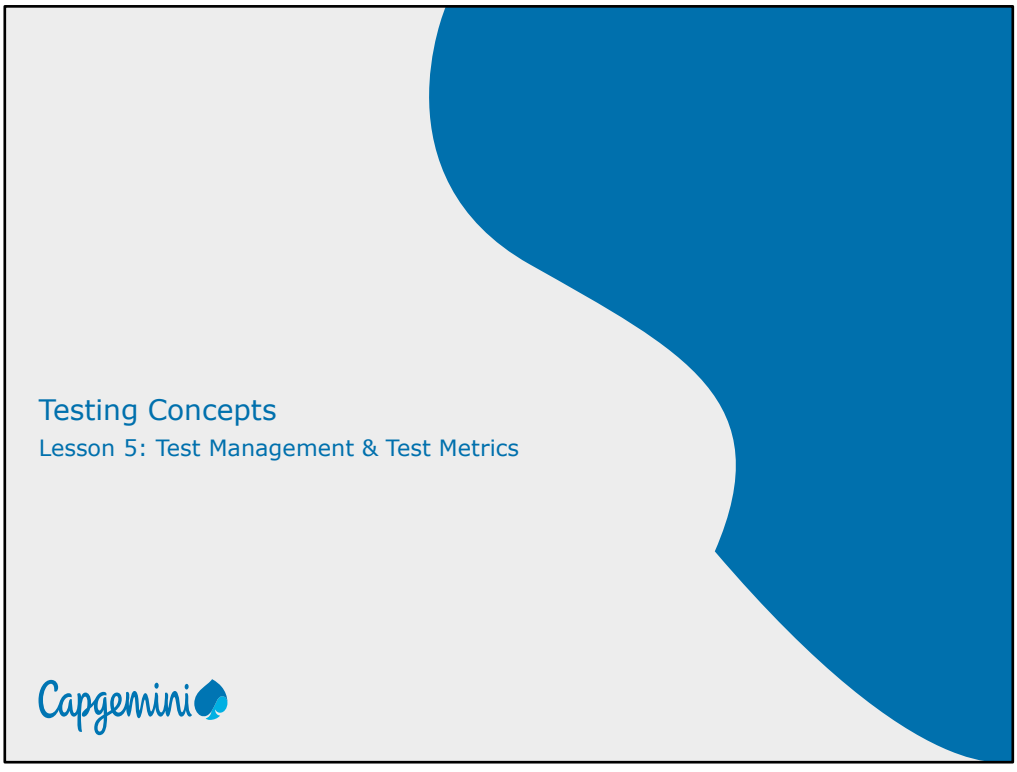
Testing Concepts
Lesson 5: Test Management & Test Metrics
Capgemini

Lesson Objectives

To understand the following topics

- Test Organization
  - Independent Testing
  - Tasks of a Test Manager and Tester
- Test Planning and Estimation
  - Purpose and Content of a Test Plan
  - Test Strategy and Test Approach
  - Entry Criteria and Exit Criteria (Ready and Done)
  - Test Execution Schedule
  - Factors Influencing the Test Effort
  - Test Estimation Techniques
- Test Monitoring and Control
  - Metrics Used in Testing
  - Purposes, Contents, and Audiences for Test Reports
- Configuration Management
- Risks and Testing
  - Definition of Risk
  - Product and Project Risks
  - Risk-based Testing and Product Quality
- Defect Management

5.1 Test Organization

5.1.1 Independent Testing

- Independent testing brings a perspective which is different than that of the authors since they have different cognitive biases from the authors.
- Unbiased testing is necessary to objectively evaluate quality of a software
- Developer carrying out testing would not like to expose defects
- Assumptions made are carried into testing
- People see what they want to see.
- More effective in terms of Quality & Cost
- It is conducted by an independent test team other than developer to avoid author bias and is more effective in finding defects and failures
- The tester sees each defect in a neutral perspective
- The tester is totally unbiased
- The tester sees what has been built rather than what the developer thought
- The tester makes no assumptions regarding quality

