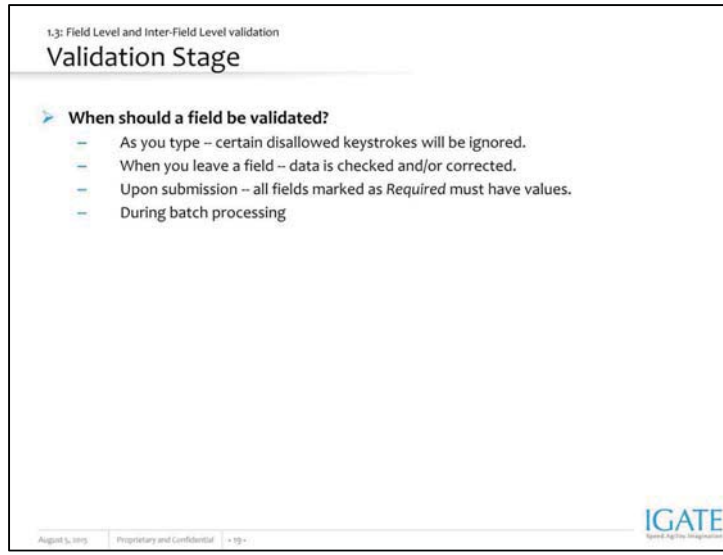
Typical Values
Q. Is there a specific list of valid values for the field?
If you have a customer type field, and you only deal with Retail, Wholesale, and Distributor customers, then providing a list for the user would almost guarantee a valid input.


1.3: Field Level and Inter-Field Level validation
Dependent Validation
Dependent Field Level Validation:
With dependent validation, the validation rule of a field depends on one or more other fields.
E.g. A date range where date1 must be smaller than date2. There is no way to validate this with field-level validation since you never know if the user is going to change the other field before committing. Therefore, dependent validation in general can only be performed when the user commits his work.
IGATE

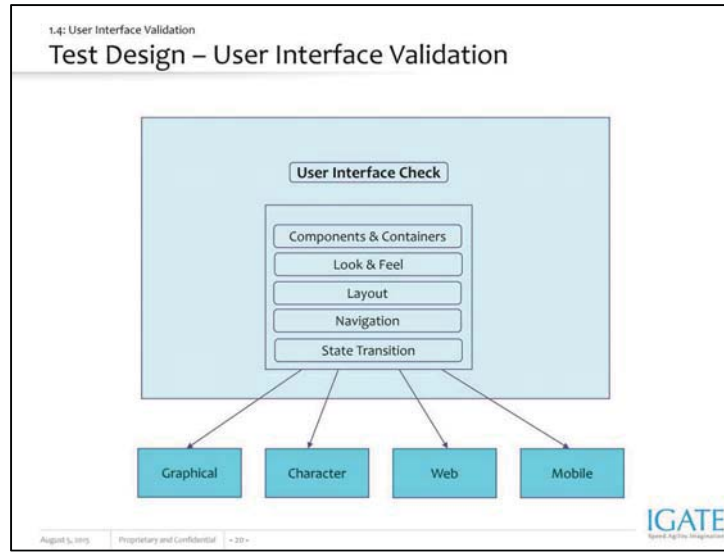


Example
Age gets calculated automatically based on birth date and current date.

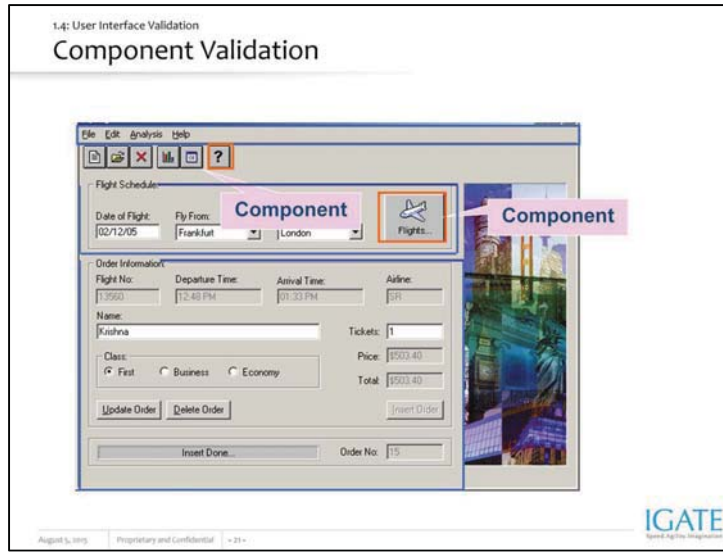
What is this a conditional field?
For instance, if the user indicates on a form that the customer wishes to have his purchase shipped instead of picked up, the shipping address would be a conditional field.



Batch processing – Executing a series of non interactive jobs all at one time.
An example of batch processing is the way that credit card companies process billing. The customer does not receive a bill for each separate credit card purchase but one monthly bill for all of that month's purchases. The bill is created through batch processing, where all of the data are collected and held until the bill is processed as a batch at the end of the billing cycle.



Forms are mainly used to collect user entered data to the system. We enter our login data or use one signup form to enter data as a user to a database table or to a form processing script. There are different components like text box, radio buttons, checkbox, combo box or drop down list boxes for users to enter data to a system.

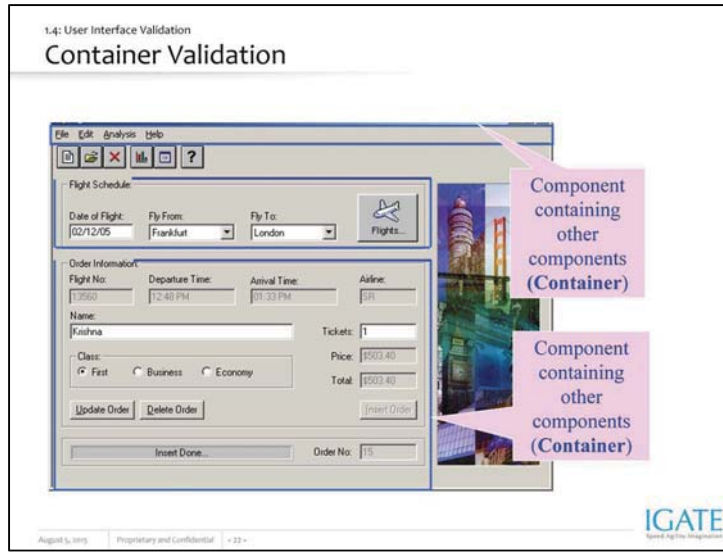


Guidelines to be followed when preparing GUI test cases

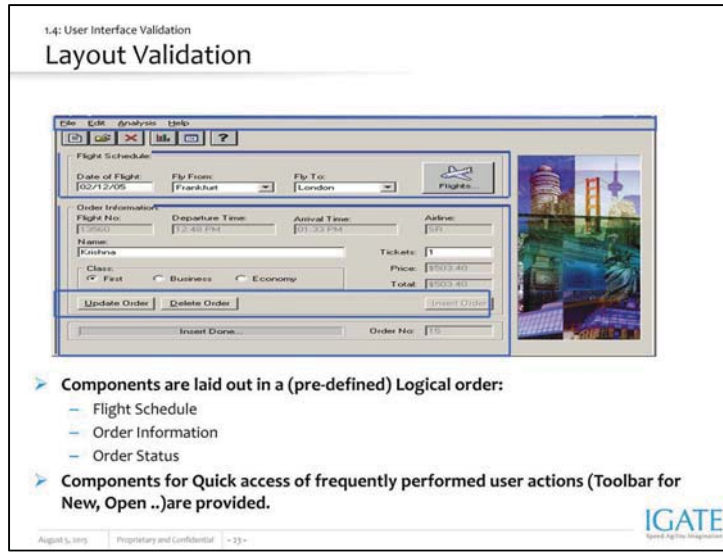
Check for availability of all the objects on the application.

Check for alignment of all the objects, even though customer does not specify them in the requirements.

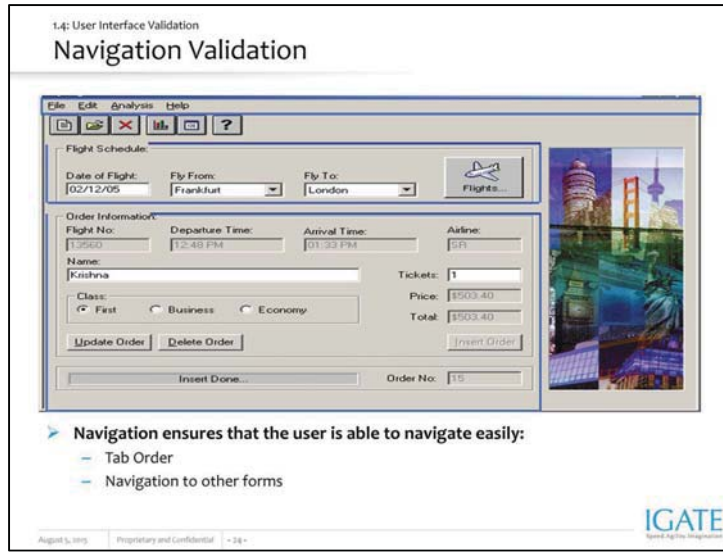
Check for consistency of the objects (Color, appearance, resolution, spelling etc) And any such feature that can be tested just by observing or a defect that can be avoided by just looking and pointing out in the development stage will fall under GUI test cases.



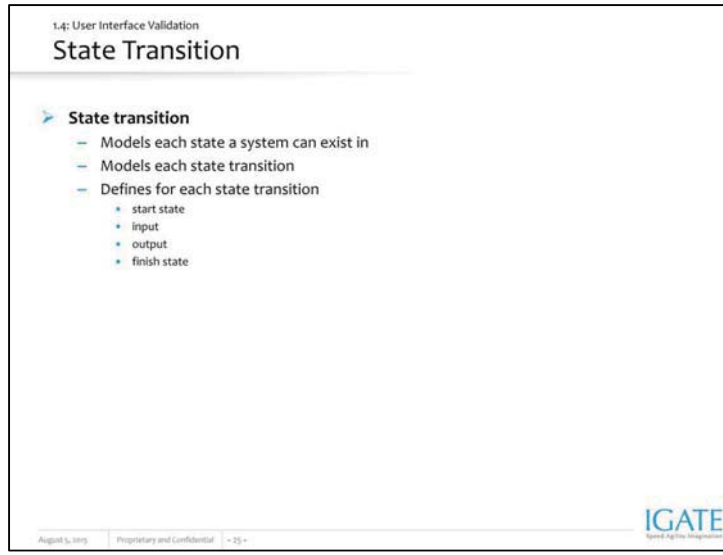
This is validating container controls such as Panel. Components are arranged in the order in which they are defined in the parent container's component sequence.



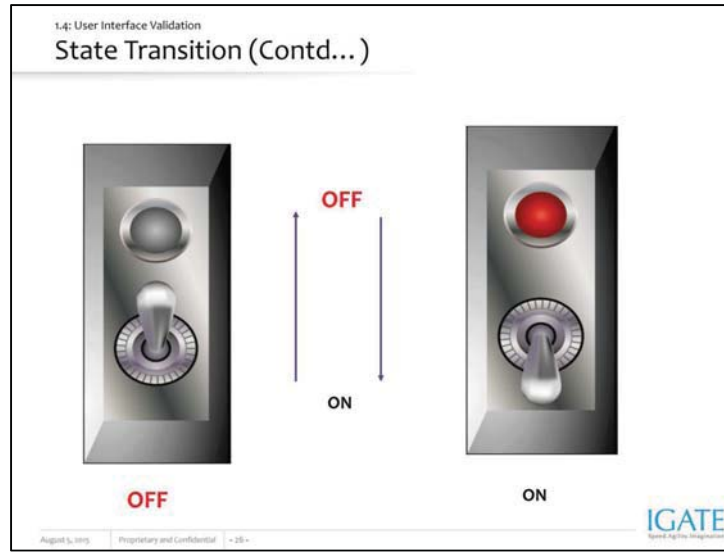
Layout validation is the process of automatically determining how components should be arranged on screen, how to explicitly set component size and position. It is the responsibility of a container's skin. When laying out a form, keep in mind that the order in which you do so is important. Typically, screen readers read from top-to-bottom, left-to-right.



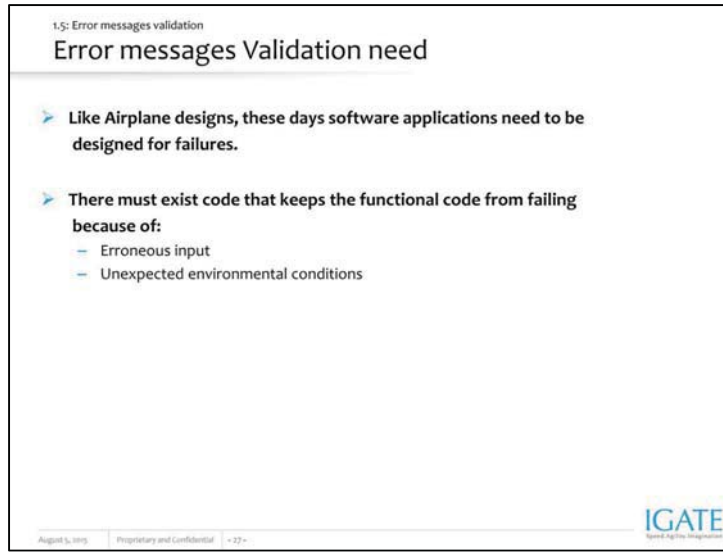
Navigation is used to give the visitor the option of navigating between several different pages, frequently visited pages. The tab order defines the order in which elements will receive focus when navigated by the user via the keyboard. The tabbing order may include elements nested within other elements.

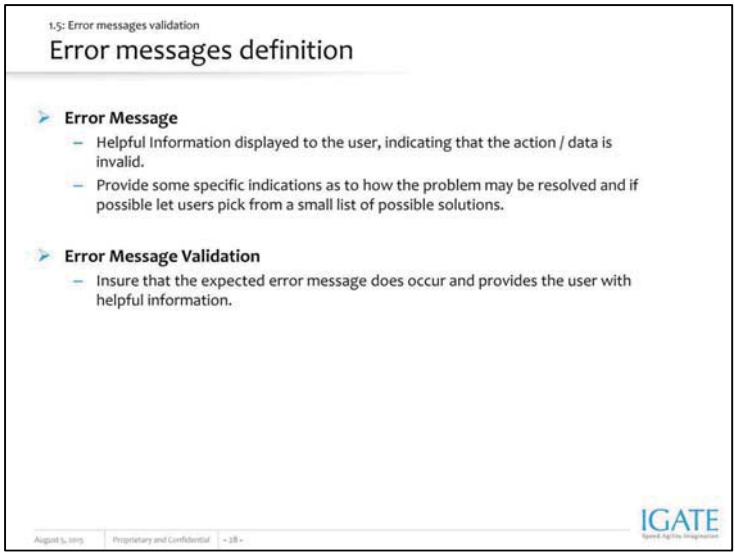


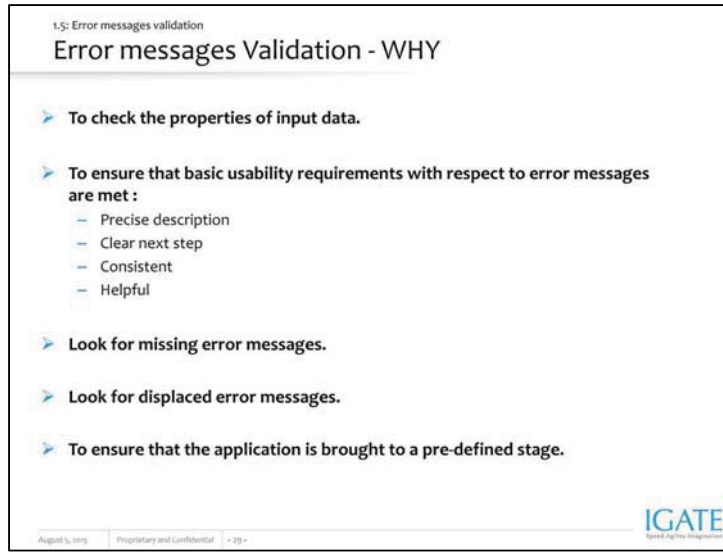
State transition testing focuses on the testing of transitions from one state of an object to another state. Incorporate the state transition testing preparation into the test scenario building process.
For example, include tests of an account transitioning from open to closed and an account transitioning from closed to open (e.g., account closed in error).

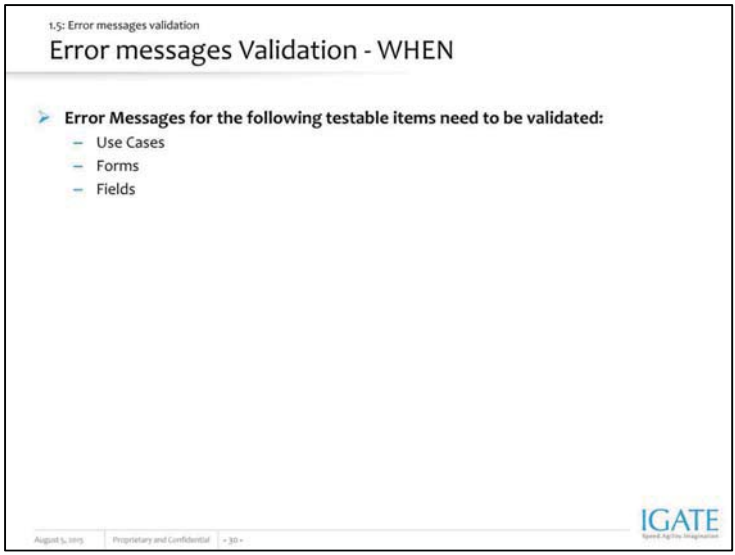


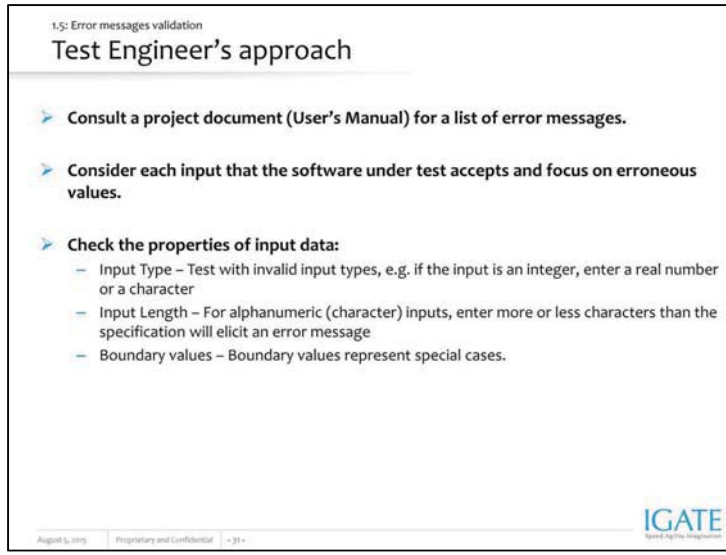
Input: switch on
Output: light on
There can be different states to be considered here.
start state: **on,** input: **switch off,** output: **light off,** finish state: **off**
start state: **off,** input: **switch on,** output: **light on,** finish state: **on**

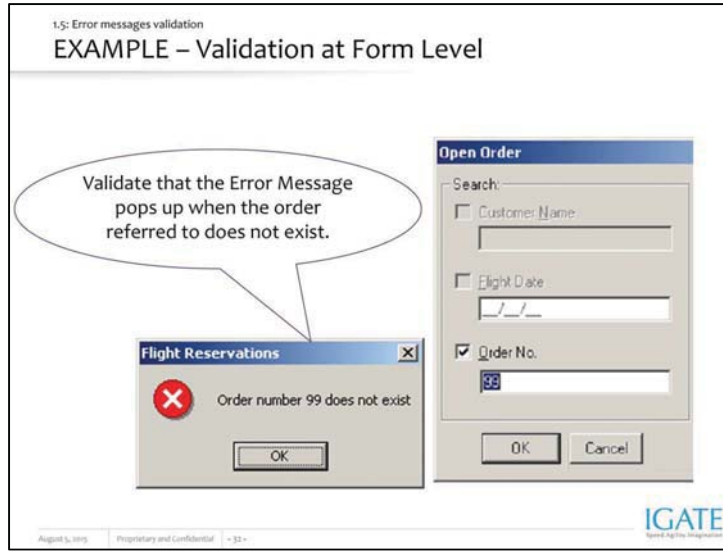

1.5: Error messages validation
Error messages Validation need
Like Airplane designs, these days software applications need to be designed for failures.
There must exist code that keeps the functional code from failing because of:
Erroneous input
Unexpected environmental conditions
IGATE


1.5: Error messages validation
Error messages definition
Error Message
Helpful Information displayed to the user, indicating that the action / data is invalid.
Provide some specific indications as to how the problem may be resolved and if possible let users pick from a small list of possible solutions.
Error Message Validation
Insure that the expected error message does occur and provides the user with helpful information.
IGATE

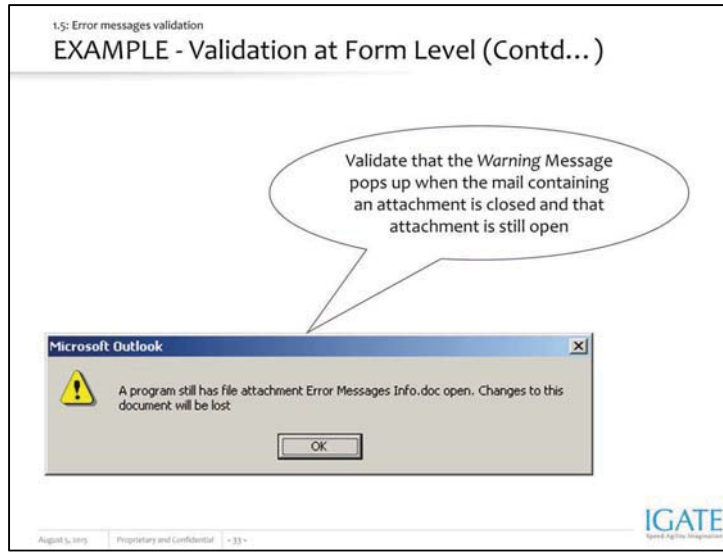

1.5: Error messages validation
Error messages Validation - WHY
To check the properties of input data.
To ensure that basic usability requirements with respect to error messages are met :
Precise description
Clear next step
Consistent
Helpful
Look for missing error messages.
Look for displaced error messages.
To ensure that the application is brought to a pre-defined stage.
IGATE


1.5: Error messages validation
Error messages Validation - WHEN
Error Messages for the following testable items need to be validated:
Use Cases
Forms
Fields
IGATE

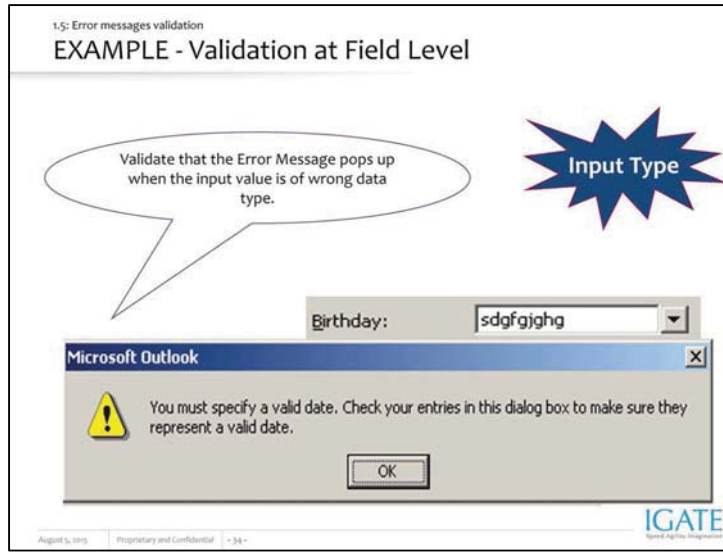

1.5: Error messages validation
Test Engineer's approach
Consult a project document (User's Manual) for a list of error messages.
Consider each input that the software under test accepts and focus on erroneous values.
Check the properties of input data:
Input Type – Test with invalid input types, e.g. if the input is an integer, enter a real number or a character
Input Length – For alphanumeric (character) inputs, enter more or less characters than the specification will elicit an error message
Boundary values – Boundary values represent special cases.
IGATE
August 5, 2015
Proprietary and Confidential
- 31 -



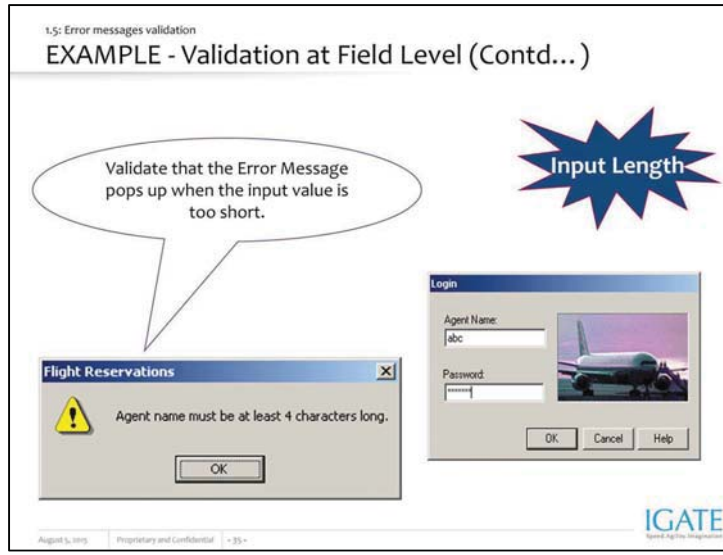
Note: An attempt is made to open an order that does not exist.



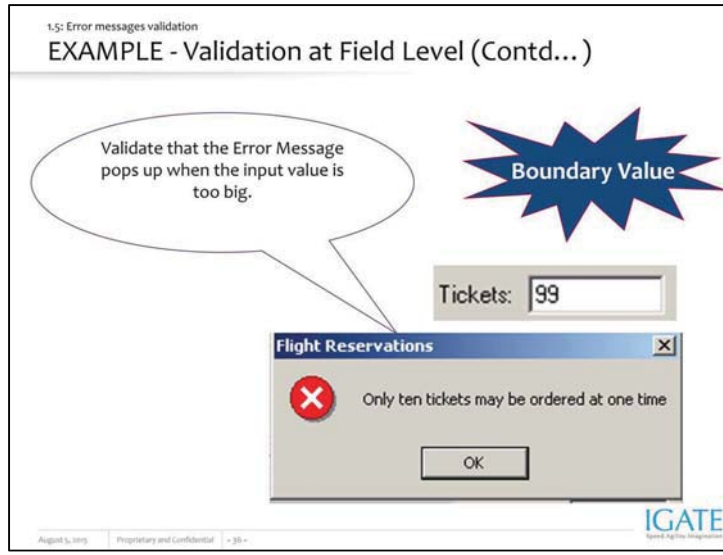
Note: Here a warning is displayed.

