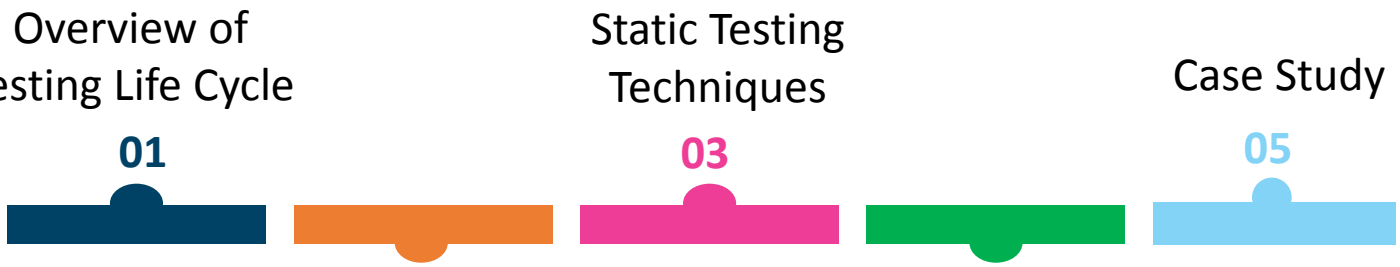
Brighter Blue

Basic Testing: Day 1

1. Overview of Testing Lifecycle
2. Types of Testing
3. Static testing techniques
4. Testing Design Techniques and Approach
5. Case Study

IBM

Agenda



01 Overview of Testing Life Cycle

02 Types and Levels of Testing

03 Static Testing Techniques

04 Testing Design Techniques and Approach

05 Case Study

© Copyright IBM Corporation 2015

Brighter Blue

01 Overview of Testing Lifecycle

IBM

What is a **test**?

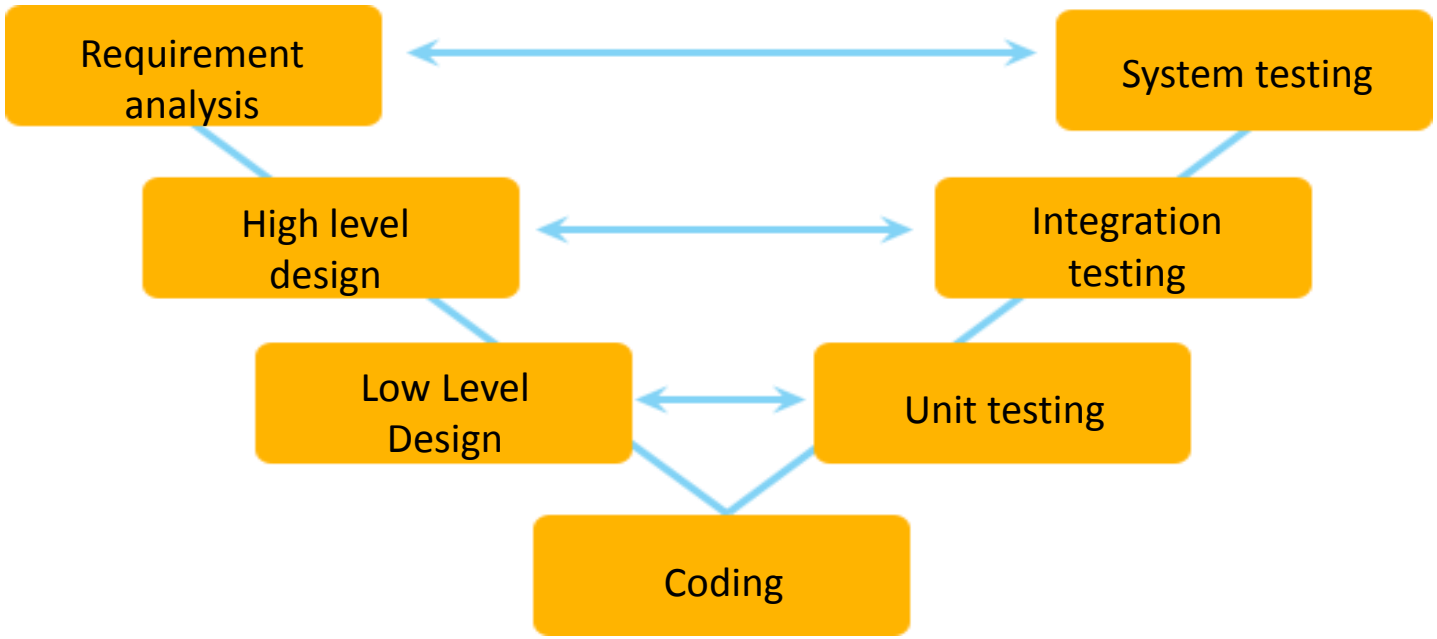
Test

A test is an activity in which a system or component is **executed under special conditions**, the **results are observed or recorded**, and an **evaluation is made** of some aspects of the system or component.

© Copyright IBM Corporation 2015

Where testing fits in SDLC

- The left side of the model is Software Development Life Cycle or SDLC
- The right side of the model is Software Test Life Cycle or STLC
- Each phase comprises independent set of development and testing activities
- A good example of development lifecycles following an iterative method can be an Agile Development



© Copyright IBM Corporation 2015

SDLC and Verification and validation model

The different phases of Software Development Cycle are:

- Requirement gathering stage: In this stage one needs to gather as much information as possible about the details and specifications of the desired software from the client.
- Design stage plan: In this stage programming languages like Java, PHP, .NET, and database like Oracle, MySQL, and so on that would be suited for the project, also high-level functions and architecture.
- Built stage: After design stage, it is built stage, that is nothing but actually code the software.
- Deployment stage: In this stage one needs to deploy the application in the respective environment.
- Testing stage: In this stage software testing is done.
- Maintenance stage: Once your system is ready to use, you may require to change the code later on as per customer request.

© Copyright IBM Corporation 2015

SDLC and Verification and validation model (continued)

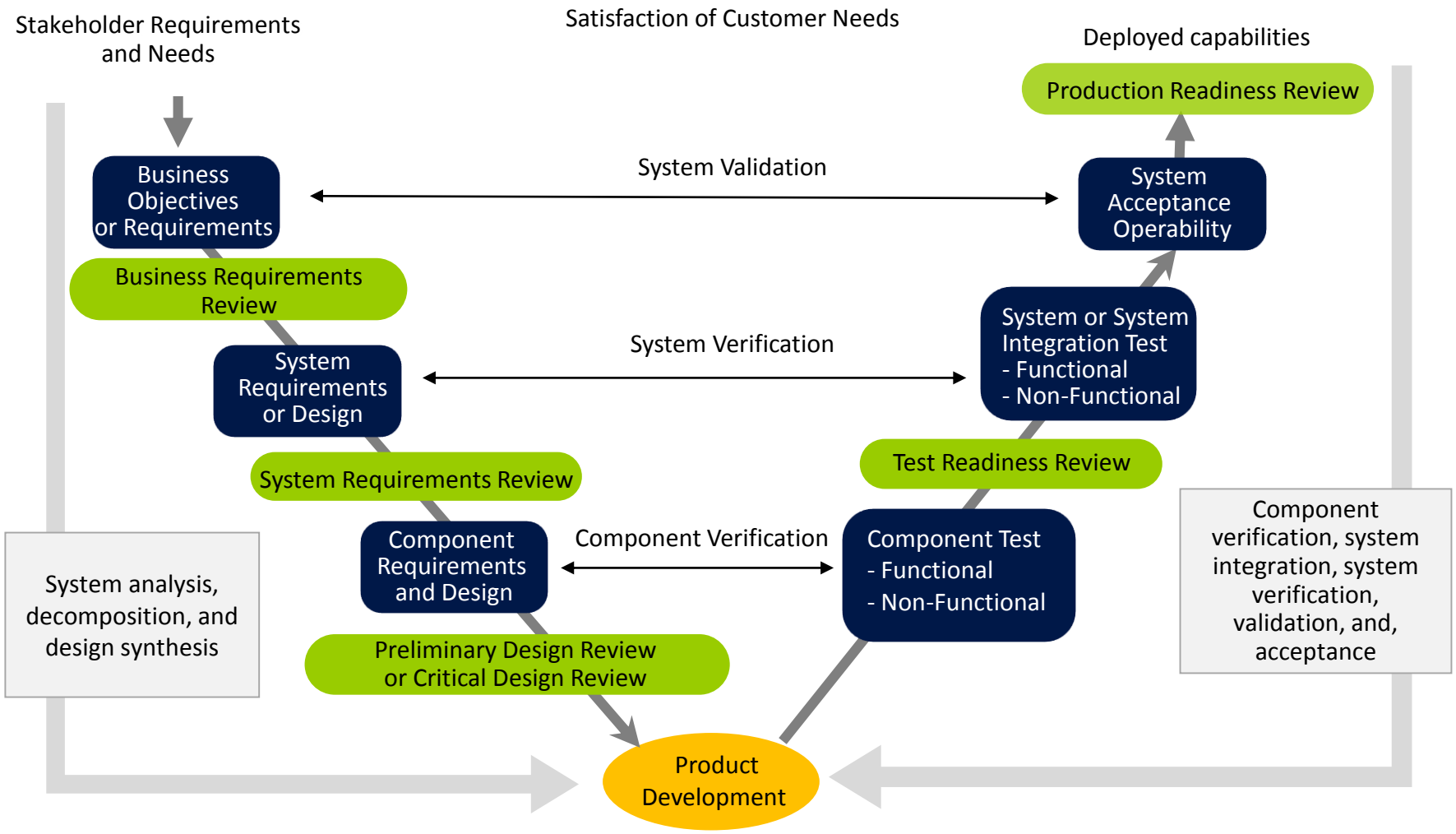
- Deployment stage: In this stage one needs to deploy the application in the respective environment.
- Testing stage: In this stage software testing is done.
- Maintenance stage: Once your system is ready to use, you may require to change the code later on as per customer request.

Conclusion:

- Testing is not a stand-alone activity, and it has to adapt the development model chosen for the project.
- In any model, testing should performed at all levels i.e. right from requirements until maintenance.

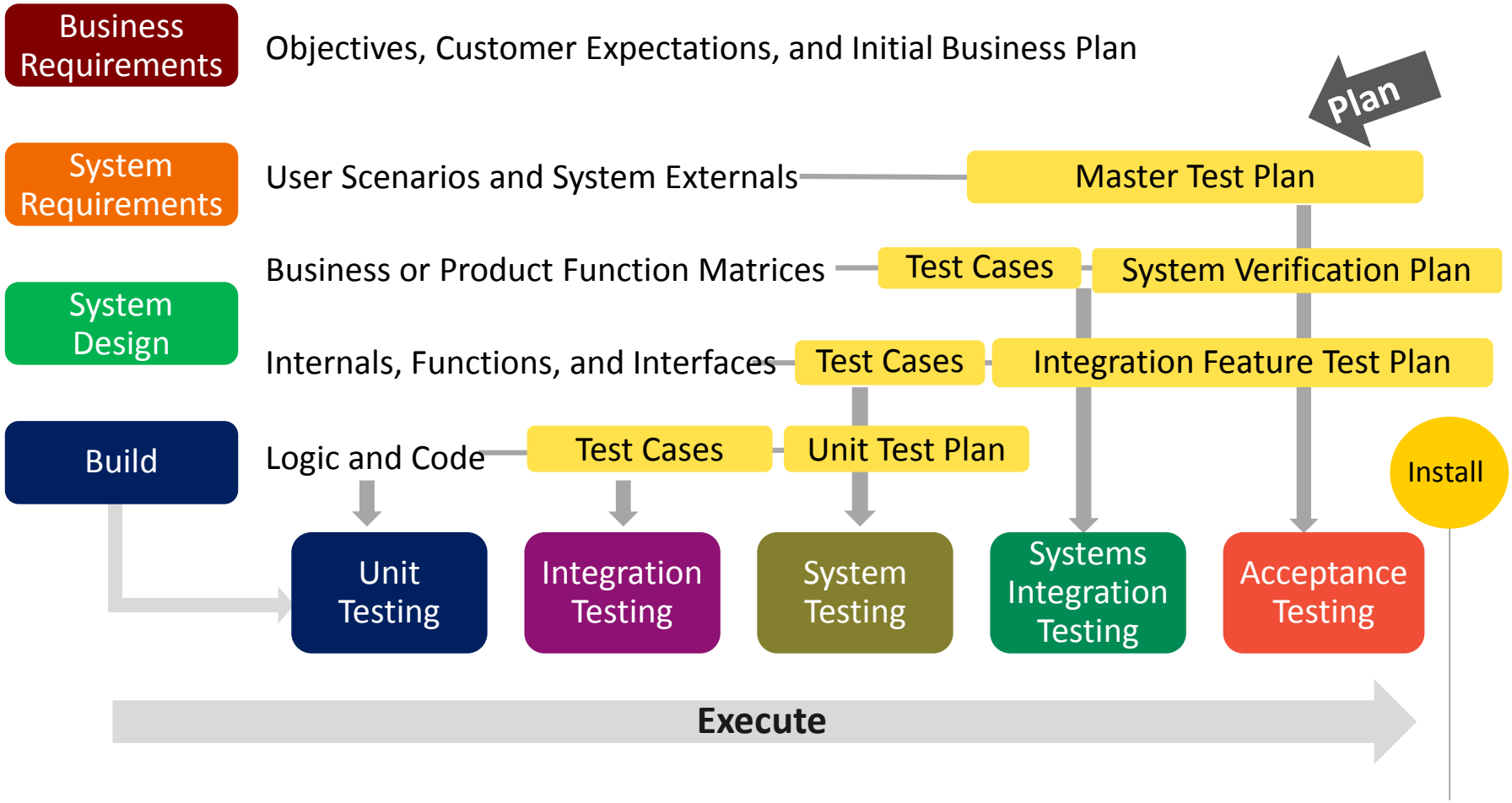
© Copyright IBM Corporation 2015

Verification and validation model



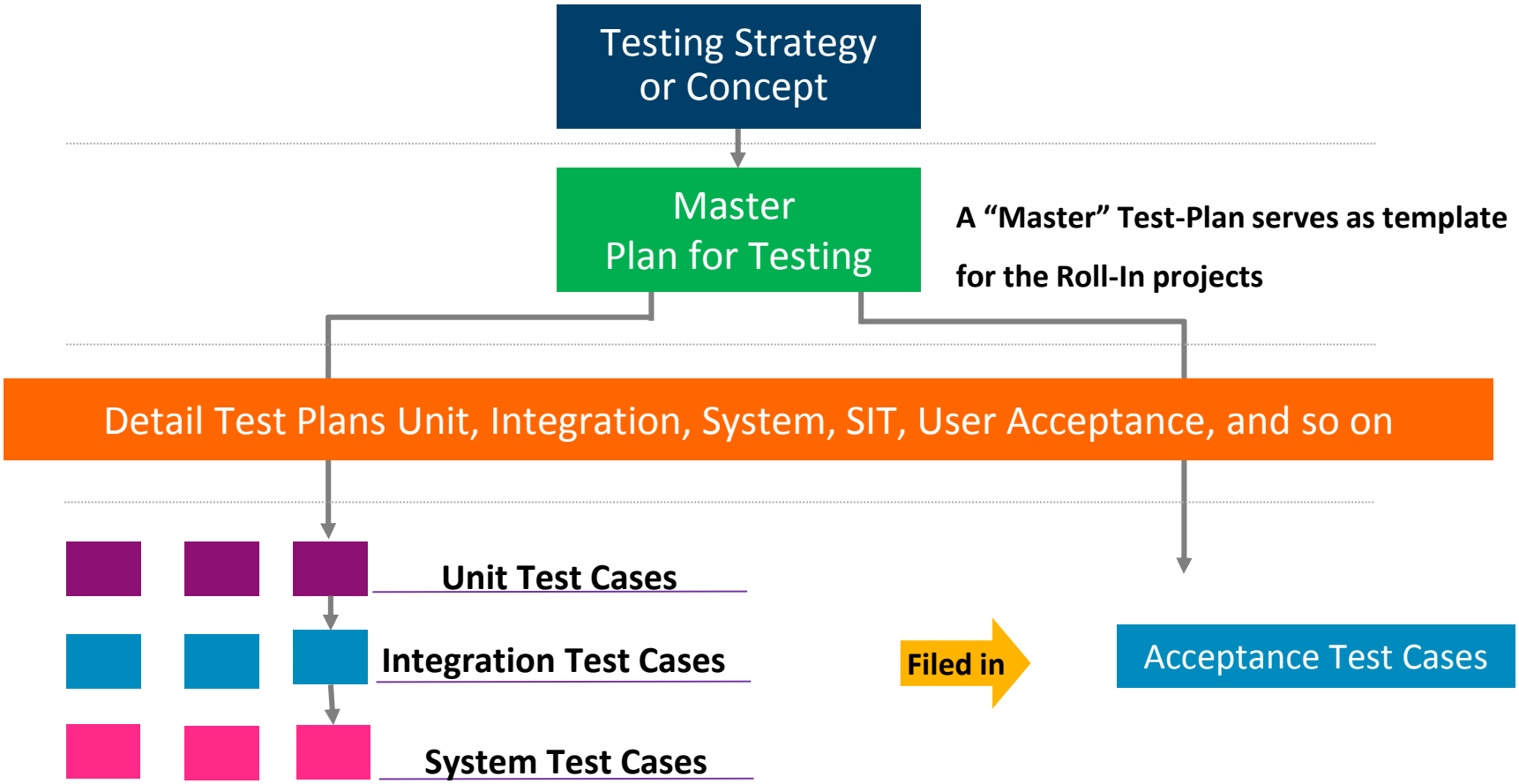
© Copyright IBM Corporation 2015

Mapping of project phase to test deliverables to testing activity



© Copyright IBM Corporation 2015

Documentation hierarchy



© Copyright IBM Corporation 2015

Approach and processes

Manual Testing Approach

There are four key steps for each testing category to successful testing

- Planning the test
- Preparing the test
- Test execution
- Test analysis and results

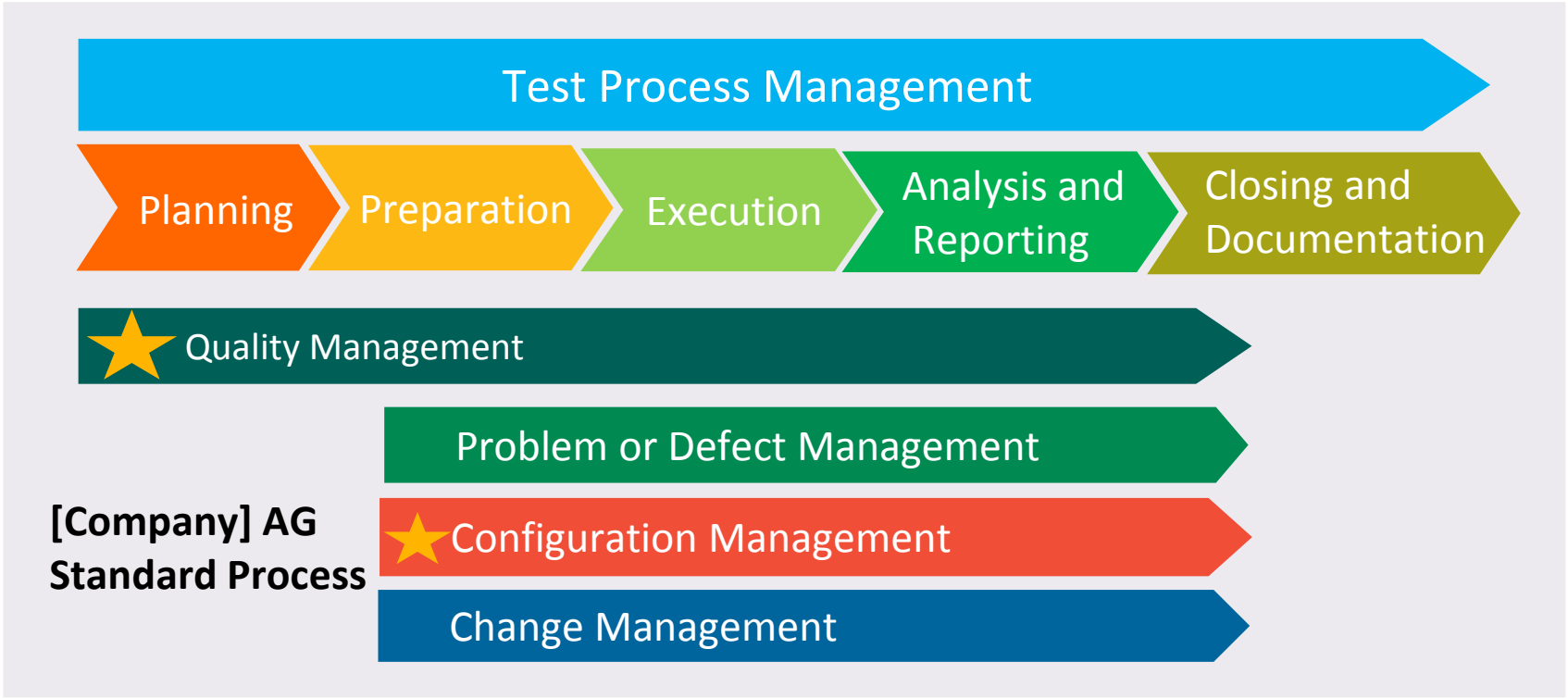
These are additional steps

- Reporting the test status
- Test documentation

Test team member may be involved in one or more of these steps during the project lifecycle. As a result, all team members should have an overall awareness of each of these steps and the responsibilities of the individuals involved.

© Copyright IBM Corporation 2015

Approach and processes (continued)



© Copyright IBM Corporation 2015

Main steps / activities in testing

Brighter Blue

- Building up the test team
- Planning testing activities
- Creation of test data
- Requirement traceability matrix
- Review the test cases and test data with client
- Development of test cases
- Test readiness review
- Execute testing
- Analysis
- Reporting
- Closing and documentation

© Copyright IBM Corporation 2015

Testing lifecycle team structure

What is a testing team?

An effective testing team includes a combination of members with extensive:

- Testing expertise
- Tools expertise
- Database expertise
- Domain or technology expertise

© Copyright IBM Corporation 2015

Activity

Brighter Blue

Remember and recite the testing lifecycle term definitions given in the lists provided below:

Testing Lifecycle Term Definitions- A

Testing Lifecycle Term Definitions- E

Testing Lifecycle Term Definitions- C

Testing Lifecycle Term Definitions- D

© Copyright IBM Corporation 2015

© Copyright IBM Corporation 2015

Spot Quiz Sample

Brighter Blue

01

Software testing activities should start:

A	As soon as the code is written
B	During the design stage
C	When the requirements have been formally documented
D	As soon as possible in the development lifecycle

© Copyright IBM Corporation 2015

Test planning

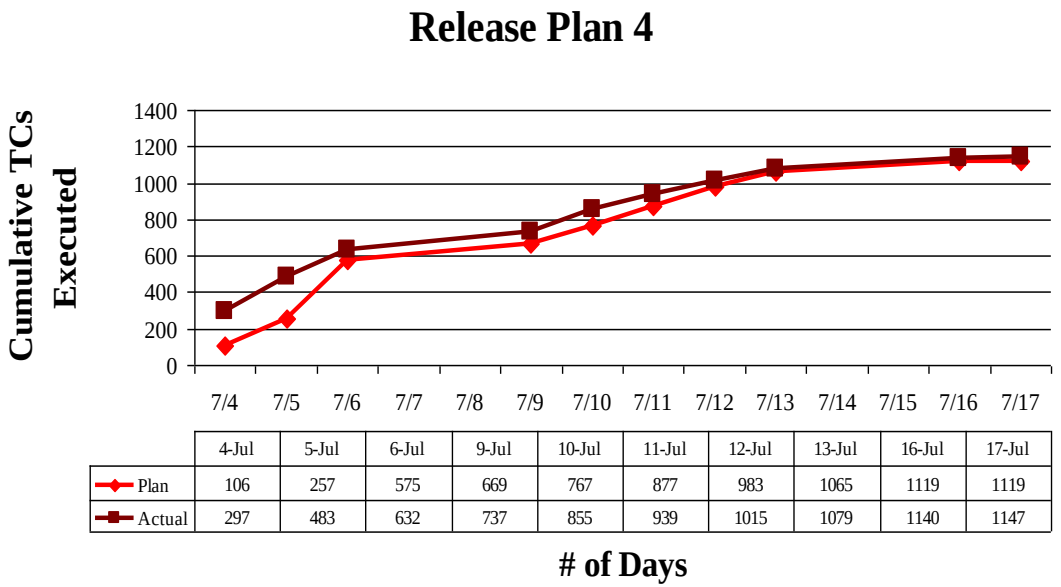
Typical test plan includes:

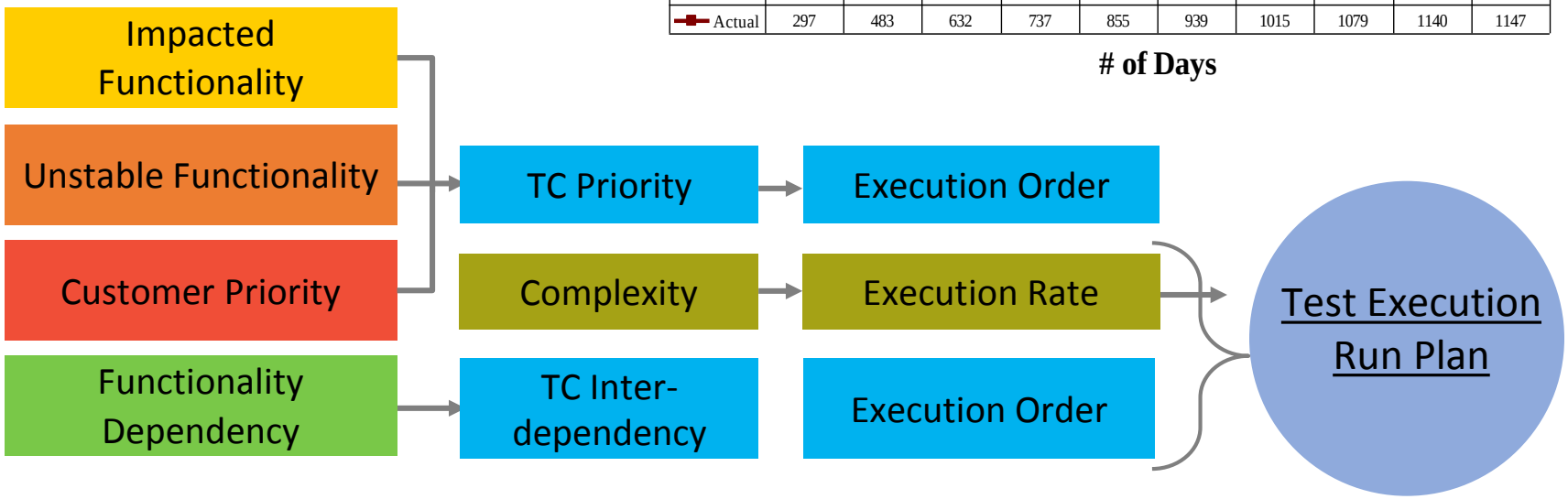
- Test plan identifier
- Introduction
- Test items
- Features to be tested
- Features not to be tested
- Approach
- Testing strategy
- Item pass or fail criteria
- Suspension criteria and resumption criteria
- Test deliverable
- Automation scope
- Testing tasks
- Effort estimation
- Environmental needs
- Roles and responsibilities

© Copyright IBM Corporation 2015

Test planning: **Test run plan**

The execution order of test cases depends on many factors and is shown as below:





© Copyright IBM Corporation 2015

Test planning: **Test run plan (continued)**

Pre conditions :

- Requirements baselines
- Project Plan finalized
- Test items
- Features to be tested
- Features

© Copyright IBM Corporation 2015

Test strategy

It is a statement of overall approach of testing to meet the business and test objectives.

It is a plan level document and has to be prepared in the requirement stage of the project.

It identifies the **methods, techniques, and tools** to be used for testing.

It can be a project specific or an organization specific.

It is critical to the success of the software development to develop a test strategy, which effectively meets the needs of the organization or project.

It must be effective enough to meet the project and business objectives.

It should define the strategy upfront before the actual testing helps in planning the test activities.

© Copyright IBM Corporation 2015

Elements of test strategy

A test strategy will typically cover the following aspects:

- Definition of test objective
- Strategy to meet the specified objective
- Overall testing approach
- Test environment
- Test automation requirements
- Metric plan
- Risk identification, mitigation, and contingency plan
- Details of tools usage
- Specific document templates used in testing

© Copyright IBM Corporation 2015

Top-down testing (1 of 3)

Top-Down testing

In this approach, testing is conducted from main module to sub module. If the sub module is not developed, a temporary program called Stub is used for simulate the sub module.

© Copyright IBM Corporation 2015

Top-down testing (2 of 3)

Advantages

- Top-down testing is advantageous if major flaws occur toward the top of the program.
- Once the I/O functions are added, representation of test cases is easier.
- Early skeletal Program allows demonstrations and boosts morale.

© Copyright IBM Corporation 2015

Top-down testing (3 of 3)

Disadvantages

- Stub modules must be produced.
- Stub modules are often more complicated than they first appear to be.
- Before the I/O functions are added, representation of test cases in stubs can be difficult.
- Test conditions may be impossible, or very difficult, to create.
- Observation of test output is more difficult
- Top-down testing allows one to think that design and testing can be overlapped.
- Top-down testing induces one to defer completion of the testing of certain modules.