5.1 Test Organization

5.1.1 Independent Testing (Cont..)

Benefits of Test Independence :

- Independent testers are likely to recognize different kinds of failures compared to developers because of their different backgrounds, technical perspectives, and biases.
- An independent tester can verify, challenge, or disprove assumptions made by stakeholders during specification and implementation of the system

Drawbacks of Test Independence :

- Isolation from the development team, leading to lack of collaboration, delays in providing feedback to the development team, or an adversarial relationship with the development team
- Developers may lose a sense of responsibility for quality
- Independent testers may be seen as a bottleneck or blamed for delays in release
- Independent testers may lack some important information about the test object)

Many organizations are able to successfully achieve the benefits of test independence while avoiding the drawbacks.

5.2 Test Planning and Estimation

5.2.1 Purpose and Content of a Test Plan

- A test plan outlines test activities for development and maintenance projects.
- Test Planning is influenced by the test policy and test strategy of the organization, the development lifecycles and methods being used, the scope of testing, objectives, risks, constraints, criticality, testability, and the availability of resources.

As the project and test planning progress, more information becomes available and more detail can be included in the test plan.
Test planning is a continuous activity and is performed throughout the product's lifecycle. (Note that the product’s lifecycle may extend beyond a project's scope to include the maintenance phase.)
Feedback from test activities should be used to recognize changing risks so that planning can be adjusted.
Planning may be documented in a master test plan and in separate test plans for test levels, such as system testing and acceptance testing, or for separate test types, such as usability testing and performance testing.

Test Plan Contents (IEEE 829)

1. Test Plan Identifier
2. References
3. Introduction
4. Test Items
5. Software Risk Issues
6. Features to be Tested
7. Features not to be Tested
8. Test Approach (Strategy)
9. Item Pass/Fail Criteria
10. Suspension Criteria and Resumption Requirements
11. Test Deliverables
12. Testing Tasks
13. Environmental Needs
14. Staffing and Training Needs
15. Responsibilities
16. Schedule
17. Planning Risks and Contingencies
18. Approvals
19. Glossary

Test Planning Activities

- Determining the scope, risks and identifying the objectives of testing
- Defining the overall approach of testing, definition of the test levels, entry and exit criteria
- Integrating and coordinating the testing activities into the SDLC activities
- Making decisions about what to test, what roles will perform the test activities, how the test activities should be done, and how the test results will be evaluated
- Scheduling test analysis and design activities
- Scheduling test implementation, execution and evaluation
- Assigning resources for the different activities
- Defining the amount, level of detail and templates for the test documentation
- Selecting metrics for monitoring and controlling test preparation and execution, defect resolution and risk issues

5.2 Test Planning and Estimation

5.2.2 Test Strategy and Test Approach

Test Strategy?

- A test strategy describes the test process at the product or organizational level.
- Test strategy means "How you are going to test the application?" You need to mention the exact process/strategy that you are going to follow when you will get the application for testing.

Test Approach:

- A test approach is the test strategy implementation of a project, defines how testing would be carried out

5.2 Test Planning and Estimation

5.2.3 Entry Criteria (Ready) and Exit Criteria (Done)

Entry Criteria :

- Availability of testable requirements, user stories, and/or models (e.g., when following a model based testing strategy)
- Availability of test items that have met the exit criteria for any prior test levels
- Availability of test environment
- Availability of necessary test tools
- Availability of test data and other necessary resources

Entry criteria (called as definition of **ready** in Agile development) : preconditions for undertaking a given test activity. If entry criteria are not met, the activity will prove more difficult, more time-consuming, more costly, and more risky.
Exit criteria (called as definition of **done** in Agile development) :
Post conditions that must be achieved in order to declare a test level or a set of tests completed.

