Topic #1 – Software Problem Management

Instructor: Jerry Gao, Ph.D., Professor
San Jose State University

TOPIC #1 – SOFTWARE PROBLEM MANAGEMENT

What Is A Software Error?

Software Problem Classification

Defects, Faults, Failures, and Errors

Software Problem Reporting

Software Problem Management

What is a software error?

Definition #1:
"A mismatch between the program and its specification is an error in the program if and only if the specification exists and is correct."

Definition #2:
"A software error is present when the program does not do what its end user reasonability expects to do." (Myers, 1976)

Definition #3:
"There can never be an absolute definition for bugs, nor an absolute determination of their existence. The extent to which a program has bugs is measured by the extent to which it fails to be useful.
This is a fundamentally human measure." (Besizer, 1984)

Mistake, Fault, Failure, and Error

Mistake
– A human action that produces an incorrect result.

Fault [or Defect]
– An incorrect step, process, or data definition in a program.

Failure
– The inability of a system or component to perform its required function within the specified performance requirement.

Error – the difference between a computed, observed, or measured value or condition and the true, specified, or theoretically correct value or condition.

A Classification of Software Errors

- User interface errors, such as output errors, incorrect user messages.
- Function errors
- Defect hardware
- Incorrect program version
- Testing errors
- Requirements errors
- Design errors
- Documentation errors
- Architecture errors
- Module interface errors
- Performance errors
- Error handling
- Boundary-related errors
- Logic errors, such as calculation errors
- State-based behavior errors
- Communication errors
- Program structure errors, such as control-flow errors

Problem Reporting

Whenever a bug or problem is found, we need to write down a problem report immediately.

<u>What are the content of a problem report?</u>

Problem ID **current software name** **release no. and version no.**

Test type **Reported by** **Reported date** **Test case ID**

Subsystem (or module name) **Feature Name (or Subject)**

Problem type (REQ/Design/Coding, …) **Problem severity (Fatal/Major/Minor, ..)**

Problem summary and detailed description:

Cause analysis **How to reproduce?** **Attachments**

Problem Reporting

How to track, control, and manage issued problems?

- **Systematically track and maintain reported problems in a repository.**
- **Define & implement a problem management process for problem analysis.**

Characteristics of a problem report:

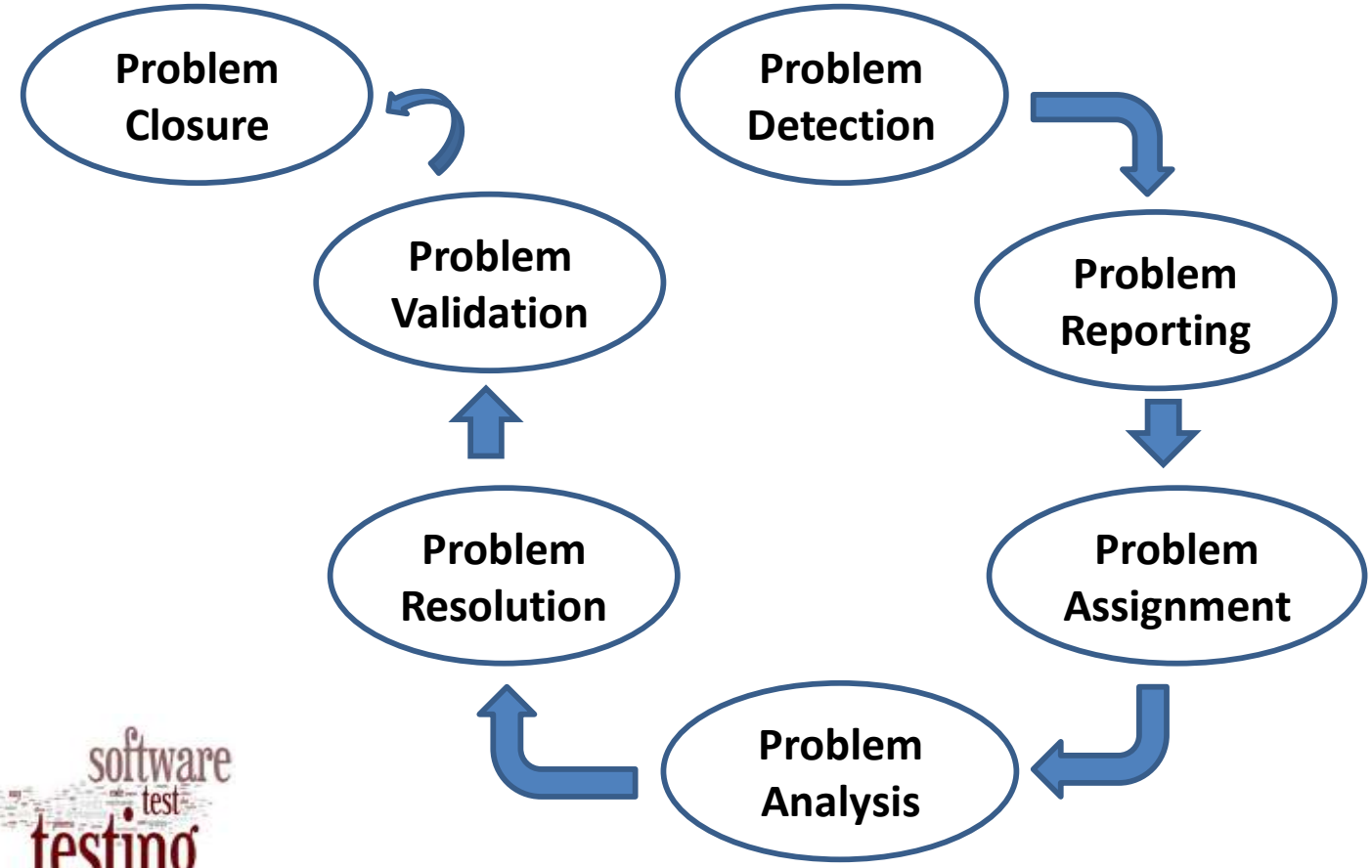
- **Simple and understandable**
- **Reproducible**
- **Traceable and numbered**
- **Non-judgmental**

Problem analysis:

-Finding the most serious consequences
-Finding the simplest and most general conditions
-Finding alternative paths to the same problem
-Finding related problems

TOPIC #1 – SOFTWARE PROBLEM MANAGEMENT

Software Problem Management Process



Topic #2 – Software Test Design

Instructor: Jerry Gao, Ph.D., Professor
San Jose State University

Topic #2 – Software Test Design

- Software Test Design Basics
- Software Test Design Principles
- Software Test Case Templates
- Software Testing Myths
- Software Testing Limitations

Software Test Case Design Basics - I

Software test design is an important task for test engineers.

A good test engineer always know:
→ How to come out quality test cases and
→ How to perform effective tests to uncover as many as bugs in a very tight schedule.

What do you need to come out an effective test set ?

- Choose a good test model and an effective testing method
- Apply a well-defined test criteria
- Generate a cost-effective test set based on the selected test criteria
- Write a good test case specification document

Software Test Case Design Basics- II

What is a good test case?

- **It must have a high probability to discover a software error**
- **It is designed to aim at a specific test requirement**
- **It is generated by following an effective test method**
- **It must be well documented and easily tracked**
- **It is easy to be performed and simple to spot the expected results**
- **It avoids the redundancy of test cases**

Software Test Case Template

What content should be included in a test case?

Test Case ID: **Test Item:**
Wrote By:(tester name) **Documented Date:**
Test Type: **Test Suite#:**
Product Name: **Release & Version No.:**

Test case description: **Operation procedure:**

Pre-conditions: **Post-conditions:**

Inputs data and/or events: **Expected output data & events:**

Required test scripts:

Software Test Design Principles

•Principle #1: Complete testing is impossible.

•Principle #2: Software testing is not simple.

•Principle #3: Testing is risk-based.

•Principle #4: Testing must be planned.

•Principle #5: Testing requires independence.

