Test Plan

Last updated: Mar-2014

Agenda

- **Duration:** *2 Hours*
- **Audience:** *Testers*
- **Content:**
    1. Test plan process
    2. Test plan structure
    3. Quiz

Objectives

After the course, student will:

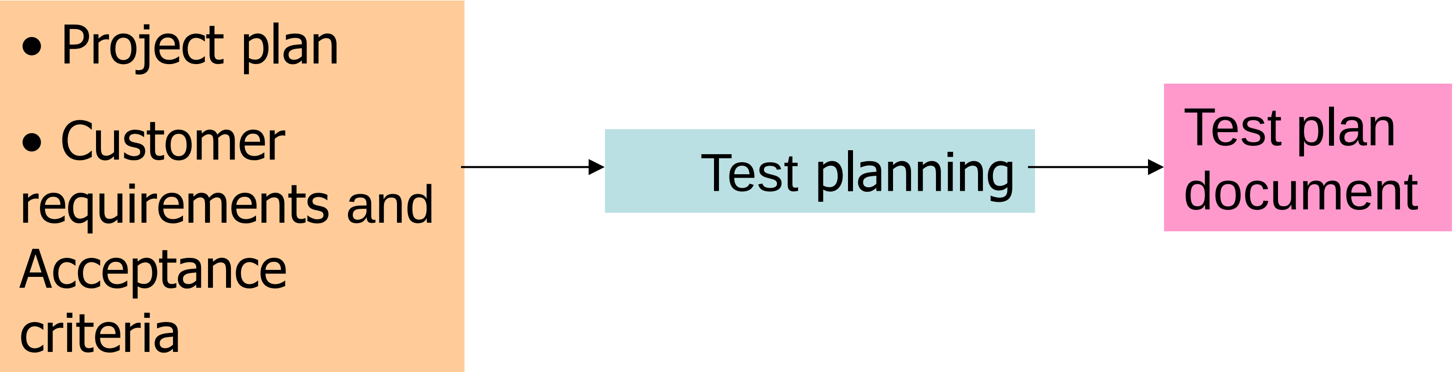
- Understand the way to create a Test Plan:
  - Input
  - Output
- Using Test Plan checklist to review a Test Plan
- Know Test tools available to use

□ ...

Test Plan Objective

- Identify testing scope
- Identify testing risks
- Identify acceptance criteria
- Obtain commitment from affected groups
- Brings in Effective collaboration among the teams
- Specify deliverables
- Define test strategy
- Define the Responsibilities
- Decide on Automation Early
- Decide on Test Metrics & Manage through Metrics

Test Plan Process



- Project plan
- Customer requirements and Acceptance criteria

Test planning

Test plan document

Input of Test Plan

- Project Plan: what information should be get?
  - When requirement specification is available
  - When detail design is available
  - When the first testing task can start
  - Date of releases
- Customer requirements and Acceptance criteria
  - SRS: Software Requirement Specification
  - Acceptance criteria

Test Plan Structure

1. Introduction
2. Define requirements for test base on acceptance criteria
3. Define test strategy base on requirements for test: test types, stage, tools
4. Define resources and responsibilities

   - Define the system: hardware & software for testing
5. Define Test milestones
6. Define deliverables of test: TP, TC, TR

Test Plan - Introduction

- **Purpose**: Briefly about the purpose and organization of the documents
- **Background information**: Briefly information of the project
- **Document reference**: List all documents used to create TP
- **Scope of testing**:
  - List stages of testing
  - List of test types
  - List any assumptions
  - Plan defects
- **Constraints**: List any constraints on test environment, resource, schedules, test tools
- **Risk list**: List any risks that may affect the design or execution of testing