5.2.3 Entry Criteria and Exit Criteria (Cont..)

Exit Criteria :

- Planned tests have been executed
- A defined level of coverage (e.g., of requirements, user stories, acceptance criteria, risks, code) has been achieved
- The number of unresolved defects is within an agreed limit
- The number of estimated remaining defects is sufficiently low
- The evaluated levels of reliability, performance efficiency, usability, security, and other relevant quality characteristics are sufficient

Entry and exit criteria should be defined for each test level and test type, and it will differ based on the test objectives.

Even without exit criteria being satisfied, it is also common for test activities to be curtailed due to the budget being expended, the scheduled time being completed, and/or pressure to bring the product to market.

It can be acceptable to end the testing under such circumstances, if the project stakeholders and business owners have reviewed and accepted the risk to go live without further testing.

Example : Entry Criteria for Functional Testing

- Integration Testing is complete and sign-off is received by Project team.
- Integration test results are provided to the QA team within the Integration Execution & Signoff artifact.
- Development team provides a demonstration of application changes prior to promotion to QA Environment
- Code is delivered and successfully promoted to the Functional/System Test Environment as described in Master Test Plan
- Functional/System Test planning is detailed, reviewed and approved within the Master Test Plan
- Smoke /Shake down test has been completed to ensure test environment is stable for testing.
- Functional/System Test Cases are created, reviewed and approved within the RBC Enterprise approved tool (HP QC)
- Test data is ready for Functional/System Testing

Example : Exit Criteria for Functional Testing

- All high and medium risk tests identified in the detailed test plan are executed, including interface testing
- All planned testing is complete and documented
- Functional/System test execution results are captured
- All known defects have been entered into the defect tracking tool
- There are no known severity one or severity two defects
- Action plans have been created for outstanding severity three and four defects
- Appropriate signoffs are obtained
- Location of test cases, automated test scripts, defects and Functional/System Execution & Signoff artefact are detailed within the SCM plan.
- Any known deviations from the BRD and SRS are documented and approved

5.2 Test Planning and Estimation

5.2.4 Test Case Execution Schedule

Pre-execution activities

Setting up the Environment

- Similar to production environment
- Hardware (e.g. Hard Disk, RAM, Processor)
- Software (e.g. IE, MS office)
- Access to Applications

Setting up data for Execution

- Any format (e.g. xml test data, system test data, SQL test data)
- Create fresh set of your own test data
- Use existing sample test data
- Verify, if the test data is not corrupted
- Ideal test data - all the application errors get identified with minimum size of data set

Test Case Execution

Pre-execution Activities

Test data to ensure complete test coverage

Design test data considering following categories:

- No data
  - Relevant error messages are generated
- Valid data set
  - Functioning as per requirements
- Invalid data set
  - Behavior for negative values
- Boundary Condition data set
  - Identify application boundary cases
- Data set for Performance, Load and Stress Testing
  - This data set should be large in volume

Test Case Execution

Setting up the Test Environment

There are various types of Test Environments :

- Unit Test Environment
- Assembly/Integration Test Environment
- System/Functional/QA Test Environment
- User Acceptance Test Environment
- Production Environment

Test Case Execution

Before starting Execution

Validate the Test Bed

- Environment
  - Hardware (e.g. Hard Disk, RAM, Processor)
  - Software (e.g. IE, MS office)
- Access
  - Access to the Application
  - Availability of Interfaces (e.g. Printer)
  - Availability of created Test Data
- Application
  - High level testing on the application to verify if the basic functionality is working
  - There are no show-stoppers
  - Referred to as Smoke/Sanity/QA Build Acceptance testing

Test Case Execution

During Execution

Run Tests

- Run test on the identified Test Bed
- Precondition
- Use the relevant test data

Note the Result

- Objective of test case
- Action performed
- Expected outcome
- Actual outcome
- Pass/Fail (according to pass/fail criteria)

Compare the Input and Output

- Validate the data (e.g. complex scenarios, data from multiple interfaces)