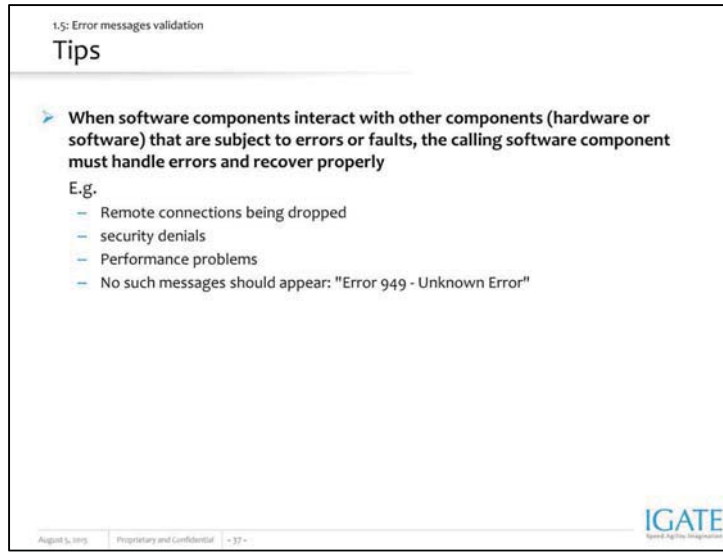


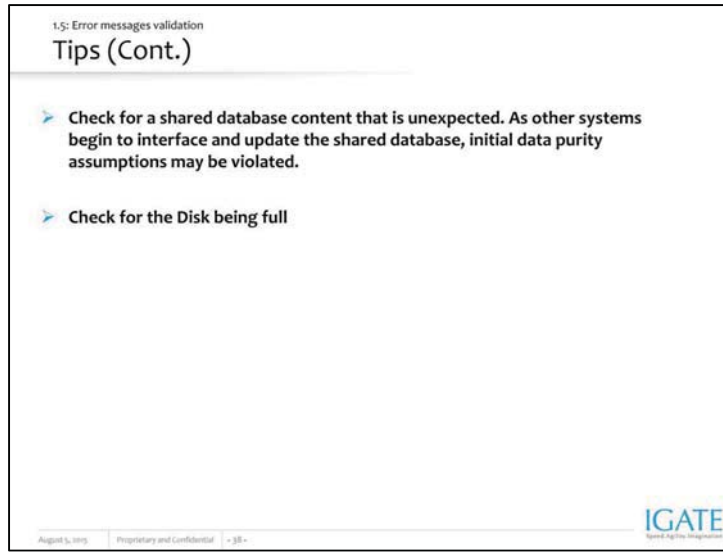
Note: Input type expected is date.



Note: Here the number of characters entered are less than the specification of the field i.e. Agent Name


1.5: Error messages validation
EXAMPLE - Validation at Field Level (Contd...)
Validate that the Error Message pops up when the input value is too big.
Boundary Value
Tickets: 99
Flight Reservations
Only ten tickets may be ordered at one time
OK
IGATE


1.5: Error messages validation
Tips
When software components interact with other components (hardware or software) that are subject to errors or faults, the calling software component must handle errors and recover properly
E.g.
Remote connections being dropped
security denials
Performance problems
No such messages should appear: "Error 949 - Unknown Error"
IGATE
August 5, 2015
Proprietary and Confidential
- 37 -


1.5: Error messages validation
Tips (Cont.)
Check for a shared database content that is unexpected. As other systems begin to interface and update the shared database, initial data purity assumptions may be violated.
Check for the Disk being full
IGATE

Summary

- **In this lesson, you have learnt:**
  - There are different testing phases in testing life cycle like
    - Test Planning
    - Test Design
    - Test Development
    - Test Automation
    - Test Execution
  - UI testing covers Component, Container, Layout, Navigation & State transition.

Summary

- **Following validations are considered for form level validation**
  - Domain Constraints
  - Real World Rules
  - Inter Form Dependencies
  - NULL checks
  - Functionality Check

IGATE

Summary

- **There are basically two types of field validations:**
  - Independent Validations
  - Dependent Validations
- **Error Message Validation ensures that the expected error message does occur and provides the user with helpful information.**

IGATE

Review Question

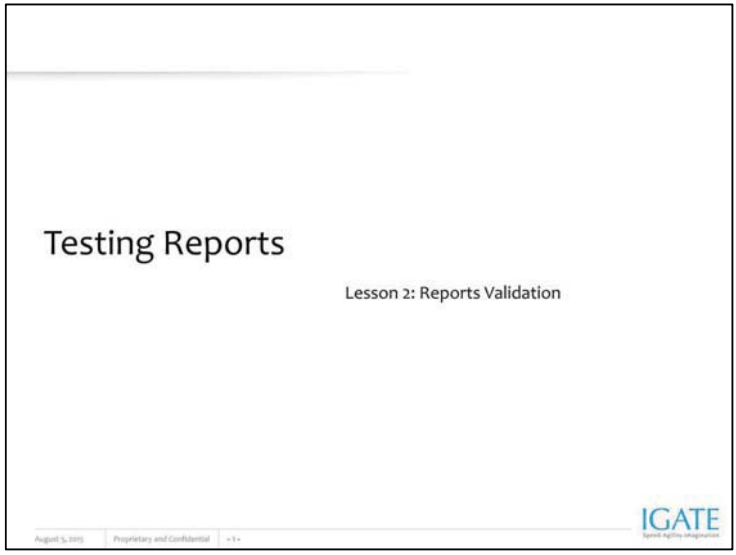
- **Question 1: _____ would be domain/project specific.**
- **Question 2: An employee has only one name, is in one department is a Domain constraint.**
  - True / False
- **Question 3: A postcondition for a use case should be true regardless of which flows were executed**
  - True / False

IGATE

Review Question: Match the Following

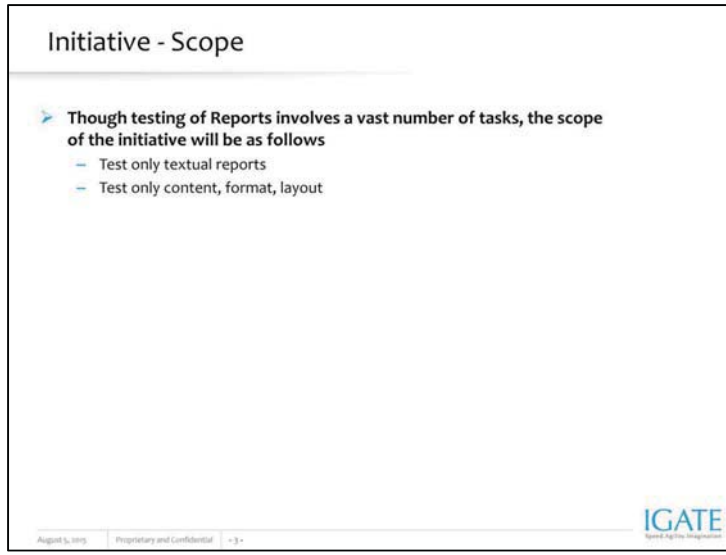
1. Test automation	A. Test Traceability Matrix and Test coverage
2. Test development	B. Expectedly go wrong
	C. Record the results of the test accurately
3. Test planning	D. Most important stage of the testing life cycle
4. Test design	E. Strategic documentation of tasks

IGATE


Testing Reports
Lesson 2: Reports Validation
IGATE

Lesson Objectives

- **To understand the following topics:**
  - What is Report
  - Approach To testing of Reports
  - How & When to test reports?
  - Purpose of Reports
  - Types of Reports
  - Testable elements (Content, Format, Look & Feel)
  - Cumulative of reports
  - Archival of reports
  - Security of Reports
  - Performance of Reports
  - Localization Issues


Initiative - Scope
Though testing of Reports involves a vast number of tasks, the scope of the initiative will be as follows
Test only textual reports
Test only content, format, layout
IGATE

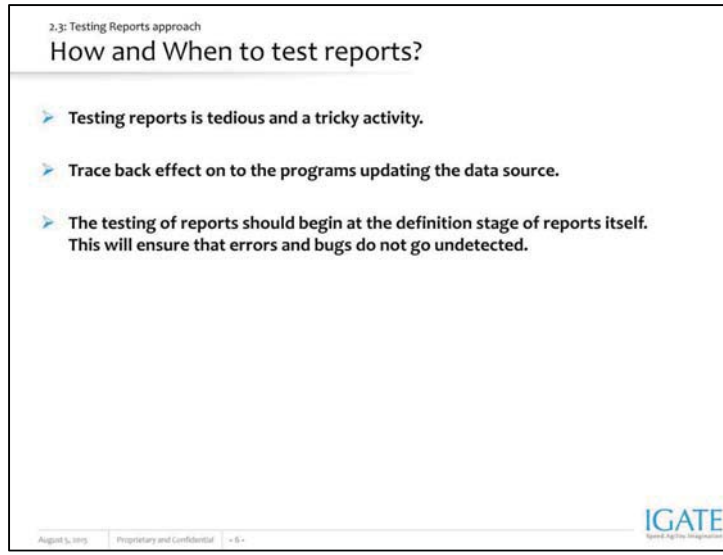
In this Presentation/Session/course, We would touch upon why automation is done, What are its benefits from QTP's point of view, QTP testing process, Introduction to the QTP interface in addition to what is mentioned.
As per software terminology,
a Report is a formatted and organized presentation of data by accessing the data source of the application.

2.2: Testing Reports approach

Approach for testing of Reports

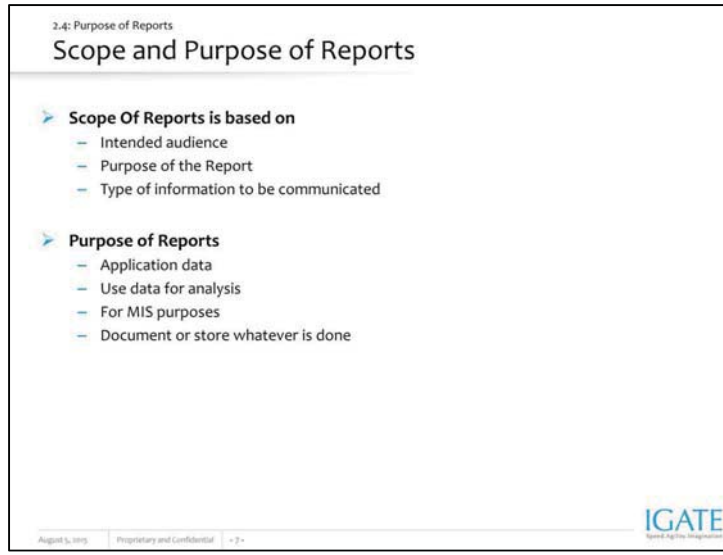
- **How & When to test Reports?**
  - Understand application
  - Study test design documents of the application
  - Study test Specifications for the same
  - Data Availability
  - Creation test cases
  - Automation of test cases(if applicable)

IGATE



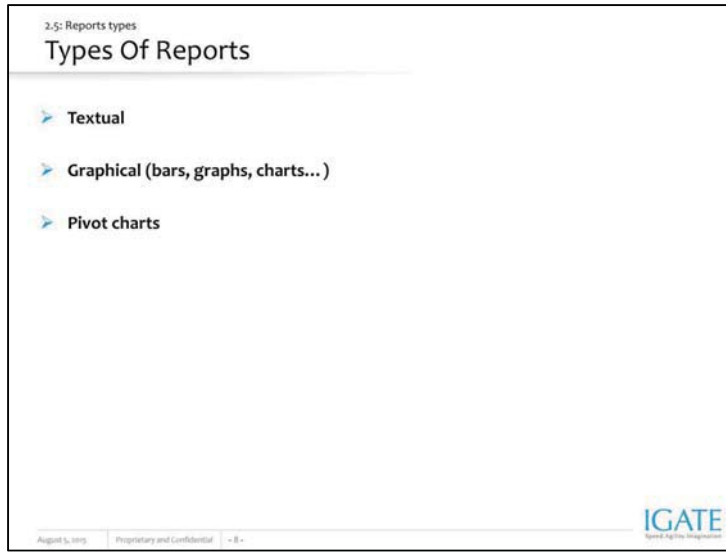
Reports are generated after some business processing is done.

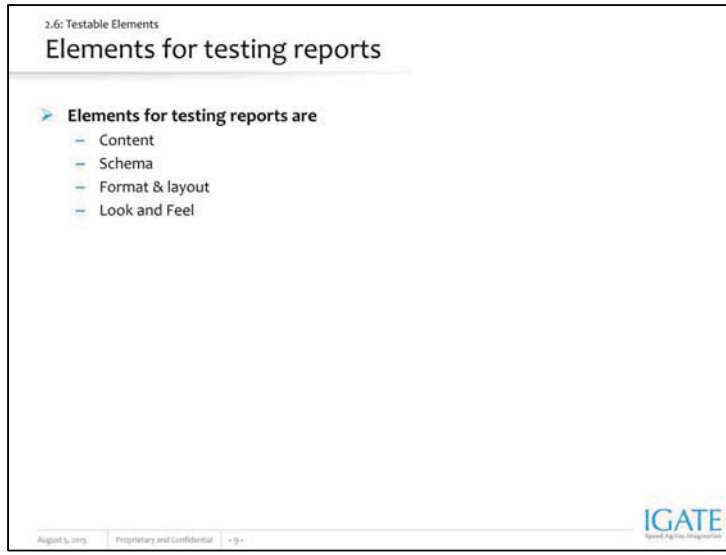
Different users of reports
- Business Analyst
- End users



Objective of testing report is

- To identify the testable elements in a report
- To identify a systematic approach for testing the testable elements
- To automate the test cases at possible scenarios

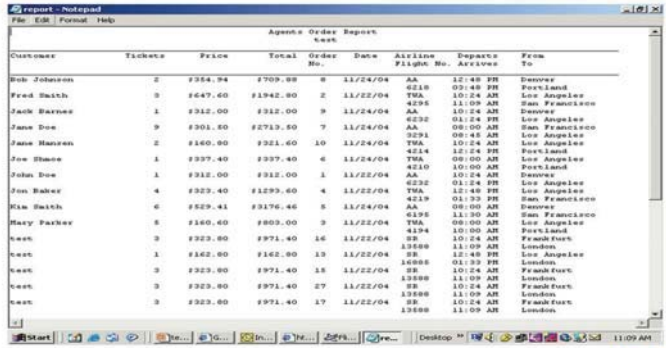

2.5: Reports types
Types Of Reports
Textual
Graphical (bars, graphs, charts...)
Pivot charts
IGATE


2.6: Testable Elements
Elements for testing reports
Elements for testing reports are
Content
Schema
Format & layout
Look and Feel
IGATE

2.6: Testable Elements

Testing Contents

- **Content is the data that will be printed in the report when a report is invoked.**


report - Notepad

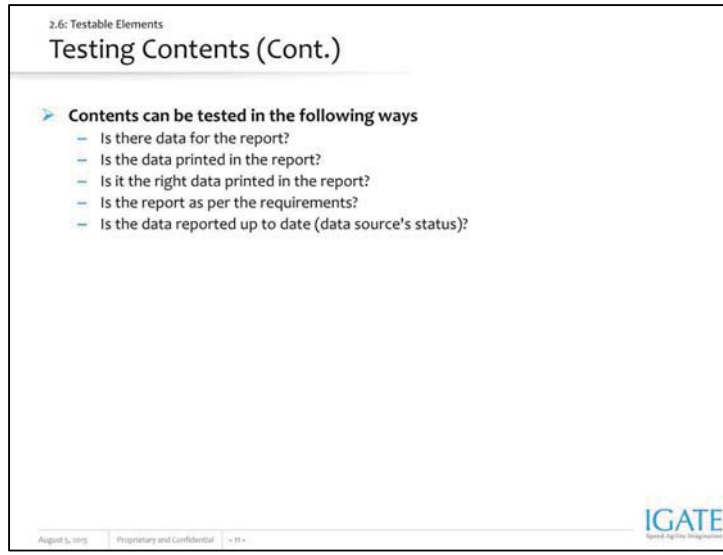
File Edit Format Help

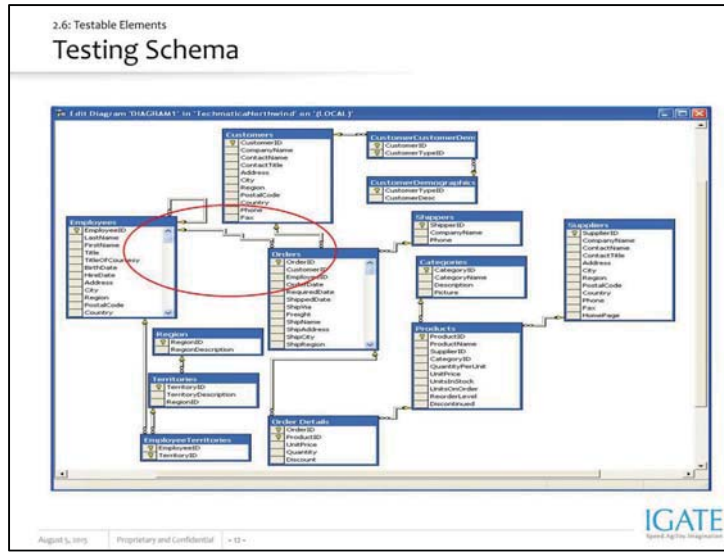
Agents Order Report
test

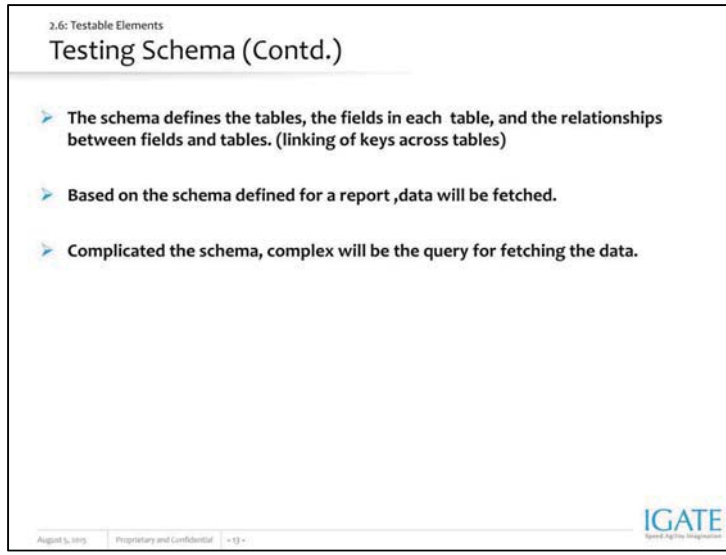
Customer	Tickets	Price	Total	Order No.	Date	Airline Flight No.	Departs Arrives	From To
Bob Johnson	2	$354.94	$709.88	8	11/24/04	AA 6210	12:48 PM 03:48 PM	Denver Portland
Fred Smith	3	$647.60	$1942.80	2	11/22/04	TWA 4295	10:24 AM 11:09 AM	Los Angeles San Francisco
Jack Barnes	1	$312.00	$312.00	9	11/24/04	AA 6232	10:24 AM 01:24 PM	Denver Los Angeles
Jane Doe	9	$301.50	$2713.50	7	11/24/04	AA 3291	08:00 AM 09:45 AM	San Francisco Los Angeles
Jane Hansen	2	$160.80	$321.60	10	11/24/04	TWA 4214	10:24 AM 12:24 PM	Los Angeles Portland
Joe Shmoe	1	$337.40	$337.40	6	11/24/04	TWA 4210	08:00 AM 10:00 AM	Los Angeles Portland
John Doe	1	$312.00	$312.00	1	11/22/04	AA 6232	10:24 AM 01:24 PM	Denver Los Angeles
Jon Baker	4	$323.40	$1293.60	4	11/22/04	TWA 4219	12:48 PM 01:33 PM	Los Angeles San Francisco
Kim Smith	6	$529.41	$3176.46	5	11/24/04	AA 6195	08:00 AM 11:30 AM	Denver San Francisco
Mary Parker	5	$160.60	$803.00	3	11/22/04	TWA 4194	08:00 AM 10:00 AM	Los Angeles Portland
test	3	$323.80	$971.40	16	11/22/04	SR 13588	10:24 AM 11:09 AM	Frankfurt London
test	1	$162.80	$162.80	13	11/22/04	SR 16065	12:48 PM 01:33 PM	Los Angeles London
test	3	$323.80	$971.40	15	11/22/04	SR 13588	10:24 AM 11:09 AM	Frankfurt London
test	3	$323.80	$971.40	27	11/22/04	SR 13588	10:24 AM 11:09 AM	Frankfurt London
test	3	$323.80	$971.40	17	11/22/04	SR 13588	10:24 AM 11:09 AM	Frankfurt London

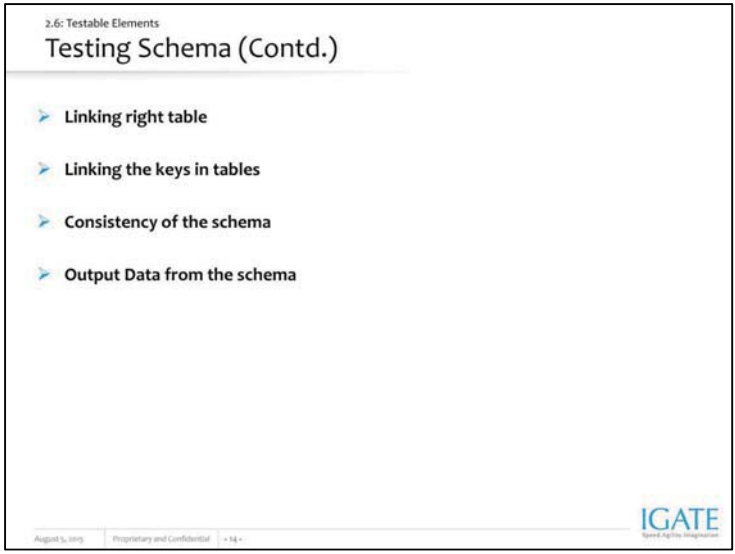
Start Desktop 11:09 AM

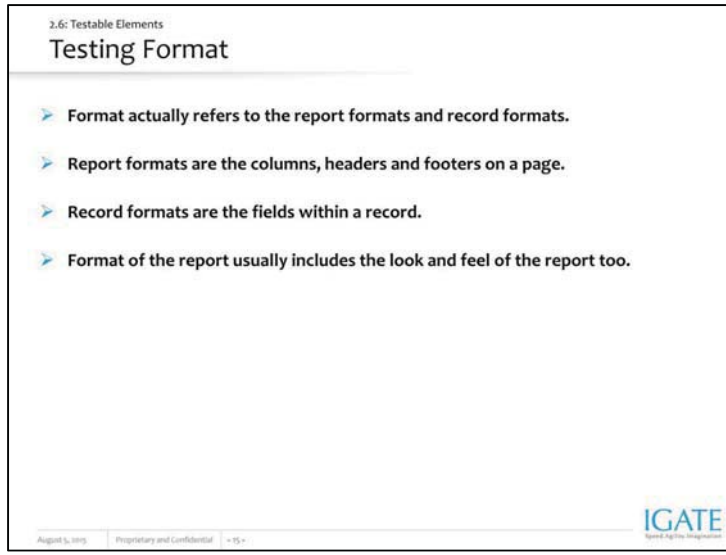
IGATE


2.6: Testable Elements
Testing Contents (Cont.)
Contents can be tested in the following ways
Is there data for the report?
Is the data printed in the report?
Is it the right data printed in the report?
Is the report as per the requirements?
Is the data reported up to date (data source's status)?
IGATE


2.6: Testable Elements
Testing Schema
IGATE


2.6: Testable Elements
Testing Schema (Contd.)
The schema defines the tables, the fields in each table, and the relationships between fields and tables. (linking of keys across tables)
Based on the schema defined for a report ,data will be fetched.
Complicated the schema, complex will be the query for fetching the data.
IGATE


2.6: Testable Elements
Testing Schema (Contd.)
Linking right table
Linking the keys in tables
Consistency of the schema
Output Data from the schema
IGATE

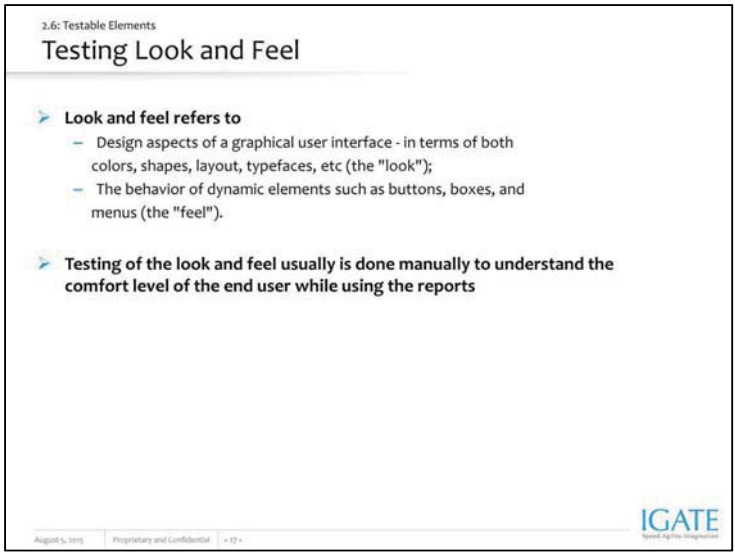

2.6: Testable Elements
Testing Format
Format actually refers to the report formats and record formats.
Report formats are the columns, headers and footers on a page.
Record formats are the fields within a record.
Format of the report usually includes the look and feel of the report too.
IGATE

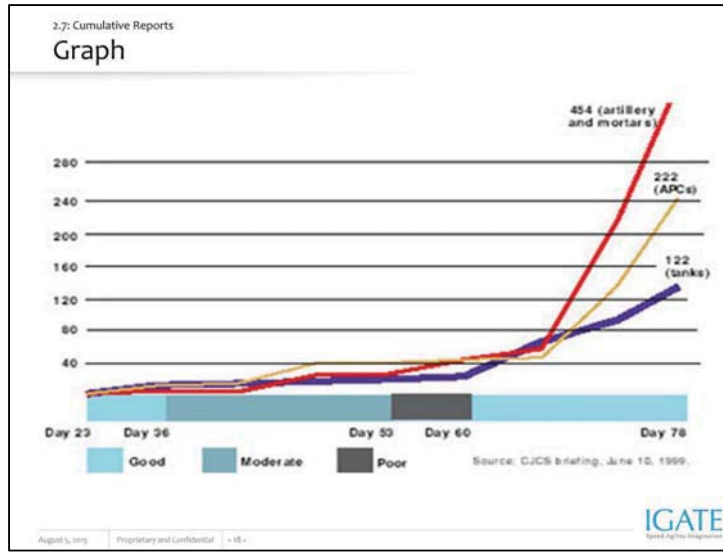
2.6: Testable Elements

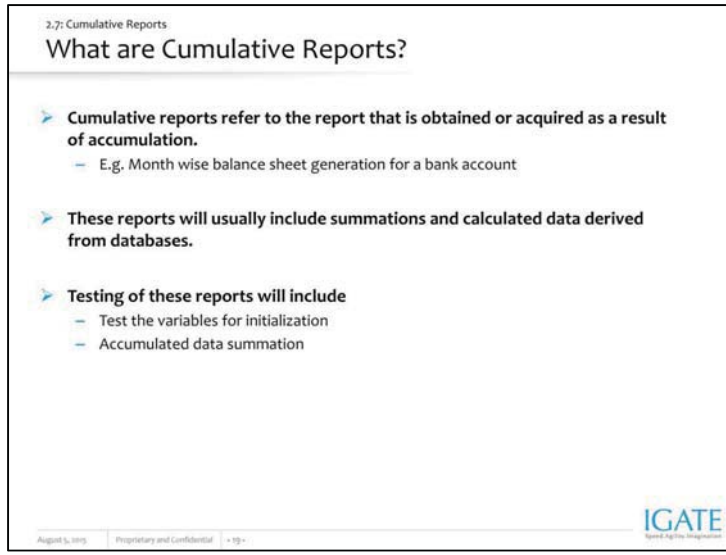
Testing Format (Contd.)