© Copyright IBM Corporation 2015

Testing strategy example

User acceptance testing will occur in two cycles. Testing starts with testing of smaller units or transactions to more complex business scenario testing	▪ **Cycle One** – includes scenarios with all basic conditions and typically uses manually created data. Some converted master data, for example, G or L accounts and cost centers, will be used. ▪ **Cycle Two** – will have added complexity or specific variations and typically uses manually created data. It also includes the testing of interfaces, enhancements, reports and forms, and bolt-ons, such as Sabrix tax package.
Single testing repository	Mercury Quality Center (QC)
Testing sign-off	▪ Test Artifacts: • Test Plan, Test Strategy, and Test Execution Plan • Test Cases and Test Data • Execution Results and Defect Summary Report ▪ Test Metrics: Time, Test coverage, Cost, Software Quality, and Testing Objectives (requirements that are covered) ▪ Reports: Test Summary Report, Causal Analysis Report, and Audit Report

© Copyright IBM Corporation 2015

Test strategy: Key points

Test strategy is a high level document, which states in detail how testing is to be carried out.

It is done in line to meet the test and business objectives.

It states the tools, methods, and techniques to be used.

© Copyright IBM Corporation 2015

Test case design: **key aspects**

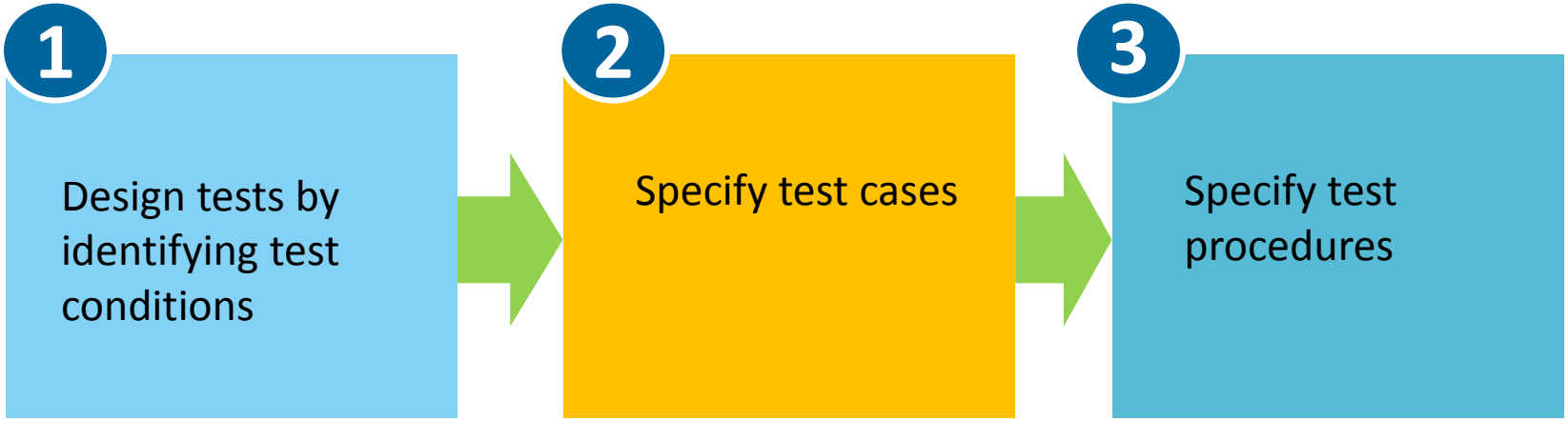
We will take a look at the following key aspects of test case design in the following slides.

1. How to identify test conditions and design test cases?
2. What are the different categories of test design techniques?

© Copyright IBM Corporation 2015

How to identify **test conditions** and design **test cases**?

The high level process to identify test conditions and design test cases is shown below.



© Copyright IBM Corporation 2015

How to identify **test conditions** and design **test cases**? (continued)

Test condition	An item or event that could be verified by one or more test cases (for example, a function, transaction, quality characteristic, or structural element)
Test cases and test data	Developed and described in detail by using test design techniques
Expected results	Produced as part of the specification of a test case and include outputs, changes to data and states, and any other consequences of the test
Test procedure (manual test script)	Specifies the sequence of action for the execution of a test

© Copyright IBM Corporation 2015

Checking test case quality

Main checkpoints

- [ ] Whether test cases covers all features
- [ ] Balance between normal, abnormal, boundary, and environmental test cases
- [ ] Balance between Black Box and White Box testing
- [ ] Balance between functional tests and performance tests

© Copyright IBM Corporation 2015

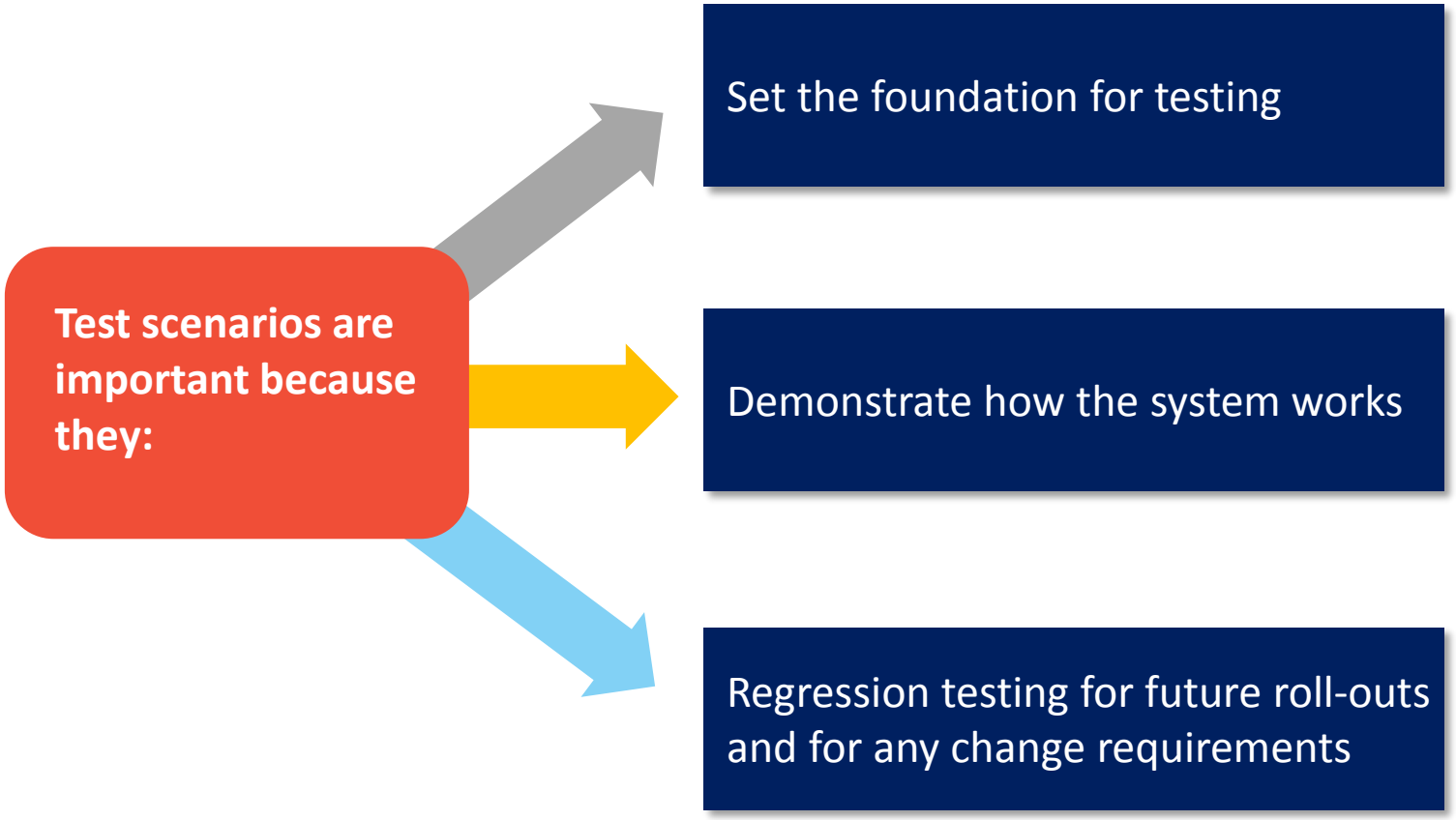
What are Test Scenarios?

Test scenarios are the processes to test, a manual non-PO invoice against a cost center. The testing team and expert users brainstormed test scenarios to test and documented them in a test matrix. Key points considered:

- Master data requirements such as vendors, cost centers, and so on
- Converted data that may be required, for example, assets, cost centers, G or L accounts, and so on
- Complexity of the test scenario to determine in, which cycle the test should be executed
- Determining which interfaces, enhancements, reports, or forms are incorporated into each scenario
- Whether a variation changes a step or the entire process that may require a new scenario

© Copyright IBM Corporation 2015

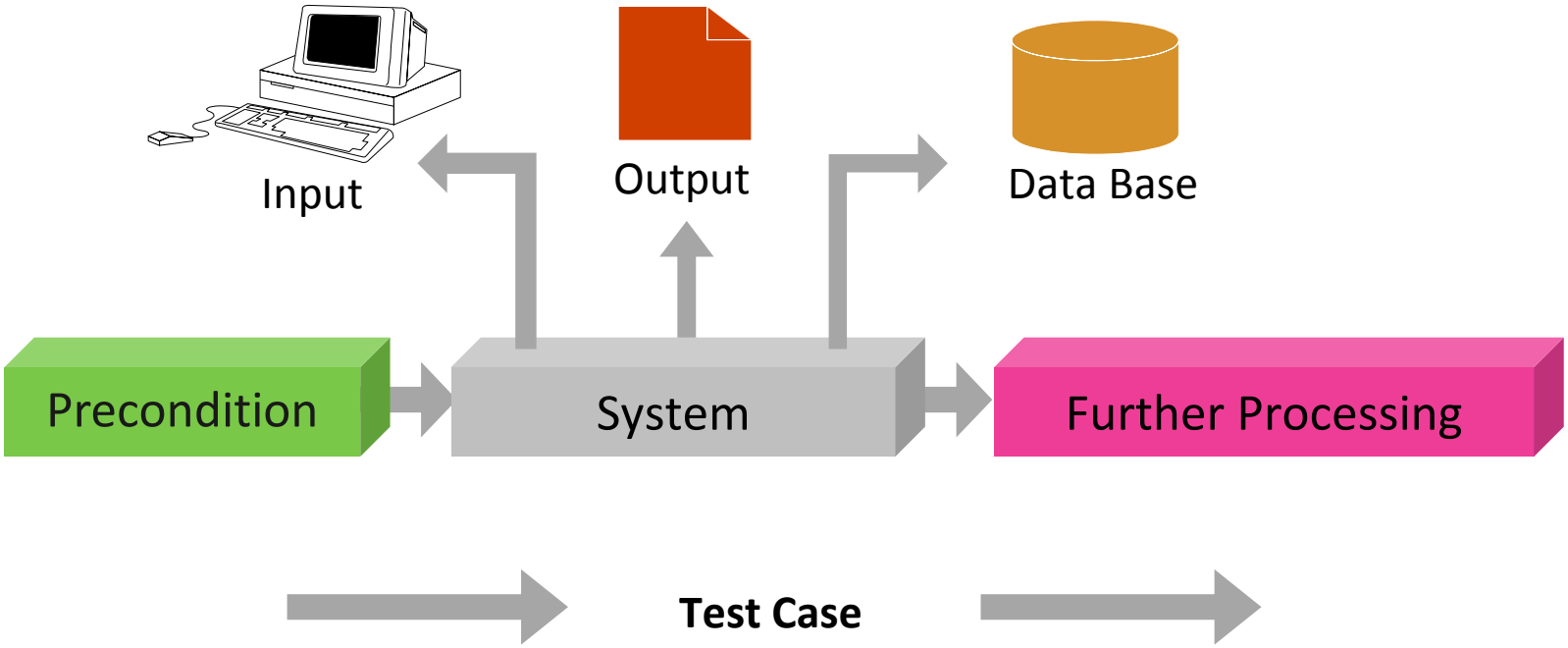
What are Test Scenarios? **(continued)**



© Copyright IBM Corporation 2015

Test case creation

Test planning and especially test case planning itself is a process.



© Copyright IBM Corporation 2015

Test case description

A test case is a set of conditions or variables under, which a tester will determine if a requirement upon an application is partially or fully satisfied.

1. At least one test case is needed for each requirement
2. Test cases have to be created for each requirement
3. Normal operation is accepted. Different test cases are there for different roles.
4. Application without formal requirement is accepted.
5. **Test Scripts** are required.
6. The **input** is **known and output** is **expected**.

© Copyright IBM Corporation 2015

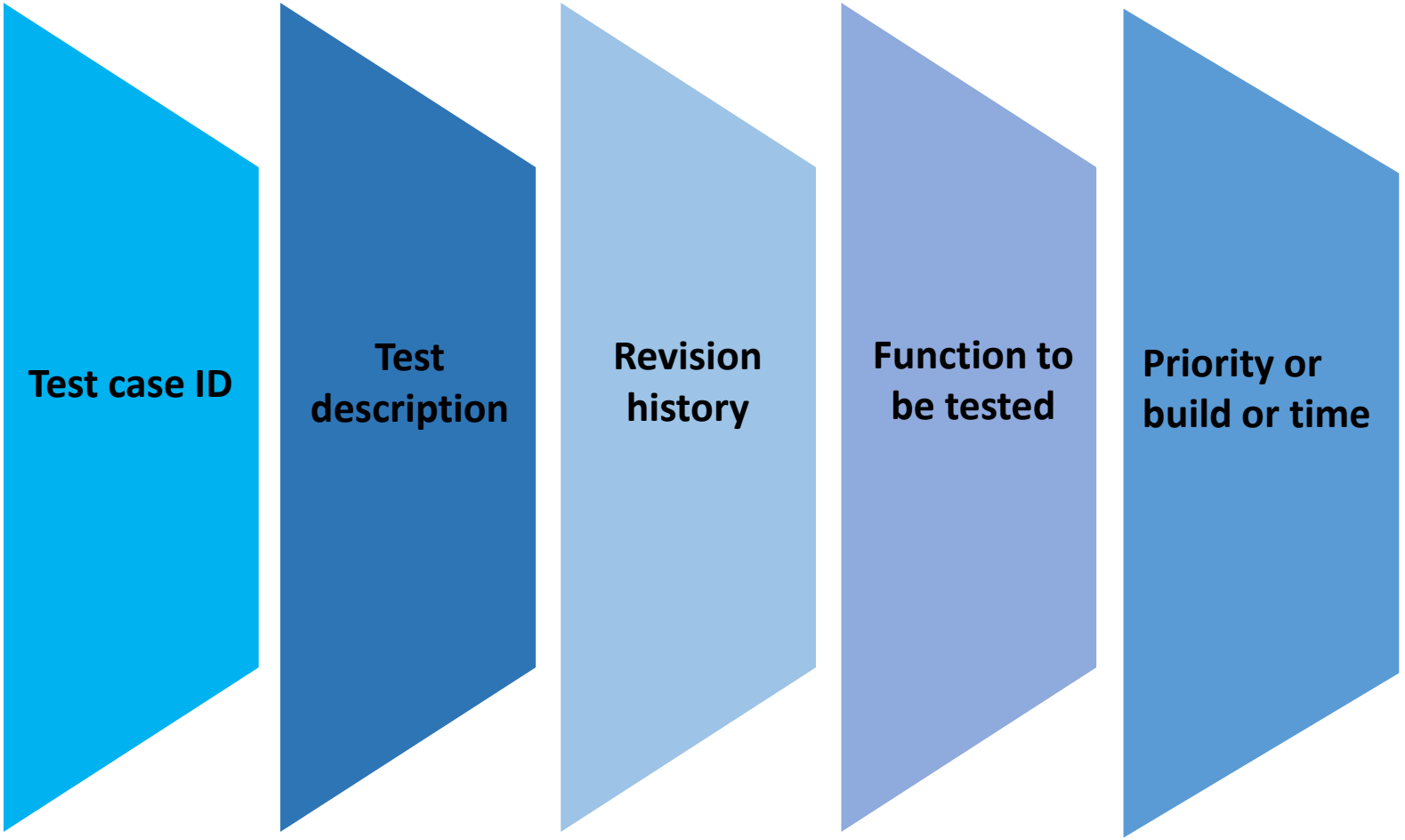
Test case format

Environment	It gives information about the testing environment.
Test Setup	It involves anything you need to set up outside of your application, for example printers, network, and so on.
Test Execution	It is detailed description of every step of execution.
Expected Results	It is the description of what you expect the function to do.
Actual Results	Passed or Failed: ▪ **If passed**– This is what actually happens when you run the test ▪ **If failed**- what is expected of the function to do" did not happen, and did not remove what you have observed"

© Copyright IBM Corporation 2015

Test case format (continued)

A typical test case sheet contains the following fields:



© Copyright IBM Corporation 2015

Test preparation

Prepare testing tool– Mercury Quality Center (QC)	▪ Attend administrator training. ▪ Create project. ▪ Set up users. ▪ Load Test Requirements including Business Process Master List or BPML and lists of interfaces, enhancements, reports or forms.
Train testers	▪ Identify testers. ▪ Conduct training.
Create test cases and test scenarios	▪ Identify test scenarios and document in test matrix. ▪ Determine test cases with variants. ▪ Create test scenarios with description or step-by-step instructions, expected results, and data requirements.
Create test schedule	▪ Develop test schedule. ▪ Obtain approval of test schedule. ▪ Develop detailed week 1 test schedule.
Build test client	▪ Create Quality Assurance Client (UQE). ▪ Input WMUS specific data (test data and or converted data) **by April 13 for** G or L Accounts, Cost centers, Profit centers, Vendors, and so on.

© Copyright IBM Corporation 2015

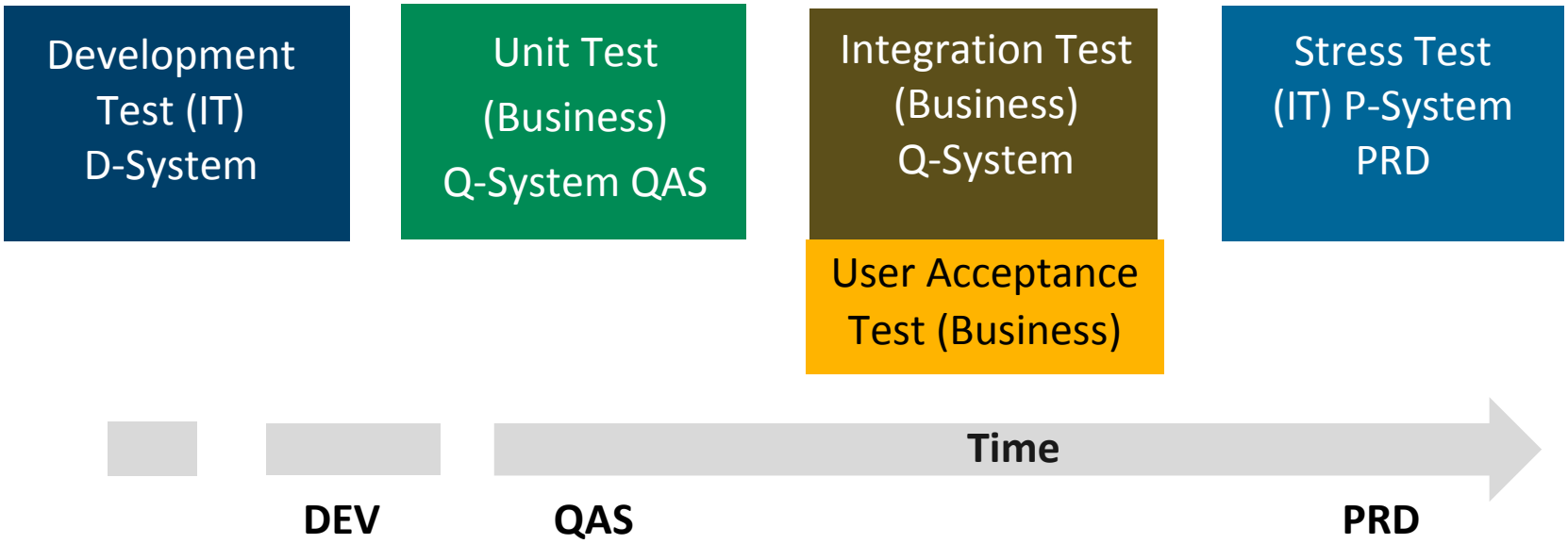
Test environments overview

Testing in three environments:

DEV D-System = Developer environment

QAS Q-System = Test environment

PRD P-System = Production environment



© Copyright IBM Corporation 2015

© Copyright IBM Corporation 2015

Spot Quiz

Brighter Blue

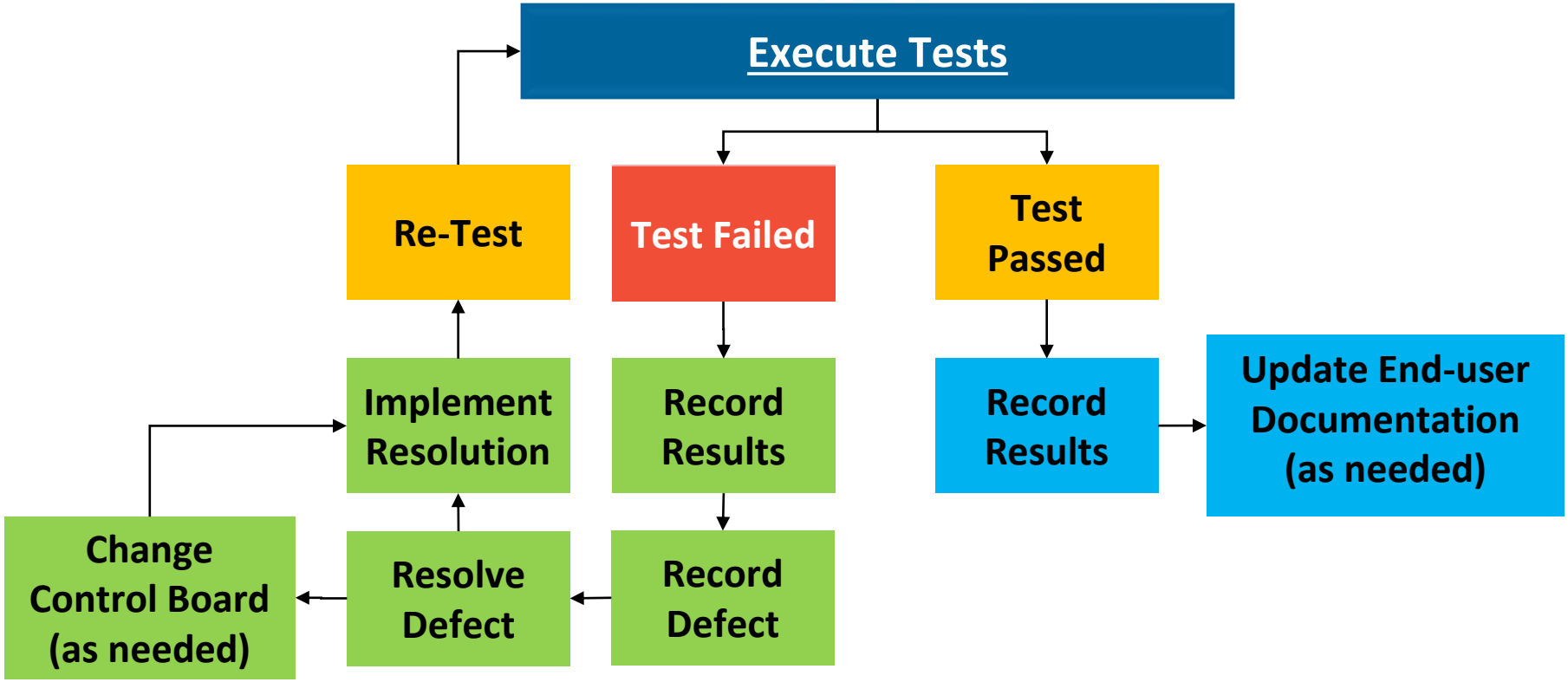
02

Test Strategy is a part of:

A	Test Cases
B	Test Plan
C	Test Design
D	Test Data

© Copyright IBM Corporation 2015

Test execution



© Copyright IBM Corporation 2015

Test execution (continued)

1. Process the test scenarios and test cases
2. Conduct the test protocol or test report
3. Document all defects
4. Classify and prioritize the defects
5. Analyze the defects (IT)
6. Document all test results
7. Use defined tools (SSM)
8. Prepare reports for management

© Copyright IBM Corporation 2015

Analysis and result documentation

Brighter Blue

- Preparing test reports or logs
- Defect classification in Defect Tracking Tool
- Validation of test results by IT
- Comparison of test results and specification
- Escalation and conflict management
- Controlling and documentation of changes

Question: did you achieve your **testing targets**?

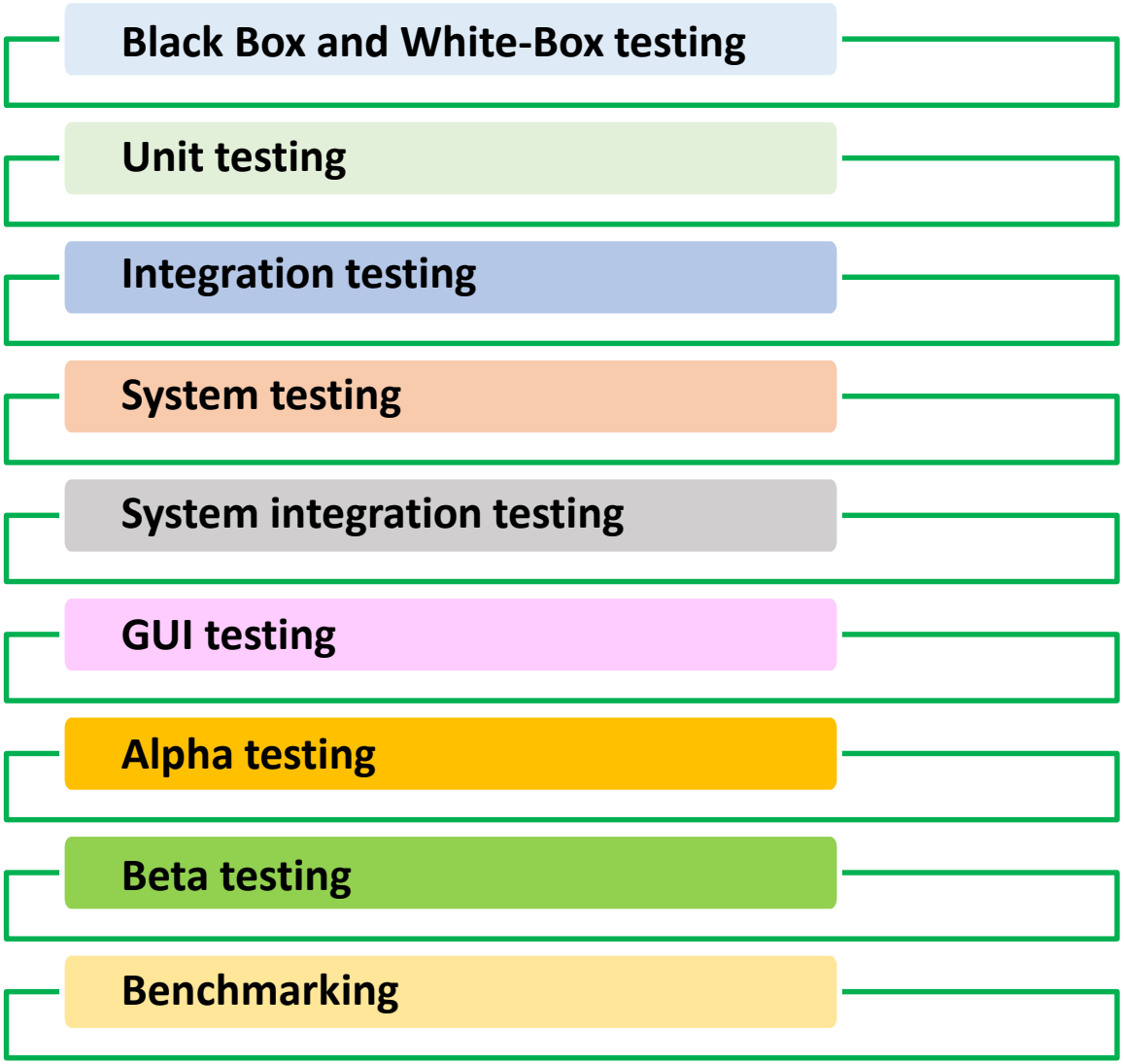
© Copyright IBM Corporation 2015

Test execution: Key points

- Test execution consists of preparing of test bed or setup, execution of planned test cases, verification of results, test tracking, and reporting and metrics collection.
- Apart from that, it also involves applying different test methods like Black Box to system testing, White Box at unit testing level and top–down and bottom–up in case of integration testing. In case of the latter, it will involve several rounds based on the stability and bugs found on the software and start or stop criteria identified.
- Regression testing will be carried out whenever the new bugs are found and fixed in the tested software to verify the impacted items.
- Test execution will involve preparation of test reports, collection of different metrics, and meeting the test objectives.

© Copyright IBM Corporation 2015

Testing levels



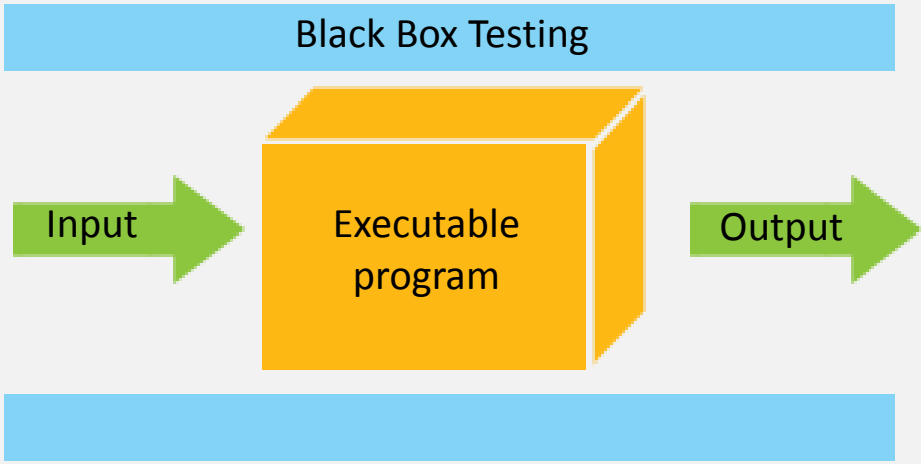
© Copyright IBM Corporation 2015

Black box and white box Testing

- **Black box testing** is either functional or non-functional, without reference to the internal structure of the component or system.
- Select test cases based on an analysis of the specification, either functional or non-functional, of a component or system without reference to its internal structure.

