Test Plan Workshop Version 2.0

By **Son Pham**

May, 2014

Duration: 3 hours

Global CyberSoft

A World of Difference

www.globalcybersoft.com

CMMI Level 4

TÜV SÜD ISO 9001

Microsoft GOLD CERTIFIED Partner

Contents

- Introduction
- Target Test Areas
- What to be tested
  - Features and Functions to be tested
  - Features and Functions not to be tested
- Test Approach
- Test Deliverable, Responsibility, Schedule
- Test Cycle Criteria
- Environment Needs
- Risk Management
- Staffing and Training Needs

What is a test plan?

A document that provides information on

- The goals and objectives of testing within the scope of the iteration (or project)
- The items being targeted
- The approach to be taken
- The resources required
- Test Cycle Criteria
- The deliverables to be produced
- Milestones
- Risk management

Introduction

- ❑ Purpose
- ❑ Scope
- ❑ Document Terminology and Acronyms
- ❑ Reference

Introduction

1.1 Purpose

- Provide a high level list of the major target test functions.
- List the Requirements for Test
- Describe the testing strategies to be employed on each target test functions
- Identify the required resources and schedule the Test execution

❑ 1.2 Scope:

- Test Plan is for release <RELEASE NO.>

Ex: Test Plan for KIKA release 1.0

- This Test Plan is for **KIKA release 1.0**

Introduction (cont)

1.3 Document Terminology and Acronyms

- Provides the definitions of any terms, acronyms, and abbreviations required to properly interpret the Test Plan

Ex: Test Plan for KIKA release 1.0

- CRF – Change Request Form
- QTY – Quality

Introduction (cont)

1.4 References

- Provides a list of the documents referenced elsewhere within the Test Plan (called test basis document)

Ex: Test Plan for KIKA release 1.0

- S0148.0-R-001-Software Requirements Specification.doc
- S0148.0-P-002-Software Development Plan.doc
- Requirement (WMSV10).doc
- HHT_Spec(WMSV10).doc
- DM_Screen(WMSV10).doc

Target Test Area

- ❑ Test Items
- ❑ Test Objectives

Target Test Areas

- *Test Items:*
  - List of what to be tested including version number, configuration requirements where needed.
  - What you are testing is what you intend to deliver to the client.

- This section can be oriented to the level of the test plan.
  - System test plan: can be
    - <Application> - <version>
    - <Component> - <version>
    - Function areas or Features
    - User documentation
  - Unit test plan: can be
    - program, unit, module or build.

Target Test Areas

- *Test Objectives:*
  - The purpose for testing on test items.
- *Examples:*
  - is to measure software quality based on system attributes.
    - Functionality (Accuracy, Suitable, Interoperability, Compliance, Security)
    - Efficiency (Time behavior, Resource Behavior)
    - …
  - is to re-test all changed or unchanged part after fixes or modification SUT

Target Test Areas

Ex: Test Plan for Project ABC release 1.0

- ***Test Items:***
  - Fight Web application **v3.2**
  - Builder application **v1.1**
  - User Operations & Installation Guides **v3.0**

- ***Test Objective***:
  - Web application **v3.2**
    - *Functionality*
    - *Time behavior*
  - Builder application **v1.1**
    - *Functionality*
  - User Operations & Installation Guides **v3.0**
    - *Correct, complete, and comprehensive*

WHAT TO TEST?

- ❑ Features to be tested?
- ❑ Features not to be tested?

Features to be tested

- List of what to be tested:
  - From User viewpoint of what the system does.
  - Set the level of risk for each feature such as (H, M, L): High, Medium and Low.

- *Example:*
  - Flight Web application:
    - Booking a Flight - High
      - UC-01: Login
      - UC-02: Research a flight
      - UC-03: Book a flight
      - UC-04: Monitor the flight booking
    - Checking a Flight - High

Features not to be tested

- List of what is not be tested from:
    - User viewpoint of what the system does.
    - a configuration management/version control view.

- *Example:*
    - Configuration: is not be included in this release.
    - Logout: Low risk, it has been tested during initial release and is considered stable.
    - Server: We will assume the server has been tested.
    - Any Legacy systems that the Flight Web Application integrates with.