- **Format of a report can be tested**
  - Is the format as per requirements?
  - Is the data printed as per the format specified?
  - Is the format conveying the required details?

IGATE


2.6: Testable Elements
Testing Look and Feel
Look and feel refers to
Design aspects of a graphical user interface - in terms of both colors, shapes, layout, typefaces, etc (the "look");
The behavior of dynamic elements such as buttons, boxes, and menus (the "feel").
Testing of the look and feel usually is done manually to understand the comfort level of the end user while using the reports
IGATE
August 5, 2015
Proprietary and Confidential
- 17 -


2.7: Cumulative Reports
Graph
454 (artillery and mortars)
222 (APCs)
122 (tanks)
280
240
200
160
120
80
40
Day 23
Day 36
Day 50
Day 60
Day 78
Good
Moderate
Poor
Source: CJCS briefing, June 10, 1999.
IGATE


2.7: Cumulative Reports
What are Cumulative Reports?
Cumulative reports refer to the report that is obtained or acquired as a result of accumulation.
E.g. Month wise balance sheet generation for a bank account
These reports will usually include summations and calculated data derived from databases.
Testing of these reports will include
Test the variables for initialization
Accumulated data summation
IGATE
August 5, 2015
Proprietary and Confidential
- 19 -

2.8: Archival Reports

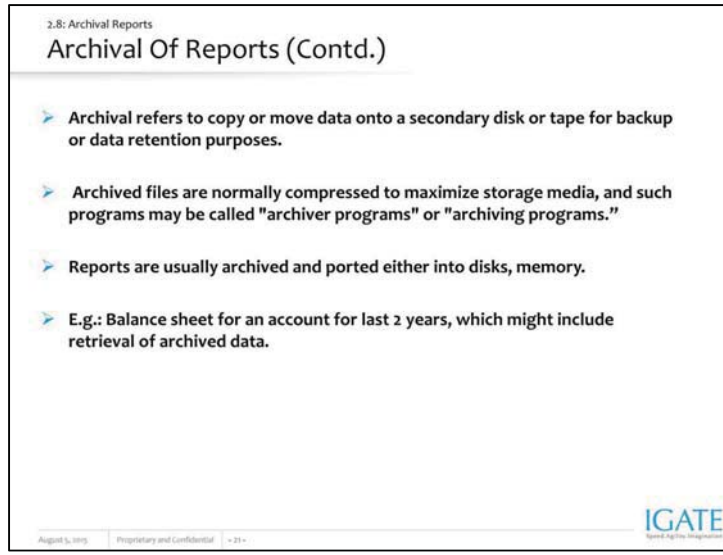
Archival Of Reports

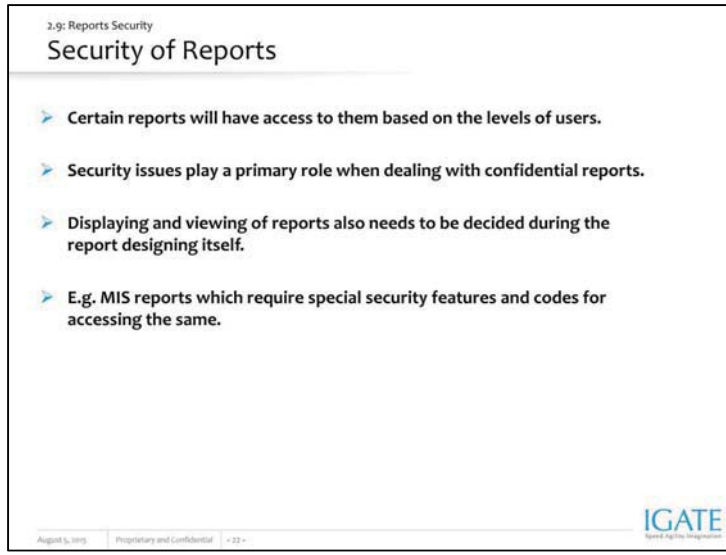
PeopleSoft Receivables
PAYMENT DETAIL
For 01-JUN-1995 through 31-DEC-1995

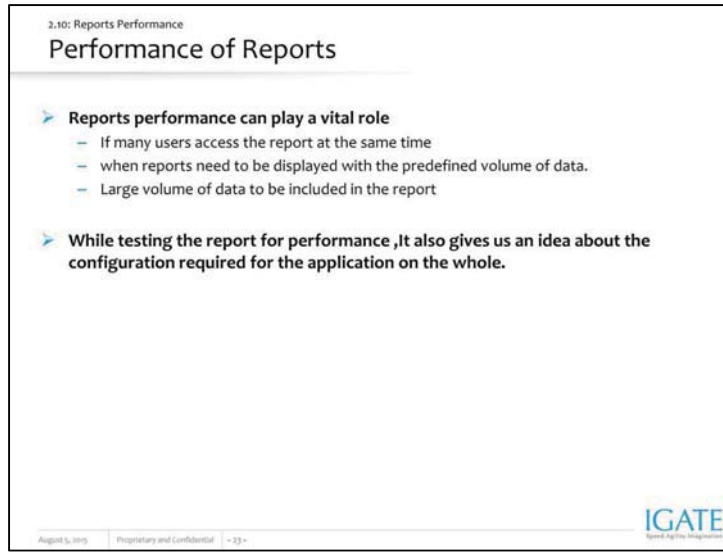
Report ID: AR20002
Deposit BU: M02--Canada Operations
Report Currency: Base Amount
Deposit ID: ALL VALUES
OPRID: ALL VALUES
Post Status: All Status

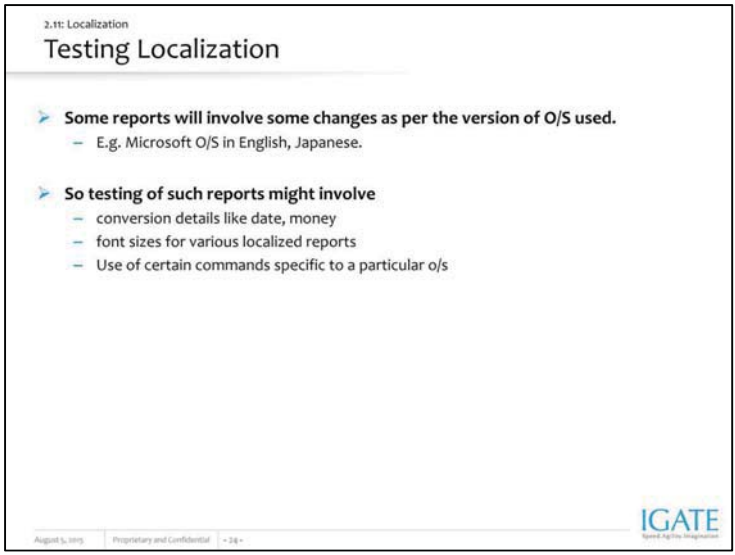
Entry Date	Deposit ID	Seq	Acctg Dt	Payment ID	Payment Amount	Status	Customer ID/Name		Item ID
11/18/1995	1	1	11/18/1995	5002391	77,344,232.00 CAD	C	50022	NewTerr	CB-010002
							50022	NewTerr	CB-010003
11/18/1995	1	2	11/18/1995	FY-1510	34,321,952.00 CAD	C	50021	P&P	CB-010001
							50021	P&P	CB-010001
11/18/1995	1	3	11/18/1995	0082500	51,211,622.00 CAD	C	50027	PC Dist	CB-010004
TOTALS FOR 1					162,877,806.00 CAD				
12/20/1995	2	1	12/20/1995	0022974	18,220,300.00 CAD	C	50007	Chelsea	CB-010008
12/20/1995	2	2	12/20/1995	22051	35,111,649.00 CAD	C	50021	P&P	CB-010007
							50021	P&P	CB-010007
12/20/1995	2	3	12/20/1995	V298	88,653,155.00 CAD	C	50027	PC Dist	CB-010005
							50027	PC Dist	CB-010005
TOTALS FOR 2					141,985,104.00 CAD				
12/21/1995	3	1	12/21/1995	24000451	12,545,808.91 CAD	C	50002	Meridien	CB-010009
TOTALS FOR 3					12,545,808.91 CAD				
TOTALS FOR M02					317,408,718.91 CAD				

IGATE


2.8: Archival Reports
Archival Of Reports (Contd.)
Archival refers to copy or move data onto a secondary disk or tape for backup or data retention purposes.
Archived files are normally compressed to maximize storage media, and such programs may be called "archiver programs" or "archiving programs."
Reports are usually archived and ported either into disks, memory.
E.g.: Balance sheet for an account for last 2 years, which might include retrieval of archived data.
IGATE


2.9: Reports Security
Security of Reports
Certain reports will have access to them based on the levels of users.
Security issues play a primary role when dealing with confidential reports.
Displaying and viewing of reports also needs to be decided during the report designing itself.
E.g. MIS reports which require special security features and codes for accessing the same.
IGATE


2.10: Reports Performance
Performance of Reports
Reports performance can play a vital role
If many users access the report at the same time
when reports need to be displayed with the predefined volume of data.
Large volume of data to be included in the report
While testing the report for performance ,It also gives us an idea about the configuration required for the application on the whole.
IGATE


2.11: Localization
Testing Localization
Some reports will involve some changes as per the version of O/S used.
E.g. Microsoft O/S in English, Japanese.
So testing of such reports might involve
conversion details like date, money
font sizes for various localized reports
Use of certain commands specific to a particular o/s
IGATE

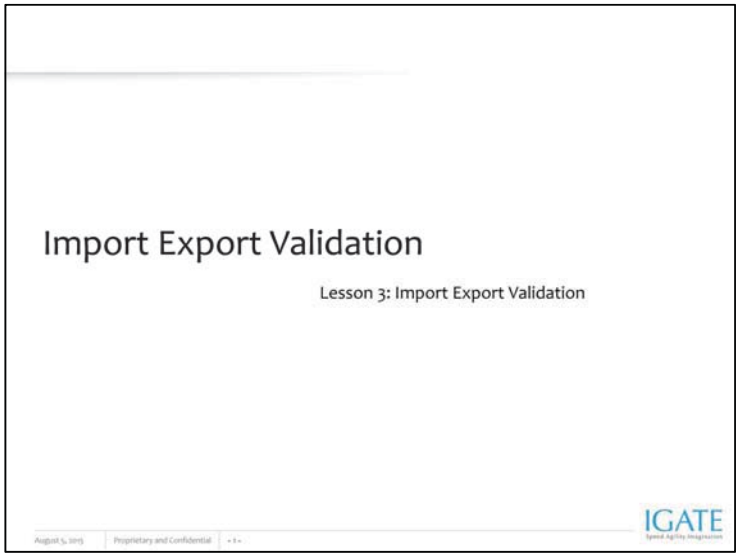
Summary

- **In this lesson, you have learnt:**
  - Use case Testing of reports is thus one of the most important aspects to be considered while testing an application since the output status of an application is verified with the help of the reports.
  - It should also be noted that report needs to have a user-friendly format (look and feel) so that the contents can be easily grasped.
  - By testing any report using the specified approach the accuracy and coverage of testing will be ensured.

Review Question

- **Question 1: What are all the types of reports available?**
- **Question 2: _____ refers to copy or move data onto a secondary disk or tape for backup or data retention purposes**
- **Question 3: Format actually refers to the _____ formats and _____ formats.**

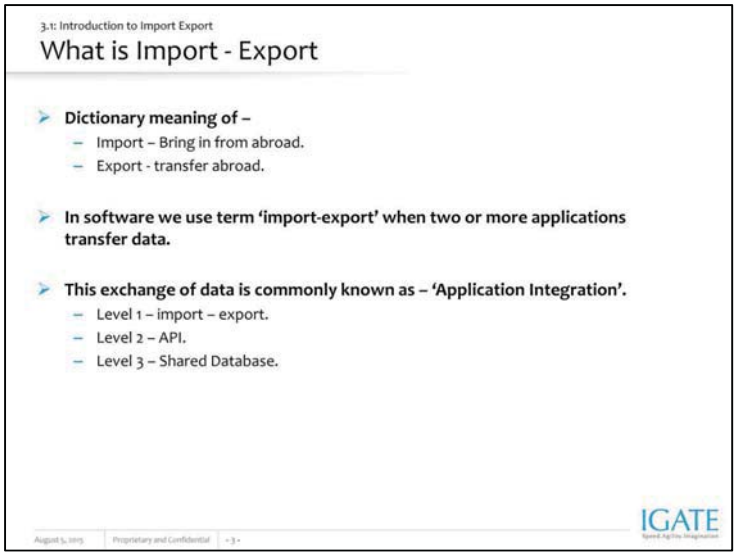
IGATE

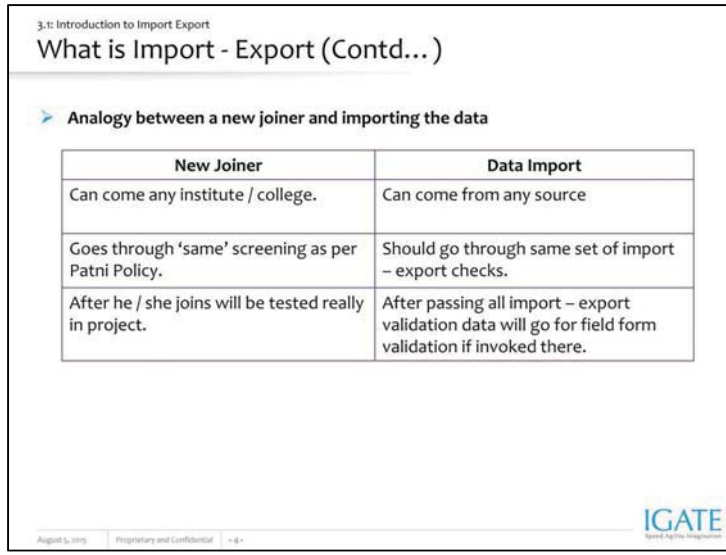

Import Export Validation
Lesson 3: Import Export Validation
IGATE

Lesson Objectives

- **To understand the following topics:**
  - Introduction to import – export
  - Types of import – export
  - Importance of import – export
  - Importance of testing import – export
  - Test Design: import – export

IGATE


3.1: Introduction to Import Export
What is Import - Export
Dictionary meaning of –
Import – Bring in from abroad.
Export - transfer abroad.
In software we use term 'import-export' when two or more applications transfer data.
This exchange of data is commonly known as – 'Application Integration'.
Level 1 – import – export.
Level 2 – API.
Level 3 – Shared Database.
IGATE

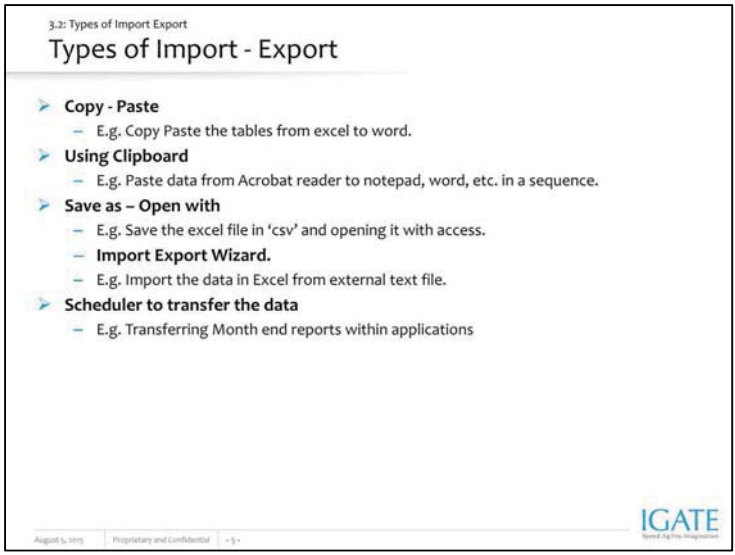


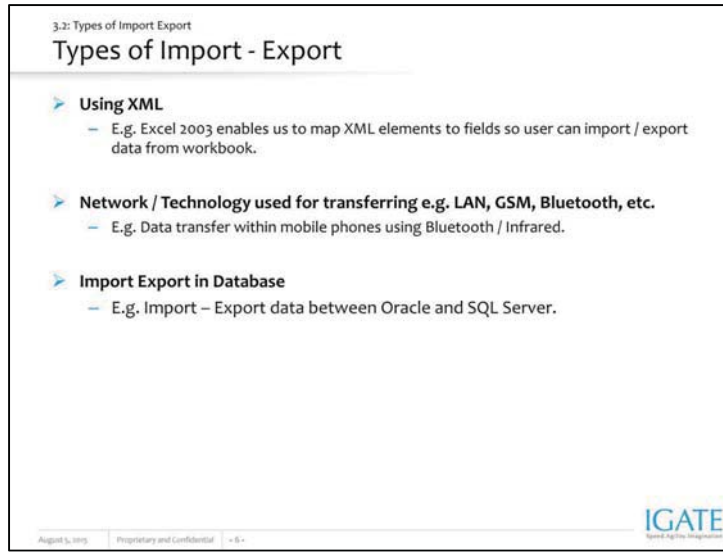
3.1: Introduction to Import Export

What is Import - Export (Contd…)

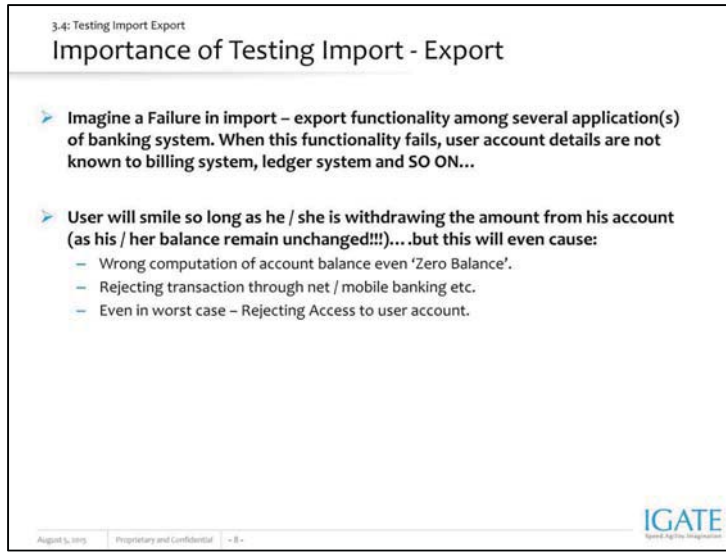
- **Analogy between a new joiner and importing the data**

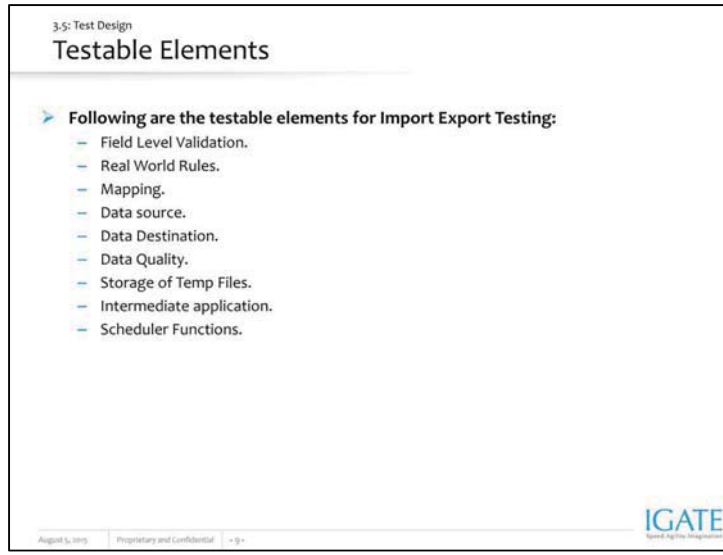
New Joiner	Data Import
Can come any institute / college.	Can come from any source
Goes through 'same' screening as per Patni Policy.	Should go through same set of import – export checks.
After he / she joins will be tested really in project.	After passing all import – export validation data will go for field form validation if invoked there.


3.2: Types of Import Export
Types of Import - Export
Copy - Paste
E.g. Copy Paste the tables from excel to word.
Using Clipboard
E.g. Paste data from Acrobat reader to notepad, word, etc. in a sequence.
Save as – Open with
E.g. Save the excel file in 'csv' and opening it with access.
Import Export Wizard.
E.g. Import the data in Excel from external text file.
Scheduler to transfer the data
E.g. Transferring Month end reports within applications
IGATE

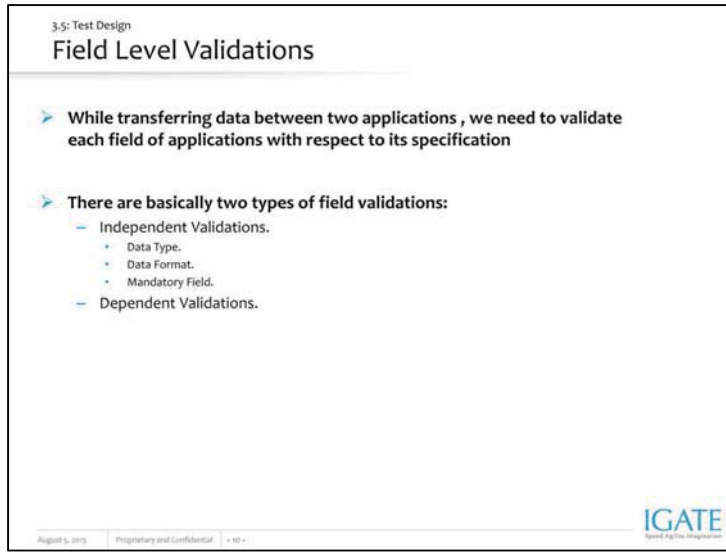

3.2: Types of Import Export
Types of Import - Export
Using XML
E.g. Excel 2003 enables us to map XML elements to fields so user can import / export data from workbook.
Network / Technology used for transferring e.g. LAN, GSM, Bluetooth, etc.
E.g. Data transfer within mobile phones using Bluetooth / Infrared.
Import Export in Database
E.g. Import – Export data between Oracle and SQL Server.
IGATE

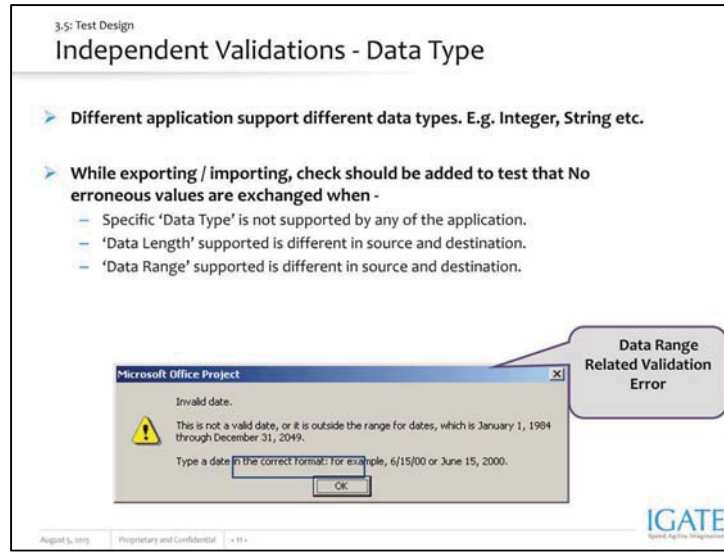

3.3: Importance of Import Export
Advantages of import - export
In a scenario when applications should work independently and still communicate following are the advantage of using import – export
Most widely used method for exchanging data between Application
Least expensive
Import Export enables application(s) talk to each other. Any sort of problem with this functionality means application(s) will fail to share information.
IGATE


3.4: Testing Import Export
Importance of Testing Import - Export
Imagine a Failure in import – export functionality among several application(s) of banking system. When this functionality fails, user account details are not known to billing system, ledger system and SO ON…
User will smile so long as he / she is withdrawing the amount from his account (as his / her balance remain unchanged!!!)… .but this will even cause:
– Wrong computation of account balance even 'Zero Balance'.
– Rejecting transaction through net / mobile banking etc.
– Even in worst case – Rejecting Access to user account.
IGATE

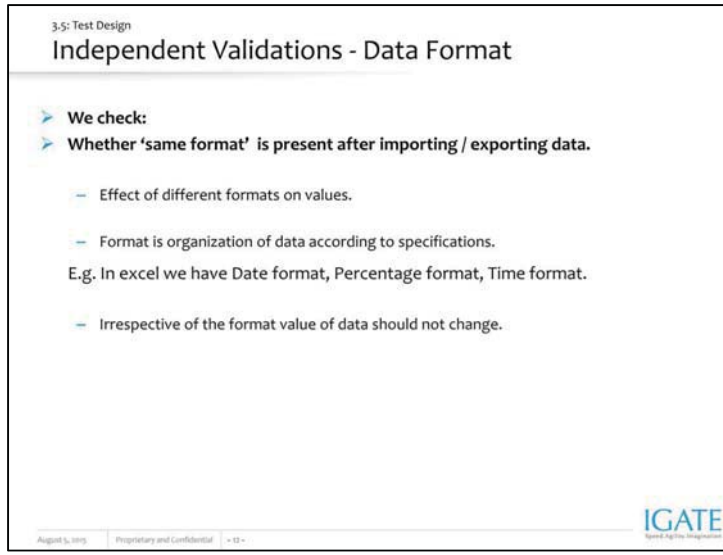


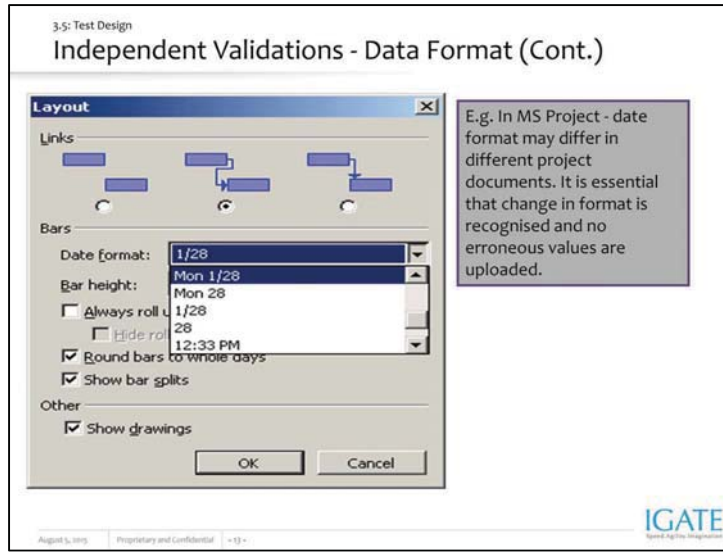
In import – export, destination application accepts external data so at this stage this data should be validated at the field and form levels.