Record the Execution

- Test data information (e.g. type of client, account type)
- Screenshots of the actions performed and results
- Video recording (HP QC Add-in)

Test Case Execution

After Execution

Report deviation

- Log Defect for Failed Test Cases
- Defect logging
  - Project
  - Summary
  - Description
  - Status
  - Detected By
  - Assigned To
  - Environment (OS, Release, Build, Server)
  - Severity
  - Priority
  - Steps to recreate and Screenshots

5.2 Test Planning and Estimation

5.2.5 Factors Influencing the Test Effort

Product Characteristics :

- The risks associated with the product
- The quality of the test basis
- The size of the product
- The complexity of the problem domain
- The requirements for quality characteristics (e.g., security, reliability)
- The required level of detail for test documentation
- Requirements for legal and regulatory compliance

Development process Characteristics :

- The stability and maturity of the organization
- SDLC model used
- Tools used
- Test approach
- Test process
- Time pressure

Test effort estimation involves predicting the amount of test-related work that will be needed in order to meet the objectives of the testing for a particular project, release, or iteration.
Factors influencing the test effort may include characteristics of the product, characteristics of the development process, characteristics of the people, and the test results.

5.2.5 Factors Influencing the Test Effort (Cont..)

People Characteristics :

- Skills and experience of the team members
- Team Cohesion and Leadership

Test Results:

- The number and severity of defects
- The amount of rework required

5.2 Test Planning and Estimation

5.2.6 Test Estimation Techniques

There are several estimation techniques to determine the effort required for adequate testing. Two most used techniques are:

- The metrics-based technique: estimating the test effort based on metrics of former similar projects, or based on typical values
- The expert-based technique: estimating the test effort based on the experience of the owners of the testing tasks.

Example:

In Agile development, burndown charts are examples of metrics-based approach as effort is being captured and reported, and is then used to feed into the team's velocity to determine the amount of work the team can do in the next iteration; whereas planning poker is an example of the expert-based approach, as team members are estimating the effort to deliver a feature based on their experience.

Example :
In sequential projects, defect removal models are examples of the metrics-based approach, where volumes of defects and time to remove them are captured and reported, which then provides a basis for estimating future projects of a similar nature; whereas the Wideband Delphi estimation technique is an example of the expert-based approach in which groups of experts provides estimates based on their experience.

5.3 Test Monitoring and Control

Why Test monitoring is necessary?

- To know the status of the testing project at any given point in time
- To provide visibility on the status of testing to other stake holders
- To be able to measure testing against defined exit criteria
- To be able to assess progress against Planned schedule & Budget

5.3 Test Monitoring and Control (Cont..)

Why Test Control is necessary?

- To guide and suggest corrective actions based on the information and metrics gathered and reported.
- Actions may cover test activity and may effect any other SDLC activity.
- Examples of test control actions include:
  - Re-prioritizing tests when an identified risk occurs (e.g., software delivered late)
  - Changing the test schedule due to availability or unavailability of a test environment or other resources
  - Re-evaluating whether a test item meets an entry or exit criterion due to rework

Test control is the response to Test Monitoring and Test Reporting that allows us to be IN CONTROL of the project.
Issues need to monitored and reported.
The process of control is the corrective actions required to put a testing effort (project) back on track.

5.3.1 Metrics used in Testing

- Efficient test process measurement is essential for managing and evaluating the effectiveness of a test process. Test metrics are an important indicator of the effectiveness of a software testing process.

Metrics should be collected during and at the end of a test level in order to assess :

- Progress against the planned schedule and budget
- Current quality of the test object
- Adequacy of the test approach
- Effectiveness of the test activities with respect to the objectives

A metric is the measurement of a particular characteristic of a program's performance or efficiency.

A quantitative measure of the degree to which a system, component or process possesses a given attribute.