HOW TO TEST?

- ❑ Test Approach

Test Approach

- The implementation of the test strategy for a specific project, it should be appropriate to the level of the plan (master, acceptance, etc.) and should be in agreement with all higher and lower levels of plans.

- Identify the testing methods/rules/processes to reach test objective:
  - Test techniques
  - Test tools
  - Test Types
  - What levels of regression testing will be done and how much at each test level?
  - What metrics will be collected?

Test Approach

- Identify the method to reach test objective (Cont)
  - Strategy for the distribution of test effort and intensity: Define the methods for the calculation of the test intensity based on risk factors
    - Product risk/ Quality risk
    - Project risk
    - Basic rules to consider risks
    - Determine sequence and time of the activities
  - Specify if there are special requirements for the testing.
    - Only the full component will be tested.
    - A specified segment of grouping of features/components must be tested together.
  - Other Rules/Process must be followed

Test Approach – Example 1

- Define Test Approach for testing new Android mobile device:
  - Test method
    - Overall checking Android Features by using **Special Star Code** to access Device Diagnostics Tool pre-installed in each Android phone (it should be provided by manufacturer)
    - Perform comparative test with similar devices
    - How to test output is a numeric (benchmark score)?
    - …
  - Test Types:
    - Combine automation and manual to test requested features
  - Test tools
    - Quarand Tool: to test

      1- CPU-Performance

      2- I/O-Input and output and represents speed for read and write to peripherals and bus operations

      3- 2D and 3D -Indicates the processing speed for graphics

Test Approach – Example 2

Risk-based approach: Analyzing

- Product risk/ Quality risk:
  - Set risk priority for each feature (Refer to Feature to be tested)
  - Basic rules to consider risks:
    - Unclear Requirement
    - Complex, difficulty
    - Required new technology
  - Determine sequence and time of the activities:
    - The test case with high priority will test first and extensively
    - The test case with lower priority will test later. If time does not permit, some of the low priority test cases may be dropped.

Template

- After analyzing, put all information into the template:

Features/ Functions	Quality Risk - Priority	Test Type	How to test
Functionality			
Login	Medium	Functional	*Using techniques:* - Equivalence Partitioning - Boundary Value Analysis - Decision Testing - Use cases Testing
Booking a fight	High	Functional	
Checking a flight	High	Functional	
….	Low	Functional	
Efficiency			
Req-01 - Multi-users log-in system at once: Number of users: 50 concurrent users Test duration: 500 seconds	High	Performance Test	Using Quick Test Pro tool to simulate 100 Virtual users connect to system at the same time
Req-02 - The total file sizes for all the files that need to be downloaded in order to view the Web page must not exceed 100k	High	Performance Test	Using Load Runner 8.1 tool

Template

Features/ Functions	Quality Risk - Priority	Test Type	How to Test
Security			
UISE1 – Input data received from the client must be parsed to make sure that it does not contain "out of bounds“ or “buffer overflow” input data (e.g. a client enters a value of 13 or 9999999999999999999999999 for a month)	High	Security Test	TBD
UISE2 - Input data received from the client must be parsed to make sure that it does not contain inappropriate meta-character sequences. e.g. &&	High	Security Test	TBD
Reliability: TBD			

Template

Features/ Functions	Quality Risk - Priority	Test Type	How to Test
Usability			
UIUS1 – Content makes up 50% to 80% of the Web page' s "screen real estate	Medium	Usability Test	TBD
UIUS2 – Medium: Critical information has not been placed on the lower portion of the Web page (if the position of this information requires the user to use scroll down, the majority of visitors are unlikely to ever read it)	Medium	Usability Test	TBD
UIUS3 – Mandatory data entry fields may be flagged with a visual cue e.g. highlight in red	Low	Usability Test	TBD
UIUS4 – Related information is grouped together on the Web page to minimized eye movement	Low	Usability Test	TBD
UIUS5 – Low: There are no competing/duplicate actions/options on the Web page, which might confuse the user or cause them to make an error	Low	Usability Test	TBD
UIUS6 – When viewed via the clients anticipated hardware/software the page fits without the need for a horizontal scroll bar	Low	Usability Test	TBD
UIUS7 – Multiple key combinations can also be entered sequentially or are mapped to a single key	Low	Usability Test	TBD