Test Stages - Types

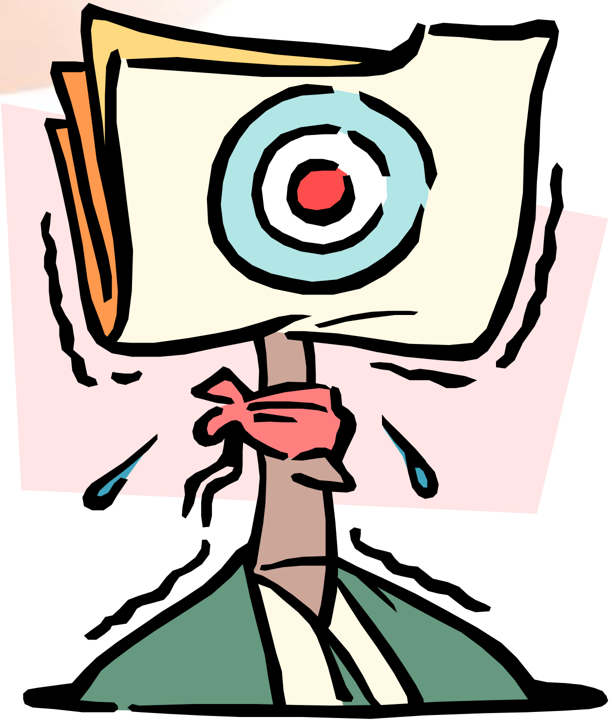
- **Test Stage:** The stages in which the test will be executed
  - Unit test
  - Integration test
  - System test
  - Acceptance test
- **Test Types:** all Types of test to execute:
  - Function test
  - User interface test
  - Performance test
  - Security and Access Control Testing
  - Regression Testing
  - ….

Test Plan – Requirement for test

- List items that have been identified as targets for testing
  - Functional requirements
  - Non-functional requirements
- List of features and functions not to be tested

Functional Requirement

- Things that a system has to do related to business flow
- Example:
  - ➢ Do a calculation: add, new, edit, delete functions
  - ➢ Make a decision: check user right
  - ➢ Make a report



Non-functional Requirement

- The qualities that a system has to have
  - Performance: response time
  - Security: access right
  - Usability: to learn using
  - Maintainability: to locate and fix bugs of operational system
  - Flexibility:
  - Reliability:
- Constraints on the test process

Test Strategy: General

- **Objective:** what needs to be ensured
- **Technique:** how testing will be executed
  - What will be tested
  - The major actions to be taken during test execution
  - The methods used to evaluate the results
- **Completion Criteria:**
  - Identify acceptance criteria for product quality
  - Identify when the testing is successfully executed
- **Special Considerations:**
  - Identify any influences/dependencies which may impact/ influence the testing described in Test strategy.
  - Criteria to stop testing (UT not good/meet requested Test coverage/…)

Function Testing

- **Objective:** Ensure proper target-of-test functionality, including navigation, data entry, processing, and retrieval
- **Technique:** based on black box techniques
  - Verifying the application and its internal processes by interacting with the application via the Graphical User Interface (GUI)
  - Analyzing the output or results
- **Completion Criteria:**
  - All planned tests have been executed
  - All identified defects have been addressed

Function Testing (cont…)

- **Special Considerations:** Identify or describe those items or issues (internal or external) that impact the implementation and execution of function test:
  - Test database
  - Criteria to stop testing (UT not good/met requested Test coverage/…)

User Interface Testing

- **Objective:**
  - Verify navigation (iE, Netscape)
  - Verify using of access methods (tab keys, mouse movements, accelerator keys)
  - Window objects and characteristics, such as menus, size, position, etc
- **Technique:**
  - Create or modify tests for each window to verify proper navigation and object states for each application window and objects
- **Completion Criteria:**
  - Each window successfully verified to remain consistent with prototype version or within acceptable standard

Data and Database Integrity Testing

- **Objective:**
  - Ensure database access methods and processes function properly and without data corruption
- **Technique:**
  - Check the returned data to ensure that the correct data was retrieved for the correct transaction
  - Check the database to ensure the data has been populated as intended, all database events occurred properly
- **Completion Criteria:**
  - All database access methods and processes function as designed and without any data corruption

Business Cycle Testing

- **Objective:**
  - Ensure proper target-of-test and background processes function according to required business models and schedules
- **Technique:**
  - All functions that occur on a periodic schedule will be executed or launched at the appropriate time
  - Testing will include using valid and invalid data
  - Each business rule is properly applied

Performance Profiling

- **Objective:** verify performance requirements have been achieved
  - Response time
  - Transaction rates
  - Other time-sensitive requirements are measured and evaluated
- **Technique:**
  - Use Test Procedures developed for Function Testing or Business Cycle Testing
  - Modify data files to increase the number of transactions or the scripts to increase the number of iterations each transaction occurs
- **Completion Criteria:**
  - Single Transaction: Successful completion of the test scripts without any failures and within the expected or required time allocation per transaction
  - Multiple transactions: Successful completion of the test scripts without any failures and within acceptable time allocation