5.3.1 Metrics used in Testing (Cont..)

Common Test Metrics include:

- Percentage of planned work done in test case preparation (or percentage of planned test cases implemented)
- Percentage of planned work done in test environment preparation
- Test case execution (e.g., number of test cases run/not run, test cases passed/failed, and/or test conditions passed/failed)
- Defect information (e.g., defect density, defects found and fixed, failure rate, and confirmation test results)
- Test coverage of requirements, user stories, acceptance criteria, risks, or code
- Task completion, resource allocation and usage, and effort
- Cost of testing, including the cost compared to the benefit of finding the next defect or the cost compared to the benefit of running the next test

https://www.tricentis.com/blog/64-essential-testing-metrics-for-measuring-quality-assurance-success/#:~:text=Software%20testing%20metrics%20are%20a,of%20the%20system%20under%20test.&text=Result%20Metrics%3A%20metrics%20that%20are,of%20an%20activity%2Fprocess%20completed.

Need of Metrics

- To track Projects against plan
- To take timely corrective actions
- To get early warnings
- It is a basis for setting benchmarks
- It is a basis for driving process improvements
- To track the process performance against business

Test Metrics data collection helps predict the long term direction and scope for an organization
Provides a basis for estimation and facilitates planning for closure of the performance gap
Provides a means for control/status reporting
Identify critical processes that will be monitored statistically
Identifies risk areas that require more testing
Provides meters to flag actions for faster and more informed decision making
Helps in identifying potential problems and areas of improvement
Provide an objective measure of the effectiveness and efficiency of testing

Types of Metrics

- **Project Metrics** - Test Coverage, Defect Density, Defect arrival rate
- **Process Metrics** - Test Effectiveness, Effort Variance, Schedule Variance, CoQ, Delivered Defect Rate, Defect Slippage or Test escape, Defect Injection Rate, Rejection Index, Resource Utilization, Review Effectiveness, Test Case Design, Rework Index, Defect Removal Efficiency.
- **Productivity Metrics** - Test case design productivity, Test case execution productivity
- **Closure Metrics** – Effort distribution metrics like Test Design Review Effort, Test Design Rework effort, KM Effort

Types of Metrics – Project Metrics

Test Coverage: The following are the test coverage metrics:

Test Design:

- # Of Requirements or # Of Use Cases covered / # Of Requirements or # Of Use Cases Planned

Test Execution:

- # Of Test scripts or Test cases executed/# Of Test scripts or Test cases Planned

Test Automation:

- # Of Test cases automated/# Of Test cases

Types of Metrics – Project Metrics (Cont..)

Defect Density

Total Defect density = (Total number of defects including both impact and non-impact, found in all the phases + Post delivery defects)/Size

Defect arrival rate:

Of Defects * 100 / # of Test Cases planned for Execution

This metric indicates the quality of the application/product under test.

Lower the value of this parameter is better.

The defect that are hampering the functionality are ‘impact’ defects. Defect those are not affecting functionality like look n feel errors, displacement errors are non-impact errors.

Types of Metrics – Process Metrics

Test Effectiveness

Of Test Cases failed (found defects)/# Of Test Cases executed

This metric indicates the effectiveness of the Test Cases in finding the defects in the product

Defect Removal Efficiency

(# of Defects found internally / Total # Of(internal + external) Defects found) * 100

It indicates the number of defects leaked after several levels of review and these defects are slipped to the customer.

Defect Injection Rate (No of Defects / 100 Person Hours)

No of Defects[phase wise] * 100/ Actual Effort[phase wise]

This is used to detect the defects injected during STLC Phases.

Types of Metrics – Process Metrics (Cont..)

Cost of quality

% CoQ = (Total efforts spent on Prevention + Total efforts spent on Appraisal + Total efforts spent on failure or rework)*100/(Total efforts spent on project)

Prevention Cost: (Green Money) :

Cost of time spent by the team in implementing the preventive actions identified from project start date to till date

Appraisal Cost: (Blue Money) :

Cost of time spent on review and testing activities from the project start date to till date

Failure Cost: (Red Money) :

Cost of time taken to fix the pre and post delivery defects. Expenses incurred in rework – Customer does not pay for this

Prevention - Money required preventing errors and to do the job right the first time is considered prevention cost. This category includes money spent on establishing methods and procedures, training workers and planning for quality. Prevention money is all spent before the product is actually built.