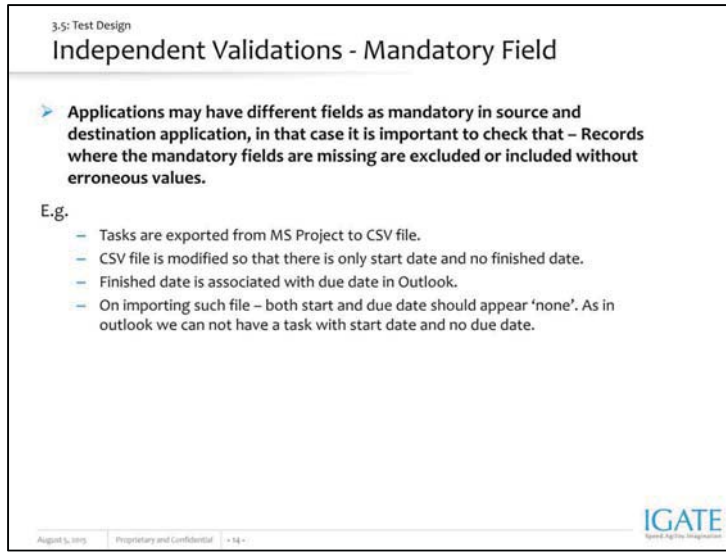

3.5: Test Design
Field Level Validations
While transferring data between two applications , we need to validate each field of applications with respect to its specification
There are basically two types of field validations:
Independent Validations.
Data Type.
Data Format.
Mandatory Field.
Dependent Validations.
IGATE



Data Type: For decimal; value representation double and long could be used.
Data Length: Character length different in source(20) and Destination (15).
Data Range: Employee Salary limit is different in two applications, like 25000 in source and 20000 in destination.

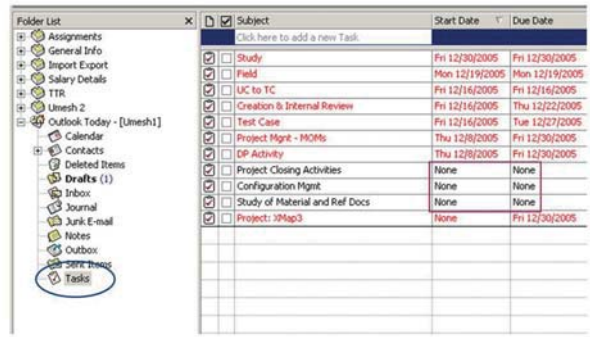

3.5: Test Design
Independent Validations - Data Format
We check:
Whether 'same format' is present after importing / exporting data.
Effect of different formats on values.
Format is organization of data according to specifications.
E.g. In excel we have Date format, Percentage format, Time format.
Irrespective of the format value of data should not change.
IGATE

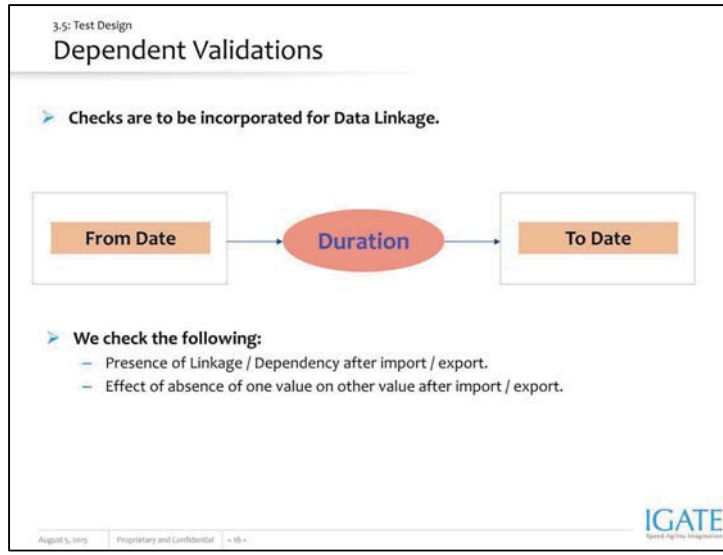

3.5: Test Design
Independent Validations - Data Format (Cont.)
Layout
Links
Bars
Date format:
1/28
Mon 1/28
Mon 28
1/28
28
12:33 PM
Bar height:
Always roll u
Hide ro
Round bars to whole days
Show bar splits
Other
Show drawings
OK
Cancel
E.g. In MS Project - date format may differ in different project documents. It is essential that change in format is recognised and no erroneous values are uploaded.
IGATE


3.5: Test Design
Independent Validations - Mandatory Field
Applications may have different fields as mandatory in source and destination application, in that case it is important to check that – Records where the mandatory fields are missing are excluded or included without erroneous values.
E.g.
– Tasks are exported from MS Project to CSV file.
– CSV file is modified so that there is only start date and no finished date.
– Finished date is associated with due date in Outlook.
– On importing such file – both start and due date should appear 'none'. As in outlook we can not have a task with start date and no due date.
IGATE

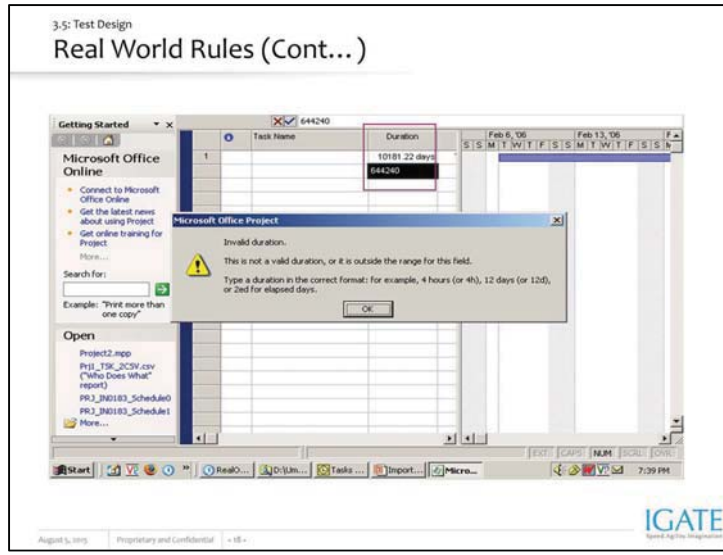
3.5: Test Design

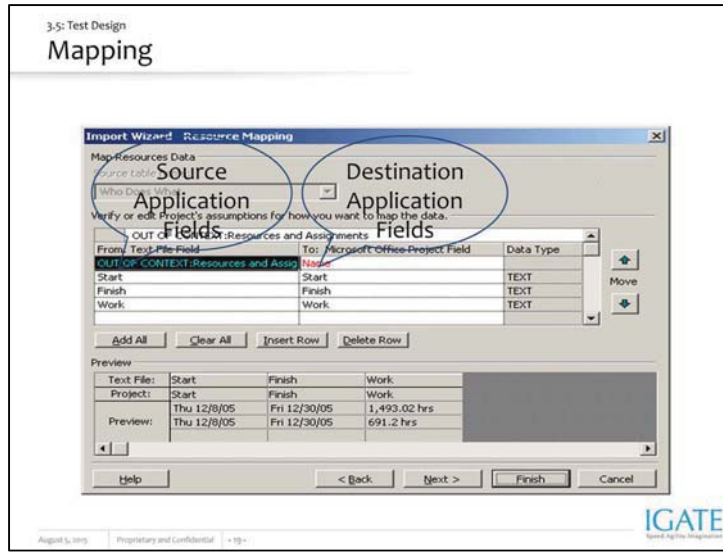
Independent Validations - Mandatory Field




3.5: Test Design
Dependent Validations
Checks are to be incorporated for Data Linkage.
From Date
Duration
To Date
We check the following:
Presence of Linkage / Dependency after import / export.
Effect of absence of one value on other value after import / export.
IGATE


3.5: Test Design
Real World Rules
This is similar to 'Form Level Validations' from Task Based Approach. All these validations are important while importing the data.
When importing only that data should be accepted which abides by certain rules / standards of destination application.
Real World Rules would be domain/project specific.
Examples:
As on 7th Feb 2006, Maximum duration allowed in MS Project is 10181.22 days; any duration beyond this should be converted to this value till 64240. Beyond this user should get the valid error.
IGATE


3.5: Test Design
Real World Rules (Cont...)
Getting Started
Microsoft Office Online
Connect to Microsoft Office Online
Get the latest news about using Project
Get online training for Project
More...
Search for:
Example: "Print more than one copy"
Open
Project2.mpp
Prj_TSK_2CSV.csv ("Who Does What" report)
PRJ_IN0183_Schedule0
PRJ_IN0183_Schedule1
More...
644240
Task Name
Duration
1
10181.22 days
644240
Feb 6, '06
Feb 13, '06
Microsoft Office Project
Invalid duration.
This is not a valid duration, or it is outside the range for this field.
Type a duration in the correct format: for example, 4 hours (or 4h), 12 days (or 12d), or 2ed for elapsed days.
OK
Start
7:39 PM
IGATE

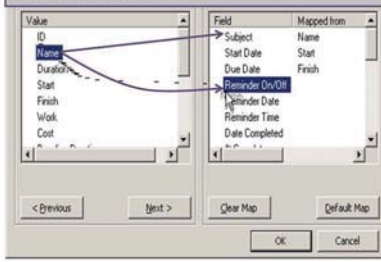

3.5: Test Design
Mapping
Import Wizard - Resource Mapping
Map Resources Data
Source table:
Who Does What
Verify or edit Project's assumptions for how you want to map the data.
OUT OF CONTEXT:Resources and Assignments
From: Text File Field
To: Microsoft Office Project Field
Data Type
OUT OF CONTEXT:Resources and Assig
Name
Start
Start
TEXT
Finish
Finish
TEXT
Work
Work
TEXT
Move
Add All
Clear All
Insert Row
Delete Row
Preview
Text File:
Start
Finish
Work
Project:
Start
Finish
Work
Preview:
Thu 12/8/05
Fri 12/30/05
1,493.02 hrs
Thu 12/8/05
Fri 12/30/05
691.2 hrs
Help
< Back
Next >
Finish
Cancel
Source Application Fields
Destination Application Fields
IGATE

3.5: Test Design

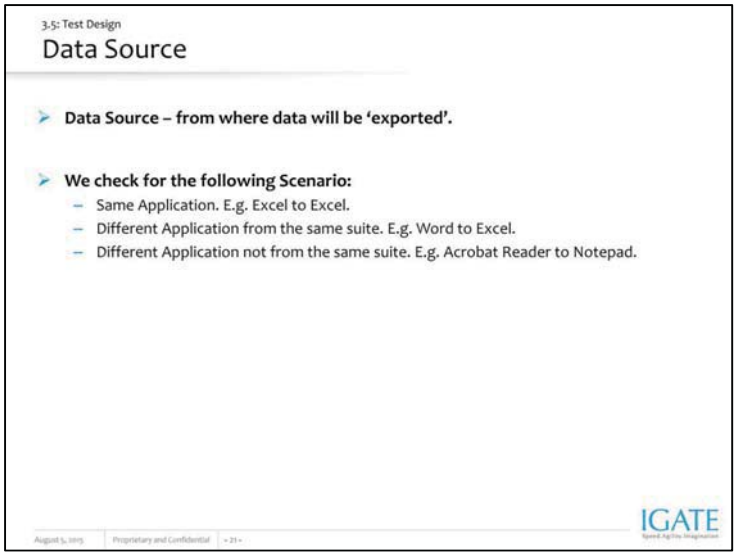
Mapping

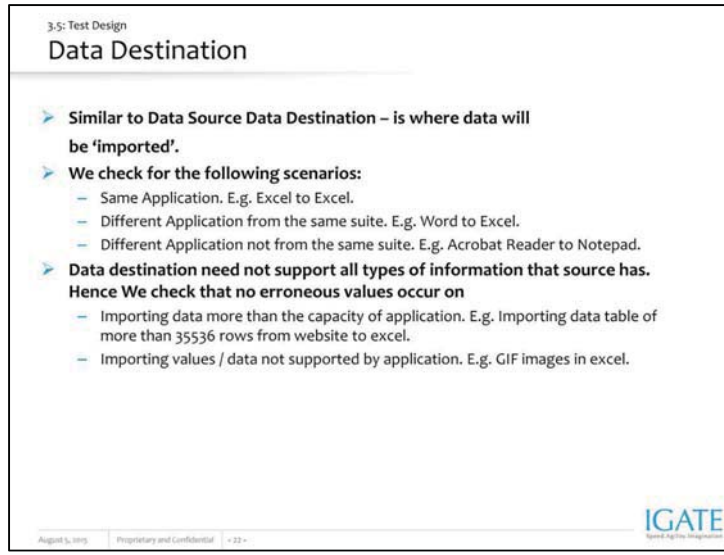
- **While transferring the data from one application to other it is required to map different destination 'fields' to source 'fields'.**
  - We check:
  - Presence of Mapping Option.
  - Effect on Data when not all the fields are mapped.
  - Effect on data when mapping is present for fields that are not present.

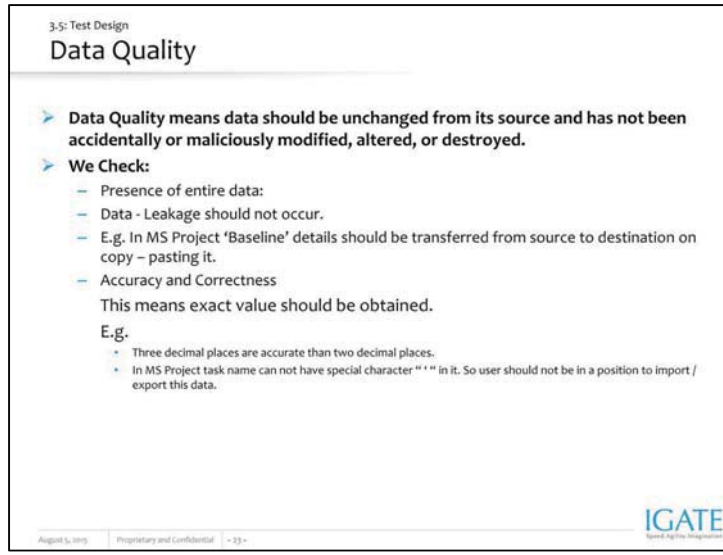
E.g. Mapping 'Task Name, to 'Reminder ON /OFF' (even allowed) creates no sense. But it's mapping to subject will transfer exact task information

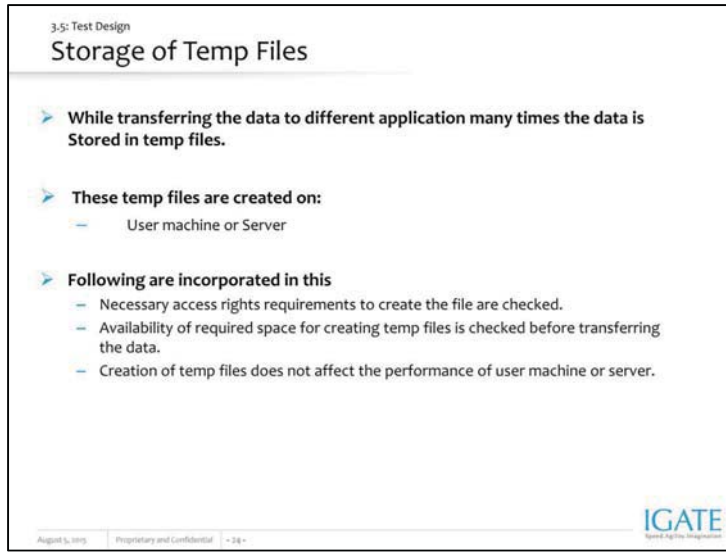


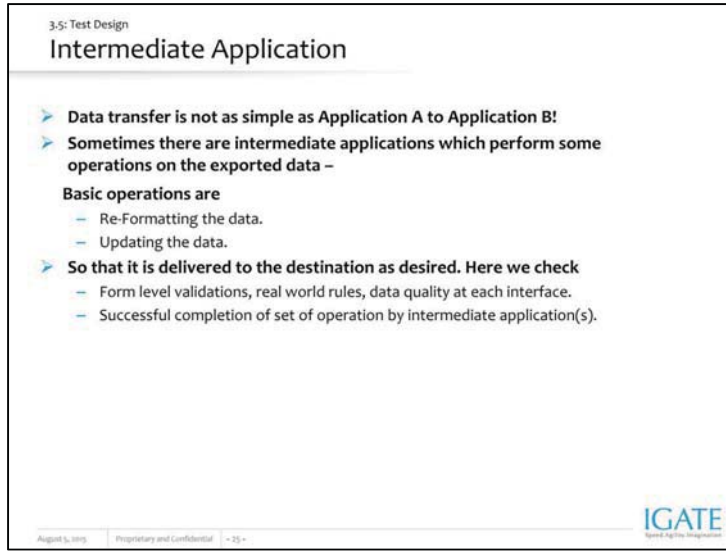
IGATE

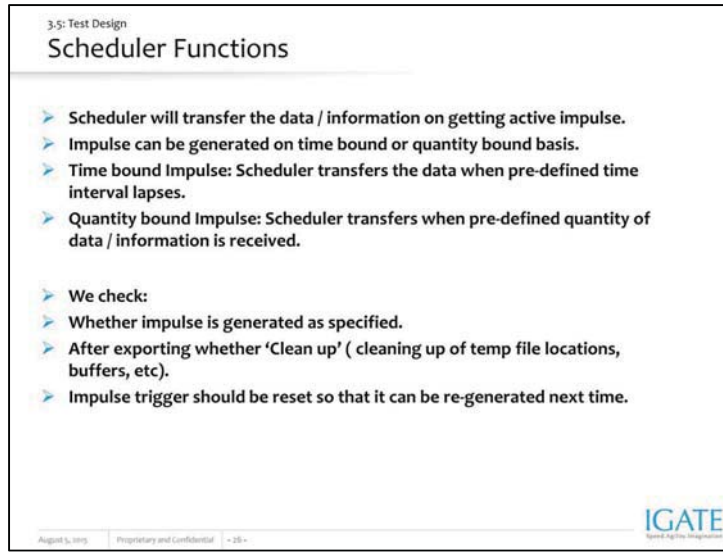

3.5: Test Design
Data Source
Data Source – from where data will be 'exported'.
We check for the following Scenario:
Same Application. E.g. Excel to Excel.
Different Application from the same suite. E.g. Word to Excel.
Different Application not from the same suite. E.g. Acrobat Reader to Notepad.
IGATE


3.5: Test Design
Data Destination
Similar to Data Source Data Destination – is where data will be 'imported'.
We check for the following scenarios:
Same Application. E.g. Excel to Excel.
Different Application from the same suite. E.g. Word to Excel.
Different Application not from the same suite. E.g. Acrobat Reader to Notepad.
Data destination need not support all types of information that source has. Hence We check that no erroneous values occur on
Importing data more than the capacity of application. E.g. Importing data table of more than 35536 rows from website to excel.
Importing values / data not supported by application. E.g. GIF images in excel.
IGATE


3.5: Test Design
Data Quality
Data Quality means data should be unchanged from its source and has not been accidentally or maliciously modified, altered, or destroyed.
We Check:
Presence of entire data:
Data - Leakage should not occur.
E.g. In MS Project 'Baseline' details should be transferred from source to destination on copy – pasting it.
Accuracy and Correctness
This means exact value should be obtained.
E.g.
Three decimal places are accurate than two decimal places.
In MS Project task name can not have special character " ' " in it. So user should not be in a position to import / export this data.
IGATE
August 5, 2015
Proprietary and Confidential
- 23 -


3.5: Test Design
Storage of Temp Files
While transferring the data to different application many times the data is Stored in temp files.
These temp files are created on:
User machine or Server
Following are incorporated in this
Necessary access rights requirements to create the file are checked.
Availability of required space for creating temp files is checked before transferring the data.
Creation of temp files does not affect the performance of user machine or server.
IGATE
August 5, 2015
Proprietary and Confidential
- 24 -


3.5: Test Design
Intermediate Application
Data transfer is not as simple as Application A to Application B!
Sometimes there are intermediate applications which perform some operations on the exported data –
Basic operations are
Re-Formatting the data.
Updating the data.
So that it is delivered to the destination as desired. Here we check
Form level validations, real world rules, data quality at each interface.
Successful completion of set of operation by intermediate application(s).
IGATE


3.5: Test Design
Scheduler Functions
Scheduler will transfer the data / information on getting active impulse.
Impulse can be generated on time bound or quantity bound basis.
Time bound Impulse: Scheduler transfers the data when pre-defined time interval lapses.
Quantity bound Impulse: Scheduler transfers when pre-defined quantity of data / information is received.
We check:
Whether impulse is generated as specified.
After exporting whether 'Clean up' (cleaning up of temp file locations, buffers, etc).
Impulse trigger should be reset so that it can be re-generated next time.
IGATE
August 5, 2015
Proprietary and Confidential
- 26 -

Summary

- **In this lesson, you have learnt:**
  - Import Export is very crucial utility in the applications.
  - Applications share data through this functionality.
  - It is mandatory to validate Import Export functionality across applications communicating to each other.

Review Question

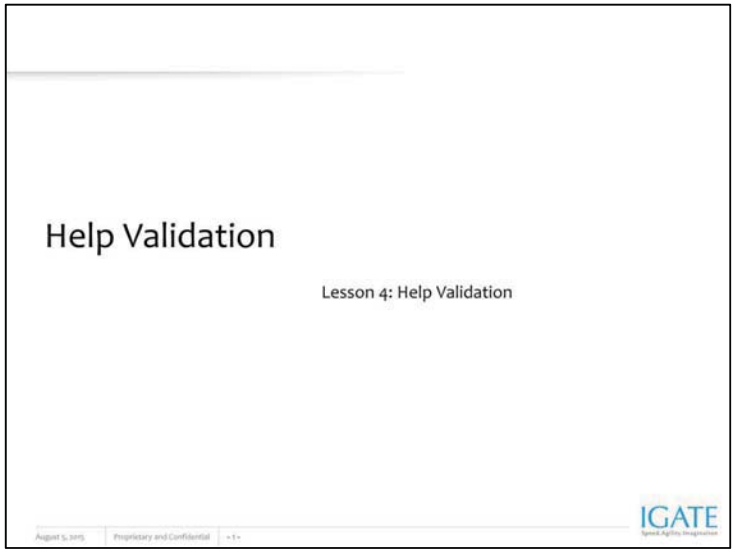
- **Question 1: Bring in from abroad is _____.**
- **Question 2: Data format means data should be unchanged from its source.**
  - True / False
- **Question 3: Availability of required space for creating temp files is checked before transferring the data**
  - True / False

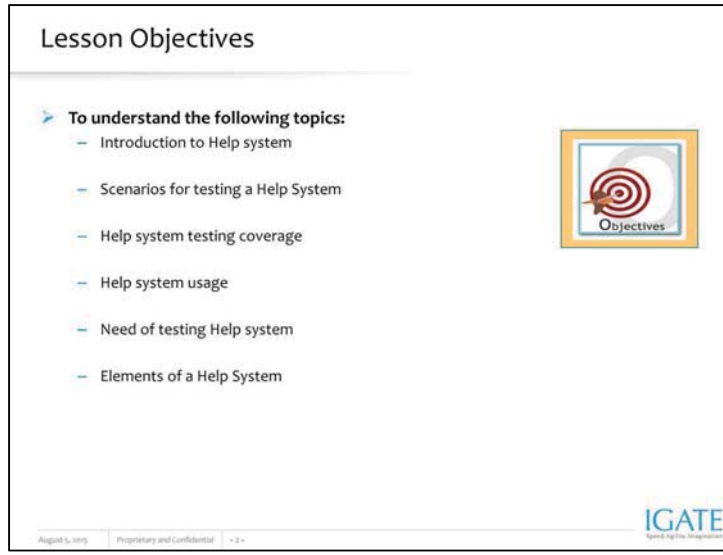
IGATE

Review Question: Match the Following

1. Data quality	A. Data type
2. Data format	B. Expectedly go wrong
3. Field level validation	C. No Data - Leakage
4. Data source	D. Organization of data
	E. Data will be exported from
	F. Data will be exported to

IGATE


Help Validation
Lesson 4: Help Validation
IGATE

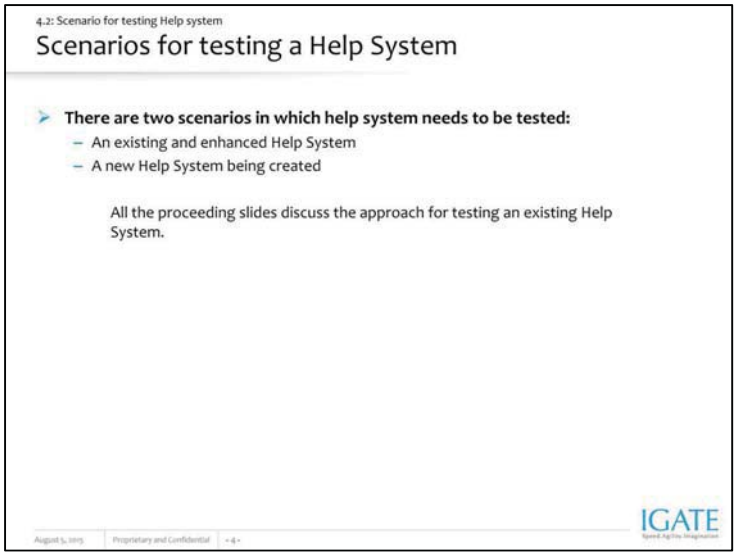

Lesson Objectives
To understand the following topics:
Introduction to Help system
Scenarios for testing a Help System
Help system testing coverage
Help system usage
Need of testing Help system
Elements of a Help System
Objectives
IGATE
August 5, 2015
Proprietary and Confidential
- 2 -

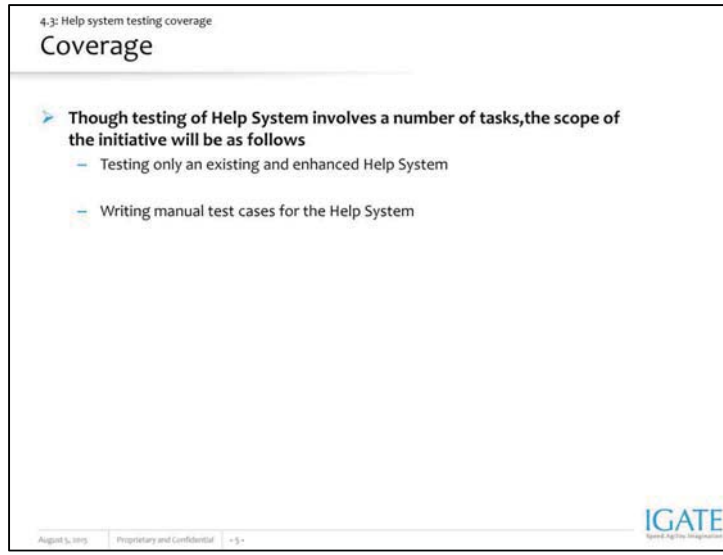
4.1: Introduction to Help system

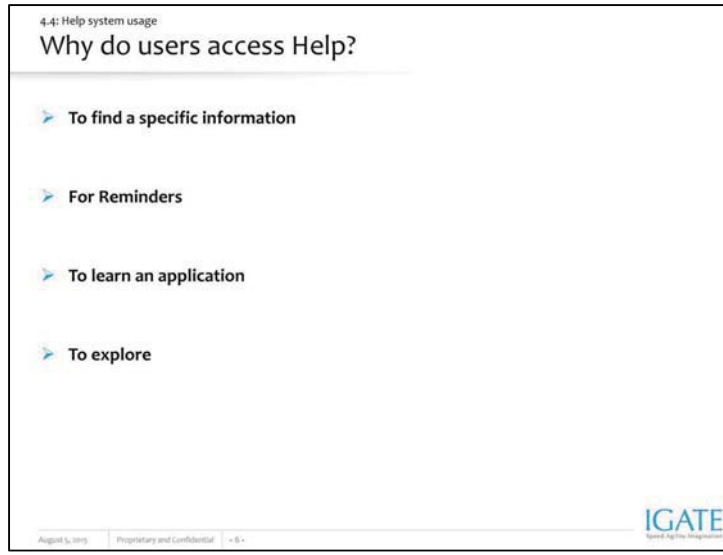
What is a Help System?