Example

- A tester, without knowledge of the internal structures of a website, tests the web pages by using a browser; providing inputs (clicks, keystrokes) and verifying the outputs against the expected outcome. (for example Gmail Page – Opening the URL enter Username Password and click on Submit - GMAIL Home Page to display).



© Copyright IBM Corporation 2015

White Box

- **White box testing:** This is the type of testing in which the internal structure/ design/Logic/ implementation of the item being tested is known to the tester. The tester chooses inputs to exercise paths through the code and determines the appropriate outputs. Programming know-how and the implementation knowledge is essential in this type of testing.
- Testing based on an analysis of the internal structure of the component or system.

© Copyright IBM Corporation 2015

White Box (continued)

- A tester, usually a developer as well, studies the implementation code of a certain field on a webpage, determines all legal (valid and invalid) AND illegal inputs and verifies the outputs against the expected outcomes, which is also determined by studying the implementation code.
- For example, Gmail page login page has User name , Password and Submit Button , so the tester will see code **of Submit button** and all the other **two text boxes Username and Password** (legal (valid and invalid) AND illegal inputs) to see if they are performing their functions or not.

Levels Applicable to:-

White Box Testing method is applicable to the following levels of software testing:

- Unit Testing: For testing paths within a unit.
- Integration Testing: For testing paths between units.
- System Testing: For testing paths between subsystems.

© Copyright IBM Corporation 2015

Types of testing

Application Integration Testing (AIT)

It is interface testing with WMUS and non-WMUS systems. Performed by functional and development teams. **Documentation tool**: Excel

User Acceptance Testing (UAT)

It is the most complex and full system test executing complete business scenarios including month end processing, reports or forms, interfaces, and enhancements. Performed by expert users with support from functional and development teams. **Documentation tool**: Quality Center

Structural Testing

It is execution of technical application, system, and database tests, such as, backup and recovery testing. **Documentation tool**: Excel

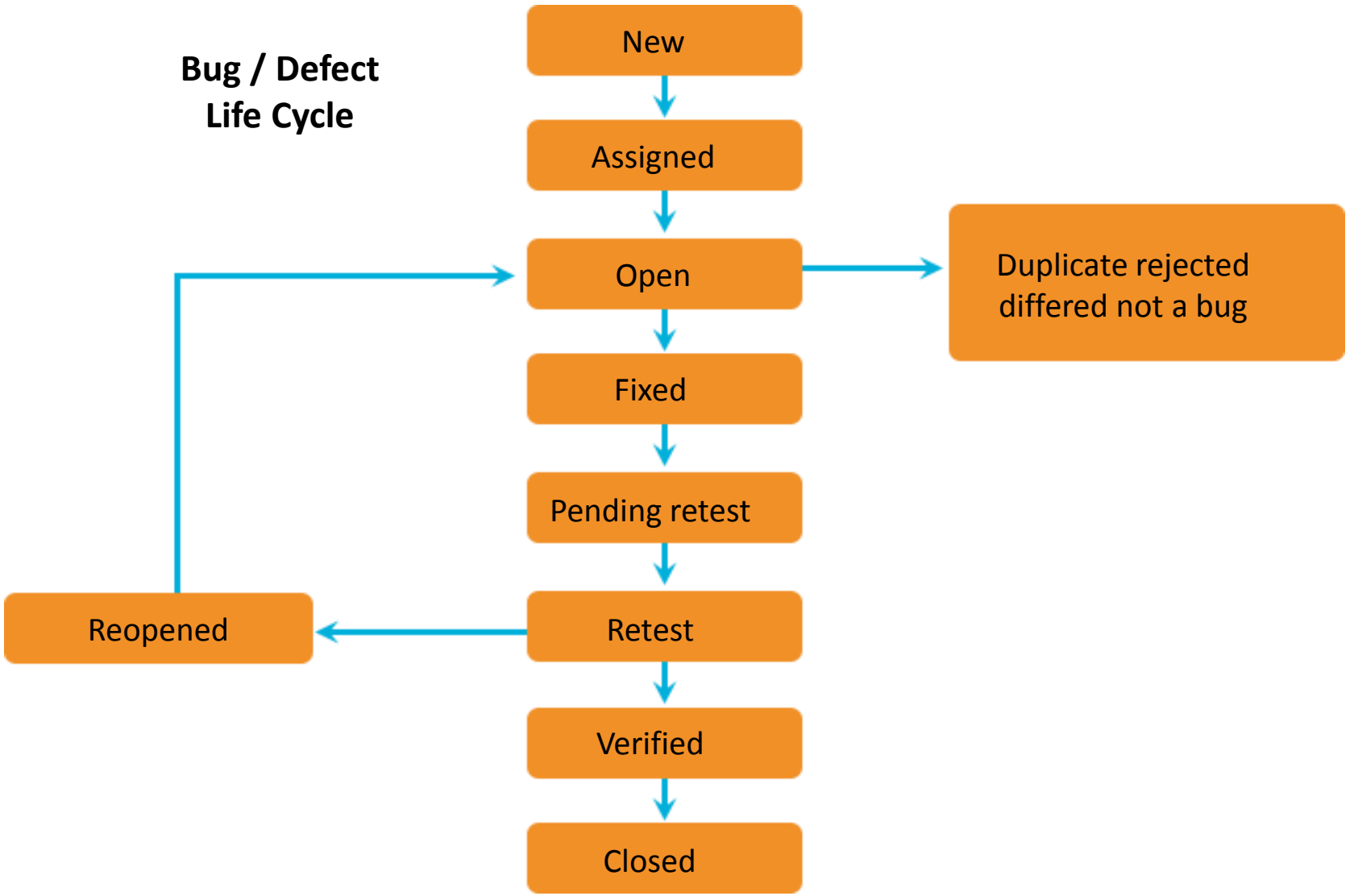
© Copyright IBM Corporation 2015

Types of testing (continued)

Conversion Testing	It is testing of conversions. It is performed by business process and functional teams, with support from development teams. **Documentation tool**: Excel
Cutover Testing	It is testing of role-based security. It is performed by business process and functional teams in parallel with other testing. Typical test cases are a subset of other test phases. **Documentation tool**: Excel
Security Testing	It is a dry run of data conversion and cut over processes, steps, and procedures. It is performed by business process and functional teams with support from development teams**. Documentation tool**: Excel

© Copyright IBM Corporation 2015

Defect workflow



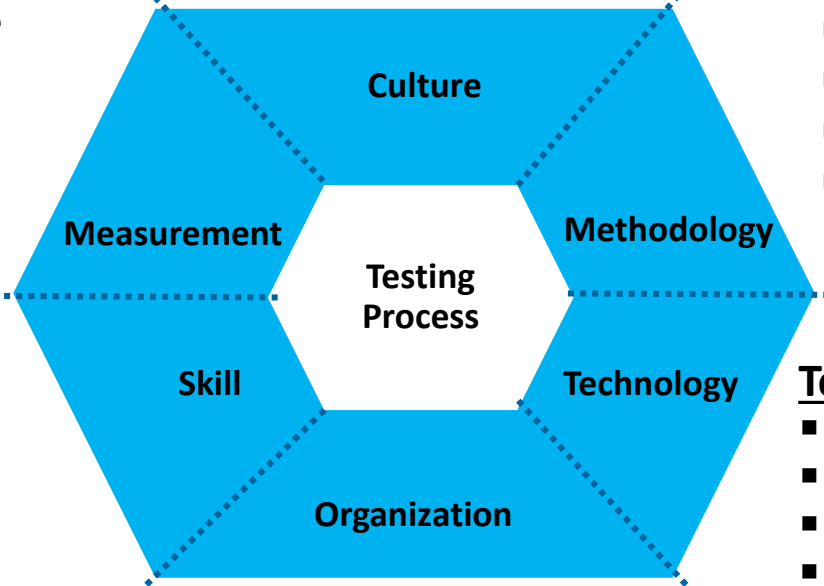
© Copyright IBM Corporation 2015

Key points

*Quality cannot be tested in the very end. It must be built-in…quality is a part of everybody's job

Statistical Based Status:
- Test Case:
  - Total number to be run
  - Total attempts to date
  - Total success to date
- Defects by severity:
  - Total found to date
  - Total solved defects
  - Total open defects

Testing Principles:
- Full lifecycle testing
- Static @ dynamic testing
- Testable requirements
- Business-driven testing
- Entry or exit criteria
- Pre-agreed quality target



Specialist Roles:
- Test Case repository
- Test Case design
- Test Case construction
- Test execution
- Test evaluation
- Test reporting
- Functional testers
- Environment builders

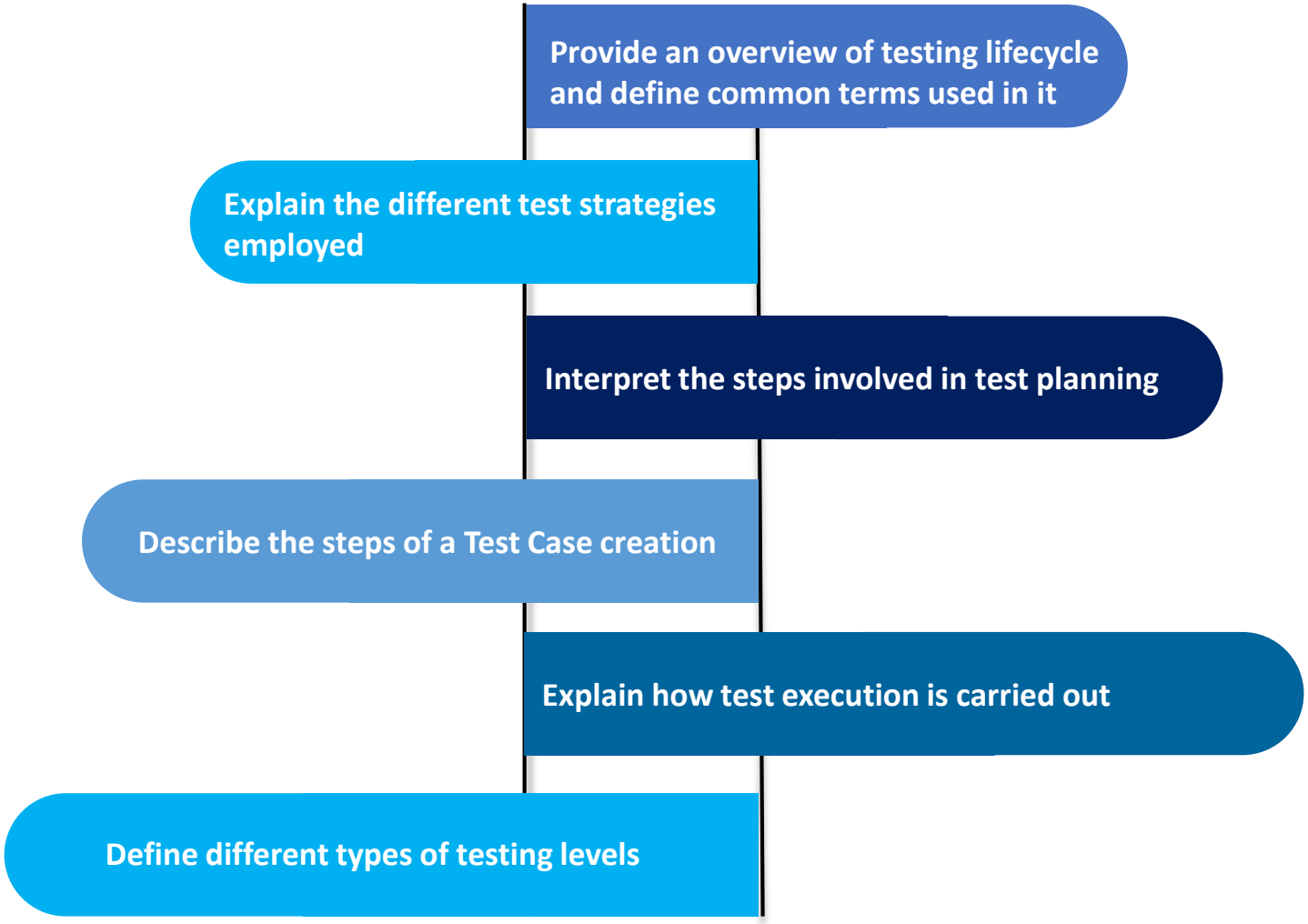
Testing Automation:
- Test Case repository
- Run execution
- Run analysis
- Status reporting
- Defect management

Independent Test Team:
- Methodology expertise
- Business expertise
- Test tool expertise

© Copyright IBM Corporation 2015

Testing lifecycle: Module summary

At the end of this module, you should now be able to:



© Copyright IBM Corporation 2015

Case study: Test lifecycle

- Test case development:
  - Several test cases will be generated as per the test strategy and plan for user interface, functionality and error handling of Global Print in W3.
  - Integration and system test cases will be generation to test for the application.
- Test report:
  - Test cases will be executed and test results will be recorded, and tracked.
  - After bugs are fixed, the system is tested again to verify the same.
  - The whole system is also tested to verify that new bugs have not been introduced.

Test Case and Defect Log Template

Test Summary Template

© Copyright IBM Corporation 2015

Case study: Test lifecycle (continued)

Prepare Requirements Traceability Matrix for “Case Study”.

Requirements
Traceability Matrix

© Copyright IBM Corporation 2015

© Copyright IBM Corporation 2015

Spot Quiz Sample

Brighter Blue

03

Which of the following is Solution Defining activity and not the Build activity of SDLC?

A	Feasibility study
B	Coding
C	Testing
D	Maintenance

© Copyright IBM Corporation 2015

Spot Quiz Sample

Brighter Blue

04

During the software development process, at what point can the test process start?

A	When the code is complete
B	When the design is complete
C	When the software requirements have been approved
D	When the first code module is ready for unit testing

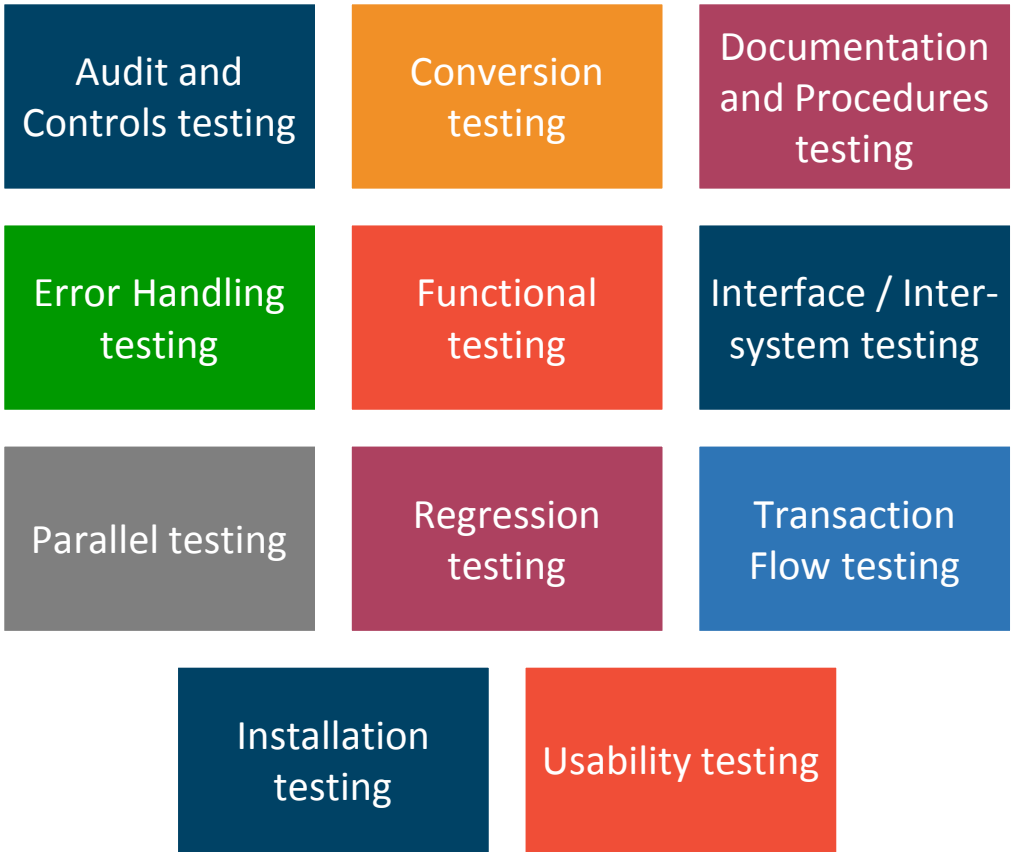
© Copyright IBM Corporation 2015

Brighter Blue

02 Types of Testing

IBM

Functional testing



© Copyright IBM Corporation 2015

Audit and Controls testing

It verifies the adequacy and effectiveness of controls and ensures the capability to prove the completeness of data processing results:

- Their validity would have been verified during design.
- This would have been normally carried out as part of System Testing once the primary application functions have stabilized.

Examples

- Inquiries of appropriate management, supervisor, and staff personnel
- Inspection of documents, reports, and electronic files
- Observation of the application of specific controls
- Re-performance of the application of the control by the auditors

© Copyright IBM Corporation 2015

Conversion testing

It verifies the compatibility of the converted program, data, and procedures with those from existing systems that are being converted or replaced. Their validity would have been verified during design.

Most programs that are developed for conversion purposes are not totally new. They are often enhancements or replacements for old, deficient, or manual systems.

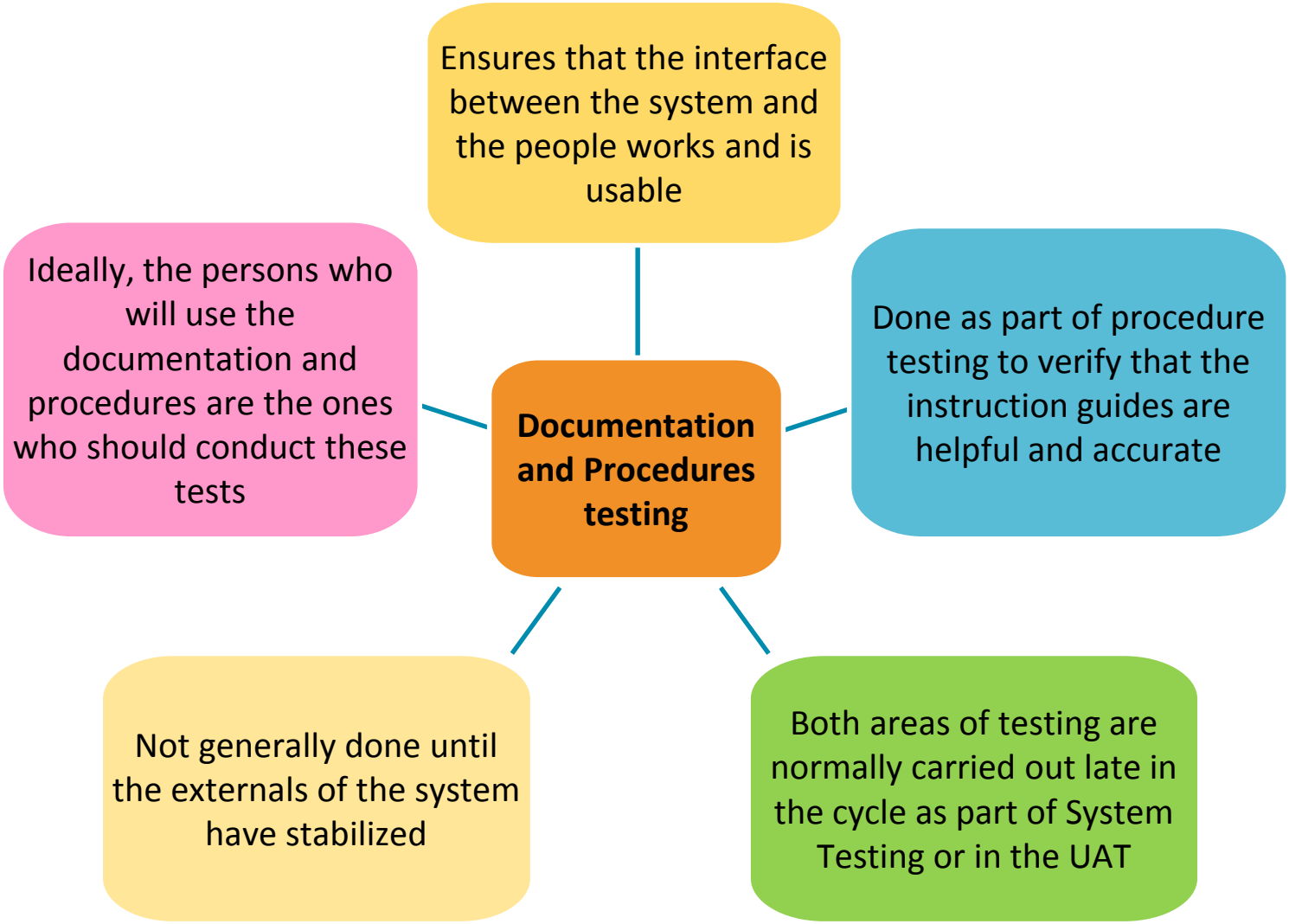
The conversion may involve files, databases, screens, report formats, and so on.

Example

- Conversion Testing is done whenever a project database is converted to a new version. A two-method verification process is needed to complete this conversion. The first method is to hire a database auditor to check value ranges and required relationships between records. The second method is to randomly select a small subset of old records and compare against a subset of new records.

© Copyright IBM Corporation 2015

Documentation and Procedures testing



© Copyright IBM Corporation 2015

Error Handling testing

- Error-handling is the system function for detecting and responding to exception conditions (such as erroneous input).

- Error handling ensures that incorrect transactions will be properly processed and that the system will terminate in a controlled, and predictable way in case of a disastrous failure.

Example

- Error-Handling testing includes the handing of run time error like input of an invalid character. Testing to confirm that application can recover from all types of error when they occur without terminating the application, or (if all else fails) gracefully terminate an affected application and save the error information to a log file.

© Copyright IBM Corporation 2015

Functions / Requirements testing

Functional testing verifies, at each stage of development, that each business function operates as stated in the **Requirements** and as specified in **the External and Internal Design** documents.

Functional testing is usually completed in System Testing so that by the time the system is handed over to the user for UAT, the test group has already verified that the system meets requirements.

Example

- This type of testing ignores the internal parts and focus on the output is as per requirement or not. Black-Box type testing geared to functional requirements of an application.

© Copyright IBM Corporation 2015

Installation testing

Any application that will be installed and run in an environment remote from the development location requires installation testing.

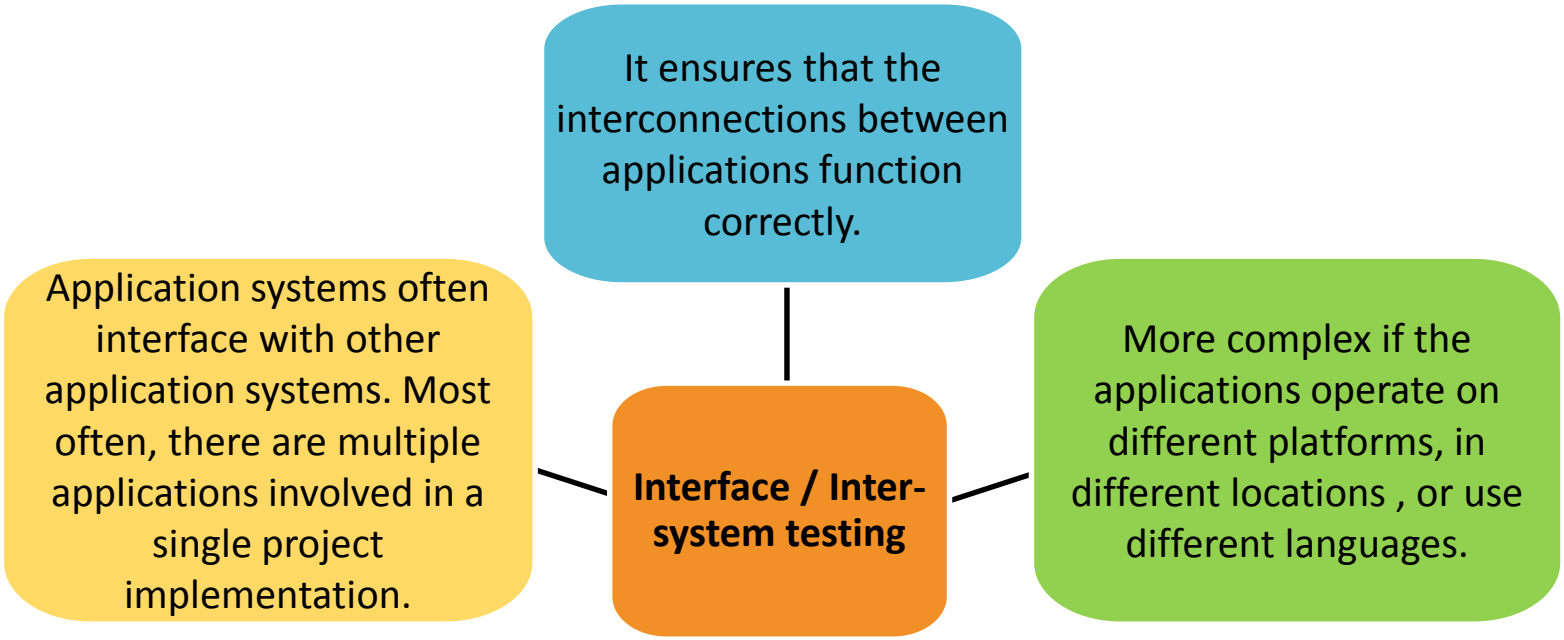
- This is especially true of network systems that may be run in many locations.
- This is also the case with packages where changes were developed at the vendor's site.
- This is necessary if the installation is complex, critical, should be completed in a short window, or of high volume, such as in microcomputer installations.
- This type of testing should always be performed by those who will perform the installation process.

Example

- Tested for full, partial, or upgrade install / uninstall processes on different operating systems under different hardware, software environment so that no issue is faced by the end user while installing the application once the code is in production.

© Copyright IBM Corporation 2015

Interface / Inter-system testing



Examples

- **Interface Testing** is performed to evaluate whether systems or components pass data and control correctly to one another.
- An **inter-system test** is the way you find out if the inter-system is working without any problem.

© Copyright IBM Corporation 2015

Parallel testing

Brighter Blue

Parallel testing compares the results of processing the same data in both the old and new systems.

Parallel testing is useful when a new application replaces an existing system, when the same transaction input is used in both, and when the output from both is reconcilable.

Parallel testing is useful when switching from a manual system to an automated system.

Examples

- Ensures that the new version of the application performs correctly.
- Ensures consistencies and inconsistencies are the same between the old and the new version.
- Ensures the integrity of the new application.
- Verifies if the data format between the two versions have changed.
- Operating new and old version of a payroll system to determine that the paychecks from both systems are reconcilable. "Running old version of application to Example :- ensure that the functions of old system are working fine with respect to the problems encountered in the new system

© Copyright IBM Corporation 2015

Sanity Versus Smoke Testing

Sanity Testing

After receiving a software build, with minor changes in code or functionality, Sanity testing is performed to ascertain that the bugs have been fixed and no further issues are introduced due to these changes. The goal is to determine that the proposed functionality works roughly as expected. If sanity test fails, the build is rejected to save the time and costs involved in a more rigorous testing.

Smoke Testing

Smoke Testing is performed after software build to ascertain that the critical functionalities of the program is working fine. It is executed "before" any detailed functional or regression tests are executed on the software build. The purpose is to reject a badly broken application, so that the QA team does not waste time installing and testing the software application.

© Copyright IBM Corporation 2015

Regression testing

Regression testing

Verifies that no unwanted changes were introduced to one part of the system as a result of making changes to another part of the system.

Examples

- It is assumed that several iterations of the system test will be done in order to test program modifications made during the system test period. A regression test will be performed for each new version of the system to detect unexpected impact resulting from program modifications.
- The regression test will be done by running all of the tests on a new version that were run on the previous version and then comparing the resulting files.

© Copyright IBM Corporation 2015

Transaction Flow testing

Transaction Flow testing

Testing of the path of a transaction from the time it enters the system until it is completely processed and exits a suite of applications.

Examples

- A transaction is a unit of work seen from a system user's point of view. It consists of a sequence of operations, some of which are performed by a system, persons or devices that are outside of the system.
- The most common testing is used to test the request a retry after user input errors. An ATM system, for example, allows the user to try, say three times, and will take the card away the fourth time.

© Copyright IBM Corporation 2015

Usability testing

It ensures that the final product is usable in a practical, day-to-day fashion.

It looks for simplicity and user-friendliness of the product.

Usability testing would normally be performed as part of functional testing during System and User Acceptance Testing.