Load Testing

- **Objective:**
  - Verify performance behavior time for designated transactions or business cases under varying workload conditions
- **Technique:**
  - Use tests developed for Function or Business Cycle Testing
  - Modify data files to increase the number of transactions or the tests to increase the number of times each transaction occurs
- **Completion Criteria:**
  - Multiple transactions or multiple users: Successful completion of the tests without any failures and within acceptable time allocation
  - Performed on a dedicated machine or at a dedicated time

Stress Testing

- **Objective:** verify functions work without error under the following stress conditions:
  - Little or no memory available on the server (RAM)
  - Maximum actual or physically capable number of clients connected or simulated
  - Multiple users performing the same transactions against the same data or accounts
- **Technique:**
  - Use tests developed for Performance Profiling or Load Testing
  - To test limited resources, tests should be run on a single machine, and RAM on server should be reduced or limited
- **Completion Criteria:** Get the specific the limited resources, the min, max number of transactions or concurrent users that make application run fail

Volume Testing

- **Objective:**
  - Test with large amounts of data to determine if limits are reached that cause the software to fail
- **Technique:**
  - Data and database integration
  - Insert data (by manual/tools/scripts)

Security and Access Control Testing

- **Application-level Security**
  - Check user right: verify that an actor/user can access only those functions or data if they have right
- **System-level Security**
  - Verify that only those users granted access to the system are capable of accessing the applications and only through the appropriate gateways
  - Check privilege of users

Regression Testing

- **Objective:**
  - Validate modified parts of the software, to make sure that the modification does not cause errors in other parts
- **Technique:**
  - Reuse the set of test cases from an existing test suite to test a modified module

Test Tool

- **List test tools will be employed for this project**
  - Tool for log defects/bugs
  - Load test tool
  - Performance test tool
  - Automatic generated data tool
  - Tools are used for Graphic test
- **Common Test tools:**
  - Eye Dropper: for colors
  - VRuler: to measure the distances
  - Unit Test tools: MAULA, JUnit, NUnit, …
  - Function test tools: Rational Robot test, Ruby-Watir
  - Load test: Rational Load test, OpenSTA

Test Plan - Resource

- **Human resource:** Test Leader, Testers, infrastructure management
  - Define worker/doer
  - Specific responsibilities/comments
- **System:** list of required software, hardware requirements
  - Server
  - Client
  - Database server

Test Plan - Estimation

- **Two techniques for estimation covered by ISTQB:**
  - Metrics-based: involves analyzing metrics from past projects and from industry data
  - Expert-based: involves consulting the people who will do the work and other people with expertise on the tasks to be done

Metric-based Estimation

- Can be as simple or sophisticated as you make it
- The simplest approach is to ask:
  - How many testers do we typically have per developer on a project?
  - Classifying the project in terms of size (small, medium or large) and complexity (simple, moderate or complex) and then seeing on average how long projects of a particular size and complexity combination have taken in the past
- Another approach: look at the average effort per test case in similar past projects and to use the estimated number of test cases to estimate the total effort

Expert-based Estimation

- Asking the individual contributors and experts involves working with experienced staff members to develop a work-breakdown structure for the project
- Called 'bottom up' estimation because you start at the lowest level of the hierarchical breakdown in the work-breakdown structure - the task - and let the duration, effort, dependencies and resources for each task add up across all the tasks

Test Estimation - Reality

- **Work Break Down:**
  - List all requirement for test (functional and non functional)
  - Divide requirement for test into 3 category of complexity: simple, average, complex
  - For each category, input estimated effort for test tasks: study requirement, create test case, test data, etc.
  - Add effort for test plan and other management works (if have)

Test Estimation - Reality

- **Historical Data:**
  - Based on effort rate of testing/project size
  - Based on domain
  - Based on market: different market -> different rates
  - Got total effort for test
  - Estimate for each tasks of test milestone

Test Plan - Milestones

- Milestone name:
  - Test planning
  - Test design
  - Test execution
  - Test evaluation
- Effort: in person-day (pd)
- Start date
- End date

Test Plan - Deliverables

- Product deliverable name
  - Test Plan
  - Test Cases
  - Test Report
- Deliverable date
- Delivered by: Tester/ Test Leader
- Delivered to: Test Leader/ PTL/QA

TP Review Checklist

- Who will use TP checklist?
- Why do have to use TP Review checklist?
  - Check template of TP
  - Check required information in TP
- Template

Resources & references

- **Resources**
  - Template_Test Plan.doc
  - Checklist_Test plan review.xls
- **References:**
  - Foundations of Software Testing.pdf
  - Guideline_Software Testing.pdf
  - Process_Test.pdf

QUESTIONS AND ANSWERS