•Principle #6: Quality software testing depends on:

- Good understanding of software products and related domain application
- Cost-effective testing methodology, coverage, test methods, and tools.
- Good engineers with creativity, and solid software testing experience

Software Testing Myths

- We can test a program completely. In short, we must test a program exhaustively.

- We can find all program errors as long as test engineers do a good job.

- We can test a program by trying all possible inputs and states of a program.

- A good test suite must include a great number of test cases.

- Good test cases always are complicated ones.

- Test automation can replace test engineers to perform good software testing.

- Software testing is simple and easy. Anyone can do it. No training is needed.

Software Testing Limits

- Due to the testing time limit, it is impossible to achieve total confidence.

- We can never be sure the specifications are 100% correct.

- We can never be certain that a testing system (or tool) is correct.

- No testing tools can copy with every software program.

- Test engineers never be sure that they completely understand a software product.

- We never have enough resources to perform software testing.

- We can never be certain that we achieve 100% adequate software testing.

Topic #3 – Software Test Management

Instructor: Jerry Gao, Ph.D., Professor
San Jose State University

Topic #3 – Software Test Management

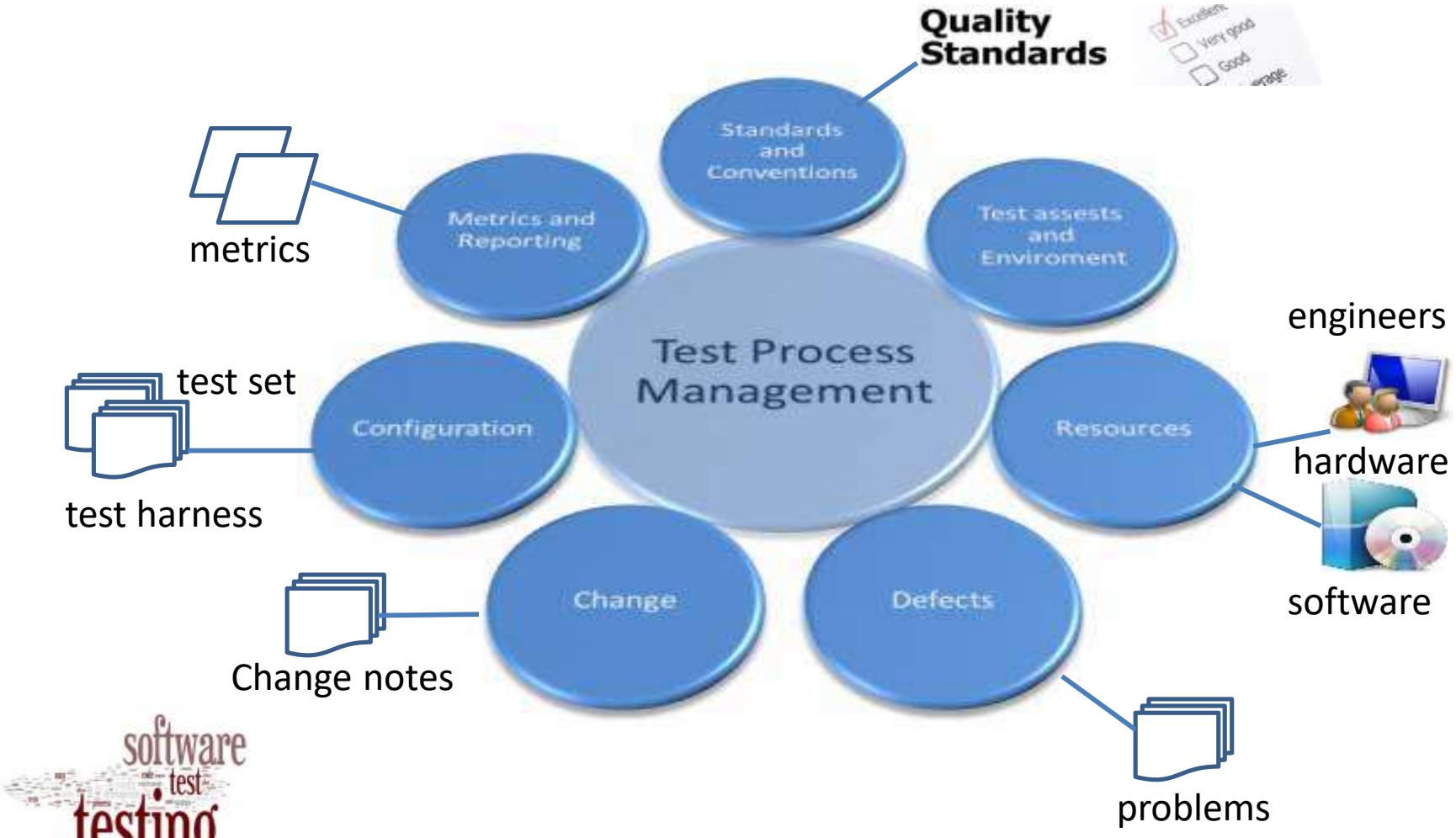
Software Test Management Process

Software Test Reporting and Analysis

Software Test Review

Software Test Management

TOPIC #3 – SOFTWARE TEST MANAGEMENT



TOPIC #3 – SOFTWARE TEST MANAGEMENT

Software Test Record Sample

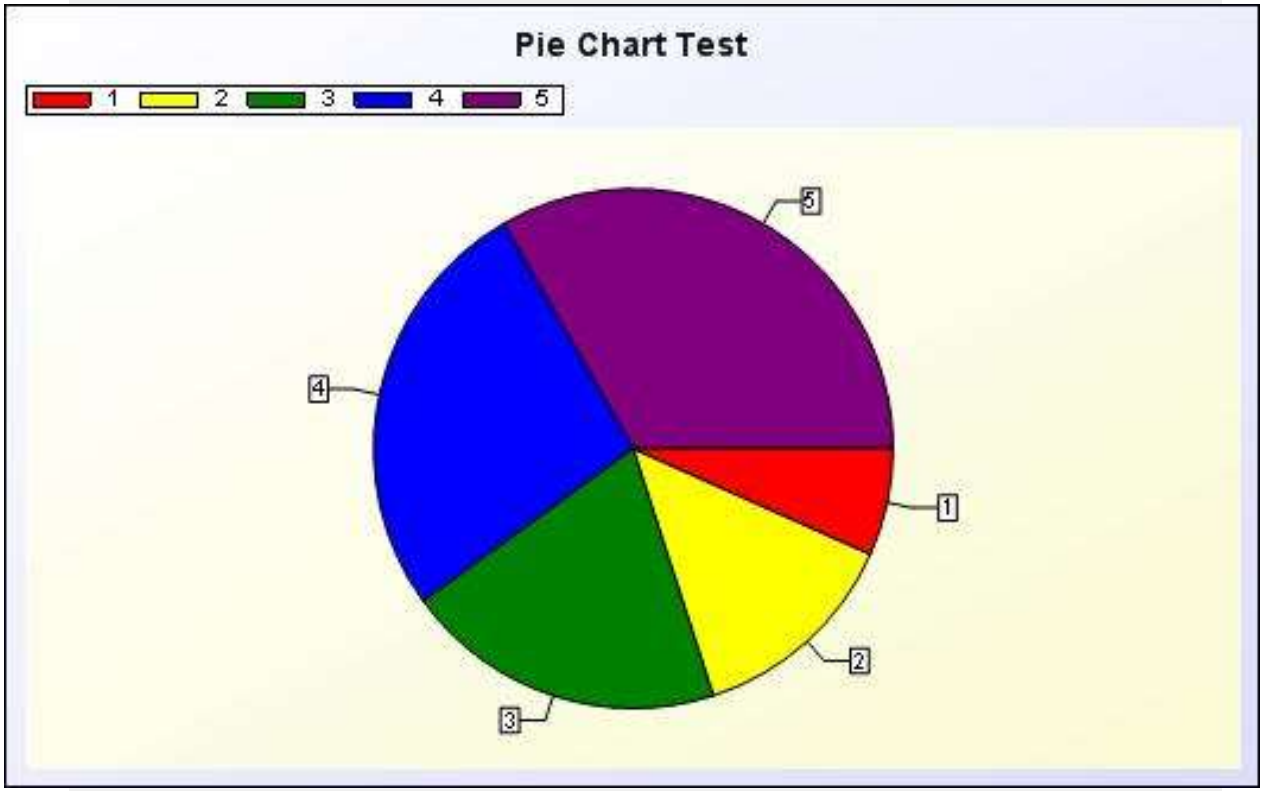
Certificates | Page 1 | Appliance Details | Additional Comments