Appraisal – Appraisal costs cover money spent to review completed products against requirements. Appraisal includes the cost of inspections, testing and reviews. This money is spent after the product or subcomponents are built but before it is shipped to the user.

Failure – Failure costs are all costs associated with defective products. Some failure costs involve repairing products to make them meet requirements. Others are costs generated by failures, such as the cost of operating faulty products, damage incurred by using them and the costs incurred because the product is not available. The user or customer of the organization may also experience failure costs.

Types of Metrics – Process Metrics (Cont..)

Effort Variance

Overall and Variance at each milestone

(Actual effort - Planned effort) / Planned effort) * 100

The purpose of this parameter is to check the accuracy of the Effort estimation process to improve the estimation process.

Schedule variance

(Actual end date - Planned end date) / (Planned end date - Plan start date + 1) * 100

It depicts the ± buffer that can be used for optimum use of resource deployment and monitor the dates committed to the client and helps in better planning of future tasks.

Types of Metrics – Process Metrics (Cont..)

Defect slippage or Test escape

(Total # Of External Defects / Total # Of Defects detected (Internal+External)) * 100

This measure helps us to know how effectively we are detecting the defects at various stages of internal testing

Rejection index

Of Defects rejected/# Of Defects raised

The purpose of this parameter is to measure the Quality of the defects raised

Types of Metrics – Process Metrics (Cont..)

Resource Utilization

Actual effort utilized in the month for project activities /Total available Effort in the month

Review Effectiveness

(No of Internal Review Defects / [No of Internal Defects+No of External Defects])*100

The purpose of this parameter is to measure how effective are our reviews in capturing all the phase injected defects.

Test Case Design Rework Index

(Test Cases with Review Comments for rework/Total Test Cases)*100

Types of Metrics – Productivity Metrics

Test case design productivity

Of Test cases (scripts) designed/ Total Test case design effort in hours

This metric indicates the productivity of the team in designing the test cases.

Test case execution productivity

Of Test cases executed/ Total Test case executed effort in hours

Effort shall include the set up time, execution time. This metric indicates the productivity of the team in executing the test cases.

Types of Metrics – Closure Metrics

Test Design Review effort

(Effort spent on Test Case design reviews / Total effort spent on Test Case design) * 100

This is used with other process metrics like "Review Effectiveness", "DRE", "Defect Injection Ratio" to plan for an adequate review effort for future projects.

Test Design Rework effort

(Effort spent on Test Case design review rework / Total effort spent on Test Case design) * 100

This can be used with other process metrics like "Effort Variance", "Schedule Variance" to plan for an adequate rework effort for future projects.

KM Effort

(Total Effort spent on preparation of KM artifacts / Total efforts for entire project) * 100

This indicates effort spent on KM that can be used to plan KM activities for future projects.

Effort Distribution
(Effort spent on a STLC Phase / Total STLC effort) * 100
This would indicate the effort distribution across STLC phases. Given the type of an engagement and application, we can re-use this in estimation & planning for future projects

5.3.2 Purposes, Contents & Audiences for Test Reports

- The purpose of test reporting is to summarize and communicate test activity information, both during and at the end of a test activity (e.g., a test level).
- The test report prepared during a test activity may be referred to as a **test progress report**, while a test report prepared at the end of a test activity may be referred to as a **test summary report (test completion report).**

Test Progress Report includes:

- The status of the test activities and progress against the test plan
- Factors impeding progress
- Testing planned for the next reporting period
- The quality of the test object

ISO standard (ISO/IEC/IEEE 29119-3) refers to two types of test reports, test progress reports and test completion reports (a.k.a. test summary reports), and contains structures and examples for each type.