Examples

Usability Testing tests the following features of the software:

- How easy it is to use the software
- How easy it is to learn the software
- How convenient is the software to end user

© Copyright IBM Corporation 2015

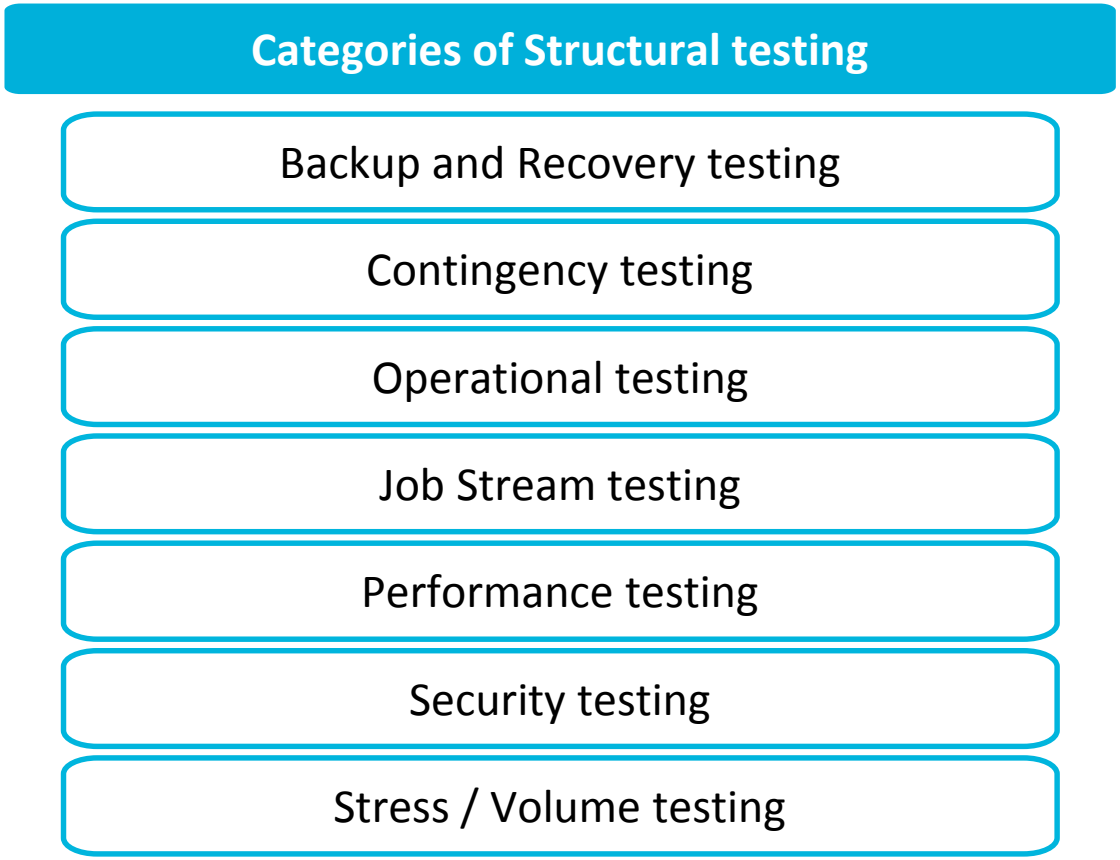
Structural testing

Ensures that the technical and "housekeeping" functions of the system work

Designed to verify that the system is structurally sound and can perform the intended tasks

© Copyright IBM Corporation 2015

Structural testing (continued)



© Copyright IBM Corporation 2015

Backup and Recovery testing

Recovery is the ability of an application to be restarted after failure.

The process usually involves backing up to a point in the processing cycle where the integrity of the system is assured and then re-processing the transactions past the original point of failure.

The nature of the application, the volume of transactions, the internal design of the application to handle a restart process, the skill level of the people involved in the recovery procedures, documentation and tools provided, all impact the recovery process.

Example

- Backup and recovery will be tested by halting the machine during stand-alone time and then following the recovery procedures.

© Copyright IBM Corporation 2015

Contingency testing

Operational situations may occur which result in major outages or **disasters**.

Some applications are so crucial that special precautions need to be taken to minimize the effects of these situations and speed the recovery process. This is called **contingency**.

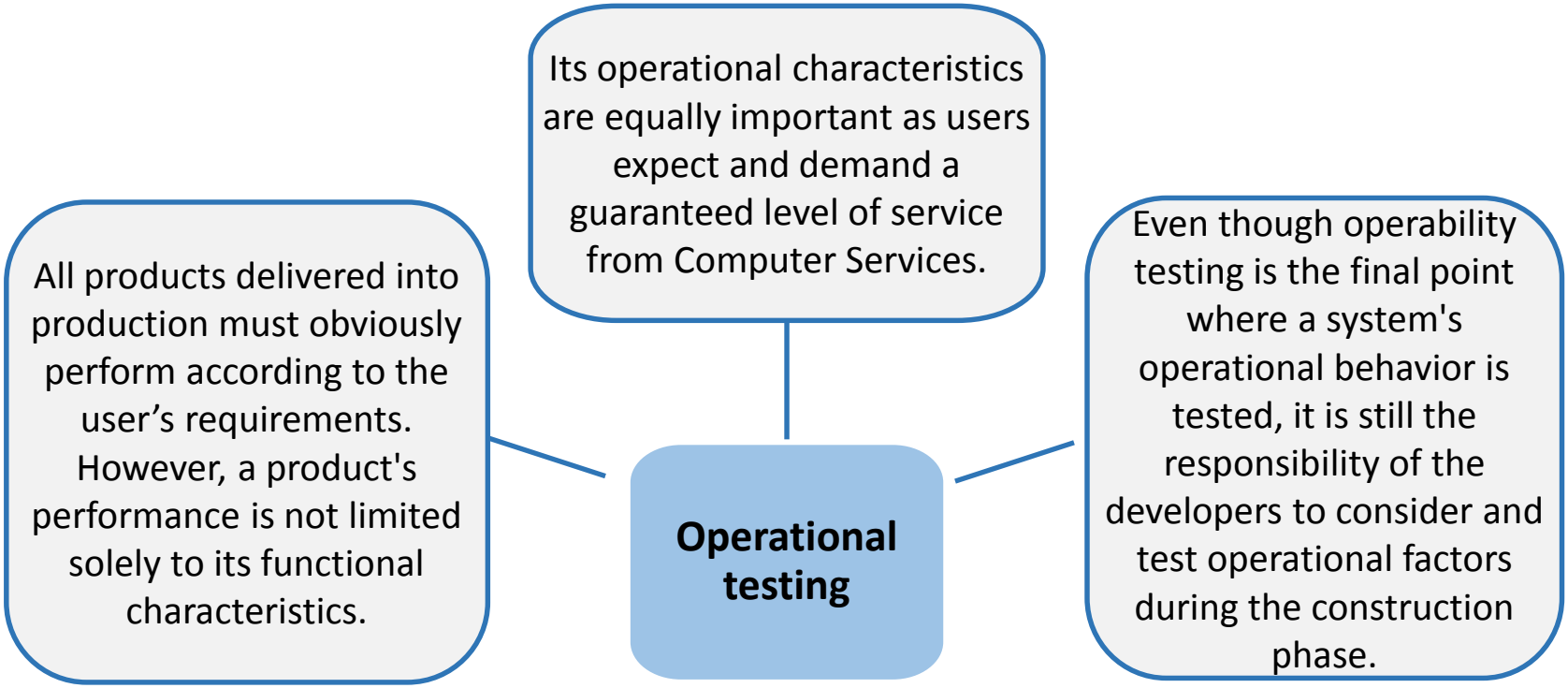
Example

Risk server overload (denial of service): The online reservation system can handle only 500 users at one time, more than 500 users attempting to access the system may result in denial of service.

- **Mitigation Plan:** An extra server to be maintained so that it could share the traffic with the main server.
- **Contingency Plan:** On failure of the two servers to handle the load, route any new user to a third server showing a message for service not available.

© Copyright IBM Corporation 2015

Operational testing



Example

- Verifying file labeling and protection functions are working properly.
- Determine system can run using document

© Copyright IBM Corporation 2015

Job Stream testing

Job Stream testing

Done as a part of operational testing (the test type not the test level, although this is still performed during operability testing):

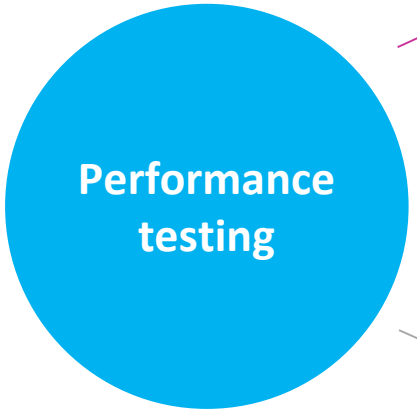
- Starts early and continues throughout all levels of testing
- Conformance to standards is checked in user acceptance and operability testing

Example

- Payroll processing–a comprehensive set of records of salaried employees, hourly employees, and a merged set of these two should be used to test payroll processing. Run each of the periodic reporting job streams at least once.

© Copyright IBM Corporation 2015

Performance testing



Performance testing is designed to test whether the system meets the desired level of performance in a production environment. Performance considerations may relate to response times, turn around times (through-put), technical design issues, and so on.

Performance testing can be conducted using a production system, a simulated environment, or a prototype.

Example

- Performance will be evaluated against the performance requirements by measuring the run times of several jobs using production data volumes.

© Copyright IBM Corporation 2015

Security testing

- Security of an application system is required to ensure the protection of confidential information in a system and in other affected systems is protected against loss, corruption, or misuse; either by deliberate or accidental actions.

- The amount of testing needed depends on the risk assessment of the consequences of a breach in security. Tests should focus on, and be limited to those security features developed as part of the system.

Example

- Attempted access without a proper password to the on-line data entry and display transactions (notification to the user of invalid authentication) will be tested.

© Copyright IBM Corporation 2015

Stress / Volume testing

Stress testing is defined as the processing of a large number of transactions through the system in a defined period of time. It is done to measure the performance characteristics of the system under peak load conditions.

Stress factors may apply to different aspects of the system, such as input transactions, report lines, internal tables, communications, computer processing capacity, throughput, disk space, I/O, and so on.

Stress testing should not begin until the system functions are fully tested and stable. The need for Stress Testing must be identified in the Design Phase and should commence as soon as operationally stable system units are available.

Example

- System is stressed beyond its specifications to check how and when it fails. Performed under heavy load like putting large number beyond storage capacity, complex database queries, and continuous input to system or database load.

© Copyright IBM Corporation 2015

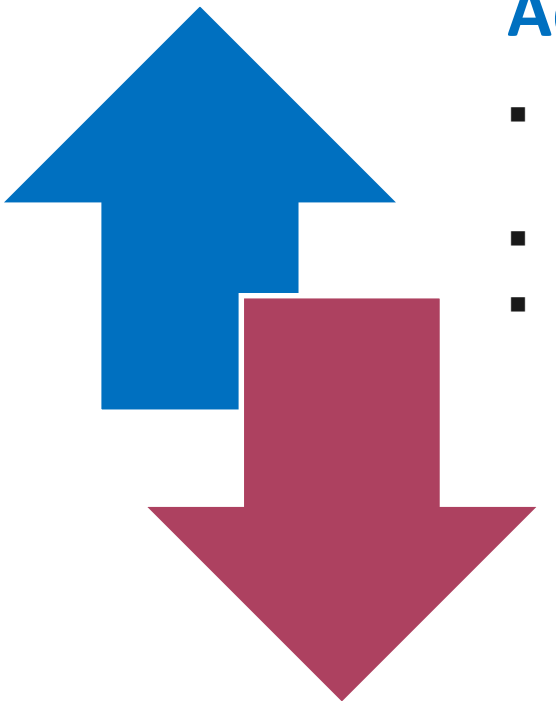
Bottom-up testing

Bottom-up testing

- Approach to integration testing where the lowest level components are tested first and then used to facilitate the testing of higher level components
- This process is repeated until the component at the top of the hierarchy is tested. In this approach, testing is conducted from sub module to main module; if the main module is not developed, a temporary program called **Drivers** is used to simulate the main module.

© Copyright IBM Corporation 2015

Bottom-up testing (continued)



Advantages:

- It is advantageous if major flaws occur toward the bottom of the program.
- Test conditions are easier to create.
- Observation of test results is easier.

Disadvantages:

- Driver Modules must be produced.
- The program as an entity does not exist until the last module is added.

© Copyright IBM Corporation 2015

Activity

Identify what type or kind of testing should be used in the following situations.

A banking institution has just bought an new software that would make their products and services
better and available to its clients. One of the features that they emphasized important is the safety
of the client data/information.

Which do you think should be the best type of testing to be performed in order to achieve the company's goal?

a. Load Testing
b. Stress Testing
c. Security Testing
d. Operational Testing
e. Contingency Testing

© Copyright IBM Corporation 2015

© Copyright IBM Corporation 2015

Spot Quiz

Brighter Blue

01 What are the two types of testing?

A	Static and Dynamic testing
B	Dynamic and Reviews testing
C	State Transition testing
D	Reviews and Inspection

© Copyright IBM Corporation 2015

Spot Quiz

Brighter Blue

02

Which of the following is a purpose for Regression testing?

A	To check that new functionality has been added or not
B	To check that the existing functionality has been changed
C	To check that no unwanted changes were introduced to one part of the system as a result of making changes to another part of the system

© Copyright IBM Corporation 2015

Spot Quiz

Brighter Blue

03 Which one is not part of Static Testing?

A	Reviews
B	Inspections
C	Walkthroughs
D	White Box testing and Black Box testing

© Copyright IBM Corporation 2015

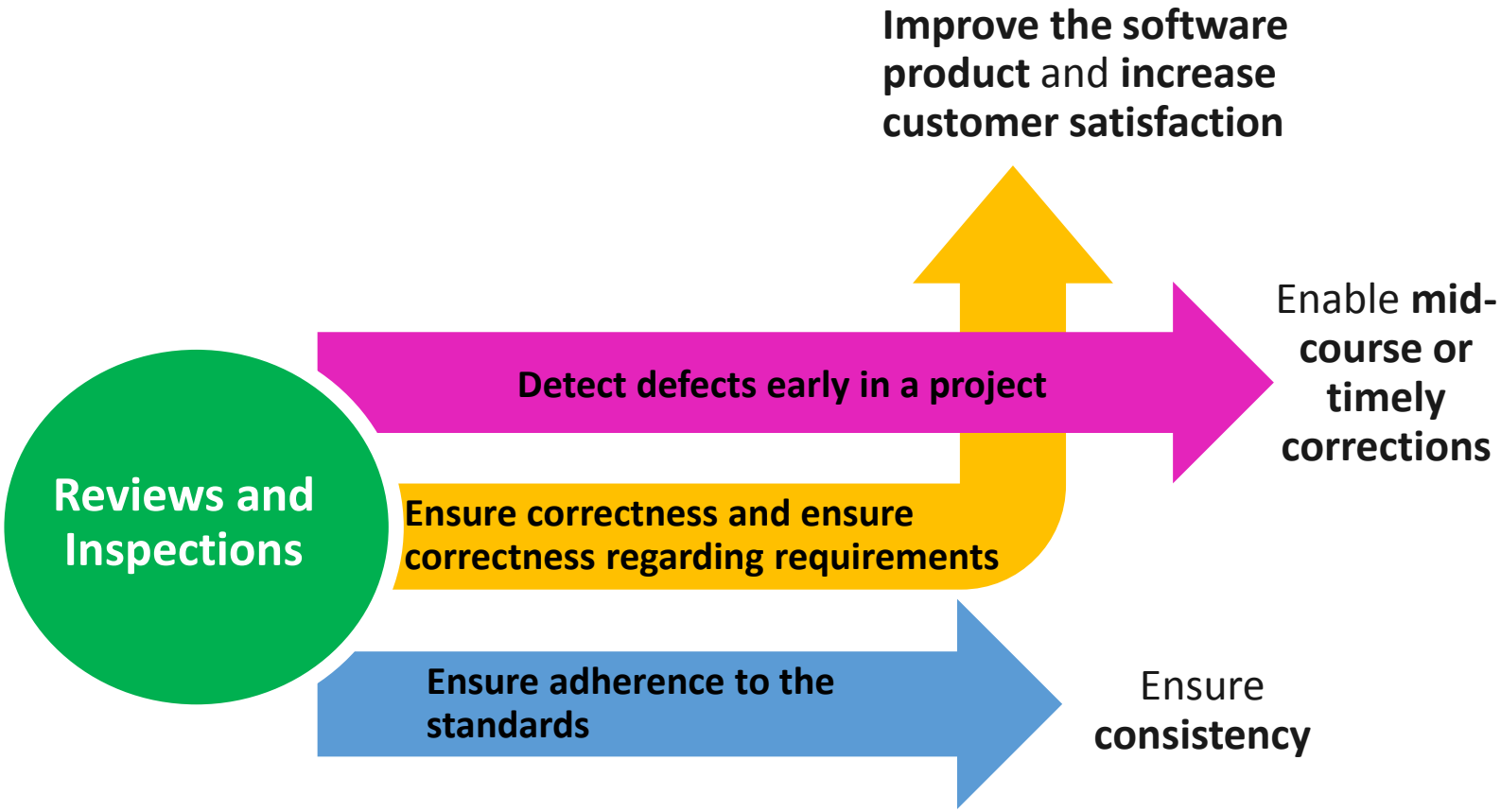
03 Static testing techniques

What are **reviews** and **inspections**

Reviews and Inspections:	Verifying or checking for correctness
Defined as:	• Activities involved in verifying a product against specified requirements • The process of identifying defects in the work product
It is:	• A methodical examination of software work products • Done over both executable and non-executable software work products • Helpful in identifying defects and areas where changes are needed • Considered a best industry practice for use on software projects

© Copyright IBM Corporation 2015

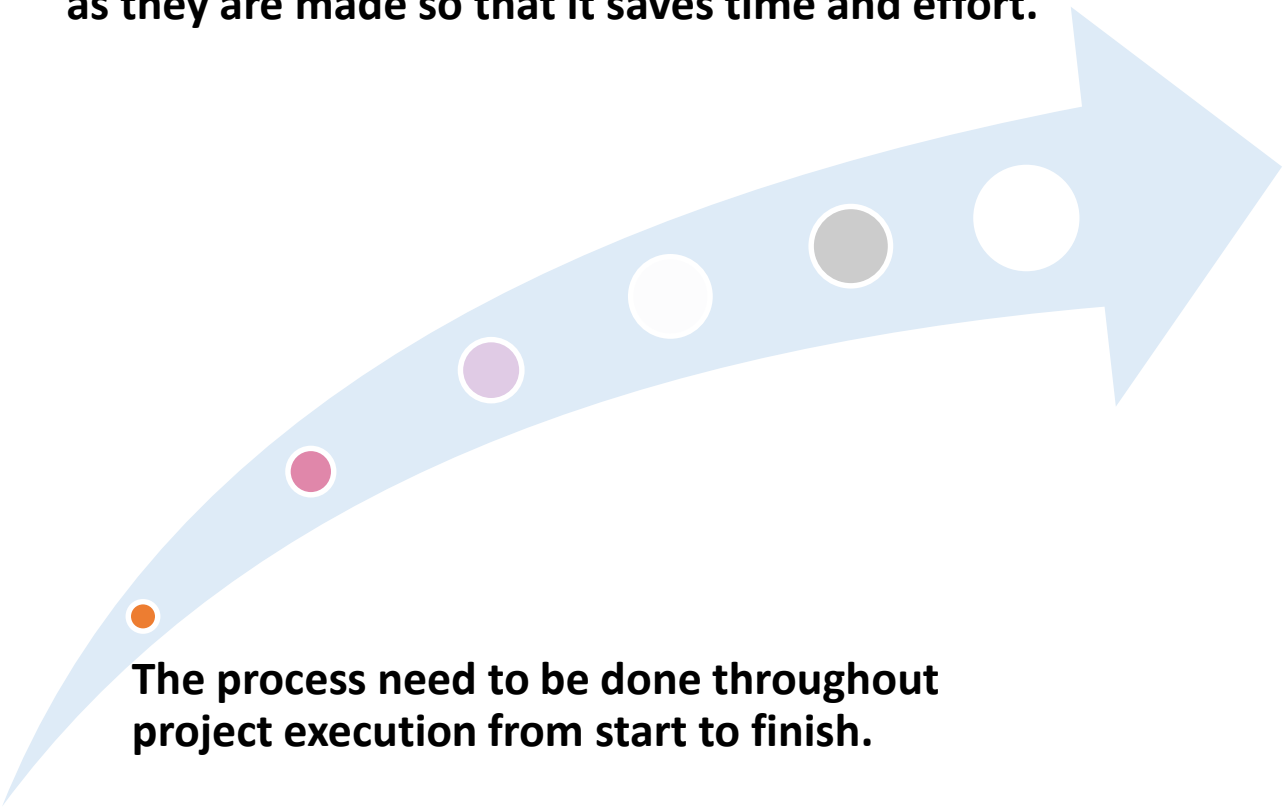
Why use Reviews and Inspections?



© Copyright IBM Corporation 2015

When to conduct Reviews and Inspections Process

Effective reviews will help detect errors as soon as they are made so that it saves time and effort.



The process need to be done throughout project execution from start to finish.

© Copyright IBM Corporation 2015

How to conduct Reviews and Inspections process

Brighter Blue

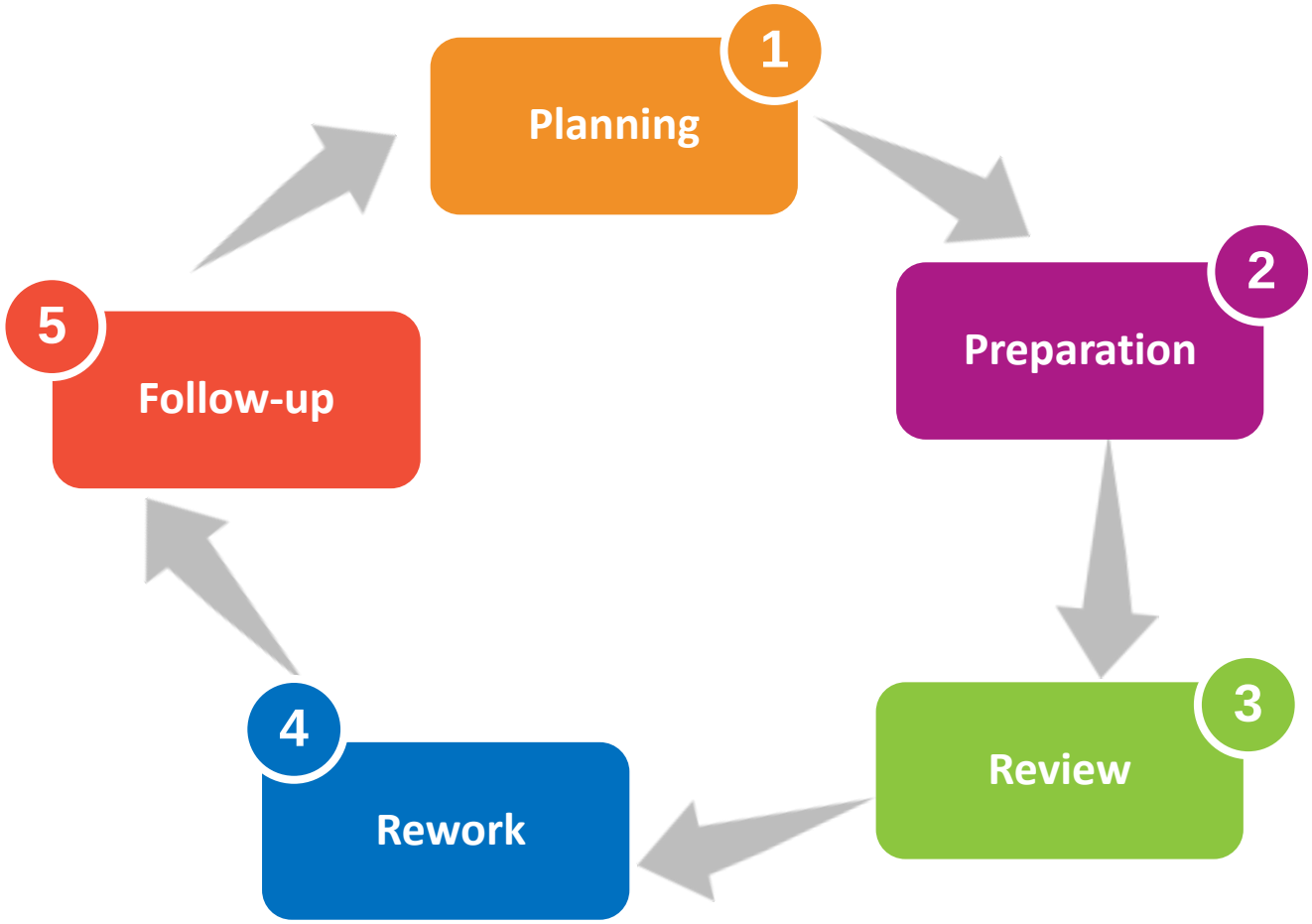
There are several methods in conducting reviews and inspections.

Peer Review

Formal Inspections

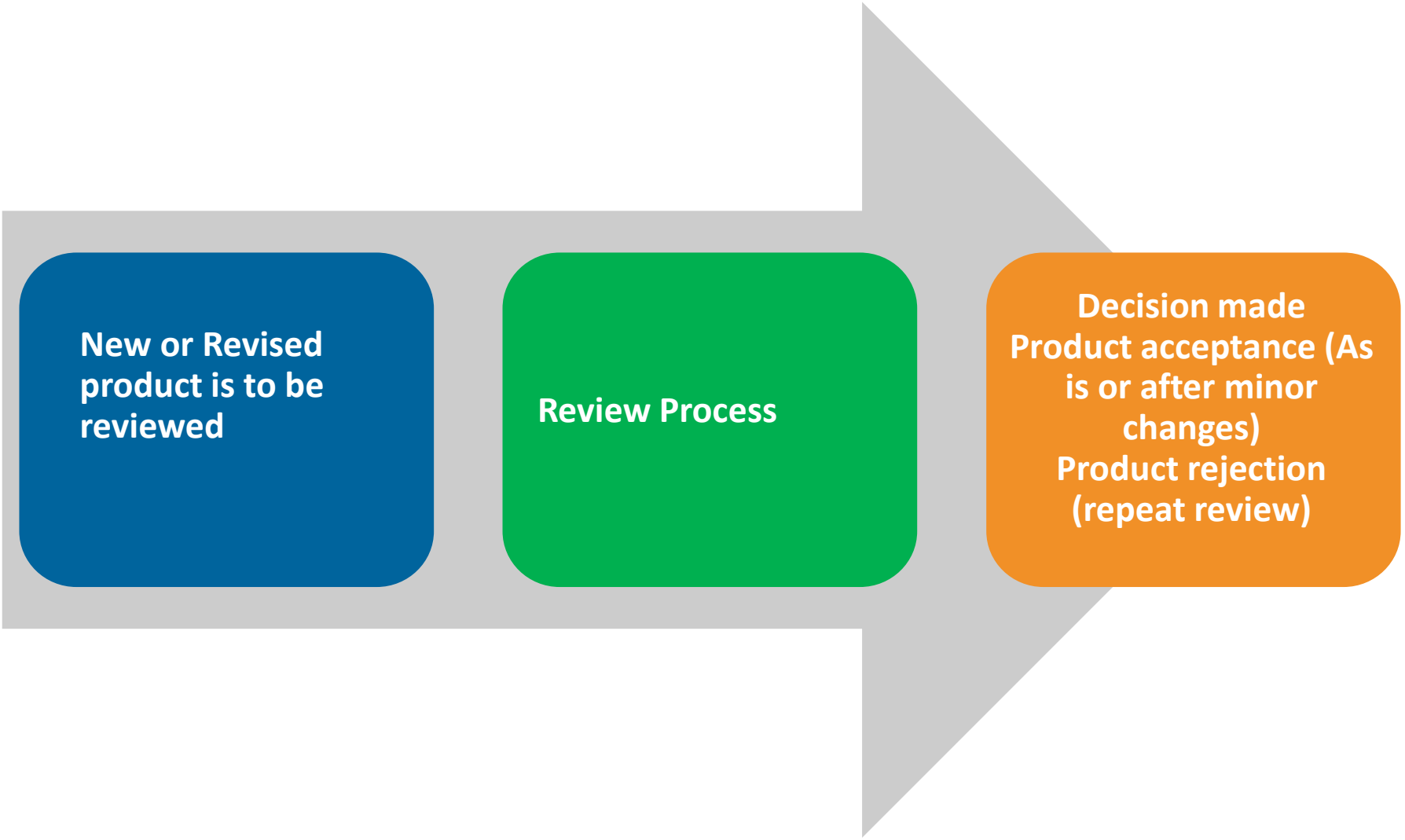
© Copyright IBM Corporation 2015

General Review Process



© Copyright IBM Corporation 2015

Entry and Exit Criteria for a Review Process



© Copyright IBM Corporation 2015

Tom's Story: Setting up the Christmas Tree (1 of 6)

Tom and his wife **Pam** are busy preparing for Christmas. They decide that Tom would buy and set up the Christmas tree, while Pam is out shopping for the gifts.

© Copyright IBM Corporation 2015

Tom's Story: Setting up the Christmas Tree (2 of 6)

© Copyright IBM Corporation 2015

Tom's Story: Setting up the Christmas Tree (3 of 6)

© Copyright IBM Corporation 2015

Tom's Story: Setting up the Christmas Tree (4 of 6)

© Copyright IBM Corporation 2015

Tom's Story: Setting up the Christmas Tree (5 of 6)

© Copyright IBM Corporation 2015

Tom's Story: Setting up the Christmas Tree (6 of 6)

By evening, Tom has assembled the tree, and is almost done putting up the star and the lights when his wife and kids walk in.

They are happy to see the tree, but Pam says "I really wanted to see a golden star and not a silver star".

Tom then switches on the lights but they are not working. Since the lights are already on the tree, he gets a ladder and checks them one by one to make sure they are ok. He then finally has to remove the extension cord to check it, and he realizes there is a problem with the extension cord.