PLIANCE DETAILS AND TEST RESULTS

Add Appliance | Edit | Delete | Find | Previous | Next | Import | Sort By | Info Bar

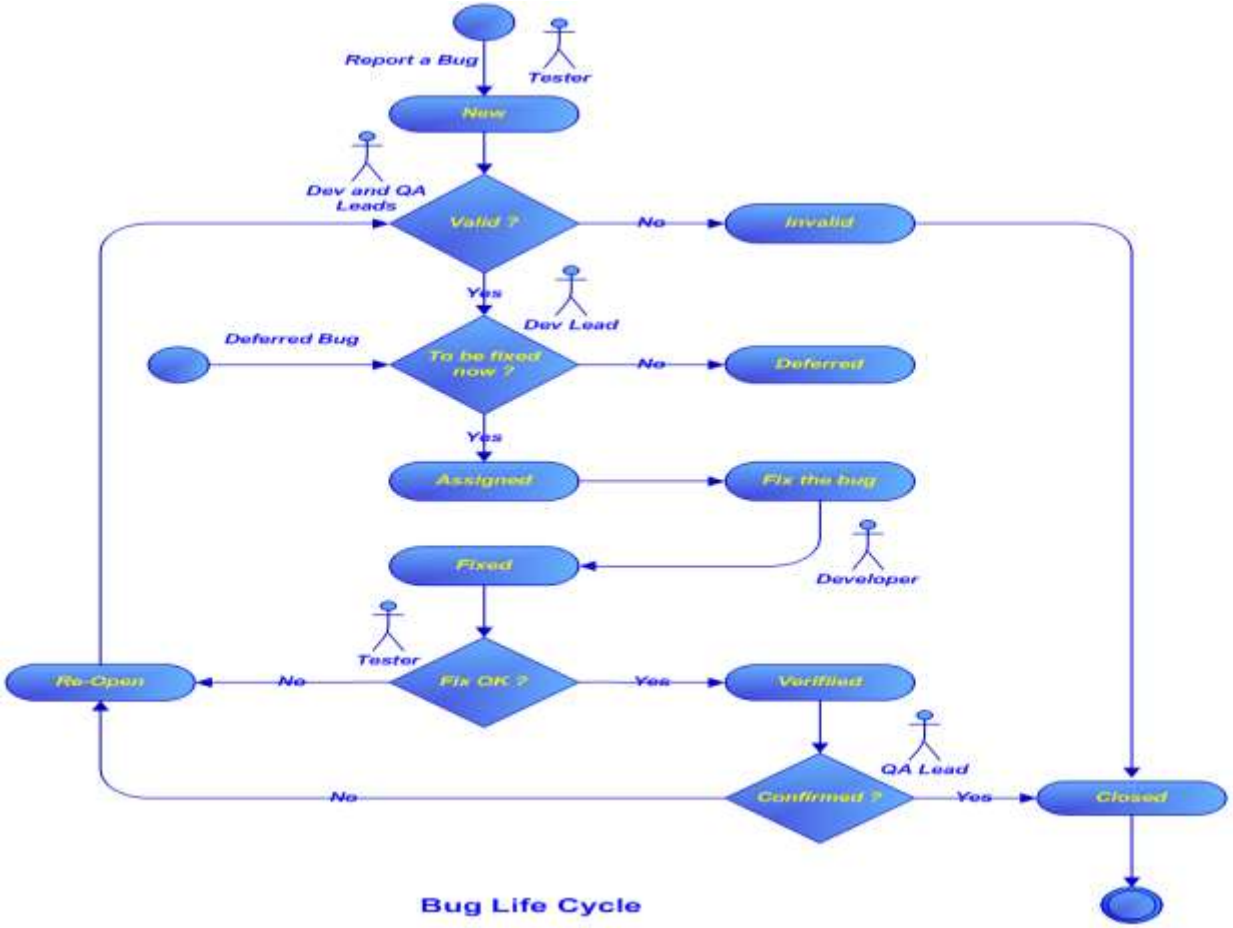
Appliance ID	Test Date	Site	Description	Location	Serial Number	Retest Period	Retest Date	Status
AP0001	29/07/2008	Joe Company 1	IEC Lead	Office	N/A	6	29/01/2009	Pass
AP0002	29/07/2008	Joe Company 1	240v Extension Lead	Board Room	N/A	6	29/01/2009	Fail
AP0003	29/07/2008	Joe Company 1	110v Extension Lead	Warehouse	N/A	6	29/01/2009	Fail
AP0004	29/07/2008	Joe Company 1	Monitor	Ground Floor Office	12345678	12	29/07/2009	Pass
AP0005	29/07/2008	Joe Company 1	Fax Machine	1st Floor Office	12345678	12	29/07/2009	Fail
AP0006	29/07/2008	Joe Company 3	Photo Copier	Meeting Room	N/A	12	29/07/2009	Pass
AP0007	29/07/2008	Joe Company 3	Printer	Board Room	87654321	12	29/07/2009	Pass
AP0008	29/07/2008	Joe Company 3	Scanner	Reception	N/A	12	29/07/2009	Pass
AP0009	29/07/2008	Joe Company 2	Kettle	Kitchen	N/A	48	29/07/2012	Pass
AP0010	29/07/2008	Joe Company 2	Fridge	Kitchen	N/A	12	29/07/2009	Pass
AP0011	29/07/2008	Joe Company 2	Microwave	Kitchen	12345678	12	29/07/2009	Pass
AP0012	29/07/2008	Joe Company 2	Amplifier	Bed Room	N/A	12	29/07/2009	Pass
AP0013	29/07/2008	Joe Company 2	Answerphone	Reception	N/A	12	29/07/2009	Pass