During test monitoring and control, the test manager regularly issues test progress reports for stakeholders

5.3.2 Purposes, Contents & Audiences for Test Reports (Cont..)

Test Summary Report includes:

- Summary of testing performed
- Information on what occurred during a test period
- Deviations from plan, including deviations in schedule, duration, or effort of test activities
- Status of testing and product quality with respect to the exit criteria or definition of done
- Factors that have blocked or continue to block progress
- Metrics of defects, test cases, test coverage, activity progress, and resource consumption.
- Residual risks
- Reusable test work products produced

When exit criteria are reached, the test manager issues the test summary report. This report provides a summary of the testing performed, based on the latest test progress report and any other relevant information.

5.3.2 Purposes, Contents & Audiences for Test Reports (Cont..)

The contents of a test report will vary depending on the project, the organizational requirements, and the software development lifecycle.

- Example : a complex project with many stakeholders or a regulated project may require more detailed and rigorous reporting than a quick software update.
- Example : in Agile development, test progress reporting may be incorporated into task boards, defect summaries, and burndown charts, which may be discussed during a daily stand-up meeting

In addition to tailoring test reports based on the context of the project, test reports should be tailored based on the report's audience.
The type and amount of information that should be included for a technical audience or a test team may be different from what would be included in an executive summary report.
In the first example, detailed information on defect types and trends may be important. In the latter example, a high-level report may be more appropriate.

5.4 Configuration Management & Configuration Control

Configuration Management :

- The purpose of configuration management is to establish and maintain the integrity of the component or system, the test ware, and their relationships to one another through the project and product lifecycle.
- A discipline applying technical and administrative direction and surveillance to identify and document the functional and physical characteristics of a configuration item

Configuration Control or Version control:

- An element of configuration management, consisting of evaluation, coordination, approval or disapproval and implementation of changes to configuration items after formal establishment of their configuration identification

5.4 Configuration Management & Configuration Control (Cont..)

To properly support testing, configuration management may involve ensuring the following:

- All test items are uniquely identified, version controlled, tracked for changes, and related to each other
- All items of testware are uniquely identified, version controlled, tracked for changes, related to each other and related to versions of the test item(s) so that traceability can be maintained throughout the test process
- All identified documents and software items are referenced unambiguously in test documentation

During test planning, configuration management procedures and infrastructure (tools) should be identified and implemented.

5.5 Risks and Testing

5.5.1 Definition of Risk

Risk:

- A factor that could result in negative consequences; usually expressed as impact and like hood
- Risks are used to decide where to start testing and where to test more.
- Testing oriented towards exploring and providing information about product risks

Risk based Testing is used to reduce risk of adverse effect occurring or to reduce the impact of adverse effect

- It draws on the collective knowledge and insight of the project stakeholders to determine the risk and the level of testing required to address those risks.

5.5.2 Project Risks

A risk related to management and control of the (test) project is called as Project Risk.

Project risk include:

- **Project issues:**
  - Delays may occur in delivery, task completion, or satisfaction of exit criteria or definition of done
  - Inaccurate estimates, reallocation of funds to higher priority projects, or general costcutting across the organization may result in inadequate funding
  - Late changes may result in substantial re-work
- **Organizational issues:**
  - Skills, training, and staff may not be sufficient
  - Personnel issues may cause conflict and problems
  - Users, business staff, or subject matter experts may not be available due to conflicting business priorities

Project risks may affect both development activities and test activities. In some cases, project managers are responsible for handling all project risks, but it is not unusual for test managers to have responsibility for test-related project risks.

Project Risks (Cont..)

- **Political issues:**
  - Testers may not communicate their needs and/or the test results adequately
  - Developers and/or testers may fail to follow up on information found in testing and reviews (e.g., not improving development and testing practices)
  - There may be an improper attitude toward, or expectations of, testing (e.g., not appreciating the value of finding defects during testing)
- **Supplier issues:**
  - A third party may fail to deliver a necessary product or service, or go bankrupt o Contractual issues may cause problems to the project

Project Risks (Cont..)