Test Deliverables, Responsibilities and Schedule

- ❑ Test Deliverable
- ❑ Responsibility
- ❑ Schedule

Test Deliverables

- Test Deliverables are documents will be generated as a result of testing activities.
  - Test plan document.
  - Test cases.
  - Test design specifications.
  - Tools and their outputs.
  - Simulators.
  - Static and dynamic generators.
  - Error logs and execution logs.
  - Problem reports and corrective actions.

Responsibilities

- Who is in charge?

  Example:
  - Who develops Test Plan
  - Who develops Test case specification
  - Who develops Automated test scripts
  - Who writes Test Report
  - Who provides the required training?
  - Who makes the critical go/no go decisions for items not covered in the test plans?
  - …

Schedule

- Schedule should be based on realistic and validated estimates.
- Set milestones for significant test activities.
- A milestone is a significant event that normally has no duration.

- Includes 3 sub items:
  - Testing task
  - Start Date
  - End Date – milestone date for testing tasks

- Note: should address questions such as:
  - What milestones we will do/ pass by?
    - What are deliverable?
    - How many test cycles?
    - …
  - What is Release date, due dates for each milestone?

Schedule- Example

- Example: Test Plan for Release 1

Test Activity	**Start Date**	**End Date**
Develop Test Plan	Jun - 08, 2006	Jun - 09, 2006
Review Test Plan	Jun - 09, 2006	Jun - 09, 2006
Develop the functional test cases	Jul - 18, 2006	Jul - 24, 2006
Develop the non-functional test cases	Jul - 27, 2006	Jul - 28, 2006
Refine Test cases	Aug - 01, 2006	Aug - 02, 2006
Review test case	Aug - 02, 2006	Aug - 02, 2006
Develop & Review Test Data	Aug - 03, 2006	Aug - 04, 2006
Prepare Test Environment for functionality test	Aug - 05, 2006	Aug - 05, 2006
Prepare Test Environment for performance test of system test	Aug - 05, 2006	Aug - 05, 2006
Execute Round 1: Functionality	Aug - 06, 2006	Aug – 09, 2006
Execute Round 2: Efficiency, Security and Usability	Aug – 10, 2006	Aug -12, 2006
Execute Round 3: All tests that have been specified in Test Specification section	Aug - 15 2006	Aug - 16, 2006
Final Summary report for Release 1	Aug - 16, 2006	Aug - 16, 2006

Test Cycle Criteria

- ❑ Entry Criteria
- ❑ Continuous Criteria
- ❑ Exit Criteria
- ❑ Resume Criteria
- ❑ Abnormal Termination Criteria
- ❑ Example

Entry Criteria

- The purpose of entry criteria is to define when the system move into a particular test phase.
- The entry criteria should address questions such as:
  - Are the necessary documentation, design, requirements information available that will allow tester to operate system and judge correct behavior?
  - Is the system ready for delivery?
  - Are the supporting utilities, accessories available in forms that tester can use?
  - Is the system at the appropriate level of quality? It implies that some or all of previous test phase has been successfully completed.
  - Is the test environment such as lab, hardware, software and system administration support ready?

Entry Criteria - Example

Examples: for system test level

- Test Cases have been approved
- The test environment has been configured and test team have right to access to system
- Bug Tracking system is in place
- First Build is successful and be completed >= 60% functions in system test plan
- Development team have completed all functions and bug fixes scheduled for the build.
- Development team have unit-tested all functions and bug fixes scheduled for the build.
- ..
- Pass sanity test

Continuous Criteria (optional)

- The purpose of continuous criteria is to define the conditions and situations that must prevail in the testing process to allow testing to continue effectively and efficiently.

Example: System can continue if

- All software released RM is accompanied by release Notes
- The test environment must remain stable. No changes is made to the system, whether in source code, configuration files, setup instruction or processes.
- The open bug backlog ("quality gap") remains less than 100. The average time to close a bug remain less than 14 days.
- The installable and stable test releases must be delivered regularly and properly and The change to the system under test must be known and controlled.
- Bug review meeting every three days/ twice-weekly.