Pam tries to place a gift near the base of the tree when she notices a crack on the stand at the base of the tree that is holding it up. No sooner than she points it out that

© Copyright IBM Corporation 2015

Activity

What were the lessons learned?

© Copyright IBM Corporation 2015

Activity

Brighter Blue

Lessons learned:

1. Early checking of the parts of the fake tree and the extension cord would have saved a lot of time and effort.

2. Checking the items right at the store would have saved even more time and effort.

© Copyright IBM Corporation 2015

Translating the real world scenario into **project scenario**

Brighter Blue

The assembling of the Christmas tree is similar to a software project.

Reviews are required to ensure that IBM's understanding matches with the client's specifications.

The code is developed inline with the design which, in turn, is produced inline with the requirements gathered.

If the requirements are gathered wrong, the defect is passed on to the design, and if there is an error in the design, it is also reflected in the code.

In fact, defective parts, when assembled, do not function as a whole and a great deal of effort is wasted on detecting the source of the defect.

Thus, we see the importance of reviews.

© Copyright IBM Corporation 2015

The **Delhi Metro Mishap**

- **Lack of proper inspection** caused a major mishap at Delhi Metro rail site. The committee found out that there was a deficiency in the design of the cantilever arm and that the concrete did not have adequate strength due to lack of its adequate curing.
- Six persons were killed when an under-construction over-bridge of Delhi Metro collapsed in Zamarudpur area of South Delhi.

© Copyright IBM Corporation 2015

The **Delhi Metro Mishap** (continued)

Brighter Blue

Contributing factors to this incident:

- Lack of supervision
- Compromising on the quality
- Bypassing the inspection process

© Copyright IBM Corporation 2015

Myths about Reviews and Inspections

Brighter Blue

Myths versus realities:

Myth	Reality
Reviews and inspections require a lot of extra effort, and thus increase the cost of a project.	If reviews are done in a timely manner and by members who are well-versed with the product being reviewed, it requires minimal effort.
Hard work alone results in quality. Everyone always tries to produce high quality.	Even if everyone genuinely works hard to produce high quality products, manual error is always possible.
Reviews and inspections can be done only by senior staff.	Anyone who is well-versed about the product being reviewed can do the reviews.

© Copyright IBM Corporation 2015

Spot quiz

Brighter Blue

01 Reviews and inspections are done for:

A. Executable software work products

B. Non-executable software work products

C. Released software products

© Copyright IBM Corporation 2015

© Copyright IBM Corporation 2015

Spot quiz

Brighter Blue

02 Why are reviews done?

A	To detect and correct defects
B	To be consistent with customer requirements
C	To adherence to standards
D	To accommodate new requirements
E	To measure contribution of each team member

© Copyright IBM Corporation 2015

Spot quiz

Brighter Blue

03

Which of the following is true about reviews?

A	Reviews increase the cost of a project since it requires a lot of extra effort.
B	A hardworking and sincere team does not require reviews.
C	Reviews are one of the best industry practices.
D	Reviews can only be done by senior staff.

© Copyright IBM Corporation 2015

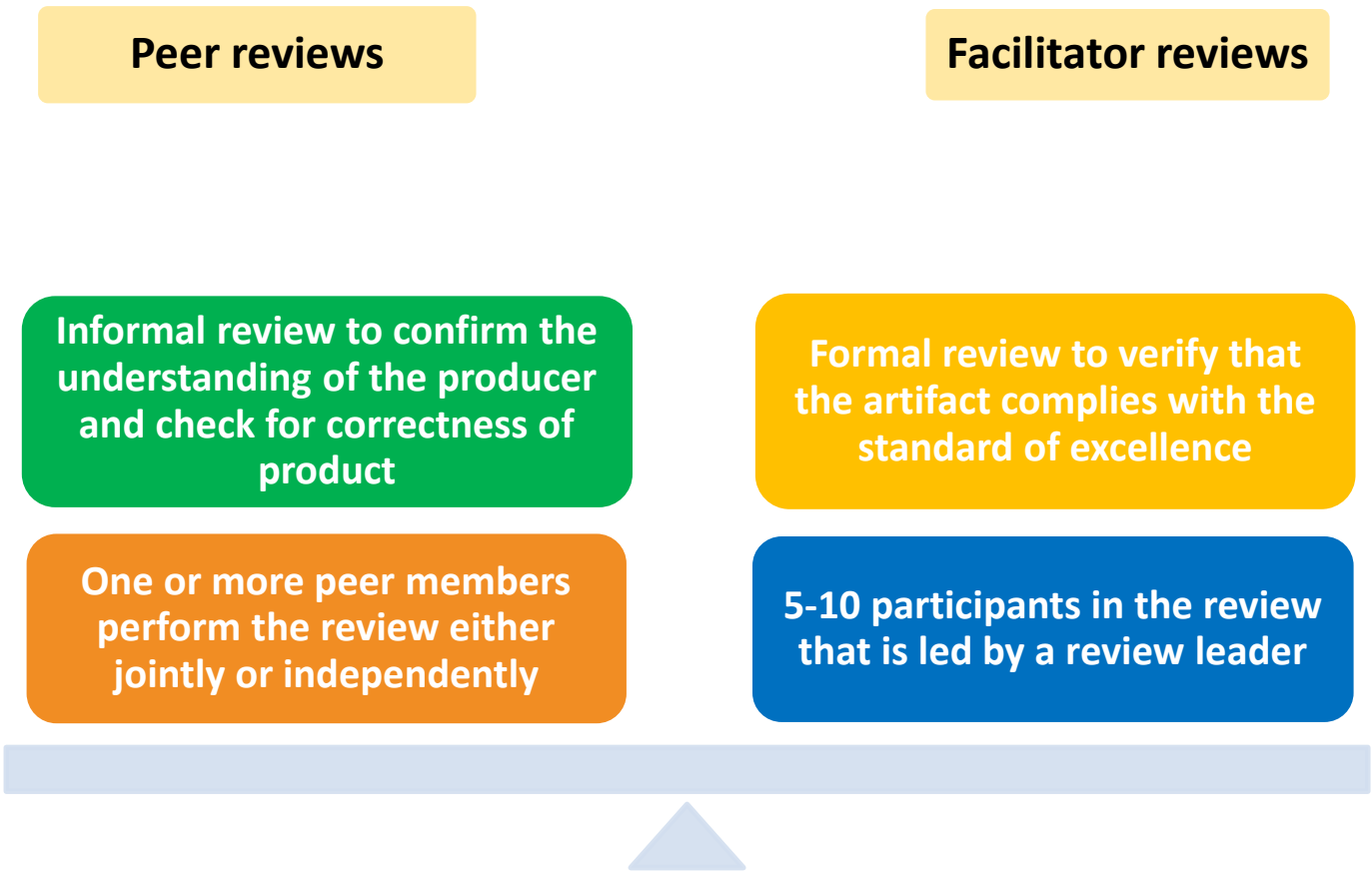
Types of reviews

There are various types of reviews that are based on different factors as given below.

Method of review	• Peer or buddy reviews • Facilitation reviews or inspections
Product reviewed	• Requirement review • Design review • Code review • Test Results review
Purpose	• Process reviews • Project Management reviews • Critical Thread reviews

© Copyright IBM Corporation 2015

Peer Reviews versus Facilitator Reviews



© Copyright IBM Corporation 2015

Factors to decide which type of reviews to use

Factors that decide on which type of review to use:

- **Work product type**
- **Business criticality**
- **Work product complexity**

© Copyright IBM Corporation 2015

Peer reviews: Characteristics

This is a less formal verification technique.

It is the author who initiates the session.

There may be several peer reviews in each software development lifecycle activity.

© Copyright IBM Corporation 2015

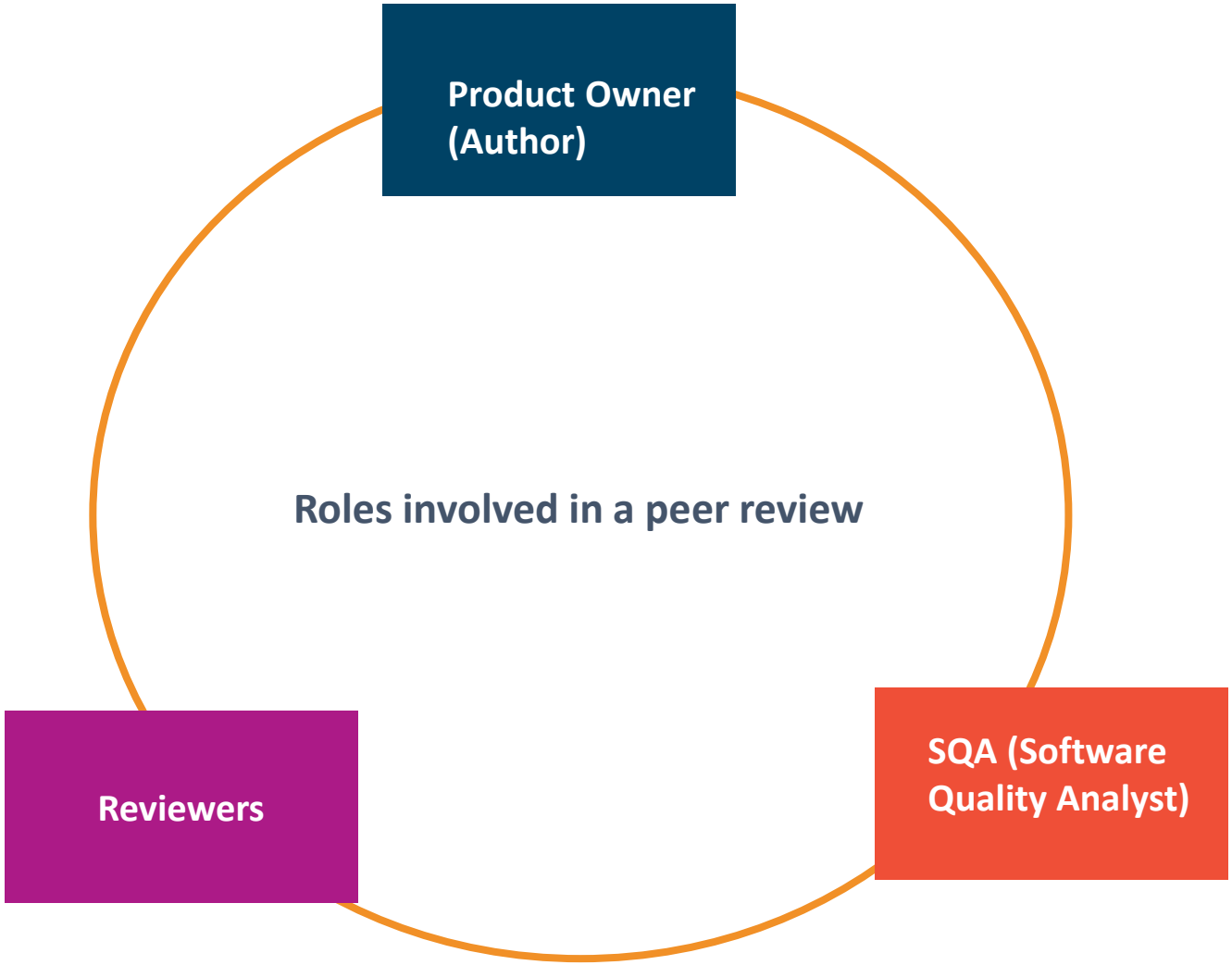
Expected end results of peer reviews

Participants - reviewer(s) and author - agree on the approaches taken, product, and engineering practices applied.

Completeness and correctness of capabilities and features of the reviewed product are obtained.

© Copyright IBM Corporation 2015

Facilitation reviews: **Roles and responsibilities**



© Copyright IBM Corporation 2015

Peer review: **Roles and responsibilities**

Product Owner (Author):

- Initiates and schedules reviews
- Ensures adherence to review process
- Identifies reviewers and distributes the product
- Ensures rework is carried out
- Fills in quality records

Reviewers:

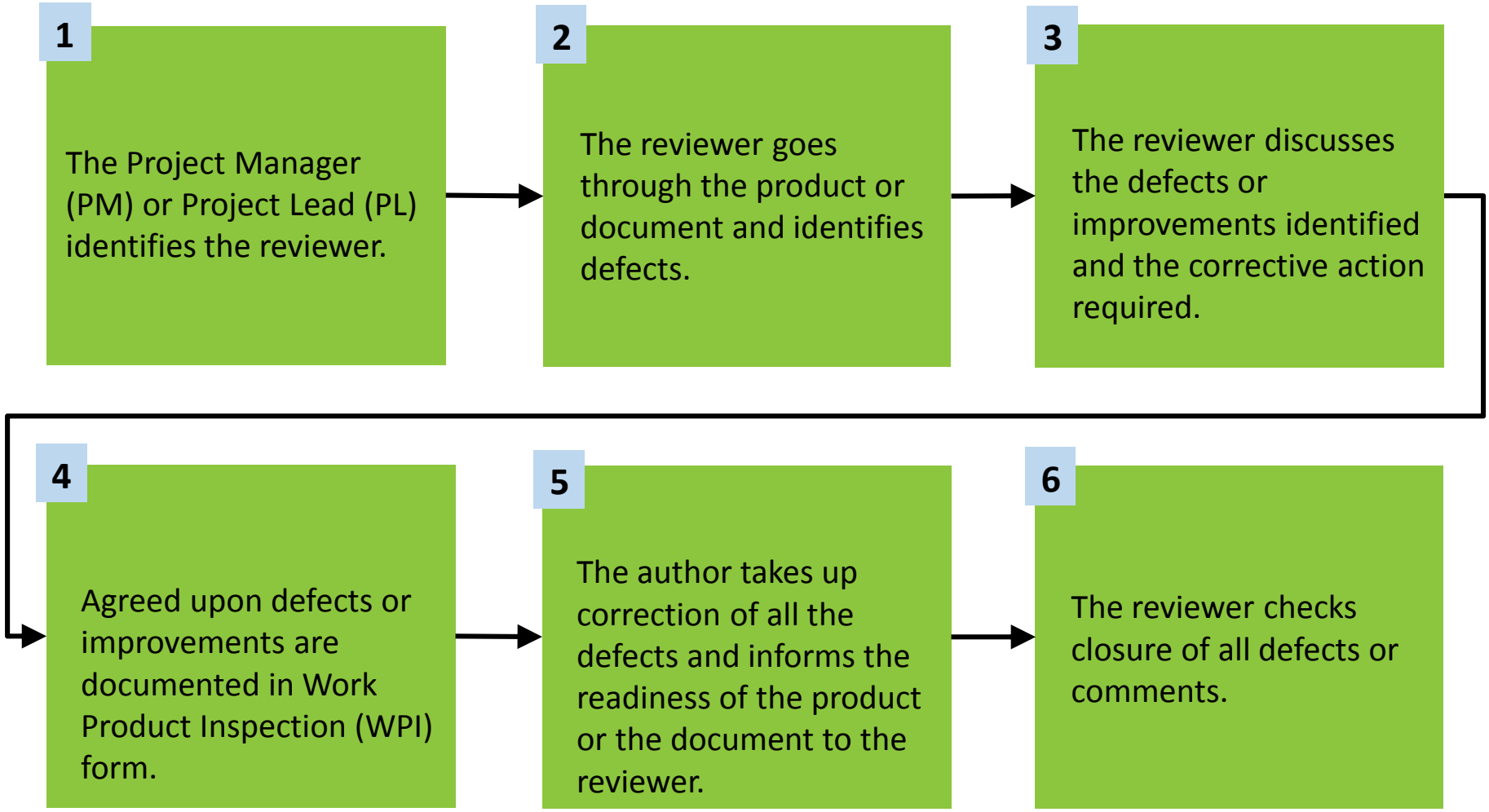
- Documents the review findings and completes the “Review Summary Report”
- Reviews the product
- Can also approve the product

SQA (Software Quality Analyst):

- Reviews the product and communicates the comments to the review leader
- Tracks the rework and resolution list to closure
- Signs off the review summary report

© Copyright IBM Corporation 2015

Peer review **procedure**



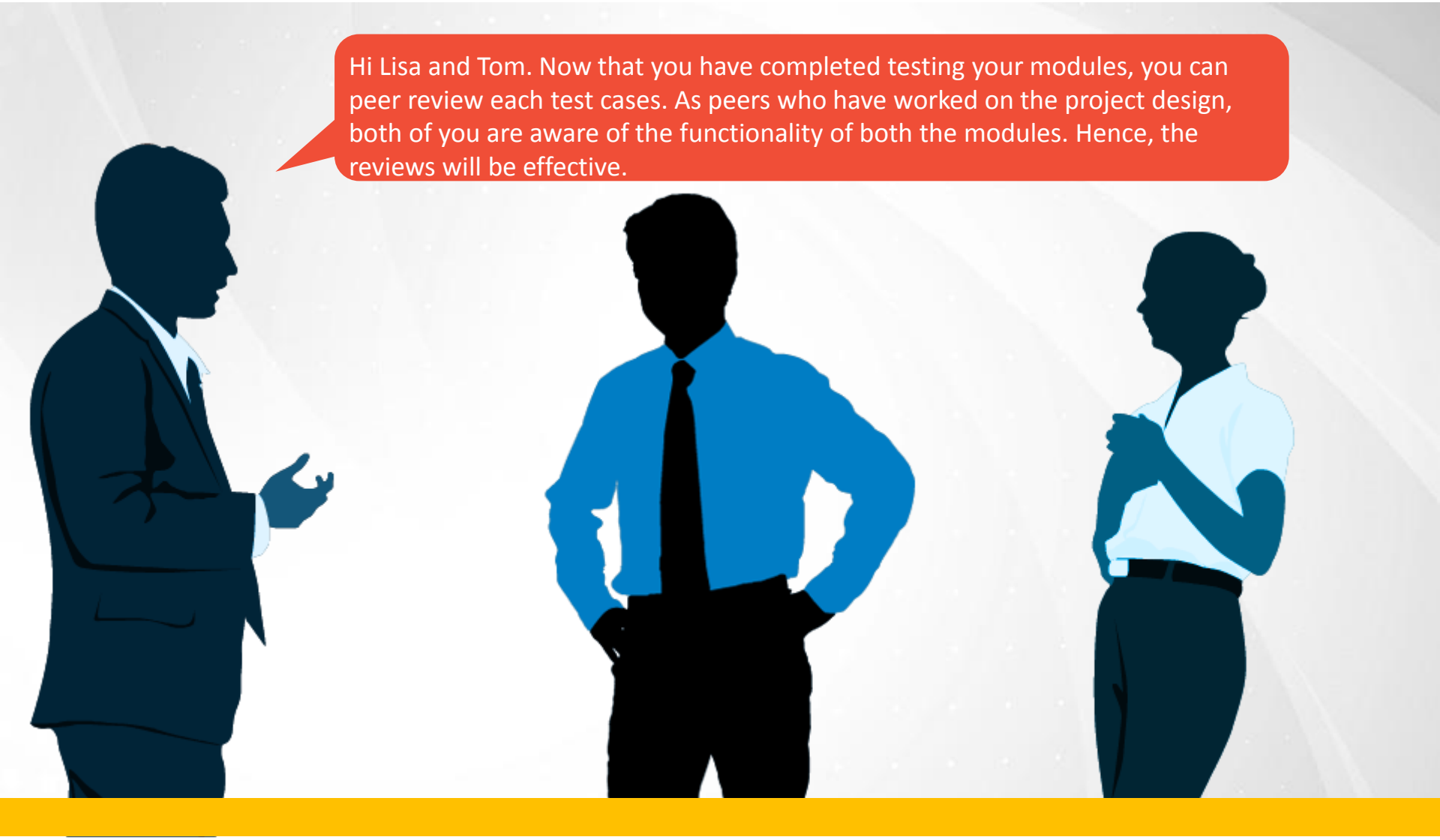
© Copyright IBM Corporation 2015

Project A - Peer Review

John is a Project Lead handling **Project A**. **Lisa** and **Tom** are working on modules 1 and 2 respectively, and have just completed the testing of their modules.

© Copyright IBM Corporation 2015

Project A - Peer Review (continued)

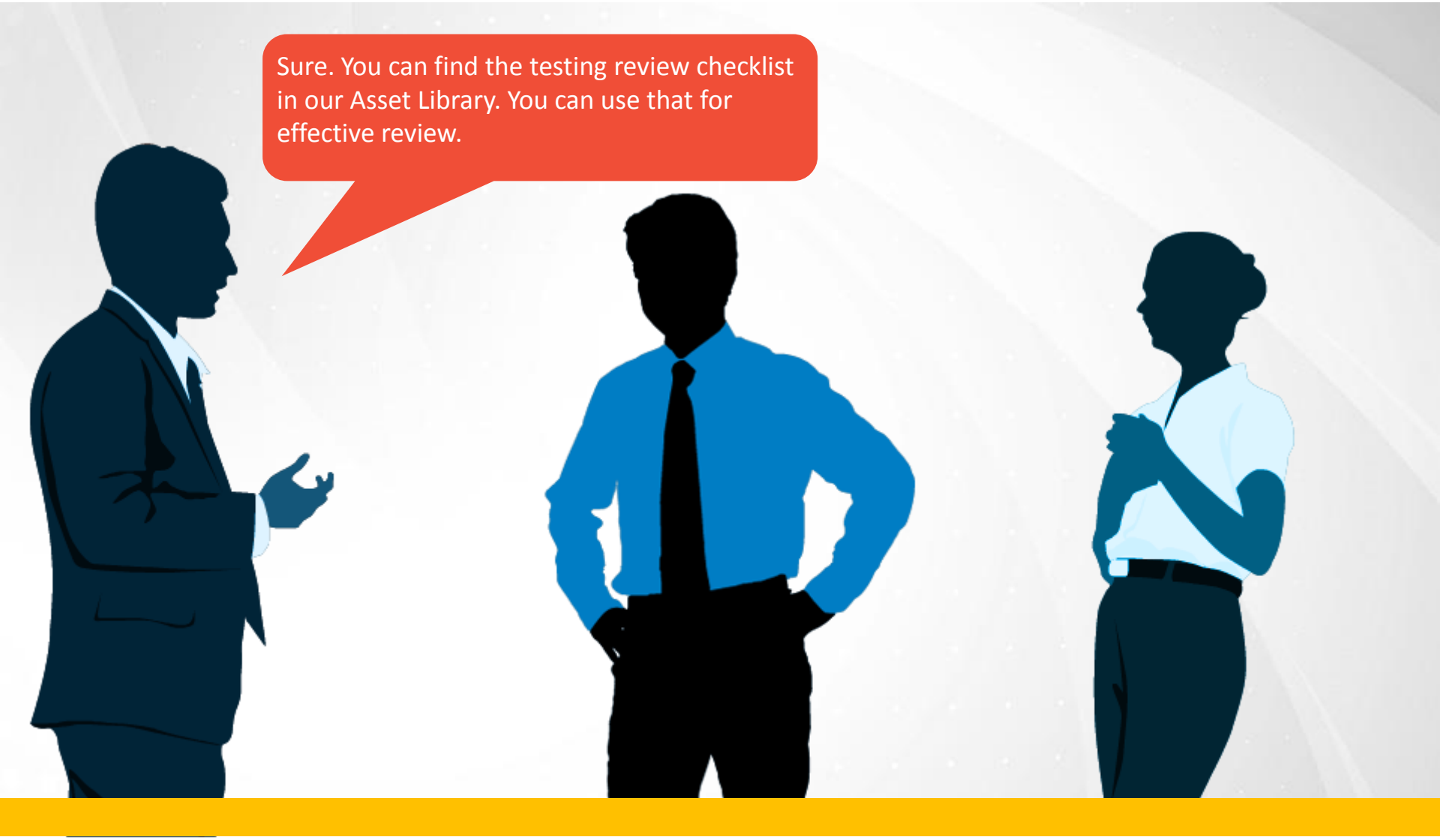


© Copyright IBM Corporation 2015

Project A - Peer Review (continued)

© Copyright IBM Corporation 2015

Project A - Peer Review (continued)

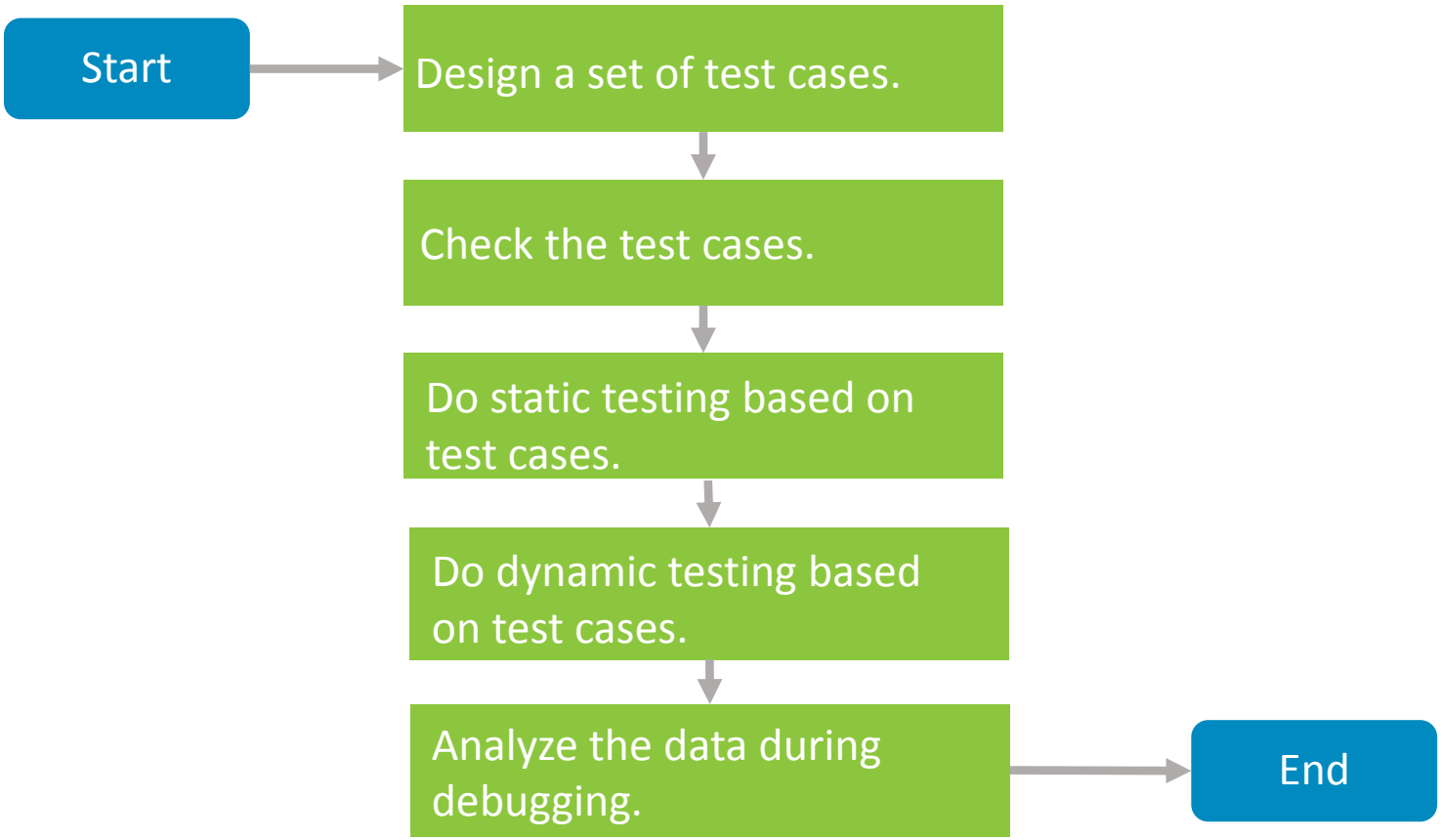


© Copyright IBM Corporation 2015

Project A - Peer Review (continued)

© Copyright IBM Corporation 2015

Simple practical **way** for **testing** and **debugging**



© Copyright IBM Corporation 2015

Peer Review Scenario

Based on the scenario on Lisa and Tom's peer review, there are two things that came up (if we look at the test case closely):

- ✓ Spelling of “Error” in error message is wrong.
- ✓ Error message has been copy-pasted. Second error message needs to be changed.

© Copyright IBM Corporation 2015

Review of Test cases example

A tester has to test for a function as:

Function to send order by Email

- Fields
- Email ID
- Recipients name
- Click Button
- Send

Test case

- Enter the Email ID of person
- Enter the recipient name
- Click on Submit button

© Copyright IBM Corporation 2015

Review of Test cases example (continued)

Test Suite ID	TC001.
Test Case ID	TCHome_01
Test Case Summary	To send an email to recipient on the order placed
Related Requirement	NA
Prerequisites	Order should be placed.
Steps to execute	Enter Email id, Enter recipient name, and click Submit .
Test Data	The test data, or links to the test data, that are to be used while conducting the test.
Expected Result	The email is sent
Actual Result	The actual result of the test; to be filled after executing the test.
Status	Pass or Fail. Other statuses can be ‘Not Executed’ if testing is not performed and ‘Blocked’ if testing is blocked.
Remarks	Any comments on the test case or test execution.
Created By	The name of the author of the test case.
Date of Creation	The date of creation of the test case.
Executed By	The name of the person who executed the test.
Date of Execution	The date of execution of the test.
Test Environment	The environment (Hardware/Software/Network) in which the test was execute

© Copyright IBM Corporation 2015

Review comments

Spelling Recipient is not correct in the test case summary and steps to test.

Expected result should always be having "Should “added – like –”Email should be sent to recipient.”