- **Technical issues:**
  - Requirements may not be defined well enough
  - The requirements may not be met, given existing constraints
  - The test environment may not be ready on time
  - Data conversion, migration planning, and their tool support may be late
  - Weaknesses in the development process may impact the consistency or quality of project work products such as design, code, configuration, test data, and test cases
  - Poor defect management and similar problems may result in accumulated defects and other technical debt

5.5.2 Product Risks

Product Risk: it is directly related to the test object

- product risks include:
- Software might not perform its intended functions according to the specification
- Software might not perform its intended functions according to user, customer, and/or stakeholder needs
- A system architecture may not adequately support some non-functional requirement(s)
- A particular computation may be performed incorrectly in some circumstances
- A loop control structure may be coded incorrectly
- Response-times may be inadequate for a high-performance transaction processing system
- User experience (UX) feedback might not meet product expectations

5.5.3 Risk-based Testing and Product Quality

Product Risks identified during Risk-based testing are used to:

- Determine the test techniques to be employed
- Determine the extent of testing to be carried out
- Prioritize testing in an attempt to find the critical defect as early as possible
- Determine whether any non testing activities could be employed to reduce risk

Risk Management activities provide a discipline approach to :

- minimize the product failure:
- Assess and reassess what can go wrong (risks)
- Determine what risks are important to deal with
- Implement action to deal with those risks

A risk-based approach to testing provides proactive opportunities to reduce the levels of product risk.
It involves product risk analysis, which includes the identification of product risks and the assessment of each risk's likelihood and impact.
The resulting product risk information is used to guide test planning, the specification, preparation and execution of test cases, and test monitoring and control.
Analyzing product risks early contributes to the success of a project.

5.6 Defect Management

Need of Defect Management Process :

- one of the objectives of testing is to find defects, defects found during testing should be logged.
- Any defects identified should be investigated and should be tracked from discovery and classification to their resolution - e.g., correction of the defects and successful confirmation testing of the solution, deferral to a subsequent release, acceptance as a permanent product limitation, etc.
- In order to manage all defects to resolution, an organization should establish a defect management process.
- This process must be agreed with all those participating in defect management, including designers, developers, testers, and product owners.

During the defect management process, some of the reports may turn out to describe false positives, not actual failures due to defects. For example, a test may fail when a network connection is broken or times out. This behavior does not result from a defect in the test object, but is an anomaly that needs to be investigated. Testers should attempt to minimize the number of false positives reported as defects.

Objectives of Defect Report

- Provide developers and other parties with information about any adverse event that occurred, to enable them to identify specific effects, to isolate the problem with a minimal reproducing test, and to correct the potential defect(s), as needed or to otherwise resolve the problem
- Provide test managers a means of tracking the quality of the work product and the impact on the testing (e.g., if a lot of defects are reported, the testers will have spent a lot of time reporting them instead of running tests, and there will be more confirmation testing needed)
- Provide ideas for development and test process improvement

Summary

In this lesson, you have learnt:

- Test management is covered from a skills perspective, focusing on test execution and defect reporting and handling.
- Managing the Test Activities
- Role of test manager and testers
- Test planning activities
- Template of Test document artifacts such as test plan and test case designs
- Entry and exit criteria of tests
- Test execution activities
- Test Metrics
- Configuration Management

Review - Questions

Question 1: The degree of independence to which testing is performed is known as _____.

Question 2: there can be different test plans for different test levels. (True / False)

Question 3: Exit criteria are used to report against and to plan when to begin testing. (True / False)

Question 4: Which of the following are Test Environments

- Unit Test Environment
- QA Test Environment
- Simple Test Environment
- Product Environment

Review – Match the Following

1. Entry Criteria	A. Consultative approach
2. Exit Criteria	B. Acceptance criteria
3. Test Approach	C. Methodical approach
4. Failure based testing	D. Completion criteria