Exit Criteria

- The purpose of exit criteria is to define when testing has been completed/ stopped (totally or within a test level).
- Should refer to Quality Plan and Acceptance Criteria

Examples: for system test level

- No changes about requirement/ design occurred in the previous build.
- **Bug criteria:**
  - No critical bugs found in previous build such as panic, crash, halt, wedge, unexpected process termination or other stoppage of processing has occurred on any server software/ hardware and/or
  - No major bugs found in previous build for some special functions such as no client system have become inoperable due to a failed update and/or
  - No new bugs found after doing one round of regression test (using a little different test data) and/or
  - Number of bugs found are at the expected range (X + i%, X - i%) and/or
  - Zero defects in priority 1(Highest) test cases and/or
  - Minor bug is not over 10 bugs
- **Test Coverage/ Test Metrics**:
  - Planned test cases/ TSc are PASSED: 100%
  - Bugs are resolved (closed or deferred): 100%
  - Risk coverage/ Test coverage of significant risks: 100%
  - Requirement coverage (for system & acceptance Test): 100%

Resume criteria

- The purpose of resume criteria is to define when the system will start new test cycle for a particular test phase.

- ***Example:*** for system test level
  - Bugs were fixed and the bug status is not in abnormal termination, AND
  - PM requests to test, AND
  - Sanity test was done by RM with zero bugs

Abnormal Termination Criteria

- The purpose of abnormal termination criteria is to define whether testing should be prematurely suspended or ended for the current test cycle, or whether the intended build candidate to be tested must be altered

- ***Example:*** for system test level
  - More than 100 bugs are opened, OR
  - Any critical bug that causes stop working/running of the system, OR
  - 05 critical Bugs have been found: loss data, OR
  - Wrong configured environment

Environmental Needs

- ❑ Software and Hardware
- ❑ Productivity and Support Tool
- ❑ Test Environment Configuration
- ❑ Test Environment Validation
- ❑ Example

Environmental Need

- Hardware and Software
- Productivity and Support Tools
- Test Environment Configurations
- Test Environment Validation

Hardware and Software

- ❑ *1. Hardware and Software*

  Specifies the system resources for the test effort. It includes the following 3 three sub items:

  - Resource:
    - **The name/IP Address of the system resource**
  - Configuration:
    - **Specifies the hardware configuration**
  - Installed OS, Software:
    - **Specifies the list of OS, Software need to be installed and versions of those**

Hardware and Software

- Example:

Resource	Configuration	Installed OS, Software
KIKA Server	❑ Processor: P4- 1GHz· ❑ RAM: 2Gb MB ❑ HDD: 40 GB ❑ Network adapter 100Mb Ethernet	❑ Linux ❑ Oracle 10i ❑ JVM 1.4.2 or higher
KIKA Client	❑ Processor PIII 700MHz ❑ Ram: 512 Mb ❑ HDD 40Gb ❑ Network adapter 100Mb Ethernet	Windows 2000 Server / Professional ❑ Oracle 10i ❑ JVM 1.4.2 or higher
Automation Server	❑ Processor: P4- 1GHz· ❑ RAM: 2Gb MB ❑ HDD: 40 GB ❑ Network adapter 100Mb Ethernet	❑ Windows 2000 Server / Professional ❑ Quick Test Pro v9.2

Productivity and Support Tools

2. Productivity and Support Tools

Specifies tools to be employed to support the test process. It includes the following 4 three sub items:

- Tool’s purpose:
  - **The purpose of using each tool**
- Tool Name:
  - **Specifies the tool’s name**
  - Vendor or In-house:
    - **Specifies Vendor’s tool name**
  - Version:
    - **Specifies versions to be used**

Productivity and Support Tools

- Example:

Tool's purpose	Tool Name	Vendor or In-house	Version
Bug tracker	Microsoft Excel	Microsoft	2000
Test Report	Microsoft Word, Excel	Microsoft	2000
Functional Automation Test	Quick Test Pro	Mercury (Commercial Tool)	v9.2
Performance Test for Web application	Web Load OpenSta	Free Tool …	…
…	…	…	…