- **Help System is one of the most common and important kinds of online documentation.**
- **It facilitates quick access to specific information needed while using the product and assists the users in performing routine tasks and solving problems.**

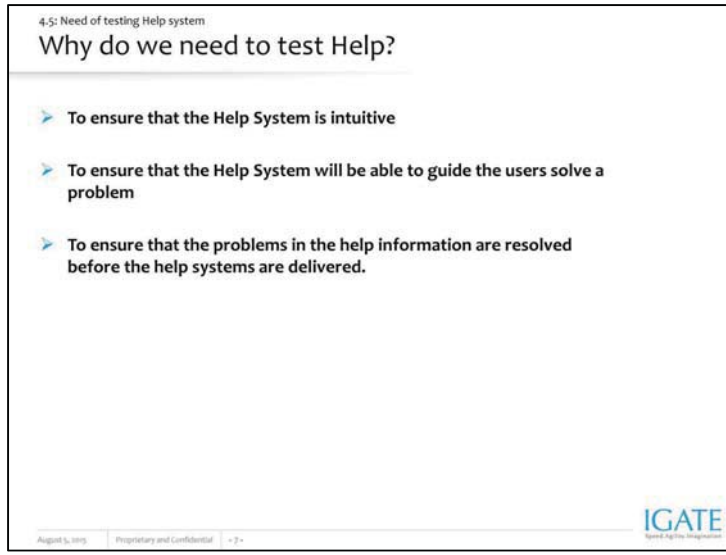
IGATE

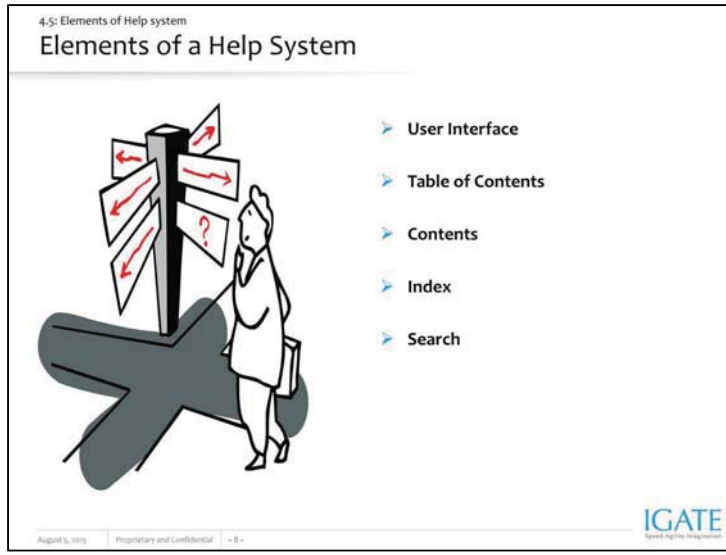

4.2: Scenario for testing Help system
Scenarios for testing a Help System
There are two scenarios in which help system needs to be tested:
An existing and enhanced Help System
A new Help System being created
All the proceeding slides discuss the approach for testing an existing Help System.
IGATE

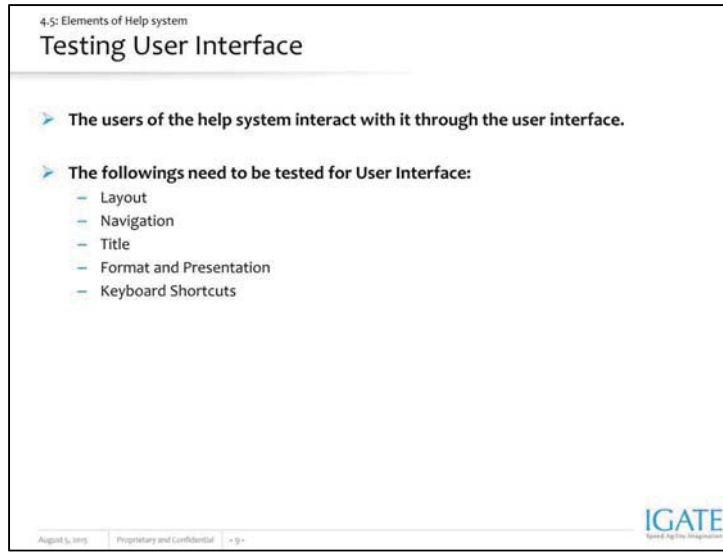

4.3: Help system testing coverage
Coverage
Though testing of Help System involves a number of tasks,the scope of the initiative will be as follows
Testing only an existing and enhanced Help System
Writing manual test cases for the Help System
IGATE

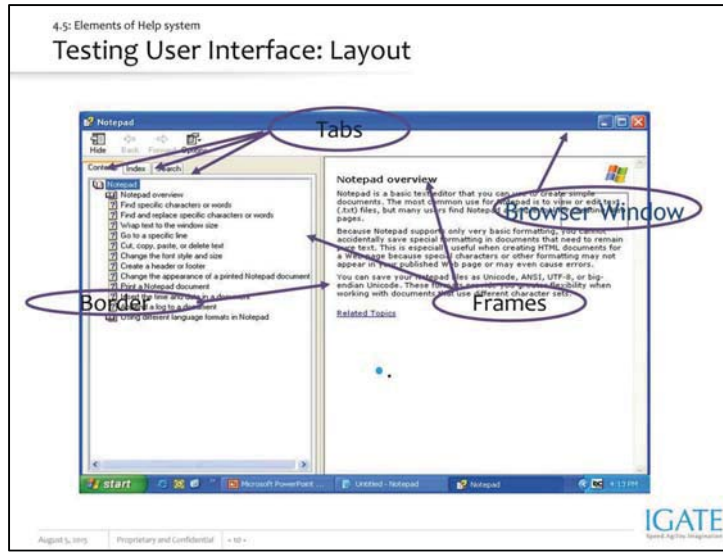


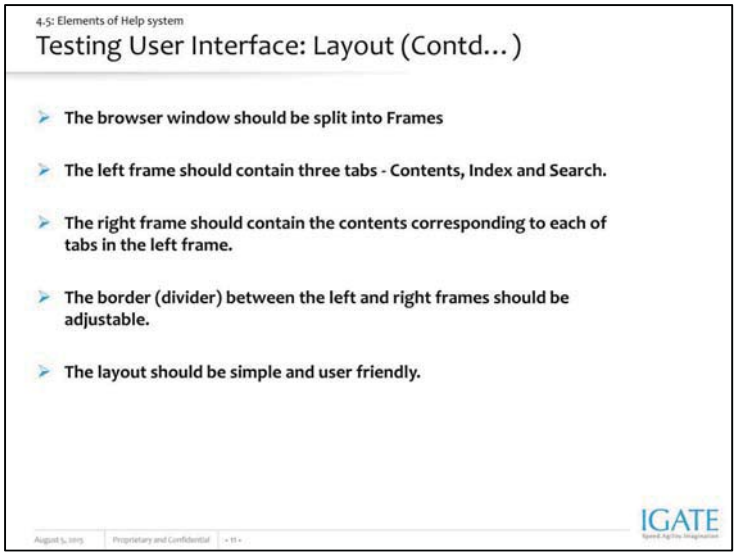
To find specific information. Many users request Help only when they have a specific question. Often, they have tried to do something on their own first and were unsuccessful.
For reminders. Many users refer to Help to remind them of something they have forgotten. For example, users frequently look up specific tasks and keyboard shortcuts.
To learn an application. Novices and other users often choose Help to learn a new application. Because Help is perceived as part of the application, it is the easiest to access.
To explore. Users sometimes browse the Help file just to see what's there, following one topic to the next, gradually tracing a path through the information and building an understanding of it. Hypertext systems like Windows Help are especially compatible with this kind of use.

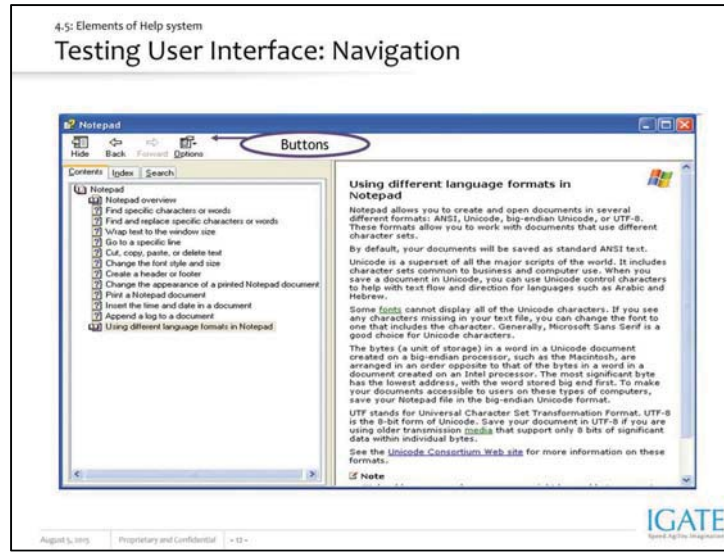

4.5: Need of testing Help system
Why do we need to test Help?
To ensure that the Help System is intuitive
To ensure that the Help System will be able to guide the users solve a problem
To ensure that the problems in the help information are resolved before the help systems are delivered.
IGATE
August 5, 2015
Proprietary and Confidential
- 7 -


4.5: Elements of Help system
Elements of a Help System
User Interface
Table of Contents
Contents
Index
Search
IGATE

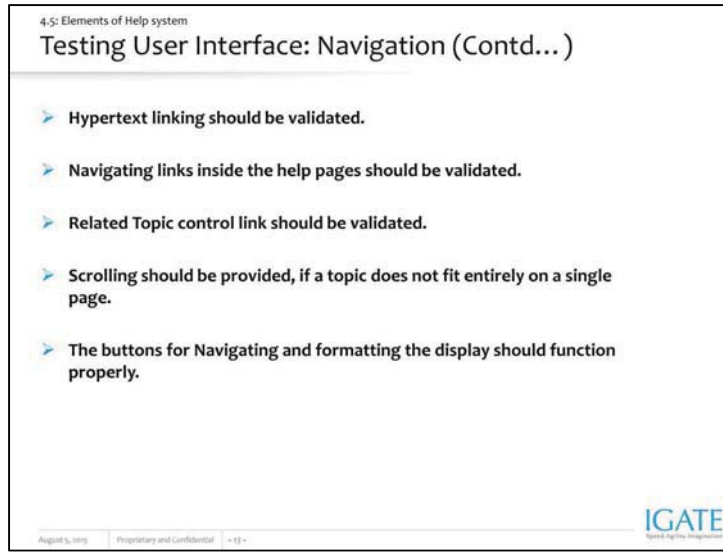

4.5: Elements of Help system
Testing User Interface
The users of the help system interact with it through the user interface.
The followings need to be tested for User Interface:
Layout
Navigation
Title
Format and Presentation
Keyboard Shortcuts
IGATE

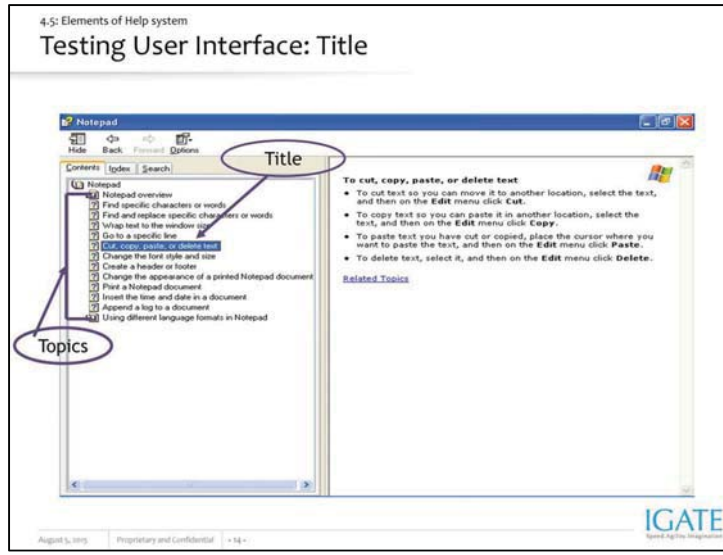

4.5: Elements of Help system
Testing User Interface: Layout
Tabs
Browser Window
Border
Frames
Notepad overview
Related Topics
IGATE


4.5: Elements of Help system
Testing User Interface: Layout (Contd…)
The browser window should be split into Frames
The left frame should contain three tabs - Contents, Index and Search.
The right frame should contain the contents corresponding to each of tabs in the left frame.
The border (divider) between the left and right frames should be adjustable.
The layout should be simple and user friendly.
IGATE
August 5, 2015
Proprietary and Confidential
- 11 -

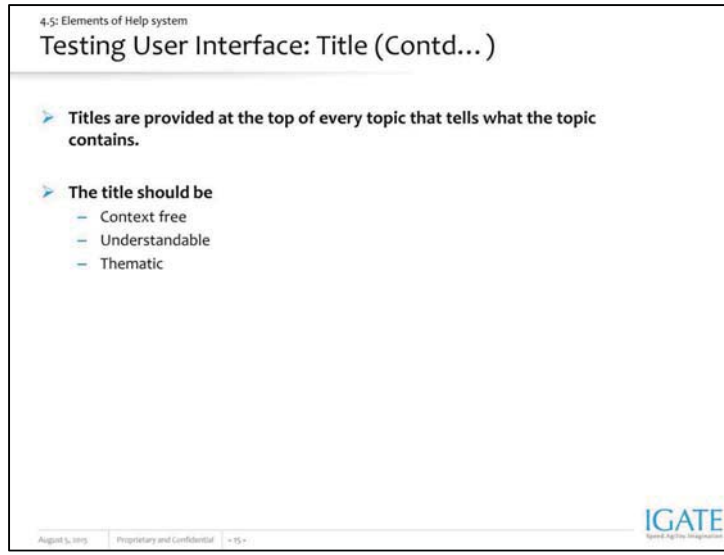


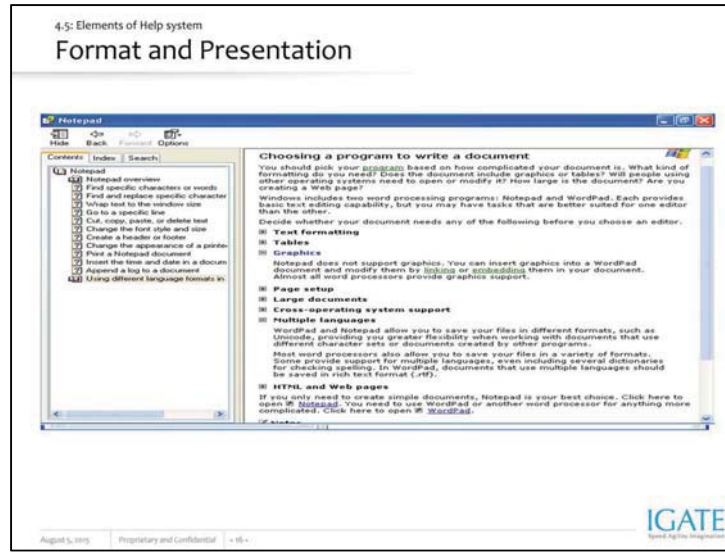
Related Topic Controls: Unlike the Keyword Link or See Also control, the Related Topic control enables Help authors to choose related topics individually to appear in a pop-up window or dialog box. The Related Topic control creates buttons or links within topics that activate a list showing multiple hyperlinks to related topics.


4.5: Elements of Help system
Testing User Interface: Navigation (Contd…)
Hypertext linking should be validated.
Navigating links inside the help pages should be validated.
Related Topic control link should be validated.
Scrolling should be provided, if a topic does not fit entirely on a single page.
The buttons for Navigating and formatting the display should function properly.
IGATE



Context Free- Doesn't depend on the topic or other surrounding information to make the sense of the name.
Understandable- Use standard grammar and terms meaningful to the users.
Thematic- Have the title summarize or preview the topic.Make the meaning of the title obvious in a glance without further reading.

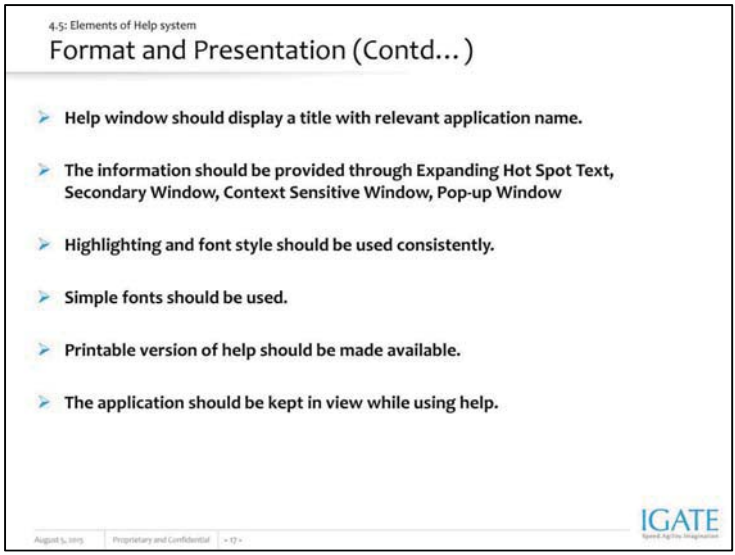

4.5: Elements of Help system
Testing User Interface: Title (Contd…)
Titles are provided at the top of every topic that tells what the topic contains.
The title should be
Context free
Understandable
Thematic
IGATE

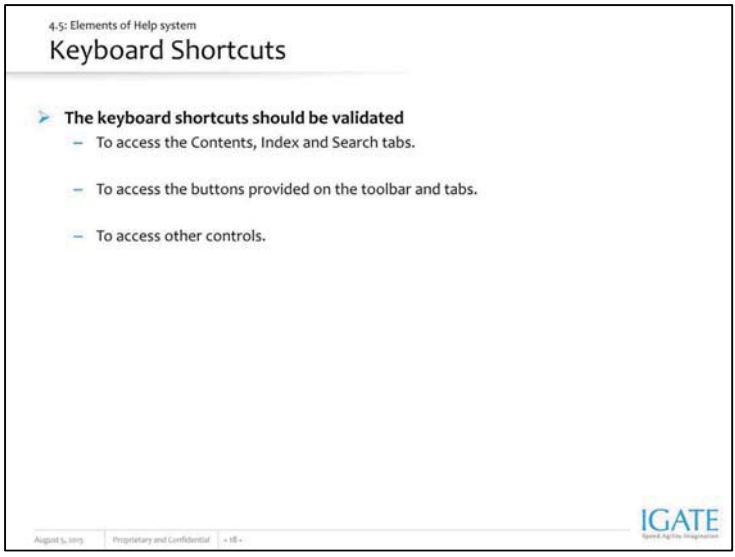


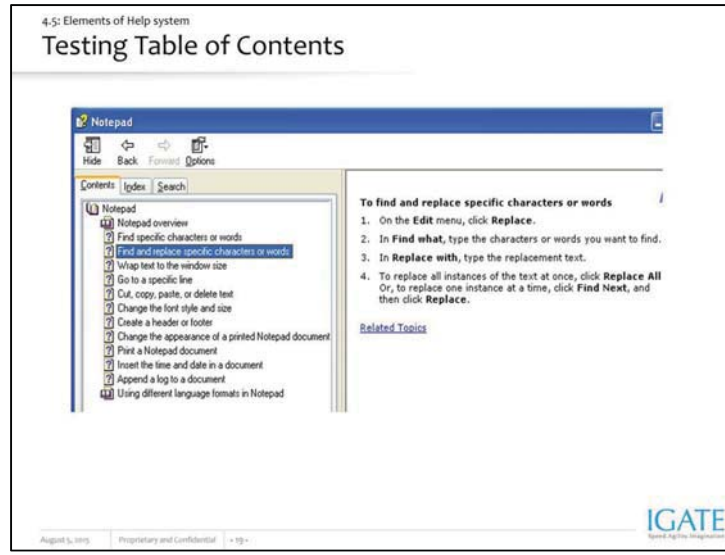
Pop-up windows are designed to display brief amounts of information off a parent main .They remain on the user's screen as long as the focus is on them. They are ideal for definitions of unfamiliar terms or brief examples and images. They appear temporarily on top of larger windows that include topics with more involved text. By appearing temporarily when, say, a user selects a pop-up hyperlink composed of an unfamiliar term, the text that displays in a pop-up enables the user to maintain her point of view in the main window rather than jump away to a topic in the parent window.
Secondary windows are 'static' windows. They remain on your user's screens until they are manually closed. They are available only through the links inside the topics. It is best to design the secondary windows so that they do not cover up the area of the application they address. You can use them for opening a a shortcut to any application from within the topic or to explain the user some settings by opening a window for settings. For e.g- Shortcut opens up the Internet Explorer settings as a secondary window, in order to explain the user the settings for the same.

Context Sensitive Window- It is composed of small blocks of text in a temporary window. These pop up windows typically appear when called as field-level, context sensitive help from the software products interface.

Expanding Hot Spot Text- This can be used to explain something with greater detail for those who need the additional information. Otherwise, the text appears only as simple bulleted or numbered text and reduces the overall length of the topic.