AP0014	29/07/2008	Joe Company 2	Cash register	Reception	87654321	12	29/07/2009	Pass
AP0015	29/07/2008	Joe Company 2	CD PlayerTC/VCR	Bed Room	87654321	12	29/07/2009	Pass
AP0016	29/07/2008	Joe Company 2	Chest freezer	Kitchen	N/A	12	29/07/2009	Pass
AP0017	29/07/2008	Joe Company 1	Chest fridge	Kitchen	12345678	12	29/07/2009	Pass
AP0018	29/07/2008	Joe Company 1	Coffee maker	Kitchen	N/A	12	29/07/2009	Pass
AP0019	29/07/2008	Joe Company 1	Convector heater	Office	N/A	12	29/07/2009	Pass
AP0020	29/07/2008	Joe Company 1	Dehumidifier	Office	12345678	12	29/07/2009	Pass
AP0021	29/07/2008	Joe Company 1	Desk light	Office	N/A	12	29/07/2009	Fail
AP0022	29/07/2008	Joe Company 1	Dish washer	Kitchen	N/A	12	29/07/2009	Pass
AP0023	29/07/2008	Joe Company 1	Dryer	Kitchen	N/A	12	29/07/2009	Pass
AP0024	29/07/2008	Joe Company 1	Electric blanket	Bed Room	N/A	12	29/07/2009	Pass
AP0025	29/07/2008	Joe Company 1	Electric blender	Kitchen	N/A	12	29/07/2009	Pass
AP0026	29/07/2008	Joe Company 1	Electric clock	Bed Room	N/A	12	29/07/2009	Pass

Software Test Analysis Report



TOPIC #3 – SOFTWARE TEST MANAGEMENT

Software Bug Management Flow



Bug Life Cycle

Software Test Review

- A test review is a typical verification activity in a test process to assure the quality of generated test cases for a software product.