Test Environment Configurations

3. Test Environment Configurations

The Test Environment Configurations needs to be provided and supported for this project. It includes the following 3 three sub items:

- ❑ Configuration Name:
  - Specifies the short name of the configuration
- ❑ Description:
  - Specifies a short description on the purpose of each configuration
- ❑ Implemented in Physical Configuration:
  - How to configure in the real system

Test Environment Configurations

- ❑ Example:

Configuration Name	Description	Implemented in Physical Configuration
Server Side	• Deploy in Linux Environment • Install Oracle 10i	• Refer to Configuration Guide document for Linux • Oracle 10i: • Database name: • User name: • Password:
Automated Server	Deploy Win XP, Install Quick Test Pro Server	• Refer to guide line for install Win XP • Quick Test Pro: • Concurrent license Server • User Name: • Password
Client Side	❑ Install Windows 2000 version ❑ JVM 1.5 or higher ❑ Install CA (Certificate Authorization)	Refer to Configuration Guide document

Test Environment Validation

4. Test Environment Validation

- Project tester is responsible for validating the installed test environment to ensure it is configured correctly, and is ready for executing test cases. All configuration mistakes, the tester must inform PM for appropriate corrective action. PM is responsible for ensuring all wrong configuration of test environment is solved.
- The table below defines the checklist/criteria to ensure that the test environment is right installed and configured

Environment Items Setting	Criteria			Note
	Correct Version	Correct Platform	App Folder Structure	
Miracle Linux 4.0	Yes	Yes		
Oracle 10g Release 2 ONE	No	No	/opt/oracle	Use Oracle 10g Release 2 Standard Edition for testing. Client approved this
JVM 1.5.0	Yes	Yes	/usr/java/jdk1.5.0_04	
Quick Test Pro 9.2	Yes	Yes	…	

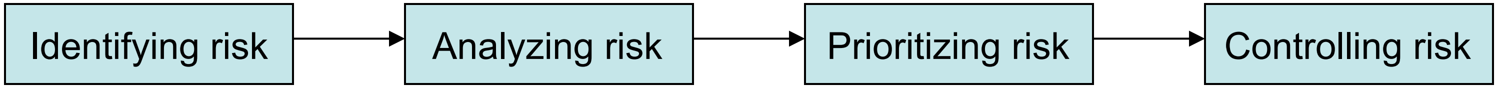
Risk Management

- ❑ Risk Overview: Definition, Types and Process
- ❑ Product Risk
- ❑ Project Risk
- ❑ Example for Risk-based approach

Risk

- Risk is a factor that could result in future negative consequences; usually expressed as impact and likelihood (probability of occurrence)
- 2 types of risks:
  - Product risk (Quality Risk): directly related to the test object (arise from the product). It is a result of problem with delivered product. It leads to many different types of damage, depending on who is affected
  - Project risk (Planning Risk): related to management and control of the (test) project (arise from the project)

    e.g. lack of staffing, strict deadlines, changing requirements
- Risk management: the process of



  - ***Identifying risk:*** assessing what can go wrong
  - ***Analyzing risk***: estimate impact and probability of occurrence
  - ***Prioritizing risk***: identify Risk Level/ the importance of a risk
  - ***Controlling risk***: implement actions to mitigate or fight those risks
    - Mitigation Strategy : The solution to mitigate the each specified risk
    - Contingency (Risk is realized): When the risk happened, which solution, strategy will be used

Product Risk – Examples

- Product risks have been considered as below:

Identifying and analyzing	Prioritizing Risk	Controlling Risk	
Quality Risks by <system test>	Priority	Mitigation Strategy	Contingency (Risk is realized)
Risk of failure: ex: harmful to human life or Insufficient qualification or poor quality			
Functionality is Inadequate functional quality and non-functionality has poor quality: - The manufacturer: incurs mostly indirect failure cost: such as cost or losses in sales due to non-performance of the contract or product liability claim, increased effort for customer hotline and support, and loss image that may lead to losing the customer altogether, declining "brand value" due to insufficient functionality, poor quality or even violations of the law enabled or supported by the product (viruses, phishing…)	1	Intensively test for feature 1: function **UC_01, UC_02**	
Customer: uses the system to provide service to users → can suffer damage through the product risk	1	intensively test for feature 2: **UC_05**	
Customer of the system: incurs almost direct failure cost: such as costs of computation errors, data loss, accounting errors, hardware damage	1	..	
Customer of the system/ User: The "safety" issue: in system as irradiation device in a hospital	1	…	
The usage frequency of a function or the probability of failure in software use			
Priority of requirements/ Quality characteristic/ Complexity of individual components/ Visibility of a failure for end user			

Product Risk – Examples

- Product Risk is mentioned in Test Approach as below:

Features/ Functions		Priority
Functionality: Inadequate functional quality – What can go wrong – Risk Level		
Feature 1	UC_01: function 1	1
	UC_02: function 2	1
	UC_03: function 3	2
	UC_04: function 4	3
Feature 2	UC_05: function 5	1
…	…	…
Non-functionality: Insufficient qualification or poor productivity		
Performance	function 1 on OS1	1
	function 1 on OS2	2
Capacity and Volume	Maximum file length	3
	Maximum table size	3
GUI	Screen 1 on OS1	1
…	…	…

Project risks - Examples

- Project Risk is mentioned in Risk section

Identifying and analyzing Risk	Prioritizing Risk	Controlling Risk	
Quality Risks by <system test>	**Priority**	**Mitigation Strategy**	**Contingency (Risk is realized)**
Human Resource Risk			
Underestimated: schedule is too tight, strict deadlines ….<More list here>	1	1. Develop test case and test data carefully to reduce time for execution 2. Optimize the test schedule and define overtime strategy at the beginning	1. Developer will help to do the test (Performance, Load test) for the final release 2. Increase testing resource (human, equipment,..)
Lack of human resource/ health is not good	1	Request soon/ back up	TBD
Test team are junior, not have domain knowledge ….<More list here>	1	Training	TBD
Not follow process: Risk at writing TP (ex: revising TP when SRS are finalized)	1	Training	TBD
Lack of cooperation btw different department	1	Training	TBD
Hardware Resource Risk			
Lack of PC	1	Request IT soon to support the system resource	Use personal resource
Internet speed is slow ….<More list here>	1	Request IS for especial line	TBD

Project risks – Examples (cont)

Identifying and analyzing Risk	Prioritizing Risk	Controlling Risk	
Quality Risks by <system test>	**Priority**	**Mitigation Strategy**	**Contingency (Risk is realized)**
Technical Risks			
SRS: - SRS is unclear, wrong, incomplete, infeasible	1	Review carefully on requirements, make more active questions and clarify with customer as soon as possible	TBD
SRS: - Too many requirement changes - Sponsors don’ t approve requirement ….<More list here>	1	Define a procedure to handle requirements Changes.	TBD
Not master in Linux Operation	2	Research and join the training about Linux for team	Ask expert to get advice
No experience about Quick Test Pro tool	3	Training QTP	Ask expert to get advice
Not have experience in ….<More list here>	2	…	…
Supplier-side risk			
Subcontractor fails to deliver. Project delays or even legal action may result from these risks	5	Out of scope ☺	TBD

Staffing and Training Needs

❑

Staffing and Training Needs

- Some training and skills are required
- Ex: The manager will ensure that the staff assigned to this project are experienced with:
  - Testing techniques
  - UML Notation
  - Automated testing tool: Quick Test Pro

Summary

- Planning must begin as early as possible to eliminate as many project risks as possible or at least make them visible early enough and to provide sufficient time to prepare test cases/ test data… Don’t wait all information is available.
- Planning must be refined in steps as soon as relevant information is available

Appendix: Course detail form

Author	Son Pham	**Duration**	3 hours
Category		**Type**	

Examination	
Intended Audience	
Pre-requisites	
Completion criteria for the course	
Criteria for granting training waivers	

Thank you

THANK YOU

Inquires regarding the above may be directed to:
Someone, Title, truonghx@gcs-vn.com