4.5: Elements of Help system
Format and Presentation (Contd...)
Help window should display a title with relevant application name.
The information should be provided through Expanding Hot Spot Text, Secondary Window, Context Sensitive Window, Pop-up Window
Highlighting and font style should be used consistently.
Simple fonts should be used.
Printable version of help should be made available.
The application should be kept in view while using help.
IGATE

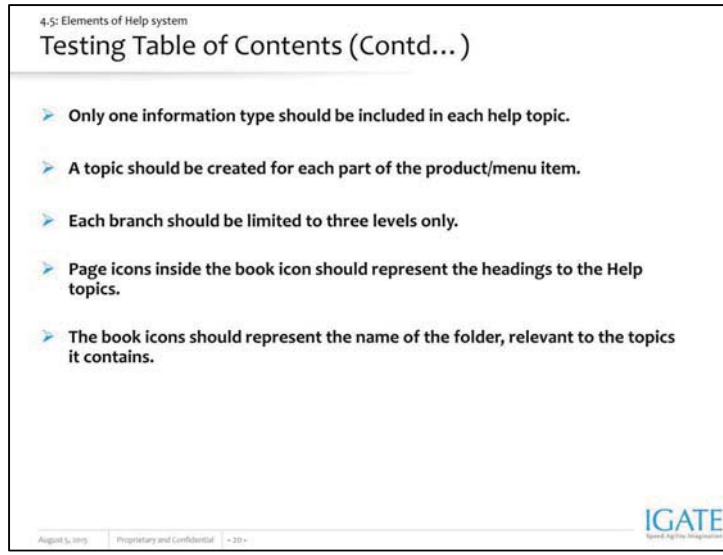

4.5: Elements of Help system
Keyboard Shortcuts
The keyboard shortcuts should be validated
To access the Contents, Index and Search tabs.
To access the buttons provided on the toolbar and tabs.
To access other controls.
IGATE

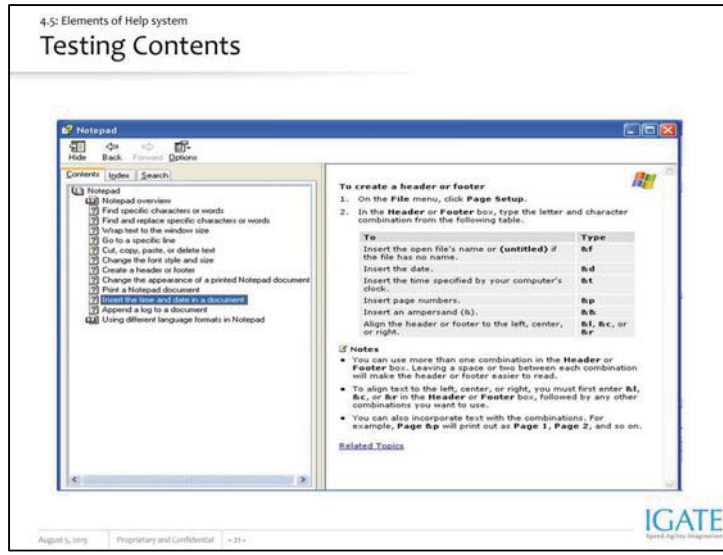


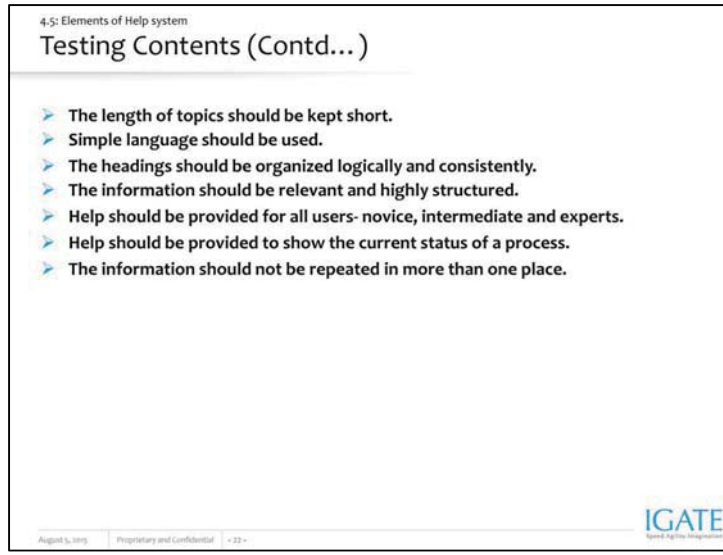
Example of Task Perspective

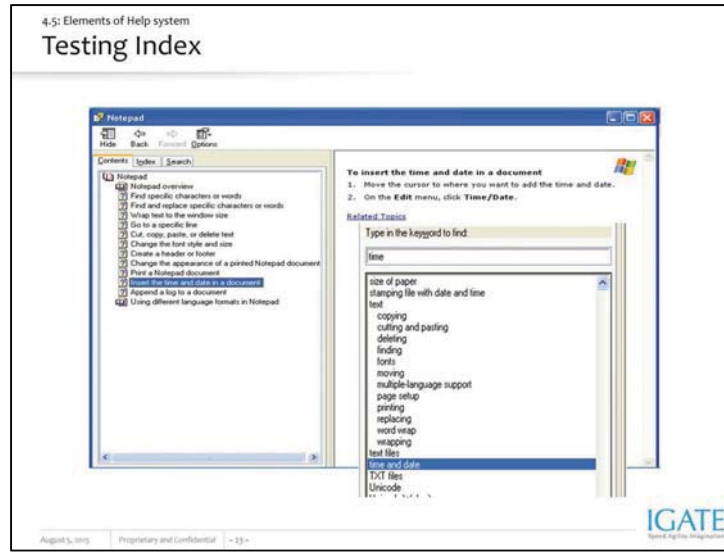
Imagine a person who has never used an ATM machine but would like to know how. Imagine the answer this person might hear after asking the ATM's engineer or creator how it works. The engineer may describe how the ATM functions electronically, in other words, the system operations and technical processes that make it work. None of this information, however, would actually help the user perform a practical task. In fact, it's unlikely that the engineer's information would interest that potential user at all. He or she wants only to know, for example, how to draw money from an ATM, or deposit a check, or use it in other ways that would transform it into a practical and useful tool. Thus, the potential user is uninterested in the engineering behind the ATM; instead, the user wants to know how to use it as a tool to fulfill a practical need.

To create good Help systems, technical writers must understand the users' needs. In so doing, technical writers will be able write Help from a task-perspective, anticipating the users' practical needs, instead of a system-perspective, which is the technical description of how something functions.


4.5: Elements of Help system
Testing Table of Contents (Contd…)
Only one information type should be included in each help topic.
A topic should be created for each part of the product/menu item.
Each branch should be limited to three levels only.
Page icons inside the book icon should represent the headings to the Help topics.
The book icons should represent the name of the folder, relevant to the topics it contains.
IGATE

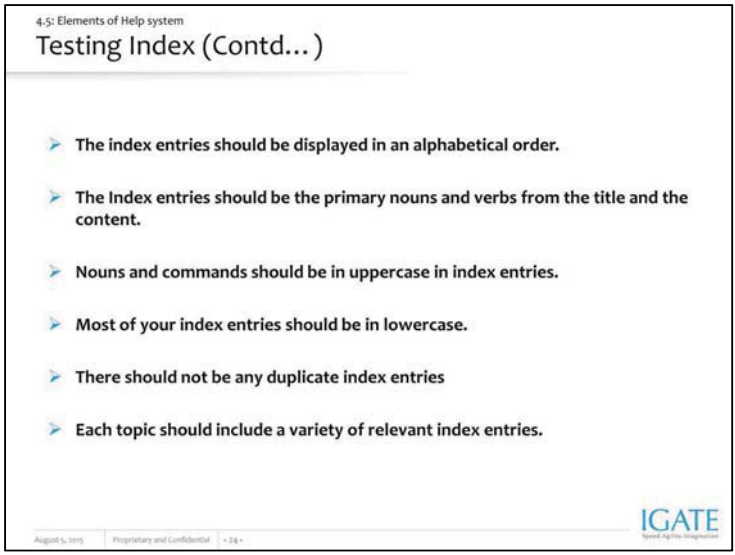

4.5: Elements of Help system
Testing Contents
Notepad
Hide
Back
Forward
Options
Contents
Index
Search
Notepad
Notepad overview
Find specific characters or words
Find and replace specific characters or words
Wrap text to the window size
Go to a specific line
Cut, copy, paste, or delete text
Change the font style and size
Create a header or footer
Change the appearance of a printed Notepad document
Print a Notepad document
Insert the time and date in a document
Append a log to a document
Using different language formats in Notepad
To create a header or footer
1. On the File menu, click Page Setup.
2. In the Header or Footer box, type the letter and character combination from the following table.
To
Type
Insert the open file's name or (untitled) if the file has no name.
&f
Insert the date.
&d
Insert the time specified by your computer's clock.
&t
Insert page numbers.
&p
Insert an ampersand (&).
&&
Align the header or footer to the left, center, or right.
&l, &c, or &r
Notes
You can use more than one combination in the Header or Footer box. Leaving a space or two between each combination will make the header or footer easier to read.
To align text to the left, center, or right, you must first enter &l, &c, or &r in the Header or Footer box, followed by any other combinations you want to use.
You can also incorporate text with the combinations. For example, Page &p will print out as Page 1, Page 2, and so on.
Related Topics
IGATE
August 5, 2015
Proprietary and Confidential
- 21 -

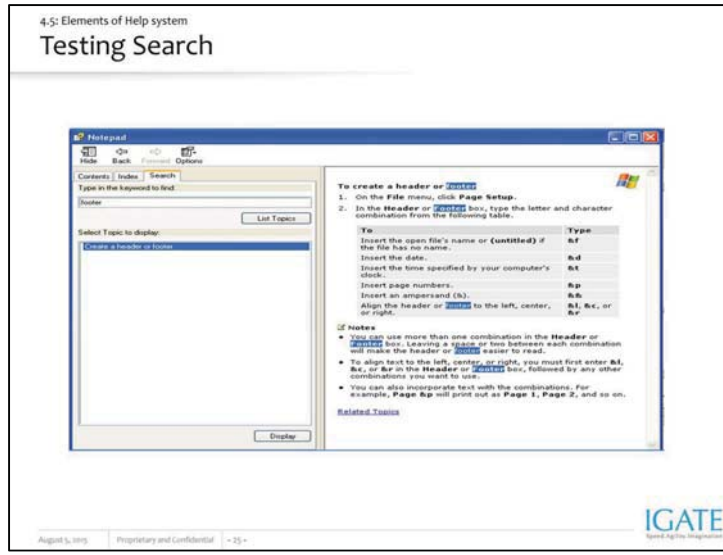

4.5: Elements of Help system
Testing Contents (Contd…)
The length of topics should be kept short.
Simple language should be used.
The headings should be organized logically and consistently.
The information should be relevant and highly structured.
Help should be provided for all users- novice, intermediate and experts.
Help should be provided to show the current status of a process.
The information should not be repeated in more than one place.
IGATE

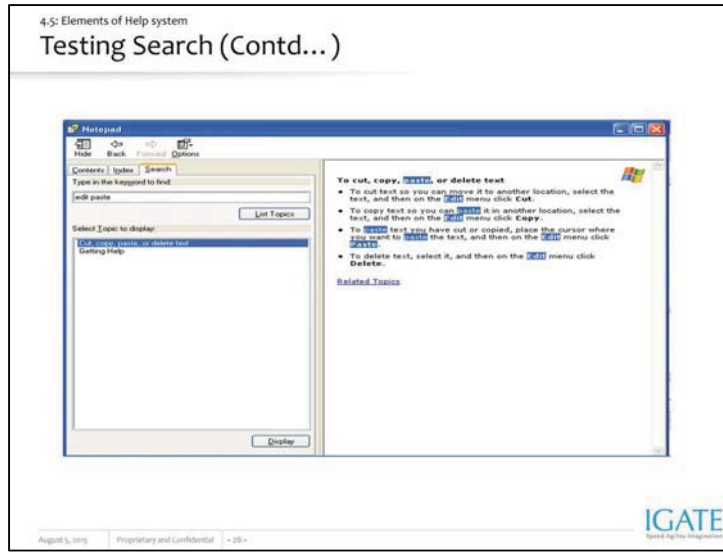


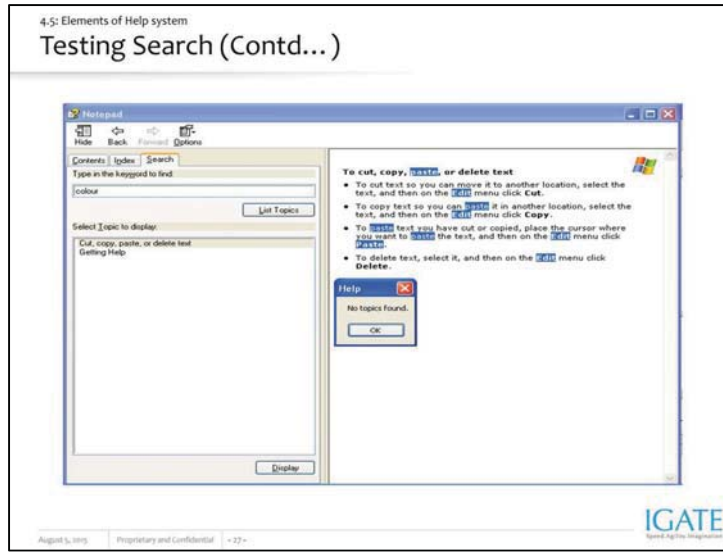
Many index entries are verb forms that express the verb-plus-object relationship. For these index entries, use gerunds (deleting files). to all your entries.
Most Help authors choose the plural because it does not require the use of an article before the noun. Using plurals is not problematic when using gerund entries as plurals (deleting files). Other nouns are singular because they are concepts (alignment) or proper nouns (Truecode). In the case of interface elements, use the plural when there are more than one of them (toolbars); use the singular when there is only one (Ruler). Lowercase most index entries but uppercase proper nouns or interface elements that are normally capitalized in documentation. These include commands (Save command) and interface objects such as Project toolbar or Date field. Use uppercase also for terms that are normally capitalized, such as ASCII.

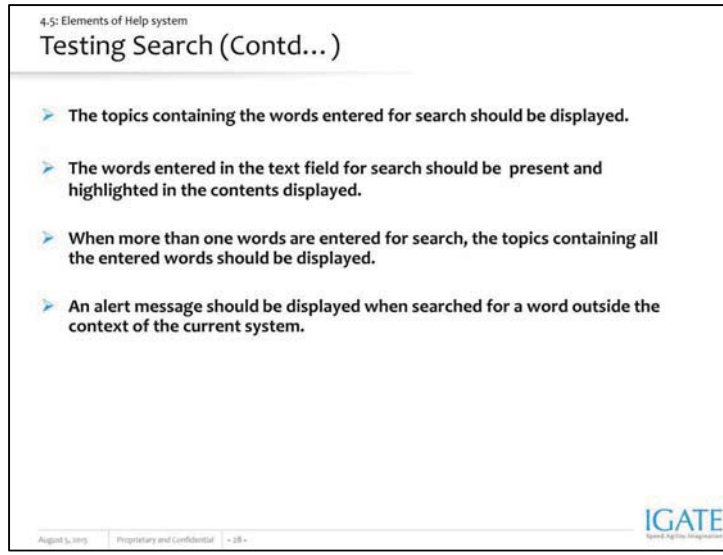
Review all index entries to look for unintentional duplications and variations of the same words. For example, avoid creating an index that includes both date and dates as separate entries, as well as similar entries, such as adding record entries and adding records. They are practically the same.


4.5: Elements of Help system
Testing Index (Contd...)
The index entries should be displayed in an alphabetical order.
The Index entries should be the primary nouns and verbs from the title and the content.
Nouns and commands should be in uppercase in index entries.
Most of your index entries should be in lowercase.
There should not be any duplicate index entries
Each topic should include a variety of relevant index entries.
IGATE


4.5: Elements of Help system
Testing Search
Notepad
Hide
Back
Forward
Options
Contents
Index
Search
Type in the keyword to find:
footer
List Topics
Select Topic to display:
Create a header or footer
Display
To create a header or footer
1. On the File menu, click Page Setup.
2. In the Header or Footer box, type the letter and character combination from the following table.
To
Type
Insert the open file's name or (untitled) if the file has no name.
&f
Insert the date.
&d
Insert the time specified by your computer's clock.
&t
Insert page numbers.
&p
Insert an ampersand (&).
&&
Align the header or footer to the left, center, or right.
&l, &c, or &r
Notes
You can use more than one combination in the Header or Footer box. Leaving a space or two between each combination will make the header or footer easier to read.
To align text to the left, center, or right, you must first enter &l, &c, or &r in the Header or Footer box, followed by any other combinations you want to use.
You can also incorporate text with the combinations. For example, Page &p will print out as Page 1, Page 2, and so on.
Related Topics
IGATE
August 5, 2015
Proprietary and Confidential
- 25 -


4.5: Elements of Help system
Testing Search (Contd...)
Notepad
Hide
Back
Forward
Options
Contents
Index
Search
Type in the keyword to find:
edit paste
List Topics
Select Topic to display:
Cut, copy, paste, or delete text
Getting Help
Display
To cut, copy, paste, or delete text
To cut text so you can move it to another location, select the text, and then on the Edit menu click Cut.
To copy text so you can paste it in another location, select the text, and then on the Edit menu click Copy.
To paste text you have cut or copied, place the cursor where you want to paste the text, and then on the Edit menu click Paste.
To delete text, select it, and then on the Edit menu click Delete.
Related Topics
IGATE
August 5, 2015
Proprietary and Confidential
- 26 -

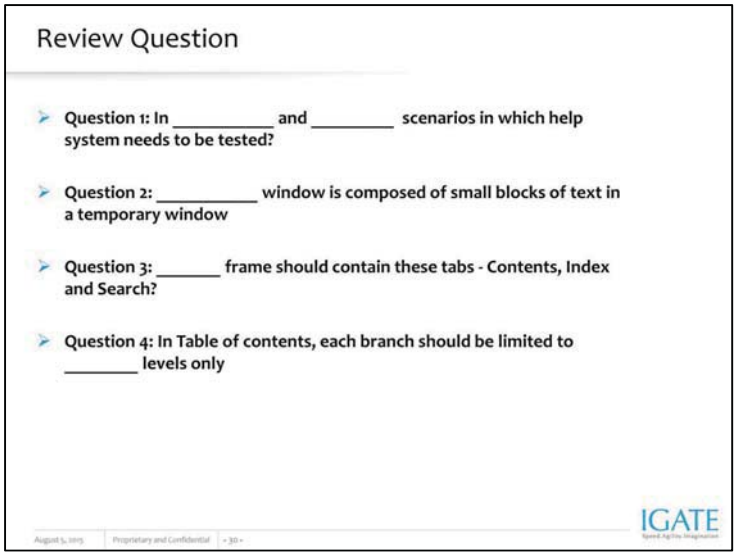

4.5: Elements of Help system
Testing Search (Contd...)
Notepad
Hide
Back
Forward
Options
Contents
Index
Search
Type in the keyword to find:
colour
List Topics
Select Topic to display:
Cut, copy, paste, or delete text
Getting Help
Display
To cut, copy, paste, or delete text
To cut text so you can move it to another location, select the text, and then on the Edit menu click Cut.
To copy text so you can paste it in another location, select the text, and then on the Edit menu click Copy.
To paste text you have cut or copied, place the cursor where you want to paste the text, and then on the Edit menu click Paste.
To delete text, select it, and then on the Edit menu click Delete.
Help
No topics found.
OK
IGATE
August 5, 2015
Proprietary and Confidential
- 27 -


4.5: Elements of Help system
Testing Search (Contd…)
The topics containing the words entered for search should be displayed.
The words entered in the text field for search should be present and highlighted in the contents displayed.
When more than one words are entered for search, the topics containing all the entered words should be displayed.
An alert message should be displayed when searched for a word outside the context of the current system.
IGATE

Summary

- **In this lesson, you have learnt:**
  - Help system is very important support system in the applications.
  - Help system helps to understand the functionality of the application.
  - It is mandatory to validate Help functionality across applications communicating to each other.

IGATE

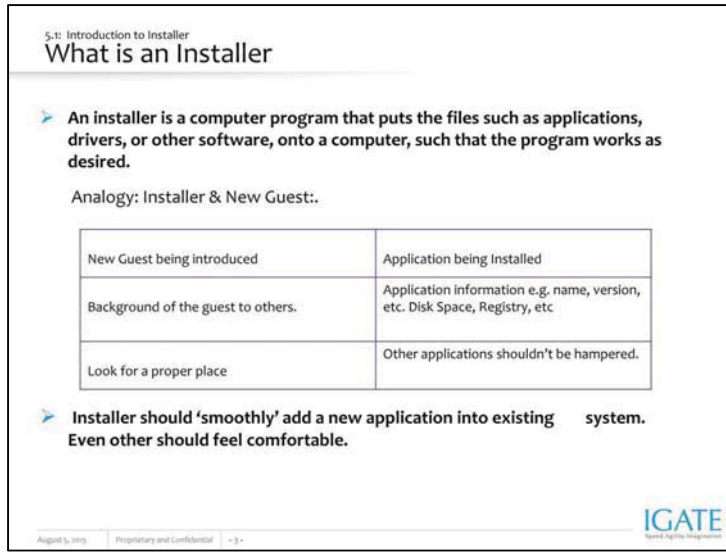

Review Question
Question 1: In _____ and _____ scenarios in which help system needs to be tested?
Question 2: _____ window is composed of small blocks of text in a temporary window
Question 3: _____ frame should contain these tabs - Contents, Index and Search?
Question 4: In Table of contents, each branch should be limited to _____ levels only
IGATE

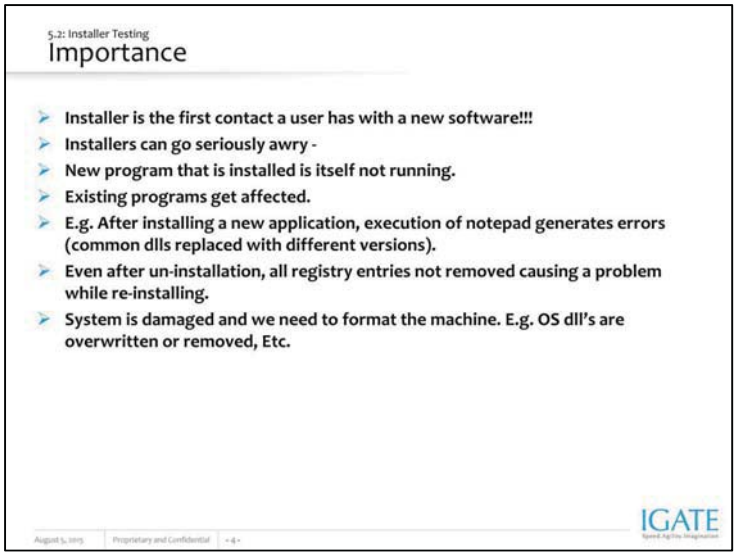

Installation Testing
Lesson 5: Installation Validation
IGATE

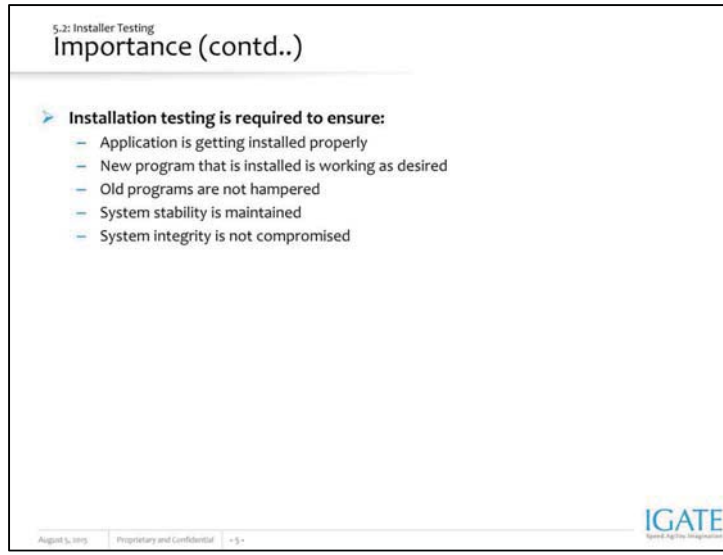
Lesson Objectives

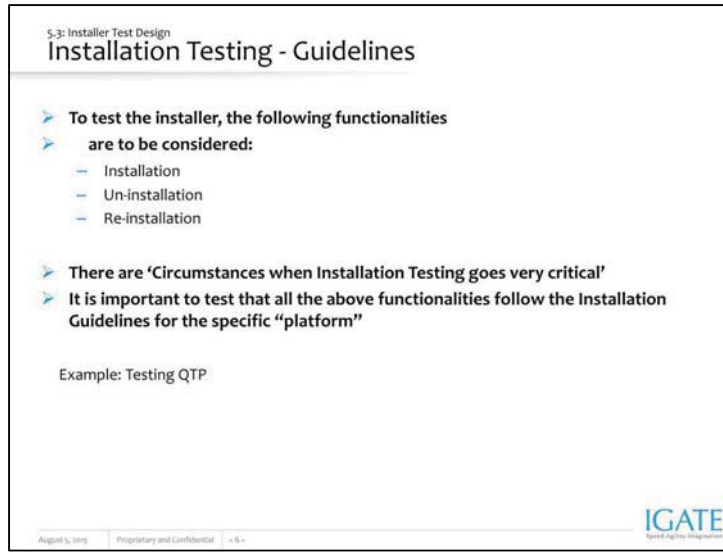
- **To understand the following topics:**
  - Introduction to Installer
  - Importance of Testing the Installer
  - Installer Test Design
    - Installation
    - Un-installation
    - Re-installation
  - Importance of Installation testing
  - OS Guidelines
  - Example – QTP Installation

IGATE

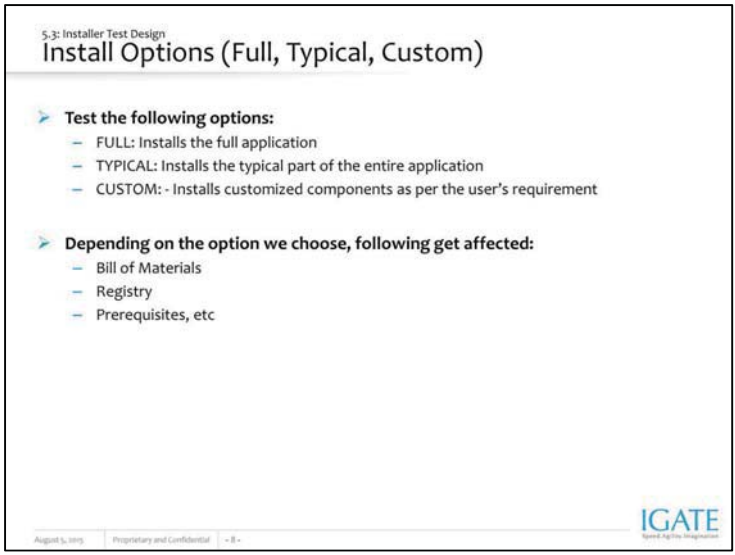

5.1: Introduction to Installer
What is an Installer
An installer is a computer program that puts the files such as applications, drivers, or other software, onto a computer, such that the program works as desired.
Analogy: Installer & New Guest:.
New Guest being introduced
Application being Installed
Background of the guest to others.
Application information e.g. name, version, etc. Disk Space, Registry, etc
Look for a proper place
Other applications shouldn't be hampered.
Installer should 'smoothly' add a new application into existing system. Even other should feel comfortable.
IGATE
August 5, 2015
Proprietary and Confidential
- 3 -

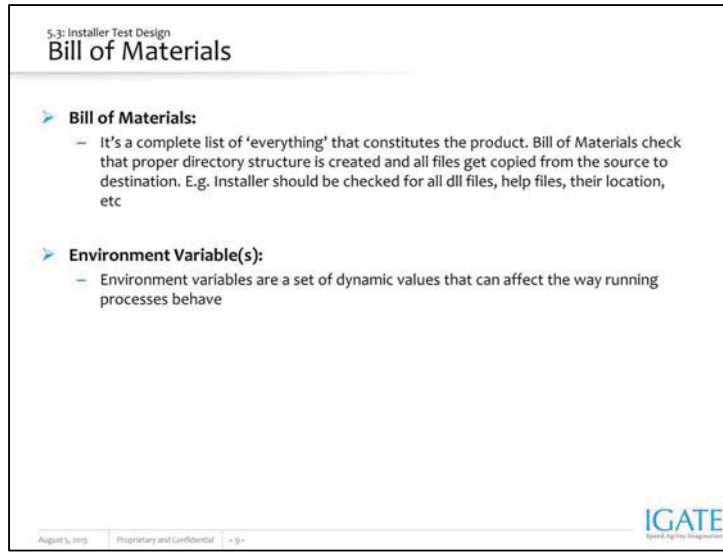

5.2: Installer Testing
Importance
Installer is the first contact a user has with a new software!!!
Installers can go seriously awry -
New program that is installed is itself not running.
Existing programs get affected.
E.g. After installing a new application, execution of notepad generates errors (common dlls replaced with different versions).
Even after un-installation, all registry entries not removed causing a problem while re-installing.
System is damaged and we need to format the machine. E.g. OS dll's are overwritten or removed, Etc.
IGATE
August 5, 2015
Proprietary and Confidential
- 4 -