- Participants in a review take full responsibility for results.

There are two types of test reviews:

- **Formal reviews:**
  - **use a well-defined review method (or technique)**
  - **inspection review and walk-through**
  - **generate formal review results**

- **Informal reviews**
  - **use a desk-check approach**
  - **generate informal review results**

TOPIC #3 – SOFTWARE TEST MANAGEMENT

Software Test Case Sample

TEST CASE REPORT

(Use one template for each test case)

GENERAL INFORMATION			
Test Stage:	☐ Unit ☐ Functionality ☐ Integration ☐ System ☐ Interface ☐ Performance ☐ Regression ☐ Acceptance ☐ Pilot Specify the testing stage for this test case.		
Test Date:	mm/dd/yy	**System Date, if applicable:**	mm/dd/yy
Tester:	Specify the name(s) of who is testing this case scenario.	**Test Case Number:**	Specify a unique test number assigned to the test case.
Test Case Description:	Provide a brief description of what functionality the case will test.		
Results:	☐ Pass ☐ Fail	**Incident Number, if applicable:**	Specify the unique identifier assigned to the incident.
INTRODUCTION			
Requirement(s) to be tested:	Identify the requirements to be tested and include the requirement number and description from the Requirements Traceability Matrix.		
Roles and Responsibilities:	Describe each project team member and stakeholder involved in the test, and identify their associated responsibility for ensuring the test is executed appropriately.		
Set Up Procedures:	Describe the sequence of actions necessary to prepare for execution of the test.		
Stop Procedures:	Describe the sequence of actions necessary to terminate the test.		
ENVIRONMENTAL NEEDS			
Hardware:	Identify the qualities and configurations of the hardware required to execute the test case.		

TOPIC #3 – SOFTWARE TEST MANAGEMENT

Software Test Review

The following deliverables must be reviewed in a software test process:

Test Plan
Test Report
Test Design Specification
Problem/bug Reports

What does test reviews accomplish?

- Test reviews provide the primary mechanism for evaluating generated test cases.

- Test reviews train and educate the participants to receive a positive effect.

- Reviews give early feedback and prevent more serious problems from arising.

- Reviews bring individual capability to light.

TOPIC #3 – SOFTWARE TEST MANAGEMENT

Software Test Management

test team management

test process management

test project management

test manager

- **Play as a leadership role in:**
  - **planning projects**
  - **setting up a direction**
  - **motivating people**
  - **build a team**
  - **manage engineers**
- **Play as a controller in:**
  - **product evaluation**
  - **performance evaluation**
  - **changing to a new direction**
- **Play as a supporter in:**
  - **assist and train engineers**
  - **train engineers**
  - **enforce and control test methods, standards, and criteria**
  - **select and develop test tools**

Software Test Management - I

- Management	**- Manage test projects** **- Manage team members** **- Manage test processes**
- Motivation	**- Motivate quality work from team members** **- Simulate for new ideas and creative solutions**
- Methodology	**- Control of setting up test methodology, process, standards.** **- Control of establishing test criteria**
- Mechanization	**- Control the selection and development of test tools** **- Mechanism for the configuration management of test suites** **- Control of setting up an integrated test environment**
- Measurement	**- Measure test cost, complexity and efforts** **- Measure engineer performance** **- Measure test effectiveness** **- Measure product quality**

Software Test Management - II

Management:	**Do you plans address testing?** **Do you know who is responsible?** **Have you published your testing policy?**
Motivation:	**Do you provide incentive for people do quality work?** **Do you encourage people to take advantage of training opportunities in testing methods?**
Methodology:	**Are your engineers trained to use test methods?** **Are you aware of new testing techniques and use them?**
Mechanization:	**Do you sufficient hardware and equipment to support testing?** **Have you provided appropriate software testing tools and aids?** **Do you evaluate automated testing aids on an ongoing basis?**
Measurement:	**Do you track errors, faults, and failures?** **Do you know what testing costs?** **Do you quantitatively measure testing performance?**