© Copyright IBM Corporation 2015

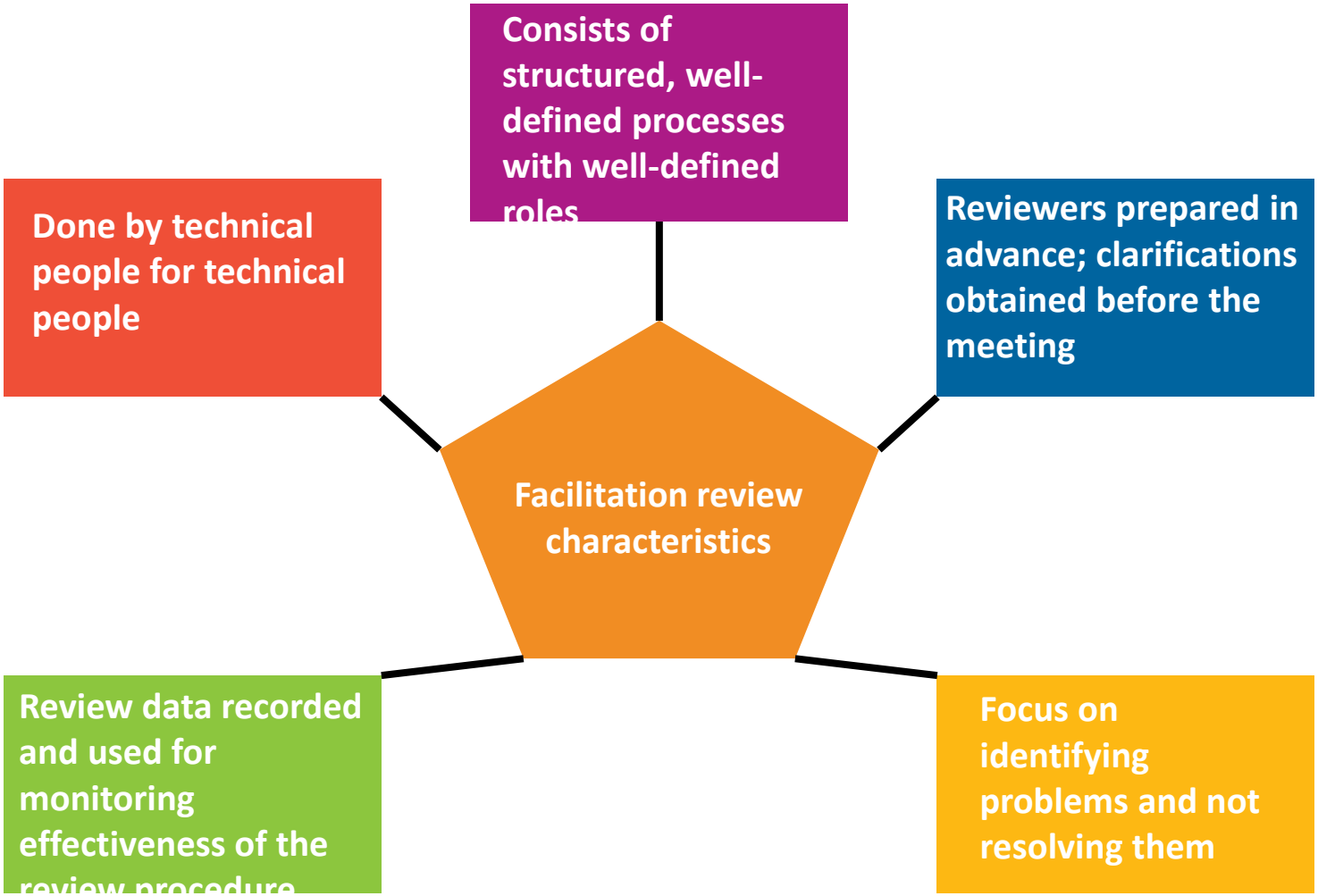
Overview of **Facilitation reviews**

Facilitation Reviews or Inspections are the most rigorous form of reviews which involve structured formal reviews.

In a lifecycle activity, the software inspection is the exit criteria.

© Copyright IBM Corporation 2015

Characteristics of Facilitation reviews



© Copyright IBM Corporation 2015

Fagan`s Inspection/Review

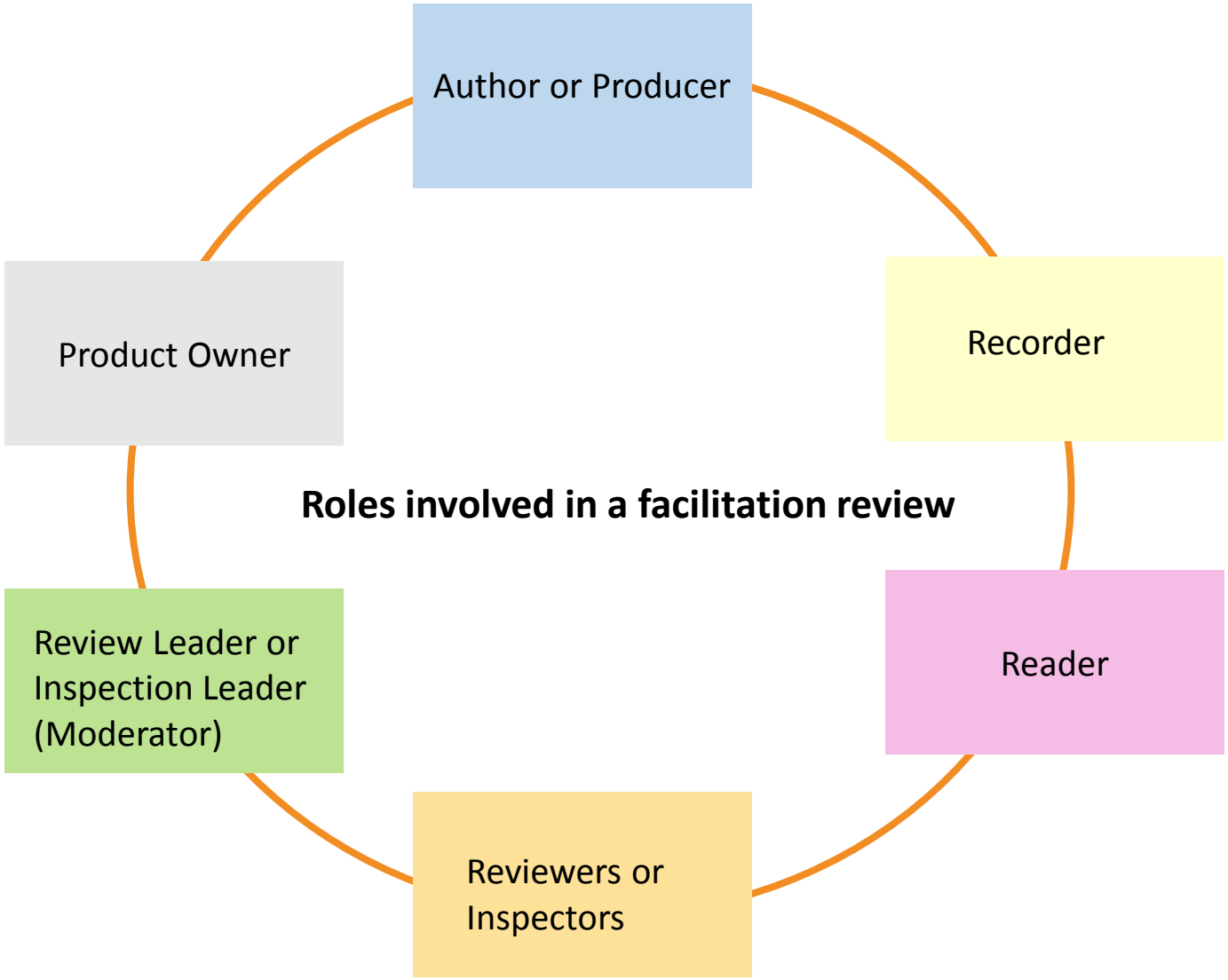
A **Fagan inspection** is a structured process of trying to find defects in development /Testing documents such as programming code, specifications, designs , Testing and others during various phases of the software development process. It is named after Michael **Fagan** who is credited with being the inventor of formal software **inspections**.

Examples of activities for which Fagan Inspection can be used are:

- Requirement specification
- Software/Information System architecture (for example DYA)
- Programming (for example for iterations in XP or DSDM)
- Software testing (for example when creating test scripts)

© Copyright IBM Corporation 2015

Facilitation reviews: **Roles and responsibilities**



© Copyright IBM Corporation 2015

Facilitation reviews: **Roles and responsibilities** (continued)

Author / Producer

- Distributes material to be reviewed
- Provides overview on product
- Provides clarifications as required
- Participates as an inspector
- Should not get defensive on an identified defect

Product Owner

- Initiates the review process
- Identifies a review leader
- Assesses the readiness for review of the work-product
- Identifies the reviewers and distributes the product for review
- Ensures rework is carried out
- Files the review records

Reader:

- Understands the material being reviewed
- Paraphrases while reading
- Sets the pace of inspection
- Participates as an inspector

© Copyright IBM Corporation 2015

Facilitation reviews: **Roles and responsibilities** (continued)

Reviewers / Inspectors:

- Review the product and communicate the comments to the review leader
- Identify one of the reviewers to consolidate the comments in the absence of the recorder
- At least one inspector (apart from the author) should be an expert on the product being reviewed.

Review Leader / Inspection Leader (Moderator)

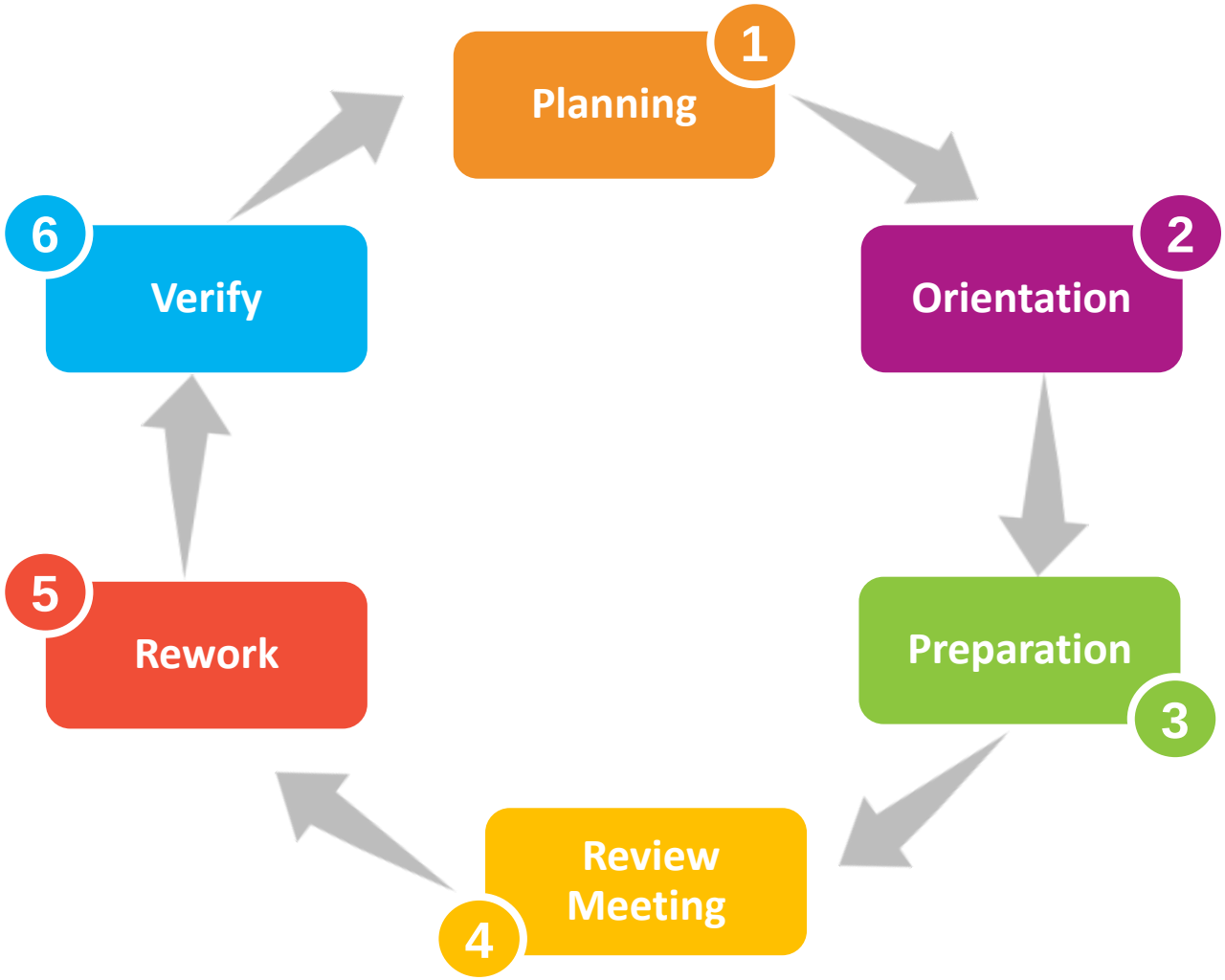
- Manages the entire process
- Should be skilled in the area of leadership and communication
- Identifies the reviewers, assigns their roles, and distributes the product
- Ensures preparedness
- Controls the review process, generates and distributes review forms or reports
- Participates as an inspector
- Takes responsibility for follow-up of rework and completion of the “Review Summary Report”
- Ensures that entry and exit criteria of the review process are met

Recorder:

- Records all review comments made at the meeting
- Classifies errors
- Takes confirmation on each recorded error
- Participates as an inspector

© Copyright IBM Corporation 2015

Facilitation Review Process



© Copyright IBM Corporation 2015

Project A - **Facilitation Review**

Mike is the Manager of Project A. **John** is the Project Leader for the same project and reports to **Mike**. There are several team members, two of whom are **Tom** and **Lisa**. The clients have just specified their requirements to the onsite coordinator **Jason**. In their weekly meeting, the following discussion ensues:

© Copyright IBM Corporation 2015

Project A - **Facilitation Review** (continued)

© Copyright IBM Corporation 2015

Project A - **Facilitation Review** (continued)

© Copyright IBM Corporation 2015

Project A - **Facilitation Review** (continued)

© Copyright IBM Corporation 2015

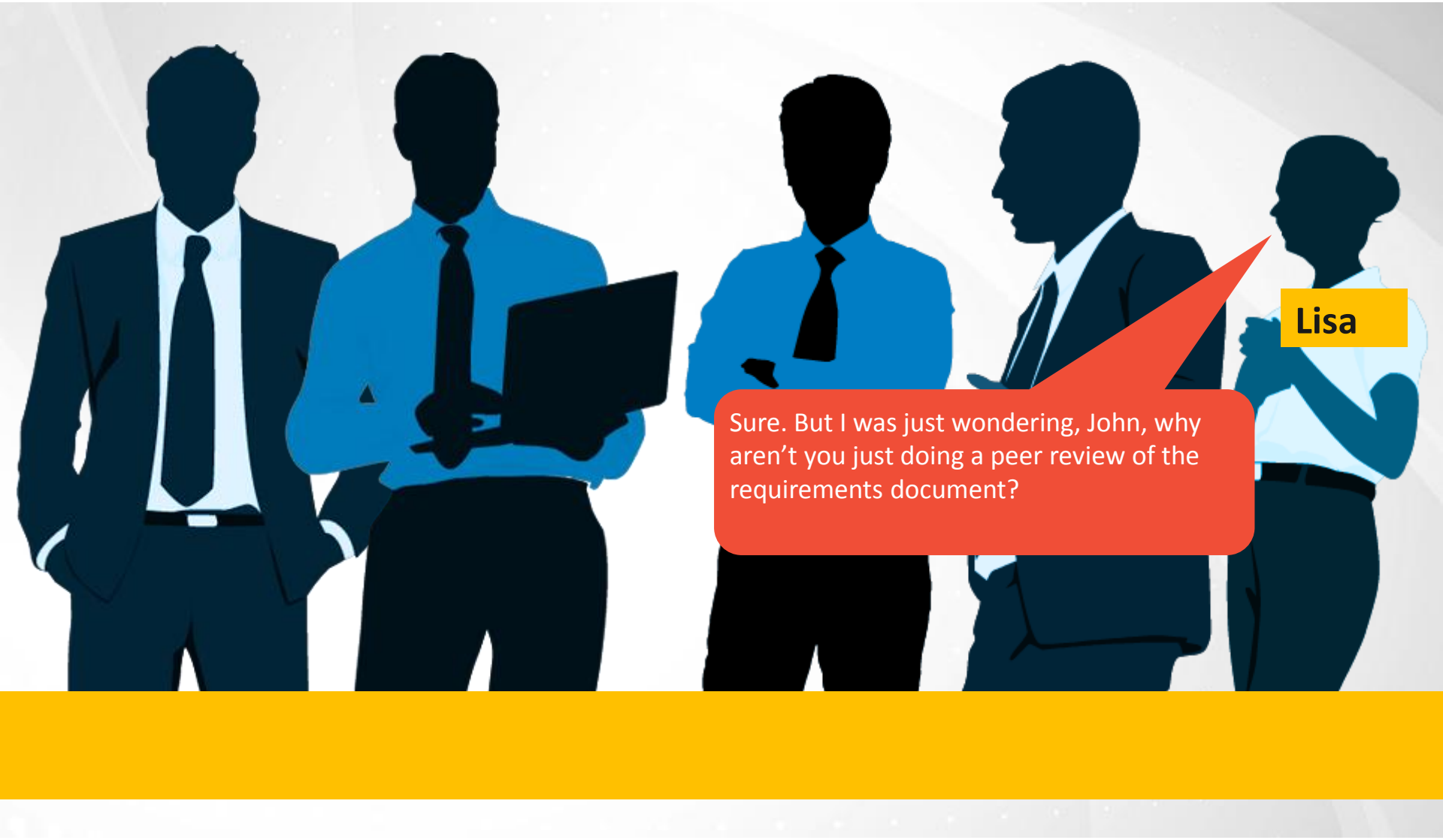
Project A - **Facilitation Review** (continued)

© Copyright IBM Corporation 2015

Project A - **Facilitation Review** (continued)

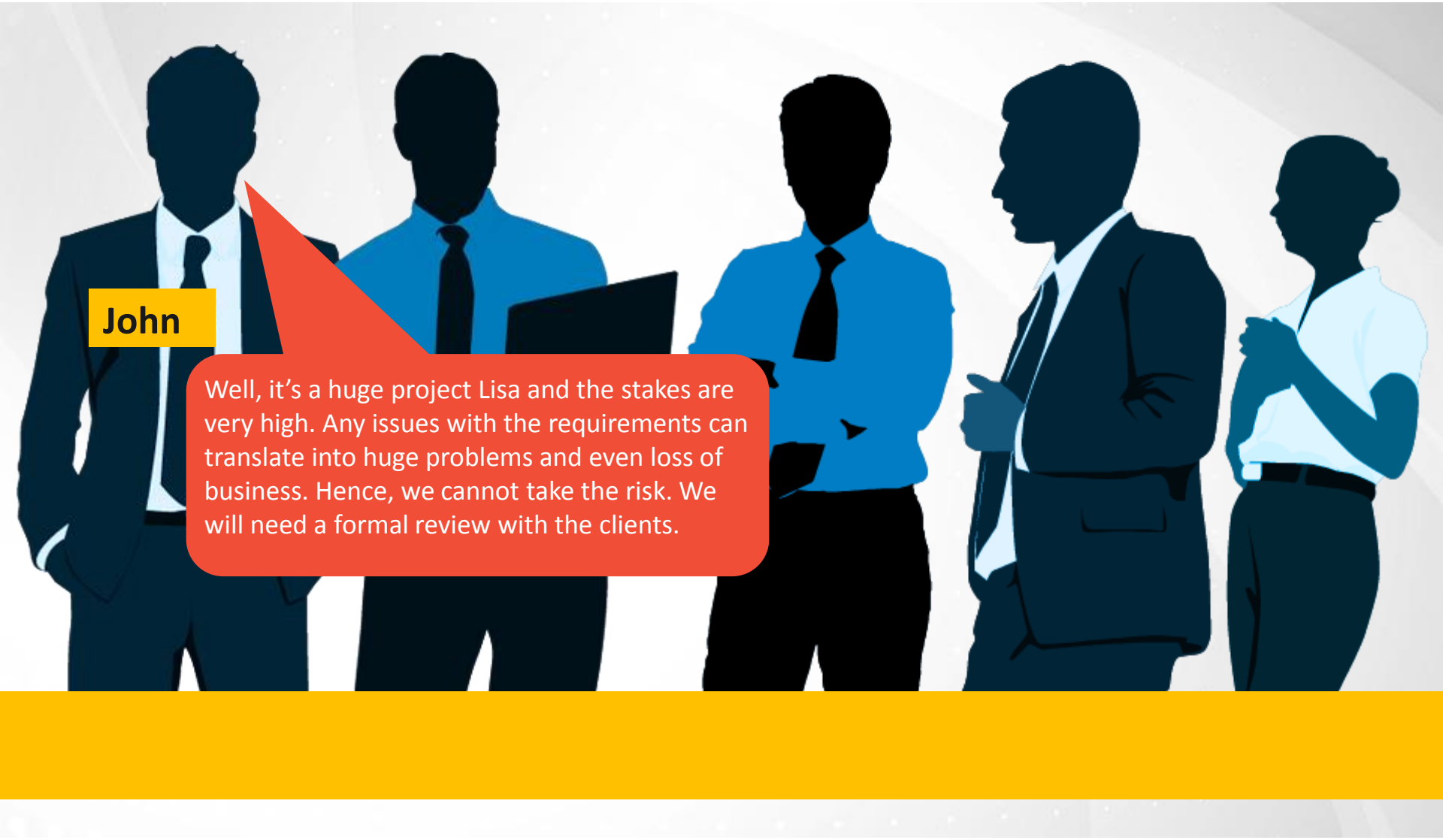
© Copyright IBM Corporation 2015

Project A - **Facilitation Review** (continued)



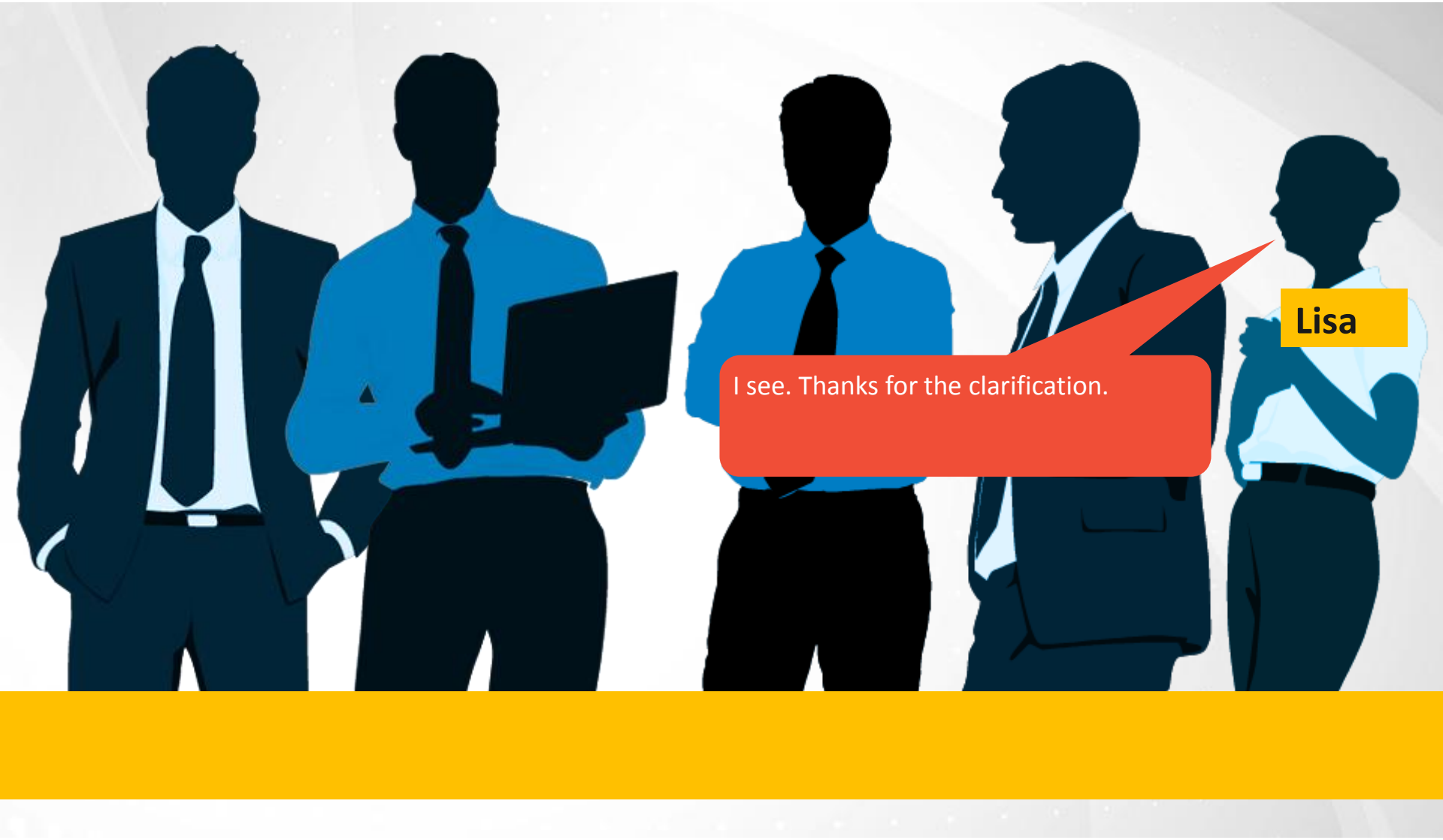
© Copyright IBM Corporation 2015

Project A - **Facilitation Review** (continued)



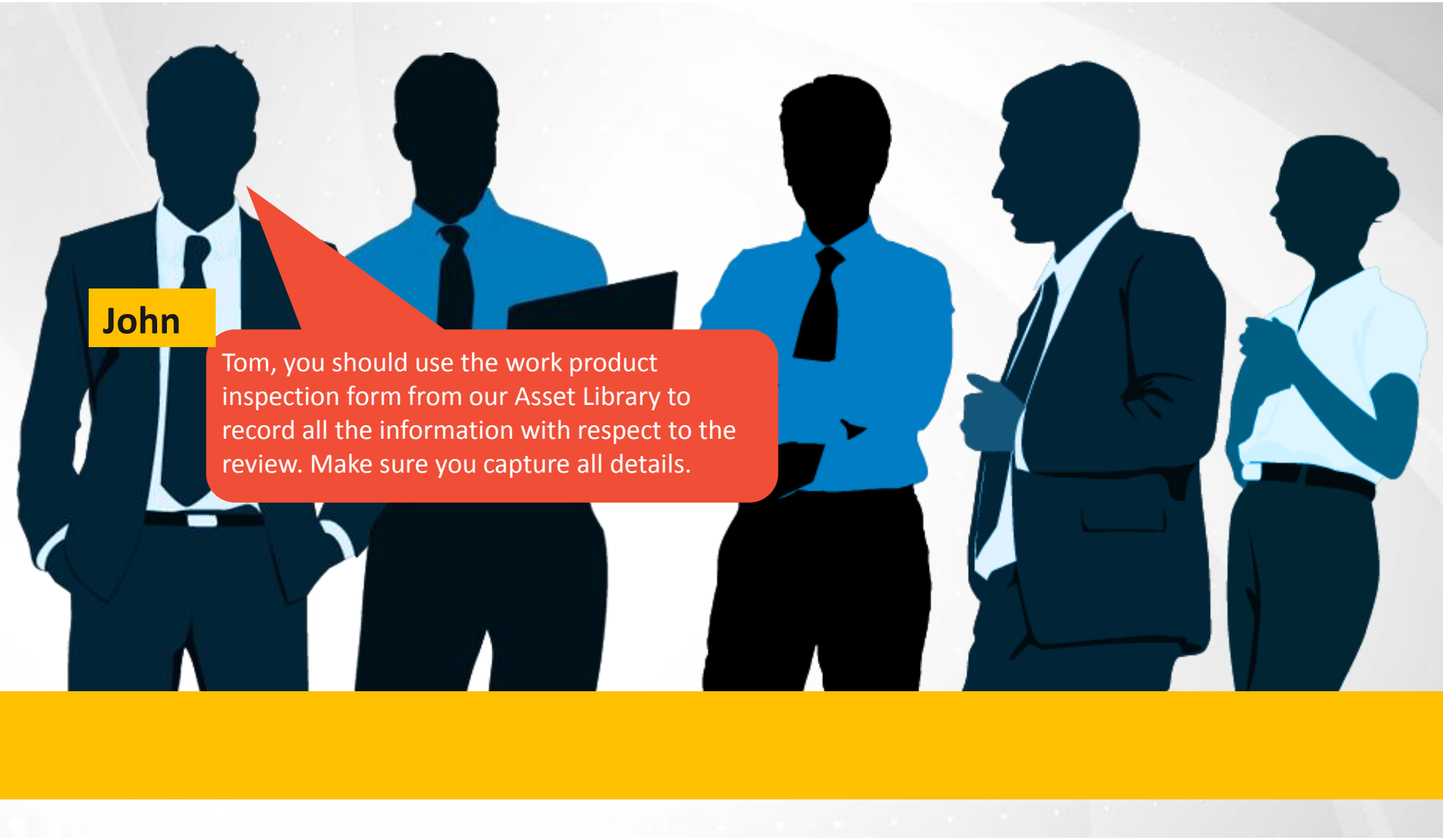
© Copyright IBM Corporation 2015

Project A - **Facilitation Review** (continued)