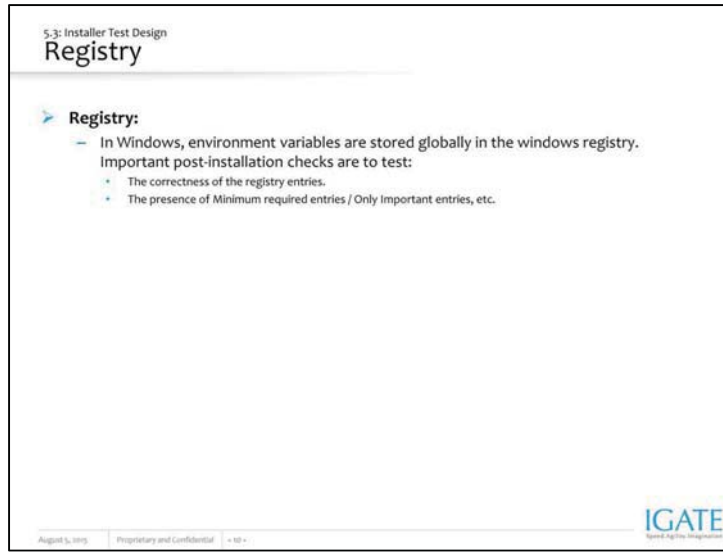

5.2: Installer Testing
Importance (contd..)
Installation testing is required to ensure:
Application is getting installed properly
New program that is installed is working as desired
Old programs are not hampered
System stability is maintained
System integrity is not compromised
IGATE


5.3: Installer Test Design
Installation Testing - Guidelines
To test the installer, the following functionalities
are to be considered:
Installation
Un-installation
Re-installation
There are 'Circumstances when Installation Testing goes very critical'
It is important to test that all the above functionalities follow the Installation Guidelines for the specific "platform"
Example: Testing QTP
IGATE


5-3: Installer Test Design
Installation checklist
Test Design for testing the installer of an application:
Install Options (Full, Typical, Custom)
Bill of Materials
Registry
Configuration(s) of supported platforms
Existence of other Software(s) on the machine
Error Recovery
Cancel / Exit Operation
Running Installer on drives other than the Default Drives
Installation Source
Backward Navigation
Prerequisites
Availability of the required Patches
Localization
IGATE


5.3: Installer Test Design
Install Options (Full, Typical, Custom)
Test the following options:
FULL: Installs the full application
TYPICAL: Installs the typical part of the entire application
CUSTOM: - Installs customized components as per the user's requirement
Depending on the option we choose, following get affected:
Bill of Materials
Registry
Prerequisites, etc
IGATE

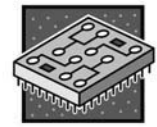

5.3: Installer Test Design
Bill of Materials
Bill of Materials:
It's a complete list of 'everything' that constitutes the product. Bill of Materials check that proper directory structure is created and all files get copied from the source to destination. E.g. Installer should be checked for all dll files, help files, their location, etc
Environment Variable(s):
Environment variables are a set of dynamic values that can affect the way running processes behave
IGATE


5.3: Installer Test Design
Registry
Registry:
In Windows, environment variables are stored globally in the windows registry. Important post-installation checks are to test:
The correctness of the registry entries.
The presence of Minimum required entries / Only Important entries, etc.
IGATE

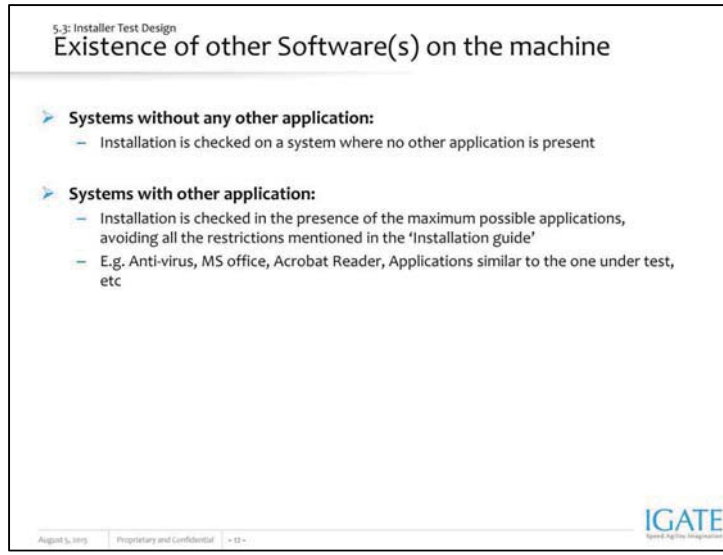
5.3: Installer Test Design

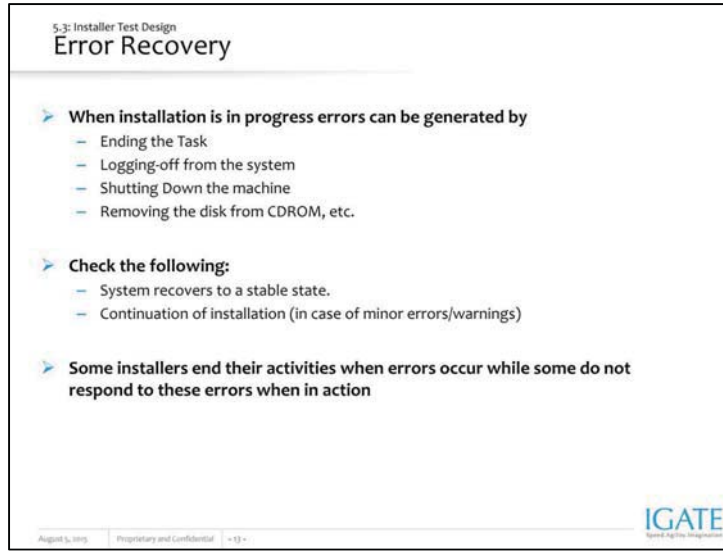
Supported Platforms - Configuration

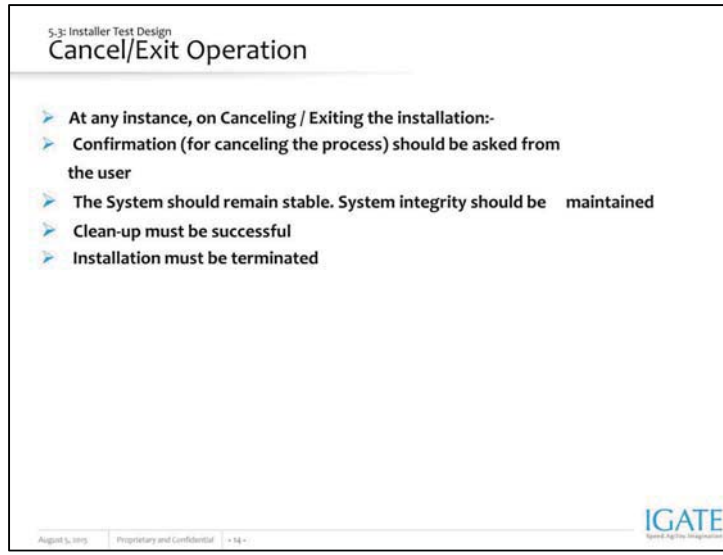
- **Installer should be checked on all supported configurations :**
  - Hardware (RAM, Speed, HDD, etc) :
  - Minimum Hardware set-up.
  - Recommended Hardware set-up.
  - Ideal Hardware Setup.
  - Operating Systems.
  - Disk File Formats – NTFS, FAT32.etc

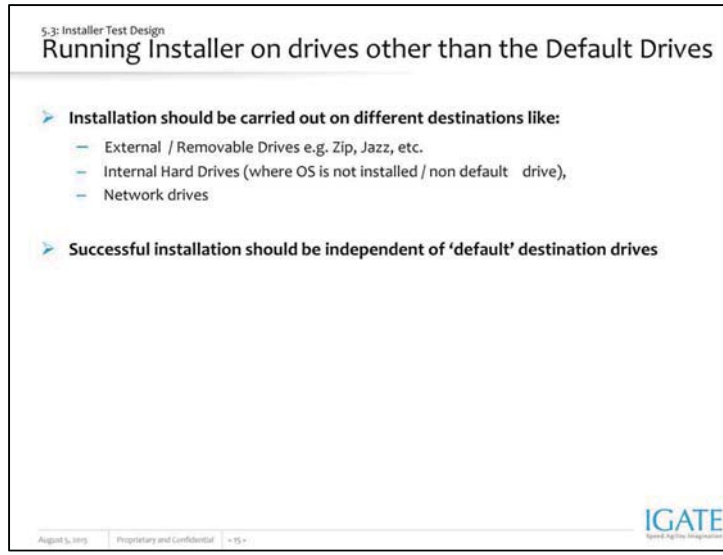


IGATE


5.3: Installer Test Design
Existence of other Software(s) on the machine
Systems without any other application:
Installation is checked on a system where no other application is present
Systems with other application:
Installation is checked in the presence of the maximum possible applications, avoiding all the restrictions mentioned in the 'Installation guide'
E.g. Anti-virus, MS office, Acrobat Reader, Applications similar to the one under test, etc
IGATE


5.3: Installer Test Design
Error Recovery
When installation is in progress errors can be generated by
Ending the Task
Logging-off from the system
Shutting Down the machine
Removing the disk from CDROM, etc.
Check the following:
System recovers to a stable state.
Continuation of installation (in case of minor errors/warnings)
Some installers end their activities when errors occur while some do not respond to these errors when in action
IGATE
August 5, 2015
Proprietary and Confidential
- 13 -


5-3: Installer Test Design
Cancel/Exit Operation
At any instance, on Canceling / Exiting the installation:-
Confirmation (for canceling the process) should be asked from the user
The System should remain stable. System integrity should be maintained
Clean-up must be successful
Installation must be terminated
IGATE
August 5, 2015
Proprietary and Confidential
- 14 -

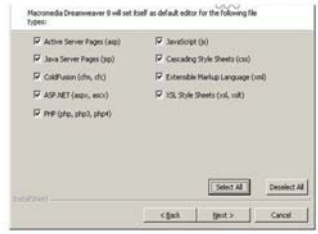

5.3: Installer Test Design
Running Installer on drives other than the Default Drives
Installation should be carried out on different destinations like:
External / Removable Drives e.g. Zip, Jazz, etc.
Internal Hard Drives (where OS is not installed / non default drive),
Network drives
Successful installation should be independent of 'default' destination drives
IGATE

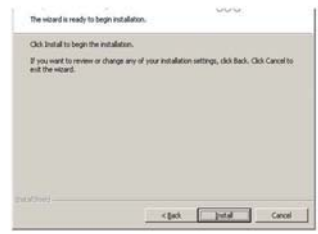

5.3: Installer Test Design
Installation Source
At Test installing the application from all the installation sources:
CD
Network Drive
Removable Drives
HDD, etc
LAN/WAN/Internet
Successful installation should be independent of 'default' installation sources.
IGATE

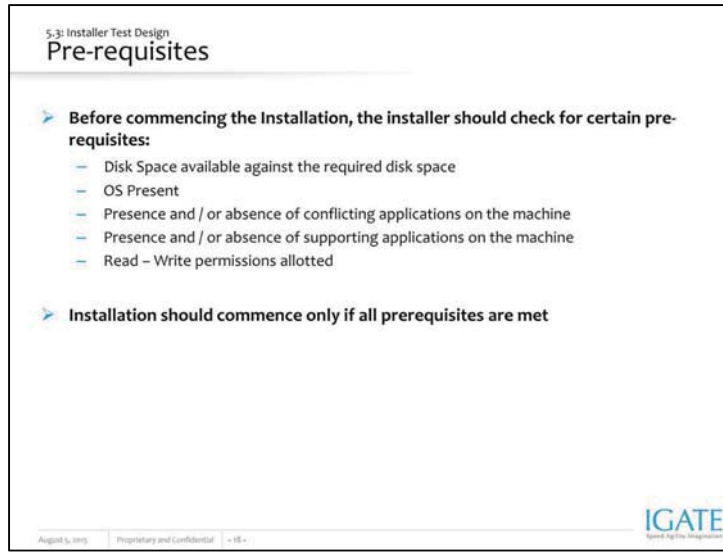
5.3: Installer Test Design

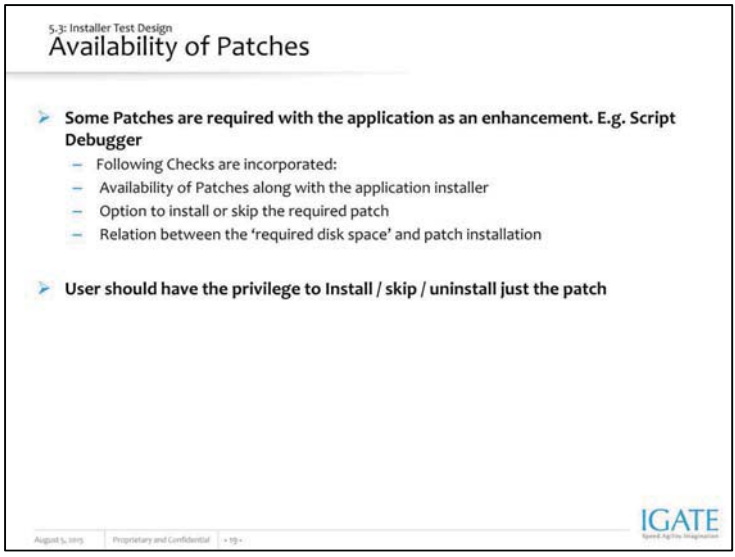
Backward Navigation

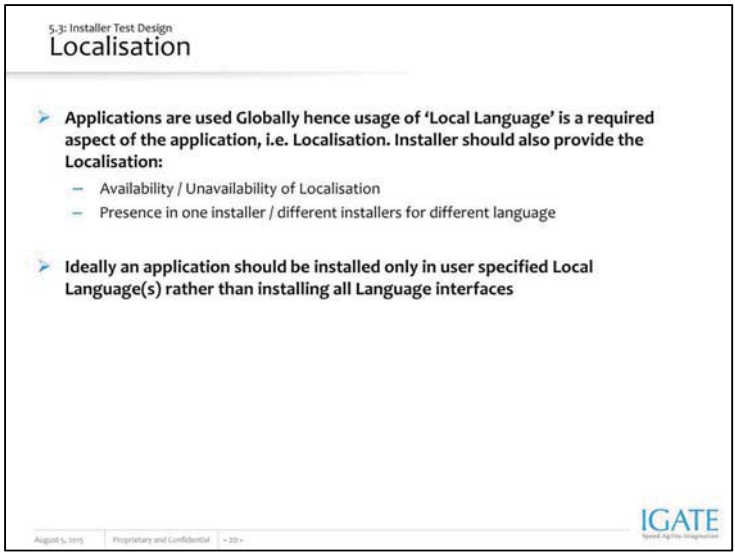
- **Ensure that the stated settings are cancelled/not cancelled/default settings are restored when the user navigates backwards (as per the specifications). Successful installation should be independent of 'default' installation sources.**

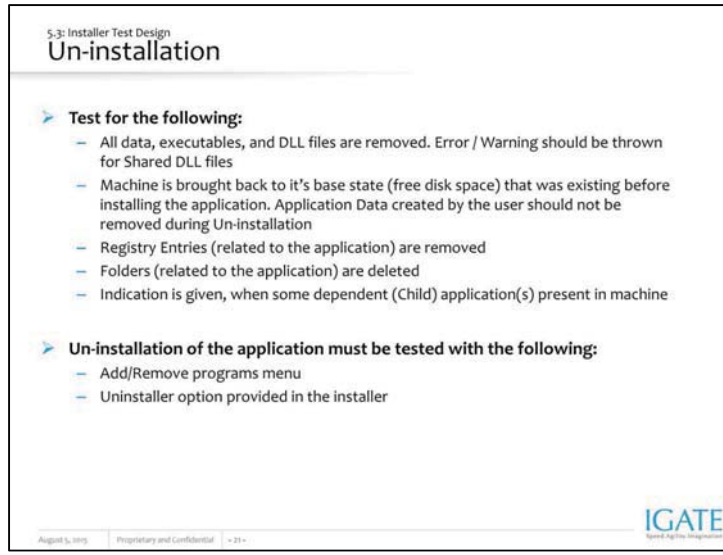




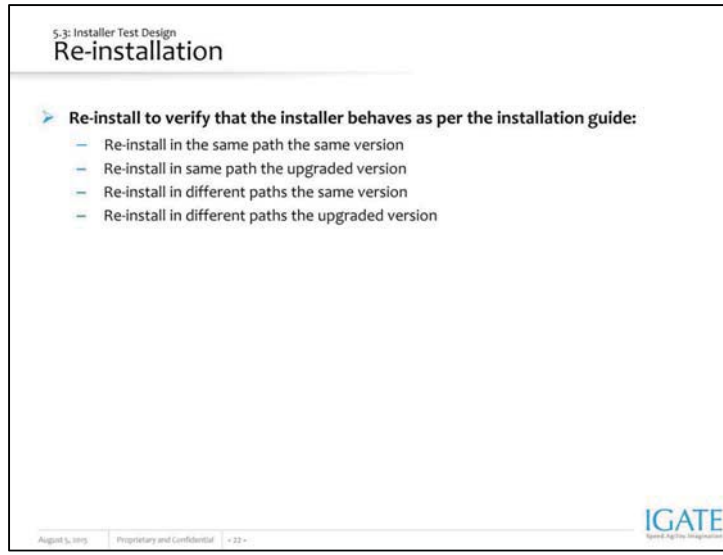

5.3: Installer Test Design
Pre-requisites
Before commencing the Installation, the installer should check for certain pre-requisites:
Disk Space available against the required disk space
OS Present
Presence and / or absence of conflicting applications on the machine
Presence and / or absence of supporting applications on the machine
Read – Write permissions allotted
Installation should commence only if all prerequisites are met
IGATE

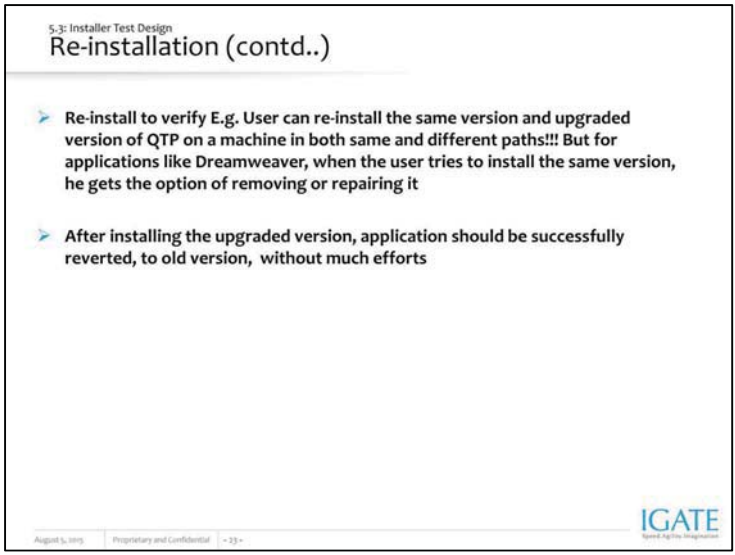

5.3: Installer Test Design
Availability of Patches
Some Patches are required with the application as an enhancement. E.g. Script Debugger
Following Checks are incorporated:
Availability of Patches along with the application installer
Option to install or skip the required patch
Relation between the 'required disk space' and patch installation
User should have the privilege to Install / skip / uninstall just the patch
IGATE

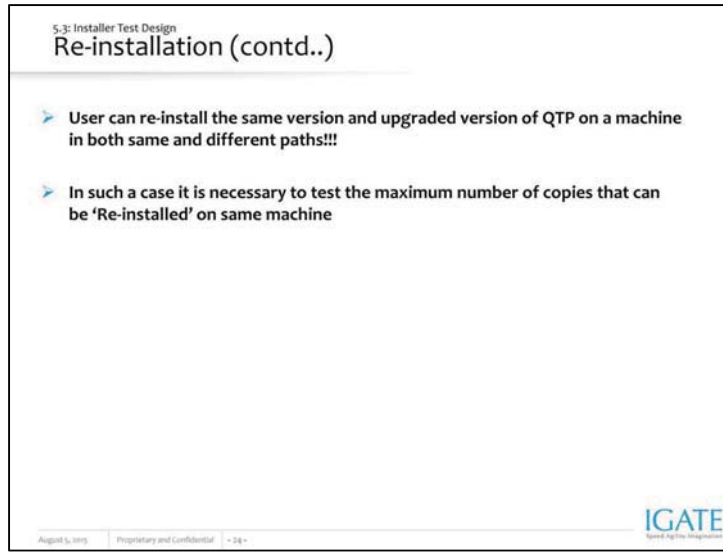

5.3: Installer Test Design
Localisation
Applications are used Globally hence usage of 'Local Language' is a required aspect of the application, i.e. Localisation. Installer should also provide the Localisation:
Availability / Unavailability of Localisation
Presence in one installer / different installers for different language
Ideally an application should be installed only in user specified Local Language(s) rather than installing all Language interfaces
IGATE

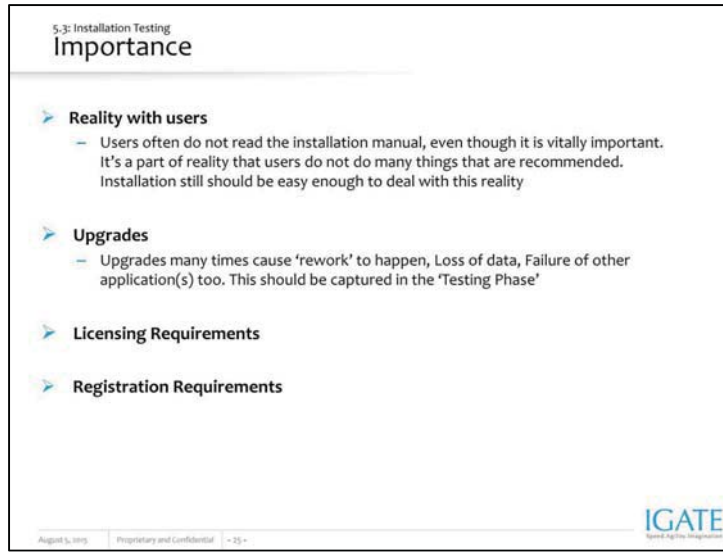


Pt 6.: If there is an application suite and we are just un-installing some application(s) from system, keeping dependent application(s); then user should get some indication before 'un-installing'.


5.3: Installer Test Design
Re-installation
Re-install to verify that the installer behaves as per the installation guide:
Re-install in the same path the same version
Re-install in same path the upgraded version
Re-install in different paths the same version
Re-install in different paths the upgraded version
IGATE


5.3: Installer Test Design
Re-installation (contd..)
Re-install to verify E.g. User can re-install the same version and upgraded version of QTP on a machine in both same and different paths!!! But for applications like Dreamweaver, when the user tries to install the same version, he gets the option of removing or repairing it
After installing the upgraded version, application should be successfully reverted, to old version, without much efforts
IGATE
August 5, 2015
Proprietary and Confidential
- 23 -


5.3: Installer Test Design
Re-installation (contd..)
User can re-install the same version and upgraded version of QTP on a machine in both same and different paths!!!
In such a case it is necessary to test the maximum number of copies that can be 'Re-installed' on same machine
IGATE
August 5, 2015
Proprietary and Confidential
- 24 -


5.3: Installation Testing
Importance
Reality with users
– Users often do not read the installation manual, even though it is vitally important. It's a part of reality that users do not do many things that are recommended. Installation still should be easy enough to deal with this reality
Upgrades
– Upgrades many times cause 'rework' to happen, Loss of data, Failure of other application(s) too. This should be captured in the 'Testing Phase'
Licensing Requirements
Registration Requirements
IGATE

5.3: Operating System (OS)

Guidelines

- Test to verify that the installer follows the OS guidelines for installation. The guidelines provide us an ideal way in which any application should be installed, un-installed & re-installed on the system.
- Installing Software on windows 2000(downloaded from the net)

Installing Software on Win 2000

- Installer Guidelines for MAC OS X (downloaded from the net)

Installer Guidelines for MAC OS X

IGATE

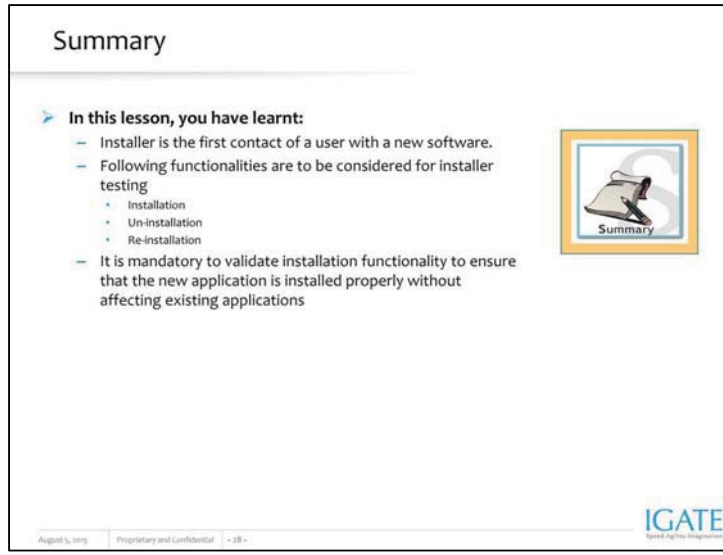
5.6: Installer Testing

Example – QTP 6.6

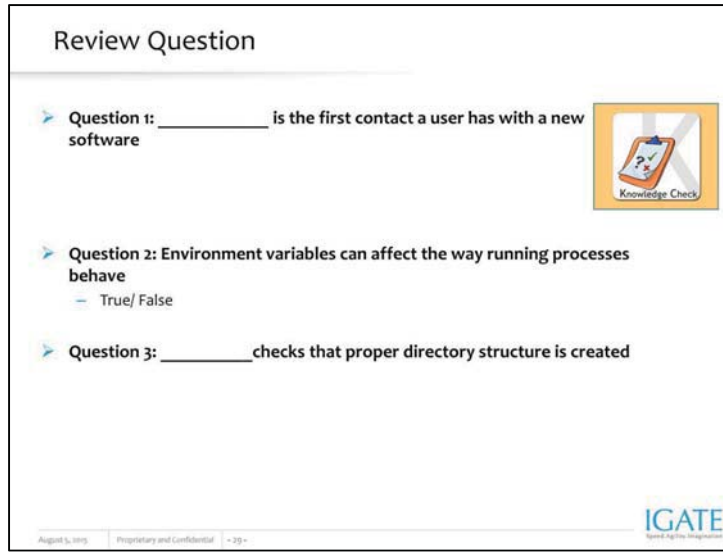
- **Testing the Installer for Quick test Pro (Ver 6.6):**

Installation Testing
- QTP

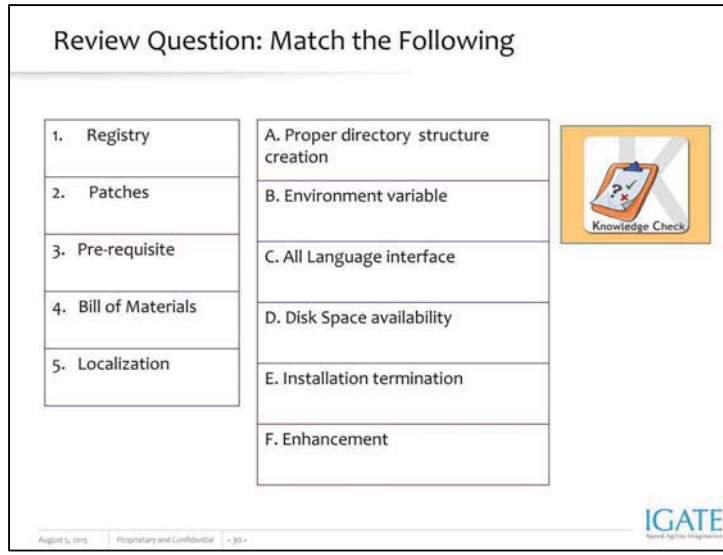
IGATE



Add the notes here.