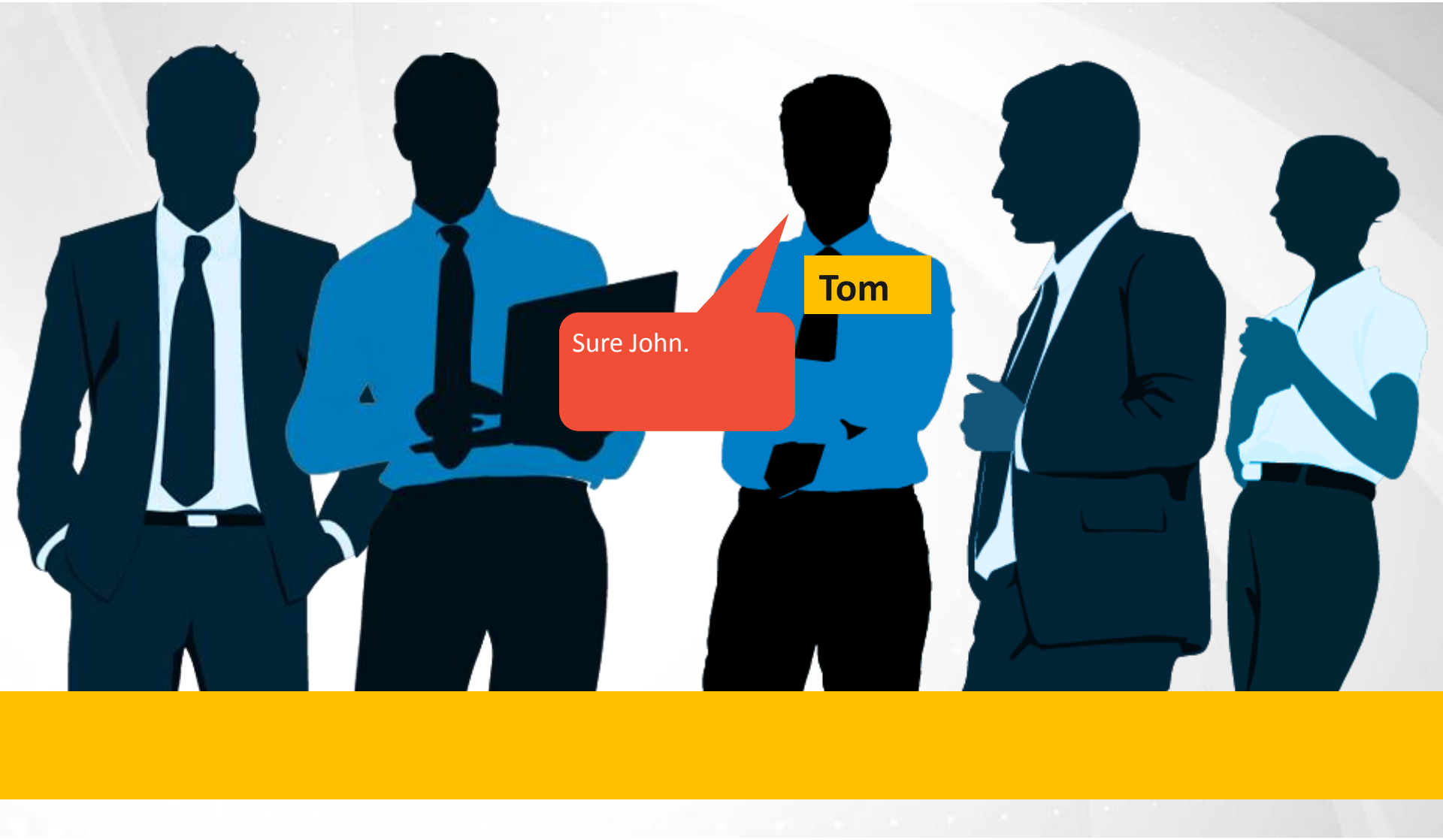
© Copyright IBM Corporation 2015

Project A - **Facilitation Review** (continued)



© Copyright IBM Corporation 2015

Project A - **Facilitation Review** (continued)



© Copyright IBM Corporation 2015

Project A - **Facilitation Review** (continued)

© Copyright IBM Corporation 2015

Facilitation Review and Peer Review: Comparison

Brighter Blue

Review Characteristics	Fagan's Review / Inspection	Peer review
Role of review leader	Required	Not required
Role of recorder	Required	Not required
Number of reviewers	>= 5 and <=10 (including the review leader)	>= 1
Advance distribution of review material (product to be reviewed and related documentation)	Required (at least 2 days in advance of review meeting)	Required (at least 2 days in advance of when the comments are due)
Formal review meeting	Required	Not required
Review moderation	Done by review leader	Done by product owner (in case multiple reviewers are used and a meeting is held)
Consolidation of review comments	Done by recorder	Done by product owner

© Copyright IBM Corporation 2015

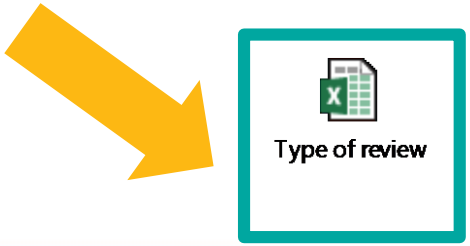
Facilitation Review and Peer Review: Comparison (continued)

Review Characteristics	Fagan's Review / Inspection	Peer review
Recording of review comments	Review defects / Resolution list must be used. (WPI Form)	Not required. (Number of reviews, review effort, and number of process improvements identified are recorded.)
Review summary report	Required, prepared by review leader	Required, prepared by product owner
Evidence of closure of review comments	Review leader verifies closure and signs on the review summary report.	Product owner completes the review summary report with all details after review closure.
Capturing review preparation time on review summary report	Required	Not required
Capturing review time on review summary report	Required	Required

© Copyright IBM Corporation 2015

Deciding type of review

Product	Review Type
Organization Level Plans	
1. First version of all plans (i.e. Process Improvement Plan, Quality Plan,Defect Prevention Plan,Technology Change Management Plan etc)	Fagan's Review
2. New Processes (i.e. First version of processes)	Fagan's Review
3. Revisions to existing Plans	Peer Review
4. Revisions to Processes	Peer Review
Proect Level Plans, Documents, Code etc	
1. Project Plans of Complex projects	Fagan's Review
2.SRS (Software Requirement Specifications) which require more than 25 person months of effort	Fagan's Review
3. All other documents/Code of Complex & Mega projects	Fagan's or Peer Review
4. Project Plan,documents and code of short term,Organic standard projects	Peer Review
5. Revised Plan of all projects	Peer Review
6. Revised documents and code of all projects	Peer Review
Testing Specific documents/deliverables	
1. Test Strategy /Master Test Plan	Fagan's Review
2. Test Plans	Fagan's or Peer Review
3. Test Specifications /Test Scenarios	Fagan's or Peer Review
4. Test cases/scripts	Fagan's or Peer Review



© Copyright IBM Corporation 2015

Work Product Inspection form: Snapshot (WPI tab)

B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
Defect No.	Page # /Location of defect	WPI Type	Severity	Defect Summary	Detail Description	Defect Creation Date	Originator name (who raised the defect)	Assigned To	Classification			Effort Spent	Comments	Defect Closure Date
									Defect type	Phase Detected	Status			
1	Page 9,section 5.2	Test Plan Review		Test Environment Section Missing	1. Specify under section 1.5 a. Unix Access b. Unix VPN c. DB Access 2. Specify WM Access Requirement under 1.6 as a. Remote VDI Access b. HPQC Work Space Access c. Project Specific Share Point Access	12/12/2011	Mohan N1	Srinivasa R Khatokar	Section Missing	Test Planning phase	O-Open			12/12/11
2														
4														
5														

© Copyright IBM Corporation 2015

© Copyright IBM Corporation 2015

Spot quiz

Brighter Blue

01 Which of the following is not a type of review?

A	Peer review
B	Test inspection
C	Process review
D	Fagan’s inspection

© Copyright IBM Corporation 2015

Spot quiz

Brighter Blue

02

Which of the following is a type of formal review?

A	Buddy review
B	Inspection
C	Test inspection
C	Peer review

© Copyright IBM Corporation 2015

Brighter Blue

03

Which of the following does not help to decide the type of review required?

A. Type of work product

B. Size of work product

C. Business criticality

D. Complexity of work product

© Copyright IBM Corporation 2015

Spot quiz

Brighter Blue

04 Which of the following is not a role in peer reviews?

A	Facilitator
B	Author
C	Reviewer
D	SQA

© Copyright IBM Corporation 2015

Spot quiz

Brighter Blue

05 In peer reviews, the author:

A	Approves the product
B	Initiates the review process
C	Reviews the product
D	Signs-off the review summary report

© Copyright IBM Corporation 2015

Spot quiz

Brighter Blue

06 Which of the following is not true about peer reviews?

A. The review activity takes more than 2 hours.

B. The reviewer identifies defects based on checklists.

C. Defects are documented in the WPI form.

D. The author takes up correction of defects.

© Copyright IBM Corporation 2015

Spot quiz

Brighter Blue

07 Which of the following is not a role in facilitation review?

A	Producer
B	Review Leader
C	Director
D	Reader

© Copyright IBM Corporation 2015

Spot quiz

Brighter Blue

08

In a facilitation review, the product owner:

A	Initiates the review process
B	Distributes material to be reviewed
C	Manages the entire review process and ensures that review process is adhered to
D	Participates as an inspector

© Copyright IBM Corporation 2015

Spot quiz

Brighter Blue

09

Which of the following is not true about a recorder in facilitation review?

A	Classifies errors
B	Understands the material being reviewed
C	Takes confirmation on each recorded error
D	Records all review comments

© Copyright IBM Corporation 2015

Measurements and Verifications

Defect classification is essential for analyzing defects and understanding their trend in order to prevent their occurrence in the future. Following measurements are collected from the process.

- ✓ Number of defects
- ✓ Effort experience on reviews
- ✓ Size of work product
- ✓ Rework effort
- ✓ Defect escapes (in form of Defect Origin)

© Copyright IBM Corporation 2015

Measurements and Verifications

The review comments or defects are classified as one of the following degree:

- ✓ Critical
- ✓ High
- ✓ Medium
- ✓ Low
- ✓ Suggestion

© Copyright IBM Corporation 2015

Measurements and Verifications

Defects are classified based on several factors

- ✓ Severity (High / Medium / Low)
- ✓ Type (for example, documentation error, logical error – refer to attachment below for details)
- ✓ Phase injected into system **during software development lifecycle**
- ✓ Phase detected into system during software lifecycle

Bitmap Image

© Copyright IBM Corporation 2015

© Copyright IBM Corporation 2015

Spot quiz

Brighter Blue

10 Which of the following measurements is not collected from a review process?

A Number of defects

B Review effort

C Rework effort

D Testing effort

© Copyright IBM Corporation 2015

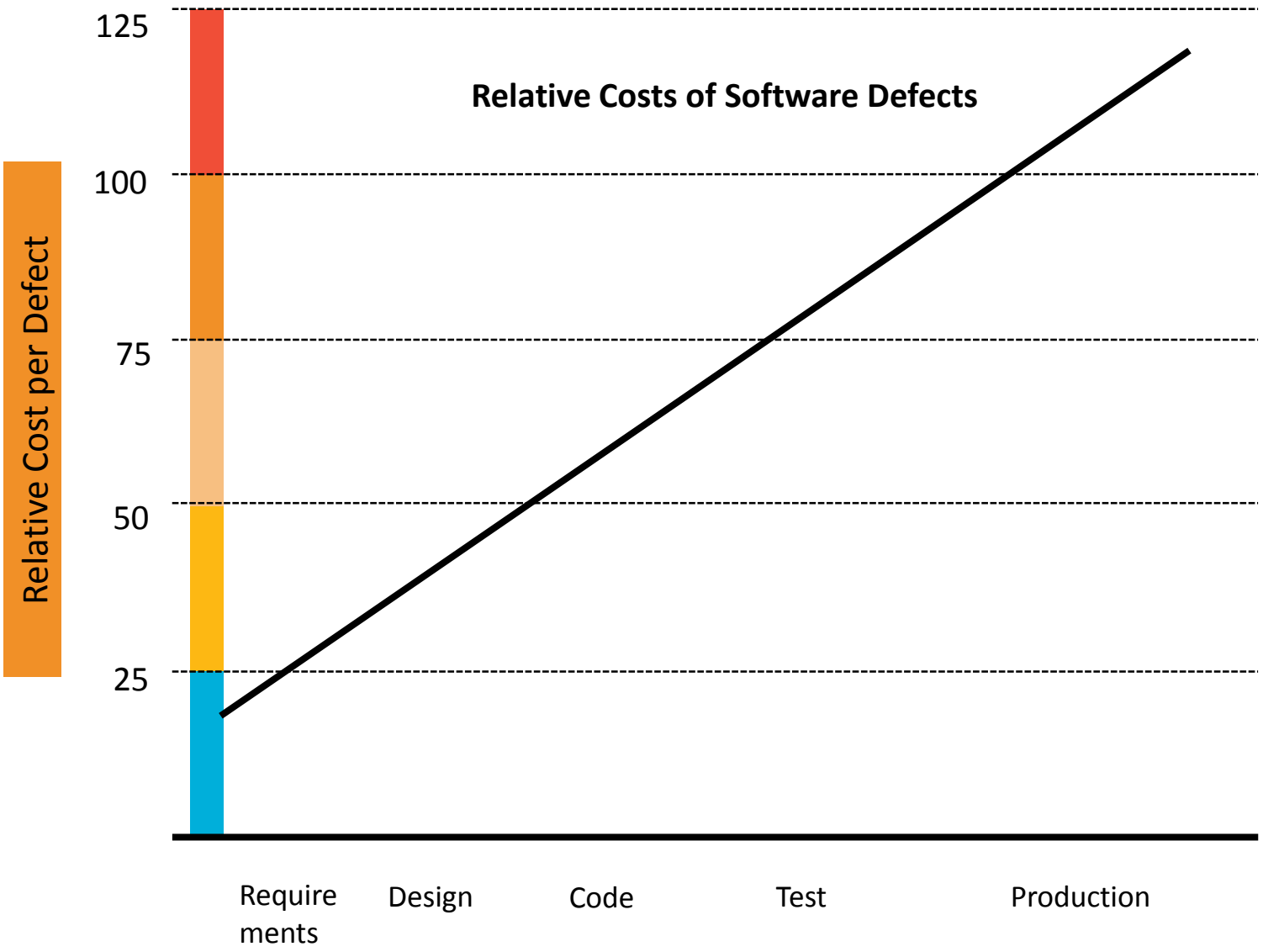
Benefits of Review

Brighter Blue

1. Reviews increase **quality**, which increases **management and customer satisfaction**.
2. Reviews help to find **errors** that cannot be detected through testing.
3. It is observed that reviews can find **60-100 % of all defects**.
4. Review data can assess or improve **quality of work product, software development process and review process.**
5. Reviews are a way of **using the diversity and power of a group of people.**
6. Reviews reduce rework and thus **reduce overall project costs**.

© Copyright IBM Corporation 2015

Benefits of reviews (continued)



© Copyright IBM Corporation 2015

Review checklist

Test Case Review Checklist			
	Yes	No	N/A
Test Overview			
1. Are the objectives of the testing activities clearly identified?			
2. Are all prerequisites and other related documents identified?			
3. Does the document describe the type(s) of tests to be addressed?			
4. Does the document describe the part of the target system to be tested?			
5. For the test type, does the document list all relevant tests necessary to be applied to the chosen part of the system?			
6. Are the validation tests clearly identified?			
7. Are the verification tests clearly identified?			
8. Are integration tests, including COTS components included?			
9. Is the test coverage sufficient?			
10. Is the test plan under version control?			
	Yes	No	N/A
Approach			
1. Does the document describe the tactics to be used for the testing activities?			
2. Are the test sequences and dependencies defined?			
3. Are the tactics and test types aligned with the Master Test Plan?			
	Yes	No	N/A
General			
1. Are the test instructions clear and easy to follow?			
2. Is the test as described likely to test what it intends to?			
3. Do the expected results reflect your expectation of the test result?			
4. Does every test case include a description of the expected output or results			
5. Are the test cases sufficiently documented so as to be 100 % reproducible?			
6. Do test cases include verification of proper return codes?			
7. Are the Test Cases under version control?			

© Copyright IBM Corporation 2015

Return on Investment (ROI)

Brighter Blue

ROI helps to justify the cost of reviews.

The model for ROI bases the savings on the cost avoidance associated with the detection and correction of defects earlier rather than later in the product evolution cycle.

A major defect that leaks from development to testing may cost as much as ten times to detect and correct.

Some defects, undetected in test, continue to leak from test to customer use, and may cost an additional ten times to detect and correct.

A minor defect may cost two to three times to correct later.

Defects leaking from code to test cost nine times more to detect and correct, and defects leaking from test to the field cost thirteen times more.

© Copyright IBM Corporation 2015

Activity

ABC software technology is working on making banking software. The team consists of a Test manager (Thomas), Tester (Gerry) & a Reviewer (Anne). Thomas provides a scenario to Gerry & asks him to prepare test cases & get them reviewed by Anne before starting the testing.

Scenario Details:
The login window of banking application consists of the following three fields.
a). User Name
b). Password
c). OK and Cancel button.

When a user gives the following URL https://www.abcbank.com, a login page opens & prompts for login name & password. When user provides the correct login & password & press ok button, the user account details will be displayed.

Review the above scenario keeping in mind informal & formal reviews process & document the positive & negative test cases.

Tip: Gerry should incorporate valid suggestions that came during review by Anne.

© Copyright IBM Corporation 2015

© Copyright IBM Corporation 2015

Spot quiz

Brighter Blue

11 Review data can assess / improve the quality of all but one of the following:

A	Work product
B	Software development process
C	Reviewer
D	Review process

© Copyright IBM Corporation 2015

Spot quiz

Brighter Blue

12 ROI stands for:

A	Return Of Investment
B	Return On Investment
C	Return On Inspection
D	Return Of Inspection

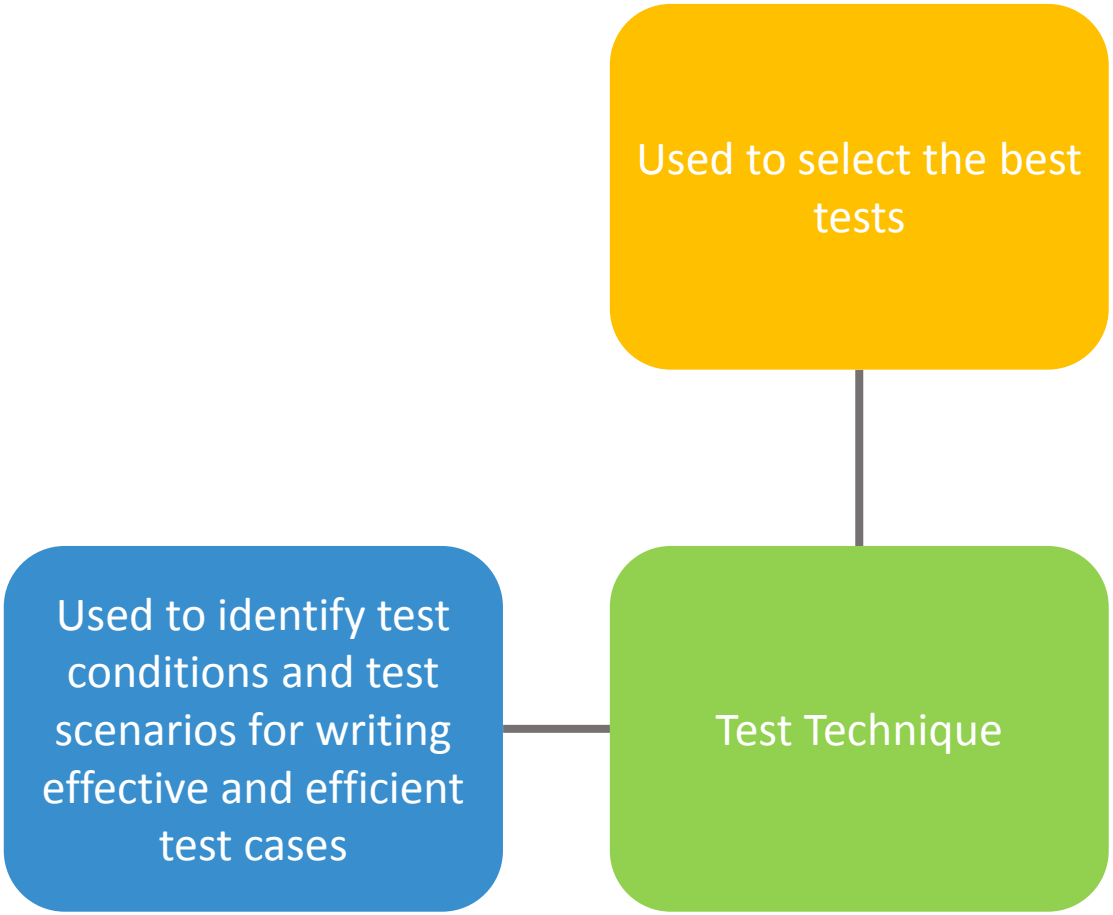
© Copyright IBM Corporation 2015

Brighter Blue

04 Testing Design Techniques and Approach

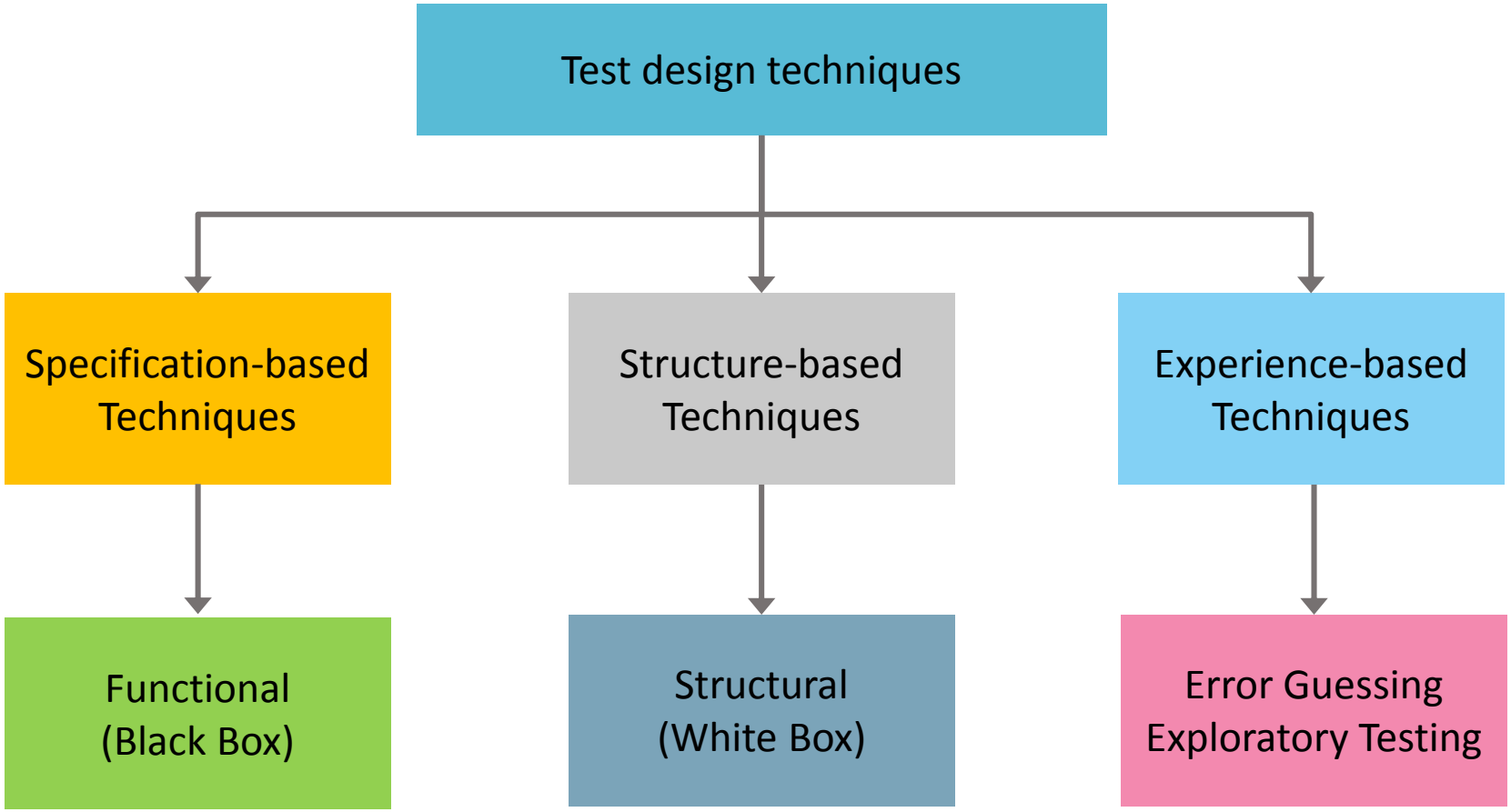
IBM

What is a **test technique**?



© Copyright IBM Corporation 2015

Categories of test design techniques



© Copyright IBM Corporation 2015

Specification-based techniques

Use models (informal or formal) from which test cases can be derived

Are based on functional / non-functional requirements

Are also called Black-box techniques

© Copyright IBM Corporation 2015

Structure-based techniques

Information about how the software is constructed is used to derive the test cases, for example, code and design.

Coverage of the software can be measured for existing test cases, and further test cases can be derived systematically to increase coverage.

Structure based techniques are also called White-box or Glass-box techniques

© Copyright IBM Corporation 2015

Experience-based techniques

Knowledge and experience of people like testers, developers, users are used to derive the test cases.

Knowledge of software, its usage, its environment, likely defects, and its distribution is required for deriving test cases.

© Copyright IBM Corporation 2015

Specification-based or black-box techniques

Also called functional, behavioral or data-driven, input / output driven, or requirements-based testing

Determines whether the combination of inputs and operations produce expected results

Test conditions are derived from the specifications.

Input data is critical.

Focus is on evaluating the function.

© Copyright IBM Corporation 2015

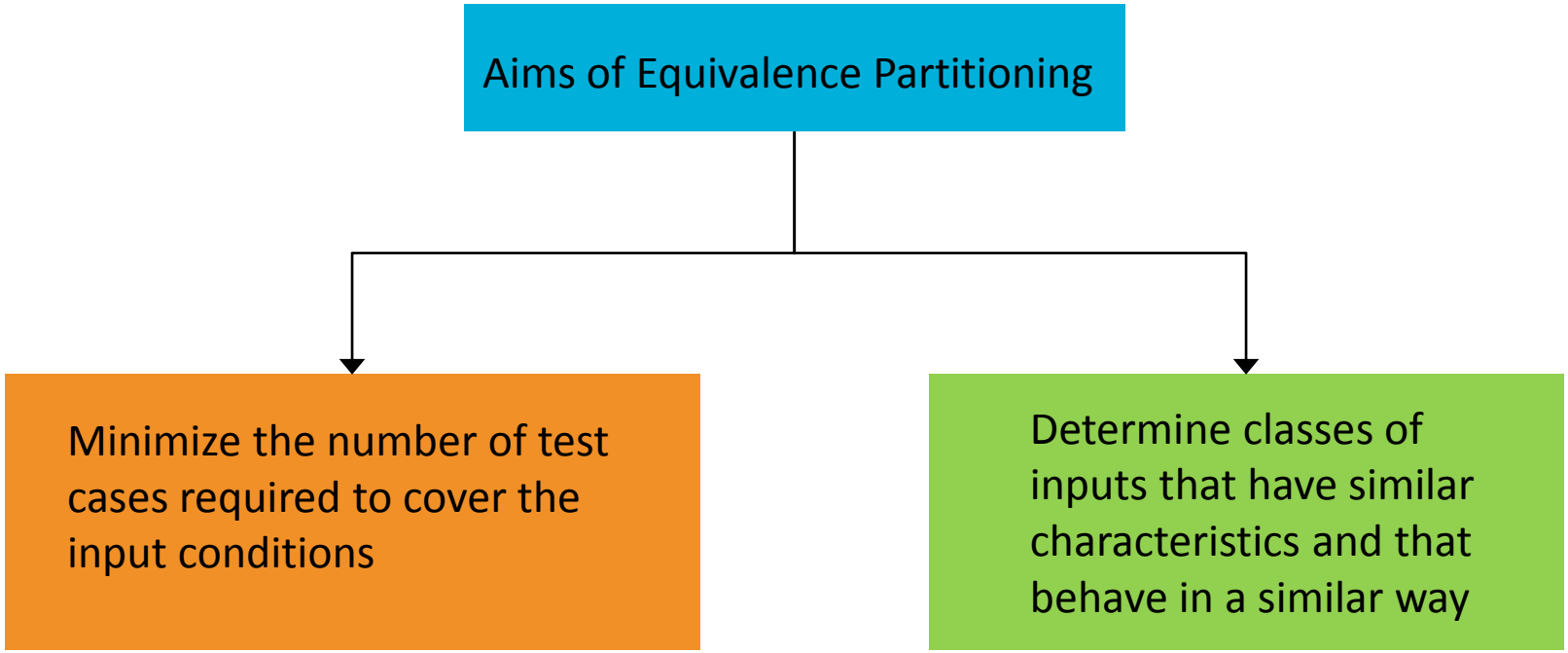
Types of **Specification-based** or **Black-box** techniques

Brighter Blue

1. Equivalence partitioning
2. Boundary value analysis
3. Decision table testing
4. Orthogonal Array Testing (OAT)
5. State transition testing
6. Use case testing

© Copyright IBM Corporation 2015

Equivalence Partitioning



© Copyright IBM Corporation 2015

Equivalence Partitioning or equivalence classes

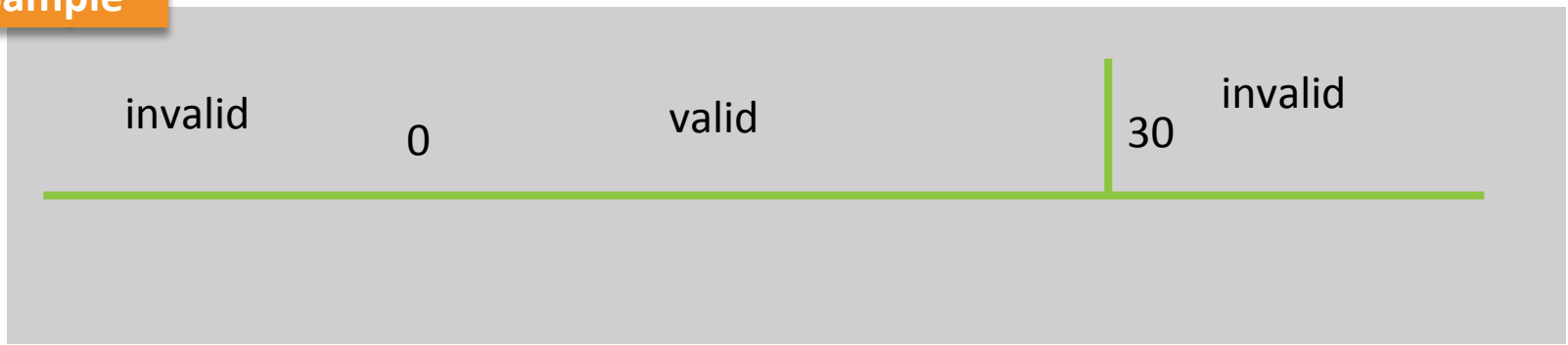
Brighter Blue

Equivalence Partitioning or Equivalence Class defines set of valid and invalid states for input condition.

The inputs and outputs are divided into equivalent classes.

Equivalence Partitioning (EP) is applicable at all levels of testing.

Sample



© Copyright IBM Corporation 2015

Equivalence Partitioning or equivalence classes (continued)

Example:

Consider the tax slabs:

Salary range	Tax
1000 to 2500	No tax
2501 to 4000	5% of the salary

In Equivalence, we divide the range as the following classes:

C1 – Invalid values:

- From 0 to 999

C2 – No tax:

- From 1000 to 2500

C3 – Tax 5% of the salary:

- From 2501 to 4000

© Copyright IBM Corporation 2015

Equivalence Partitioning: **Advantages** and **disadvantages**

Brighter Blue

Advantage	Disadvantage
▪ By identifying and testing one input of each partition, we gain a 'good' coverage with 'small' number of test cases. ▪ Testing one input of a partition should be as good as testing any inputs of the partition.	▪ Does not test every input ▪ No guidelines for choosing inputs ▪ Heuristic-based approach ▪ Very limited focus ▪ Not guaranteed that the system under test treats all sets of an equivalence class in the same way

© Copyright IBM Corporation 2015

Boundary Value Analysis

Brighter Blue

- Input values at the boundaries of the input domain are tested.
- Values at and just beyond the boundaries of the input domain are used to generate test cases to ensure proper functionality of the system.
- Variant and refinement of Equivalence Partitioning
- Also covers output conditions
- Usage of <, =, >

© Copyright IBM Corporation 2015

Boundary Value Analysis (continued)

Brighter Blue

Defects tend to lurk near boundaries.

Boundaries are good places to look for defects.

Test values at and on either side of each valid boundary

BVA enhances Equivalence Partitioning by selecting elements just on, and beyond the borders of each Equivalence Class.

Applies at all levels of testing

Sample

Invalid Boundaries	valid	Invalid Boundaries
-1(< 0) 0 1	29 30 31	

© Copyright IBM Corporation 2015

Boundary Value Analysis: **Example**

Consider the tax slabs:

Salary range	Tax
1000 to 2500	No tax
2501 to 4000	5% of the salary

In Equivalence, we divide the range as the following classes:

C1 – Invalid values:
- From 0 to 999

C2 – Valid values - No tax:
- From 1000 to 2500

C3 – Tax 5% of the salary:
- From 2501 to 4000

In Boundary Value Analysis, analyze the boundaries with the following values:

Lower boundary (1000): → 1000-1, 1000, 1000+1

Upper boundary (2500): → 2500-1, 2500, 2500+1

© Copyright IBM Corporation 2015

Boundary Value Analysis: **Advantages** and **disadvantages**

Brighter Blue

Advantage	Disadvantage
▪ Very good at exposing potential user interface / user input problems ▪ Very clear guidelines on determining test cases ▪ Very small set of test cases generated	▪ Does not test all possible inputs ▪ Does not test dependencies between combinations of inputs

© Copyright IBM Corporation 2015

Decision Table Testing

Brighter Blue

A table containing combinations of true and false

Each column corresponding to a business rule

Requirements containing logical conditions captured

At least one test case for each column

© Copyright IBM Corporation 2015

Decision table testing

Marks	Rule 1	Rule 2	Rule 3
0 - 35	Y	N	N
36 - 65	N	Y	N
66 - 74	N	N	Y
Result			
Fail	Y	N	N
Pass	N	Y	Y
First Class	N	N	Y

© Copyright IBM Corporation 2015

Orthogonal Array Technique (OAT)

A systematic, statistical way of testing

Could be applied in user interface, testing, system testing, regression, testing, and configuration testing

Orthogonal arrays are two-dimensional arrays of numbers

© Copyright IBM Corporation 2015

Orthogonal Array Technique (OAT): **Terminology**

Brighter Blue

Terminology for working with orthogonal arrays:

Runs	Runs are the number of rows in the array. This directly translates to the number of test cases
Factors	Factors are the number of columns in an array. This directly translates to the maximum number of variables that can be handled by this array.
Levels	Levels are the maximum number of values that can be taken on by any single factor.
Strength	Strength is the number of columns it takes to see each of the levels-^{factors} possibilities equally often

© Copyright IBM Corporation 2015

Example of OAT

Parameters for placing a telephone call:

Call Type	Billing	Access	Status
Local	Caller	Loop	Success
Long Distance	Collect	ISDN	Busy
International	800	PBX	Blocked

The above table defines:

4 Factors (Parameters)

3 Levels (Values)

81 Runs (3^4 = 81 different scenarios)

© Copyright IBM Corporation 2015

Pair-wise test cases for placing a phone call using OAT

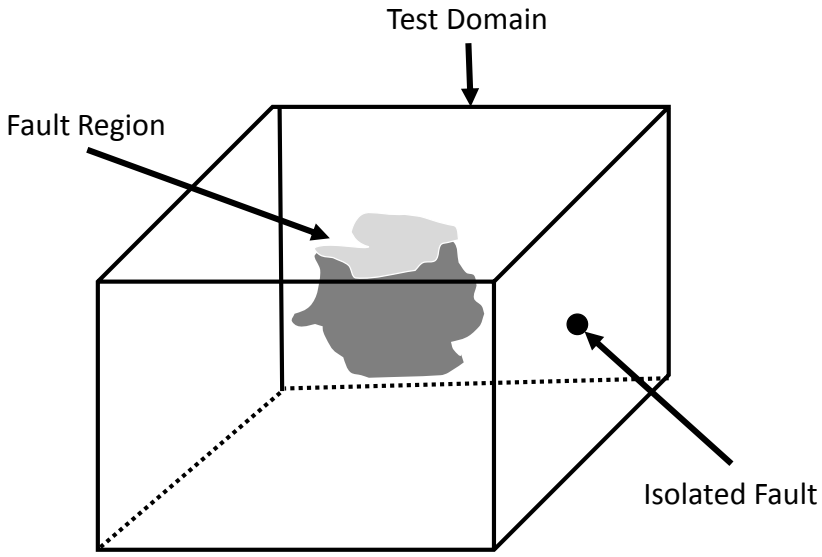
Brighter Blue

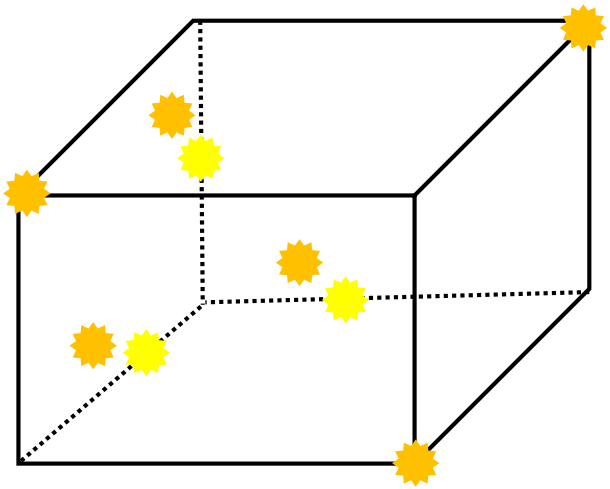
Call Type	Billing	Access	Status
Local	Collect	PBX	Busy
Long Distance	800	Loop	Busy
International	Caller	ISDN	Busy
Local	800	ISDN	Blocked
Long Distance	Caller	PBX	Blocked
International	Collect	Loop	Blocked
Local	Caller	Loop	Success
Long Distance	Collect	ISDN	Success
International	800	PBX	Success

© Copyright IBM Corporation 2015

Traditional test case **versus** test case based on OAT

Traditional Manual Data-Driven Testing	Test Case based on OAT
▪ Many scenarios to execute and time consuming ▪ High probability of some data getting missed out (during cons / test phase)	▪ Makes sure test cases are evenly / thoroughly distributed





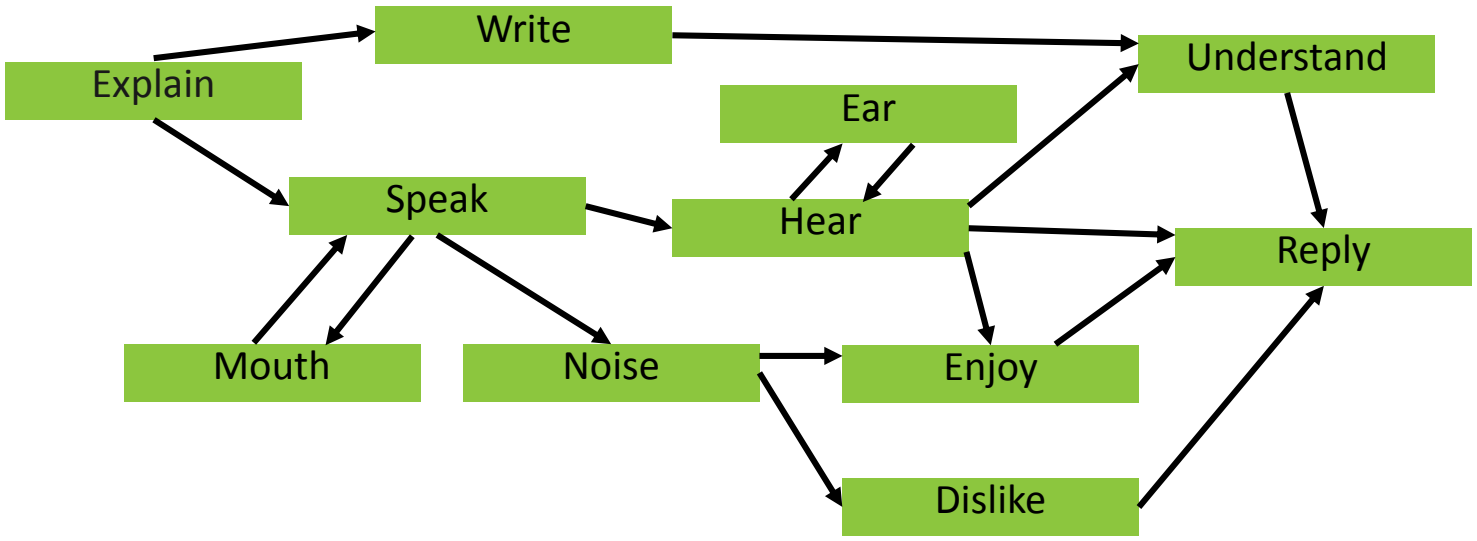
© Copyright IBM Corporation 2015

Cause-Effect Graphing

Aims of Cause-Effect Graphing

- A graphical representation of inputs (causes) with their associated outputs (effects) used in designing test cases
- Brainstorm session to find out the related causes and effects until we reach the goal
- Producing non-redundant, high-yield tests

Example:



Cause-Effect graph

© Copyright IBM Corporation 2015

Error Guessing

Brighter Blue

Aim of the Error Guessing technique

Using experience and intuition of the tester to postulate what faults might occur, and to design tests specifically to expose them

Important aspects of the procedure are as follows:

It involves making an itemized list of the errors expected to occur in a particular area of the system and then designing a set of test cases to check for these expected errors.

It is more testing art than testing science, but can be very effective given a tester familiar with the history of the system.

© Copyright IBM Corporation 2015

Error Guessing: **Advantages** and **disadvantages**

Advantage	Disadvantage
▪ Can keep track of defect history ▪ Lists out all the possible error combinations	▪ Involves tester's experience and ability

© Copyright IBM Corporation 2015

Race Conditions

Aims of Race Conditions technique

- A test case design technique which detects the system attempting to perform two or more operations at the same time
- Should track the possibilities that the developer has incorrectly assumed that a particular event will always happen before another
- Tests possible in multithreading, multiprocessing, and distributed computing

Procedure:

List the possible areas wherein the race conditions occur.

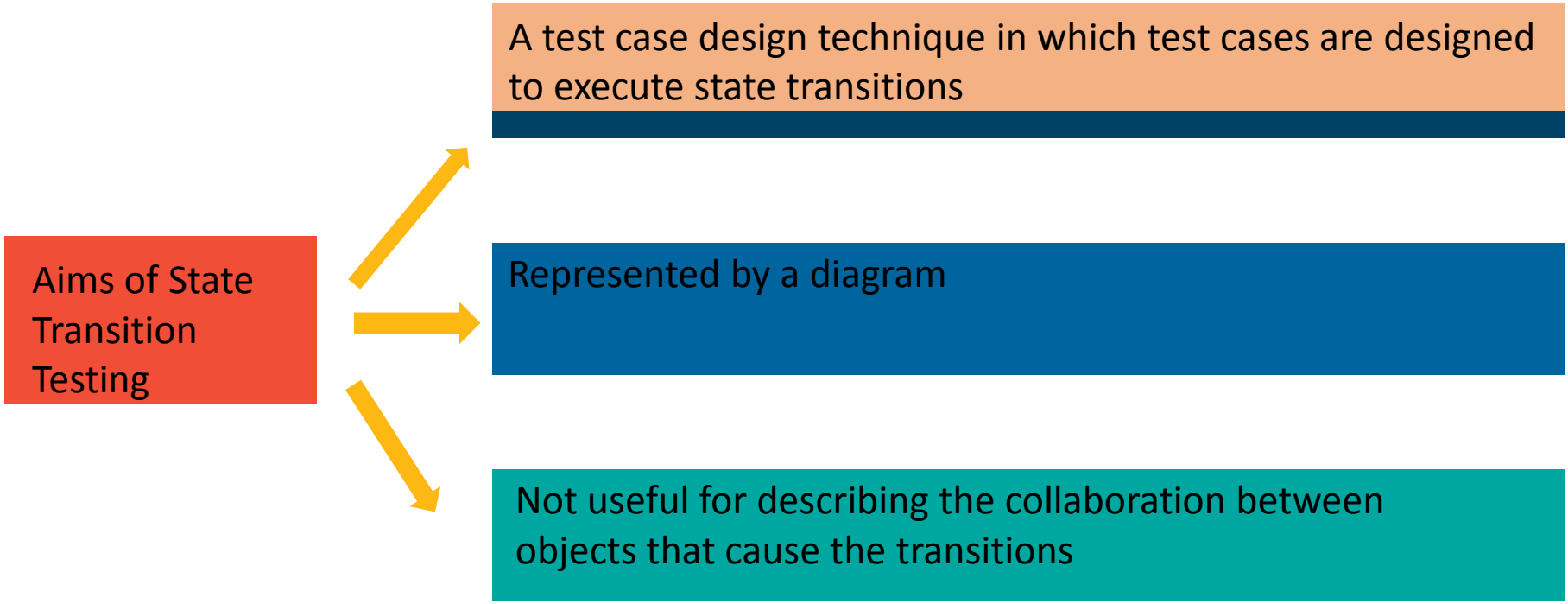
© Copyright IBM Corporation 2015

Race Conditions: **Advantages** and **disadvantages**

Advantage	Disadvantage
▪ Used for security testing ▪ Can be used for testing under heavy load with multiple programs running	▪ Difficult to detect the fault ▪ Easily prone to hackers

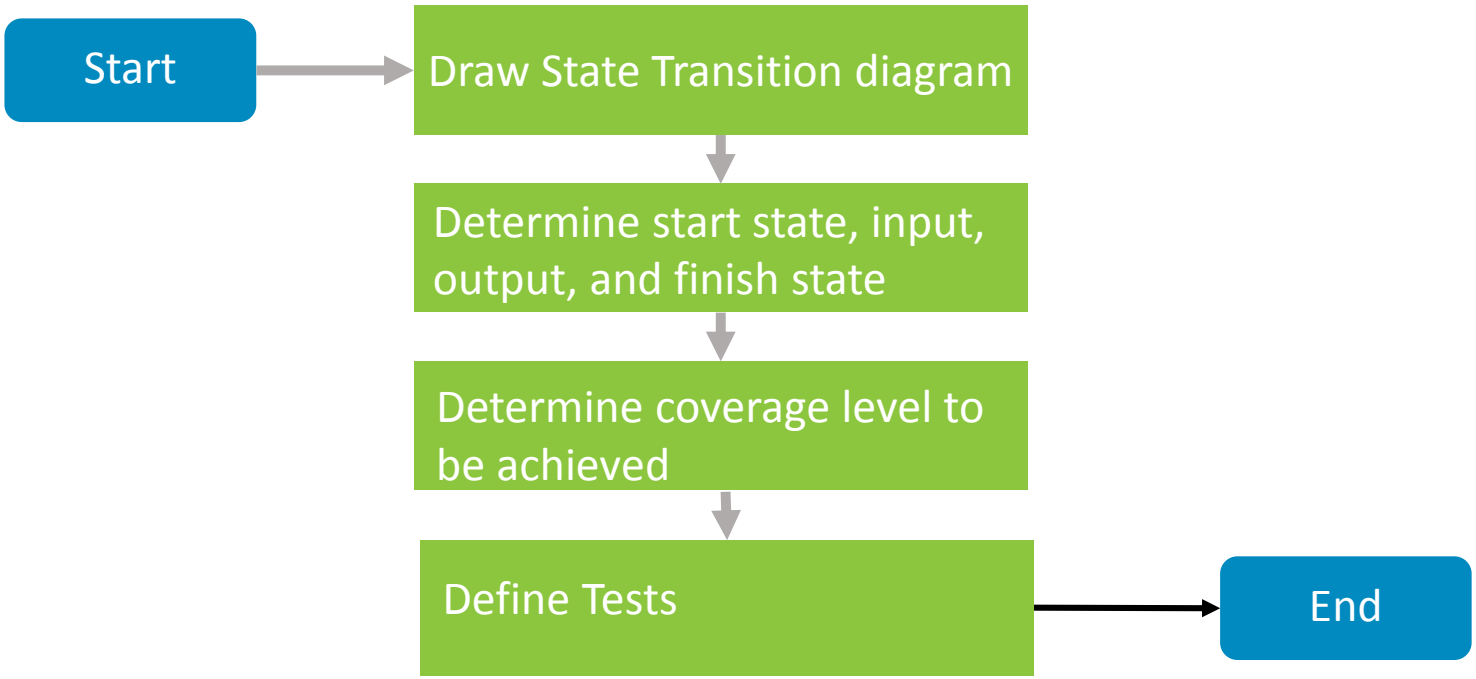
© Copyright IBM Corporation 2015

State Transition Testing: (1 of 4)



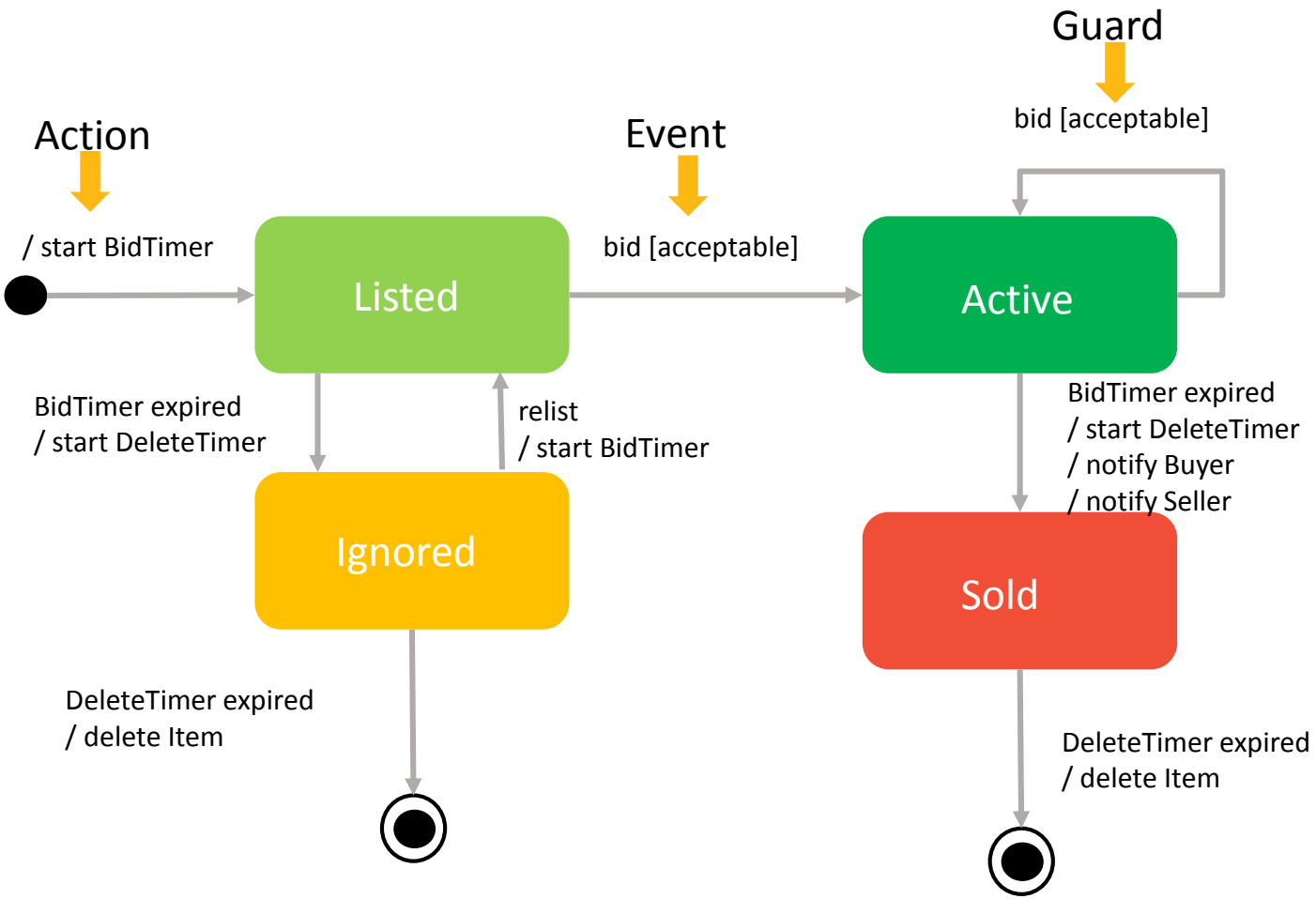
© Copyright IBM Corporation 2015

State Transition Testing: **(2 of 4)**



© Copyright IBM Corporation 2015

State Transition Testing: **(3 of 4)**



© Copyright IBM Corporation 2015

State Transition Testing: **(4 of 4)**

State:	▪ A condition during the life of an object in which it satisfies some conditions, performs some actions, or waits for some events
Event:	▪ An occurrence that may trigger a state transition
Guard:	▪ A Boolean expression which, if true, enables an event to cause a transition
Transition:	▪ The change of state within an object
Action:	▪ One or more actions taken by an object in response to a state change

© Copyright IBM Corporation 2015

Example of State Transition Testing

Consider the example of a light switch:

Initially the switch is OFF and the light is OFF. Now, the input is switch ON, and the light also switches ON. The state transition, with the initial and final states, is given in the following slide.

© Copyright IBM Corporation 2015

Representation of state transition

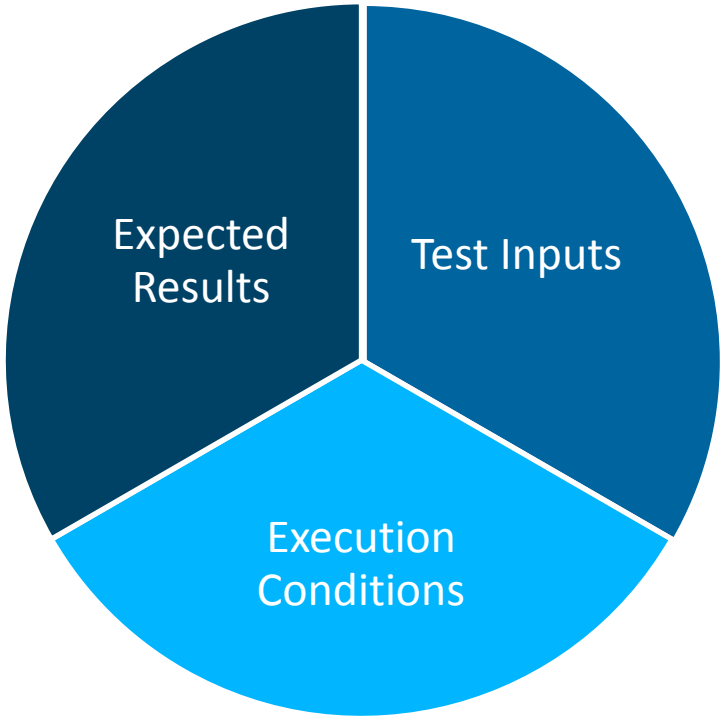
TEST

	STEP1	STEP2	STEP3
START STATE	OFF	ON	OFF
INPUT	SWITCH ON	SWITCH OFF	SWITCH ON
OUTPUT	LIGHT ON	LIGHT OFF	LIGHT ON
FINISH STATE	ON	OFF	ON

© Copyright IBM Corporation 2015

Test case **IEEE standard 610 (1990)**

Aim Developed for a particular objective, such as to exercise a particular program path or to verify compliance with a specific requirement



Test case IEEE standard 610 (1990)

© Copyright IBM Corporation 2015

What is a Test **procedure**?

Test Procedure

- The detailed instructions for the set-up, execution, and evaluation of results for a given test case.
- A test case may use more than one test procedures.

We will look at details of the procedures in the next few slides.

© Copyright IBM Corporation 2015

© Copyright IBM Corporation 2015

Spot quiz

Brighter Blue

01 Test case template consists of:

A	Use case step / ID / name
B	Bug ID
C	Step description
D	Assigned to

© Copyright IBM Corporation 2015

Spot quiz

Brighter Blue

02

In which type of testing technique do we only check the boundary values and + or – of those values?

A	Equivalence Partitioning
B	Decision Table Testing
C	Boundary Value Analysis
D	Orthogonal Array Technique

© Copyright IBM Corporation 2015

Spot quiz

Brighter Blue

03

Which of the following has detailed instructions for the set-up, execution, and evaluation of results for a given test case?

A	Test procedure
B	Test environment
C	Test technique
D	Test inputs

© Copyright IBM Corporation 2015

Spot quiz

Brighter Blue

04 Which of the following is not an aim of Cause – Effect Graphing?

A	A graphical representation of inputs with their associated outputs used in designing test cases
B	Brainstorm session to find out the related causes and effects until we reach the goal
C	Tracking possibilities of incorrect assumptions of a particular event always happening before another
D	Producing non-redundant, high-yield tests

© Copyright IBM Corporation 2015

Spot quiz

Brighter Blue

05

Which of the following is not an advantage of Boundary Value Analysis?

A	Very good at exposing potential user interface / user input problems
B	Very clear guidelines on determining test cases
C	Very small set of test cases generated
D	Lists out all the possible error combinations

© Copyright IBM Corporation 2015

Spot quiz

Brighter Blue

06 Which of the following are advantages of Equivalence Partitioning?

A. By identifying and testing one input of each partition, we gain a good coverage with small number of test cases.

B. Not guaranteed that the system under test treats all sets of an equivalence classes in the same way

C. Testing one input of a partition should be as good as testing any inputs of the partition.

© Copyright IBM Corporation 2015

05 Case Study

Activity

Brighter Blue

Role play: Day 1—Requirements review

Let us get started with the some real life case studies now. Here is what you need to do:

- Divide yourselves into groups as instructed by the facilitator
- Work within your group to assume any of the following roles:
  - Tester
  - SME
  - Observer
- Refer to the Case Study by clicking the embedded doc or the Case Study handout provided to you.
- Read the four use cases on Page 8 and 9
- Share your key takeaways from the role play with the class (30 mins)

Microsoft Word
97 - 2003 Documen

© Copyright IBM Corporation 2015
IBM

Role play: What will the Tester do?

When you play the role of the tester, you also have an SME and an observer in your team for this role play.

You need to:

- Review the test cases in the case study
- Ask questions to SMEs to get clarifications on the use cases as well
- Record the review observations and comments directly in RQM or in OPAL template first and then enter into RQM to maintain the records
- Document your observations from the conversation

© Copyright IBM Corporation 2015

Role play: What will the SME do?

- When you play the role of the SME, you also have a tester and an observer in your team for this role play

You need to:

- Read the test cases to the team and pose relevant queries to the tester
- Hear responses from the tester

© Copyright IBM Corporation 2015

Role play: What will the Observer do?

- When you play the role of the observer in this role play, you also have a tester and a SME in your team

You need to:

- Closely observe the dialogue between the tester and a SME
- Refer to the responses for the scenarios in your handout
- Document your feedback for the tester based on your observation of the role play

© Copyright IBM Corporation 2015

© Copyright IBM Corporation 2015