Add the notes here.



Add the notes here.

IGATE
Speed.Agility.Imagination

V&V - Task Based Approach
V2.0
Lab Book

Document Revision History

Date	Revision No.	Author	Summary of Changes
12/8/2009	1	Priya Rane	Material Revamp
29/6/2011	1.1	Selva Lakshmi	Material Revamp
17/07/2015	2.0	Shilpa Bhosle	Material Revamp

Table of Contents

Document Revision History 2
Table of Contents 3
Case Study – Report Testing 4
Case Study – Import Export Testing, Help System Testing & Installation Testing 7
Lab 1. Task Based Approach Basics 9
Lab 2. Task Based Approach Basics 11
Form Level Errors: *12*
QC Entry Screen 13

Case Study – Report Testing

Case Study for Report testing - Ms Project Reports

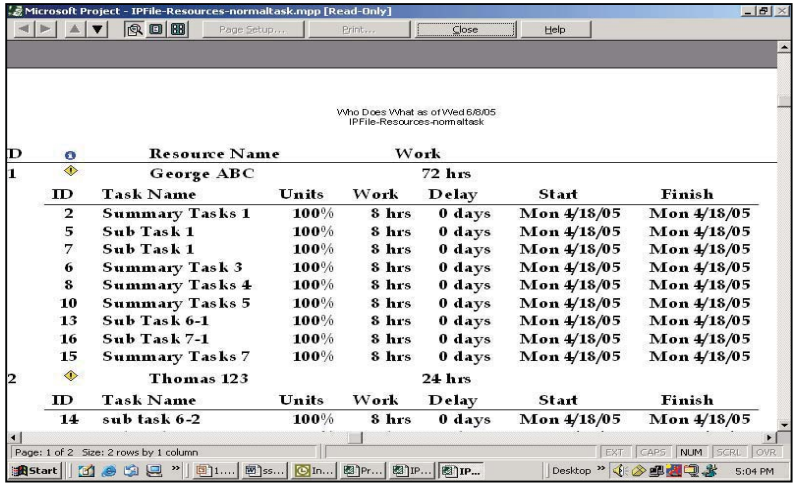
1. With Microsoft Project, Information can be viewed or printed about tasks, resources, costs, and progress in a report that suits user needs.

2. The content displayed can be changed in most reports by changing tables, filters, or both.

3. For the purpose of understanding the approach towards the testing of reports, Let us consider the following report in MS-Project ***"Who Does What Report"***

Who Does What Report?

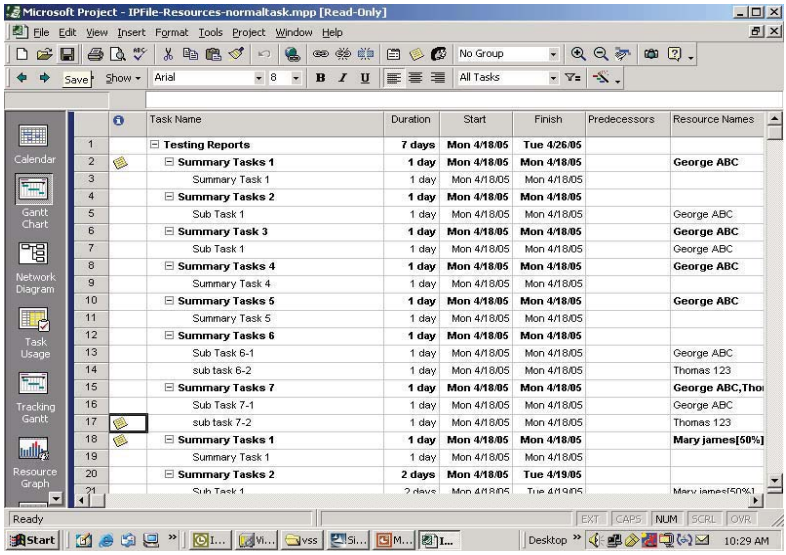
1. It gives a resource wise report.
2. Name of the resource, task name, start date, finish date, will be taken from the database or inputs.
3. Total hours of work per resource will be derived based on that start date and finish.
4. It also gives an indication if the resources need to be leveled.



Note:

1. **The Test Design Document should be discussed here**
2. **The Test Specification Document should be discussed here**

Create an input File



Testing Contents: Content can be tested as follows:

1. Assign no resource to any task and generate report
2. Assign resources and generate report
3. Rename resources and generate report
4. Unallocate resources and generate report
5. Assign multiple resources and generate report

Baseline Vs Expected

- Have a base line set and compare it with the actual result or output file that is obtained.
- This process of comparison can be automated using tools.
- The results will indicate the as to whether the comparison has passed or failed

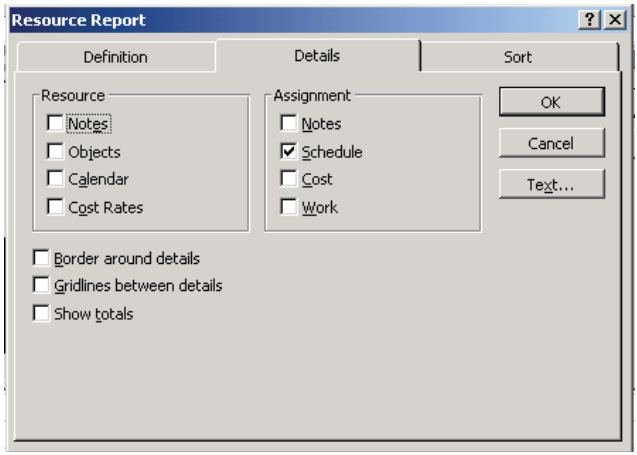
Testing Schema

The schema should be checked by inputting multiple values and check for the correctness of the schema set.
This testing should be done at the design level of the report itself.
The schema testing can be applied to the above said example as follows

1. Assign multiple resources to same task and generate report
2. Assign multiple tasks to one resource and generate report

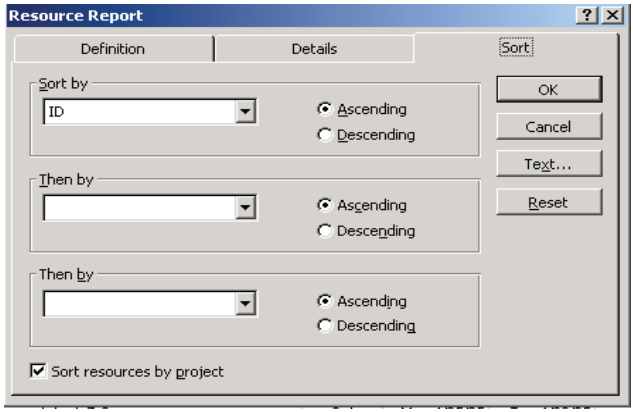
Testing Format & Layout

The format and layout of the above said report can be tested for change in the format and layout in the ways as below.



Testing User defined filters or sorting

While generating reports the user can input filters or sort orders and generate reports as required



Case Study – Import Export Testing, Help System Testing & Installation Testing

Case Study – Import Export Testing:

Example:
Import – Export: MS Project and MS Outlook.

The Import Export test cases should be discussed here:
File Name – Worksheet in ImportExport Testing.xls

Case Study – Help System testing:

Consider the help being provided by Windows Notepad.

As discussed so far in the presentation.

1. User Interface
2. Table of Contents
3. Index
4. Search
5. Contents

Approach to Testing
Inputs Required: Design Guidelines, Specifications

To test the help provided by application under test:

1. Study the help provided by the application under test.
2. Study the design guidelines are studied to form our expectations.
3. Study the Specifications provided for the application
4. Develop the test cases.

Output: Test Cases

Design Guidelines

Guidelines document would help us form the expectations for testing the help system by laying down a set of design guidelines for what to test in a Help System.
It would also specify the elements of the help system, their organization and their designing concepts.

Note – Discuss Design Guidelines here
File Name - Guidelines for a Help System.doc

Specifications
Specifications document would help us understand the features of the Help provided by the application.
It would help to form the base for comparing the expected results with and in turn for writing test cases.

Note – Discuss Specifications here
File Name - Specification For notepad Validations.doc

Test cases
The test cases for the testing the help provided by the Notepad would be created.
Discuss Test Cases validate help system for Notepad here
File Name - Worksheet in Help Validation.xls

Installation Testing - Case Study – Dreamweaver 8
Discuss Installation Test Cases here
File Name - Worksheet in Installer Testing_QTP.xls

Lab 1. Task Based Approach Basics

Goals	• Understand the process of creating Test Cases using Tasked Based Approach • Learn to apply task based approach for writing Test cases • Learn to prepare finite set of Test cases
Time	180 min

Note:

1. Refer to the Use Cases written in the Use Case Level Test Cases module to perform the lab on Task Based Approach

    a. **Login – Administrator**
    b. **Add Exam Details**
    c. **Register (New User/Applicant)**

2. The participant is expected to identify Use Case Scenario, Form Level Test Cases & Field Level Test Cases

Use Case	Form Level Validations	Field Level Validation
Login - Administrator	Form Level Validations and Use Case level validations should be identified with help of use case written in Use Case Level Test Cases Module Labs & RVFD Banking Exam Portal case study	1. All fields on the Login web page are mandatory 2. The User Name field can accept only alphabets 3. The User Name field should accept maximum 10 alphabets 4. Passwords must have at least six characters 5. Passwords must use at least three of the four available character types: lowercase letters, uppercase letters, numbers, and symbols
Add Exam Details	Form Level Validations and Use Case level validations should be identified with help of use case written in Use Case Level Test Cases Module Labs & RVFD Banking Exam Portal case study	1. All fields are mandatory on the form 2. Exam Name should accept only character Data 3. Exam Code should contain only 6 digits 4. Bank Name should accept only alphabets 5. Commencement of Online Registration field should accept only date type of value 6. Closure of Online Application should accept only date type of values 7. Exam Date field should accept only date type of values 8. Exam description field can accept alpha numeric data up to 100 characters 9. Exam description field can accept only &, *, @, #, % as a special characters

		10. Examination Fees field can accept only numeric data
Register (New User/Applicant)	Form Level Validations and Use Case level validations should be identified with help of use case written in Use Case Level Test Cases Module Labs & RVFD Banking Exam Portal case study	1. All fields are mandatory fields on the form 2. First Name, Last Name and Middle Name should contain only alphabets 3. Mobile Number should accept only 10 Digits 4. Confirm Mobile Number should accept only 10 digits 5. Email Address can accept alphanumeric values 6. Email Address can accept only one special character i.e. @ in its value 7. Email Address should accept data in a proper format of being valid email id e.g. someone@domain.com 8. The domain name can be gmail.com, yahoo.com, Hotmail.com in Email Address 9. Date of Birth should be a valid date 10. Gender can be selected as Male or Female 11. Marital Status can be selected as Married or Unmarried 12. Graduation/Equivalent passes can contain values as Graduate/ Post Graduate 13. Degree/Stream/Subject can contain the values like Commerce, Arts, Science 14. Year of Passing should accept only valid date value 15. % of Marks should accept only numeric or decimal data 16. The signature file size should be between 50 kb to 100 kb 17. The image file size should not be more than 150 kb

Lab 2. Task Based Approach Basics

Goals	• Understand the process of creating Test Cases using Tasked Based Approach for Defect Entry • Learn to apply task based approach for writing Test cases • Learn to prepare finite set of Test cases
Time	180 min

To Do - With the help of a below given Use Case for Defect Entry Screen complete this Lab

1. **Use Case Name –** Add Defect
2. **Use Case ID –** 01
3. **Scenario -** Adding a new defect for a project in PMS
4. **Description -** Add a new defect against a task
5. **Primary Actor(s) -** Employee assigned to the project
6. **Pre-Conditions -**
    a. The project is created in PMS.
    b. The user must be assigned to the project.
    c. The project's activities must be defined.
    d. The project's stages must be customized.
    e. The project task structure must be created.
7. **Basic Flow –**
    a. User selects the option to add a new defect.
    b. User specifies valid values for mandatory details:
        a. Reported Date, Reported By, Stage Detected.
        b. Task (L1, L2, L3, L4 Tasks) for which the defect is to be recorded.
        c. Type of Review / Testing.
        d. Defect Count, Defect Impact, Defect Severity, Defect Description.
        e. Specify the Status as 'Active' for the new defect.
    c. User saves the defect.
8. **Post-Conditions –**
    The defect gets saved under the project Defects List and a unique sequential Defect
    number is generated.
9. **Alternate Flow 1 -** User specifies valid values for mandatory details.
    a. User specifies all the valid optional details.
        1. Build detail
        2. Assigned To
        3. Error Category
        4. Root Cause
        5. Reason
        6. Defect source
        7. Defect priority
    b. User saves the defect.

Result - Defect gets saved and a unique sequential defect number is generated.

10. **Exceptional Flow 1** - Invalid User
    a. User not assigned to Project
    b. User selects the option to add a new defect.

Post-Conditions - Error-1

11. **Exceptional Flow 2** - Task Structure not created for the project
    a. User selects the option to add a new defect.

Post-Conditions - Error-2

12. **Exceptional Flow 3** - Project's stages are not customized
    a. User selects the option to add a new defect.

Post-Conditions: **Error-3**

Error / Warning Message Table
(Defect Entry Screen)

Use Case Level Errors:

1	Invalid User (User not assigned to this project)
2	Project Tasks are not created
3	Project's stages are not customized

Form Level Errors:

1.	Reported Date cannot be less than the project start date.
2.	Reported Date cannot be greater than the project end date.
3.	Reported Date cannot be less than the Actual start date of the Task*.
4.	Reported Date cannot be greater than the Actual end date of the Task*.
5.	Reported Date cannot be during On-Hold period for Task*.
6.	You cannot log a Fixed or Verified defect for the first time.

Field Level Errors:

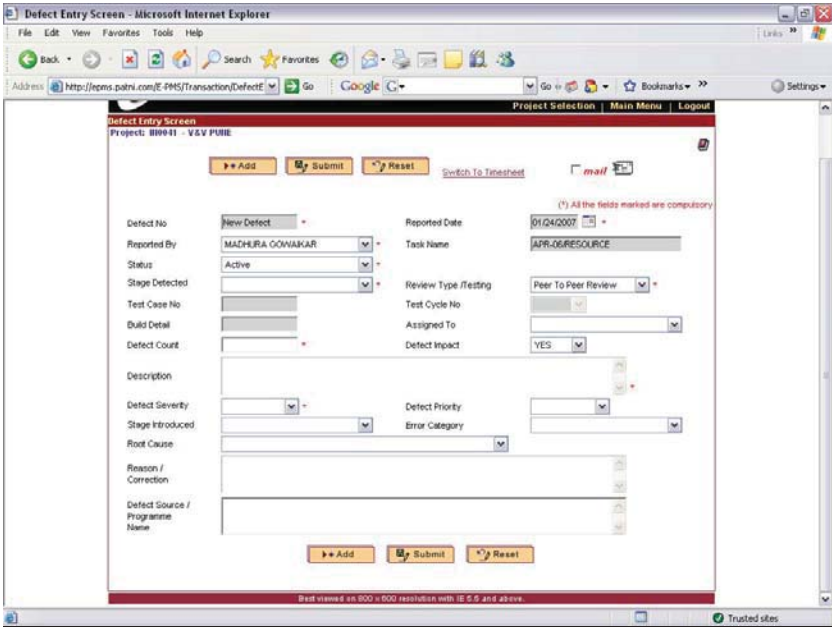
1.	Reported Date cannot be blank.
2.	Reported Date cannot be future date.
3.	L1 Task cannot be blank.
4.	L2 Task cannot be blank.
5.	L3 Task cannot be blank.
6.	L4 Task cannot be blank.
7.	Stage Detected cannot be blank.
8.	Test Case No. cannot be blank.
9.	*(Warning)* Test Cycle No. is blank. Do you want to continue?
10.	Test Cycle No. cannot be 0.
11.	Defect count cannot be 0 or Null, if there is a defect.
12.	Defect Description cannot be blank.
13.	Status cannot be blank.
14.	Verified Date cannot be blank.
15.	Verified Date cannot be future date.
16.	Verified Date cannot be less than the Reported Date.
17.	Defect Severity cannot be blank.

18.	Assigned To cannot be blank.
19.	Stage Introduced cannot be blank.
20.	Error Category cannot be blank.
21.	Root Cause cannot be blank.

* Task would be the selected lowermost task.
E.g. - If the task structure consists of levels L1, L2, L3, then the lowermost task would be L3.
i.e. L3 Task <task_name>

QC Entry Screen

Note: The Mandatory fields have been grayed out for easy understanding.



Form Level Rules – Defect Entry Screen

(Add a new defect)

Before come up with the Defect entry form, the manager has to do the entry for the project first. Consider that manager had done the entry for the project ABC with the subtasks like GUI testing, Rules testing, level wise testing.

GUI testing:

Contains GUI validation, Layout, Color scheme testing. Once these three testing is over, the GUI testing will be over.

Level wise testing:

Total levels are 4. L1, L2, L3, L4.

L1 is the first level task. Every first level task will be divided into second level task (L2). Every second level task is divided into third level task (L3). Every third level task will be divided into fourth level task (L4). This fourth level task is the final level task.

In the QC screen, which is given, the task name is L4 task. Assume that in this form, there are 4 text boxes, one for L1, one for L2, one for L3 and another one for L4.

For example, after selecting the country (L1), it should show the states list based on the selected country. After selecting the state (L2), it should show the city list based on the state selected. Finally select the city (L3).

Rules Testing:

Rule 1: Reported Date of the defect has to be within the project start date to project end date (both inclusive) duration. Project start and end date will be enter by the project manager, while adding the project.

Rule 2: Reported Date of the defect has to be within actual start date to actual end date (both inclusive) duration of the selected lowermost level Task.

Rule 3: Reported Date cannot be within On-Hold start date to On-Hold end date (both inclusive) duration of the selected lowermost level Task. Assume that GUI testing is going on, and customer wants to do some changes in the GUI. That times no need to continue the testing in GUI. The task will be in hold. During this on hold period, we cannot produce any bug. But for other functionality, we can log the defect.

Rule 4: Reported By, Assigned To fields are populated with the employees assigned to the project. Have to write the test cases for the valid values.

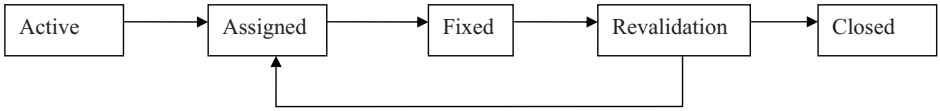
Rule 5: L1, L2, L3, L4 Task fields are populated according to the Task structure defined in the *Task Entry* Screen.

Rule 6: Stage Detected is populated based on the project stages selected in the *Customize Project Stages* screen.

Rule 7: Stage Introduced is populated based on the project stages selected in the *Customize Project Stages* screen.

Rule 8: While adding a new defect the Status cannot be 'Verified'.

Life cycle of the defect status:



Active: The entry of the new defect would be active.

Assigned: After entry was happened, the defect will be assigned to someone in the development team.

Fixed: Development team has corrected the defect, this would be fixed.

Revalidation: After fixing the defect by the development team, come for tester for re-validation. If the tester found any defects now, again it will be assigned to someone in the development team to correct it.

Closed: If no defects are there after the revalidation, the defects will be closed.

Field Level Specification – Defect Entry Screen
(Add a new defect)

Field	Control Box	Data Type	Mandatory	Length	Range	Defaults	Dependency	Validation Stage	Error / Warning (Refer the Error / Warning Message Table)	Context sensitive Help
Defect No.	Text Field- Non Editable	Integer	Yes	Auto Generated Field						
Reported Date	Spin box	Date	Yes	MM: 2 DD: 2 YY: 4	Selected lowermost task start date - Selected lowermost task end date	Current date	None	On submit	Errors 1, 2	No
Reported By	List box	None	Yes	None	Names of employees assigned to the project, CLIENT, OSC, QACELL, SQA, DFD, V&V	Employee logged on to PMS	None	None	None	No

Field	Control Box	Data Type	Mandatory	Length	Range	Defaults	Dependency	Validation Stage	Error / Warning (Refer the Error / Warning Message Table)	Context sensitive Help
L1 Task	List box	None	Yes	None	As per the task structure for the project	None	None	On submit	Error 3	No
L2 Task	List box	None	Yes	None	As per the task structure for the project	None	L1 task. (Enabled on after L1 task is selected)	On submit	Error 4	No
L3 Task	List box	None	Depends on task structure for the project	None	As per the task structure for the project	None	L2 task. (Enabled on after L2 task is selected)	On submit	Error 5	No
L4 Task	List box	None	Depends on task structure for the project	None	As per the task structure for the project	None	L3 task (Enabled on after L3 task is selected)	On submit	Error 6	No
Stage detected	List box	None	Yes	None	Depends on the stage detected list of the project	None	None	On submit	Error 7	No

Field	Control Box	Data Type	Mandatory	Length	Range	Defaults	Dependency	Validation Stage	Error / Warning (Refer the Error / Warning Message Table)	Context sensitive Help
Type of Review / Testing	List box	None	Yes	None	Peer Review, Peer to Peer Review, Self/Programmer Review, 10% Quality Probe (Review)	Peer to Peer Review	None	None	None	No
Testcase No	Text field	Alphanumeric, Spl. Char.	Depends on the selected Stage Detected	1-10	Alphanumeric, Spl. Char.	None	Mandatory if *Stage detected* = Unit Testing AND *Type of Review / Testing* = Using Test Cases	On submit	Error 8	No
Test Cycle No	Spin box	Integer	No	1-2	1-99	None	Validated only if *Stage detected* = Unit Testing AND *Type of Review / Testing* = Using Test Cases	On submit	Errors 9(warning), 10	No

Field	Control Box	Data Type	Mandatory	Length	Range	Defaults	Dependency	Validation Stage	Error / Warning (Refer the Error / Warning Message Table)	Context sensitive Help
Build Detail	Text field	Alphanumeric, Spl. Char.	No	0-16	Alphanumeric, Spl. Char.	None	None	None	None	No
Defect Count	Text field	Integer	Yes	1-5	1-32767	1	None	On submit	Error 11	No
Defect Impact	List box	None	Yes	None	Yes No	Yes	None	None	None	No
Defect description	Text area	Alphanumeric, Spl. Char.	Yes	1-255	Alphanumeric, Spl. Char.	None	None	On submit	Error 12	No

Field	Control Box	Data Type	Mandatory	Length	Range	Defaults	Dependency	Validation Stage	Error / Warning (Refer the Error / Warning Message Table)	Context sensitive Help
Status	List box	None	Yes	None	Active, Could not be repeated, Fixed, No defects found, Need not be fixed, Reported bug not a defect, Verified	Active	None	On submit	Error 13	No
Verified Date	Spin box	Date	Yes if Status = Verified	MM: 2 DD: 2 YY: 4	Reported Date to Current Date	Current Date	Verified Date must be greater than Reported Date	On submit	Errors 14, 15, 16	
Defect severity	List box	None	Yes	None	High, Medium, Low	None	None	On submit	Error 17	No
Defect Priority	List box	None	No	None	High, Low, Medium	None	None	None	None	No

Field	Control Box	Data Type	Mandatory	Length	Range	Defaults	Dependency	Validation Stage	Error / Warning (Refer the Error / Warning Message Table)	Context sensitive Help
Assigned to	List box	None	Depends on the selected Status.	None	Names of employees assigned to the project	None	Mandatory for *Status* = Fixed or Verified	On submit	Error 18	No
Stage Introduced	List box	None	Depends on the selected Status.	None	Depends on the Stage Introduced list of the project	None	Mandatory for *Status* = Fixed or Verified	On submit	Error 19	No
Error Category	List box	None	Depends on the selected Status.	None	List of predefined error categories	None	Mandatory for *Status* = Fixed or Verified	On submit	Error 20	No
Root Cause	List box	None	Depends on the selected Status.	None	List of 31 root causes	None	Mandatory for *Status* = Fixed or Verified	On submit	Error 21	No
Reason & Correction	Text area	Alphanumeric, Spl. Char.	No	0-300	Alphanumeric, Spl. Char.	None	None	None	None	No

Field	Control Box	Data Type	Mandatory	Length	Range	Defaults	Dependency	Validation Stage	Error / Warning (Refer the Error / Warning Message Table)	Context sensitive Help
Defect Source & programme Name	Text field	Alphanumeric, Spl. Char.	No	0-100	Alphanumeric, Spl. Char.	None	None	None	None	No