SNS COLLEGE OF TECHNOLOGY

COIMBATORE

AN AUTONOMOUS INSTITUTION

Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A' Grade

Approved by AICTE New Delhi & affiliated to the Anna University, Chennai

DEPARTMENT OF MCA

Course Name : 19CAE709 - SOFTWARE TESTING AND QUALITY ASSURANCE

Class : II Year / III Semester

Unit I - Introduction

Topic V – Validation Testing

Objective

Concerns

Software not in a testable mode

Inadequate time/resources

Significant problems will not be uncovered during testing

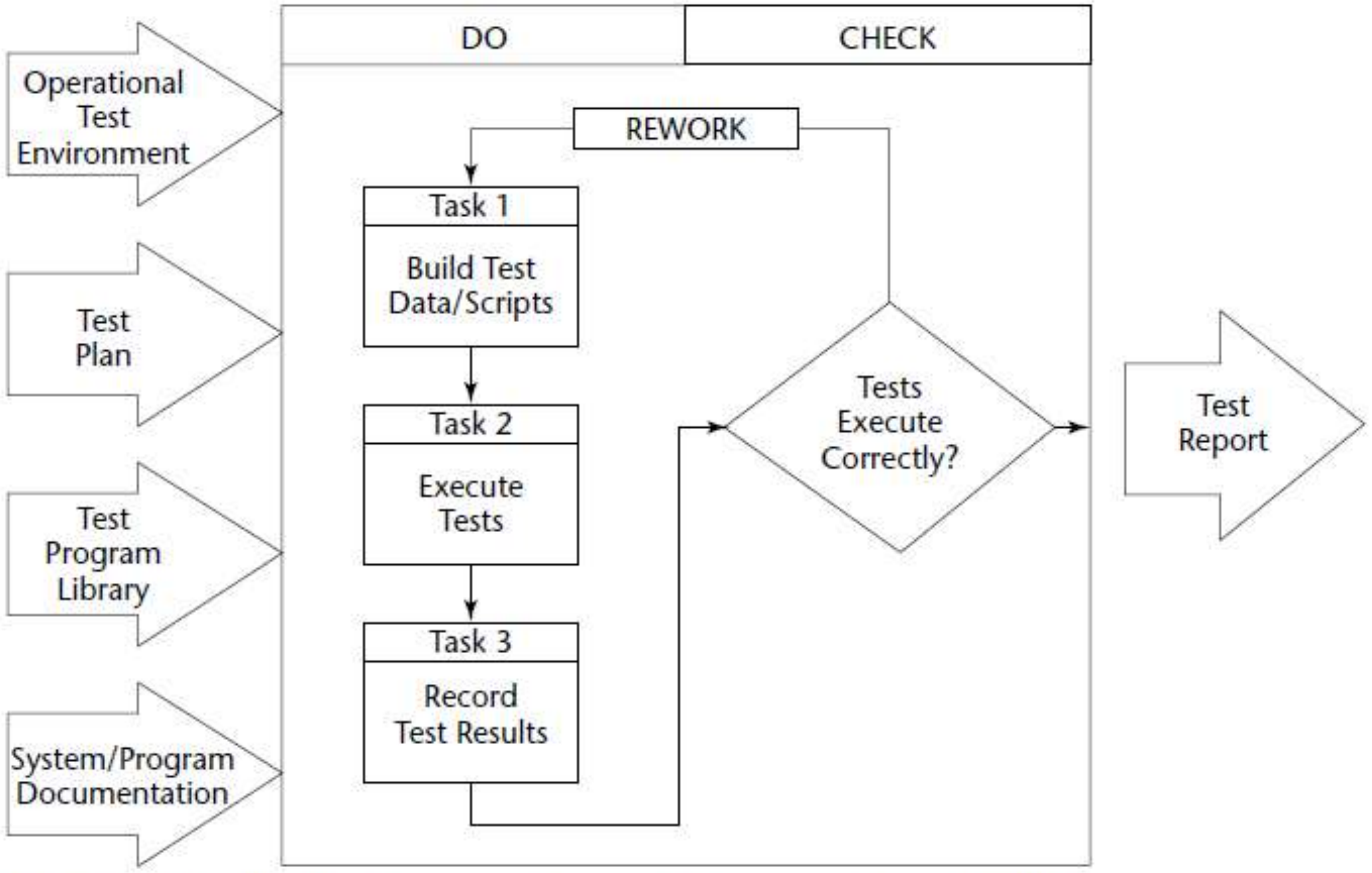


Figure 10-1 Workbench to execute dynamic tests and record results.

Input

The deliverables that are available during the validation testing include:

- System test plan (may include a unit test plan)
- Test data and/or test scripts
- Results of previous verification tests
- Inputs from third-party sources, such as computer operators

Do Procedures

1. Build the test data.
2. Execute tests.
3. Record test results

Task 1: Build the Test Data

Sources of Test Data/Test Scripts

- System documentation
- Use cases
- Test generators
- Production data
- Databases
- Operational profiles
- Individually created test data/scripts

Task 1: Build the Test Data

Testing File Design

- Tests of normally occurring transactions
- Tests using invalid data
- Tests to violate established edit checks

Task 1: Build the Test Data

Defining Design Goals

Entering Test Data

Applying Test Files Against Programs That Update Master Records

Creating and Using Test Data

1. Identify test resources
2. Identify test conditions
3. Rank test conditions
4. Select conditions for testing
5. Determine correct results of processing
6. Create test transactions
7. Document test conditions
8. Conduct test
9. Verify and correct test results

Payroll Application Example

Creating Test Data for Stress/Load Testing

1. Identify input data used by the program
2. Identify data created by the program
3. Challenge each data element for potential limitations
4. Document limitations
5. Perform volume testing

Creating Test Scripts

Data entry procedures required

Use of software packages

Sequencing of events

Stop procedures

To develop, use, and maintain test scripts, testers should perform the following five steps:

1. Determine testing levels.

2. Develop test scripts.

3. Execute test scripts.

4. Analyze the results.

5. Maintain test scripts.

Payroll Application Example

Creating Test Data for Stress/Load Testing

1. Identify input data used by the program
2. Identify data created by the program
3. Challenge each data element for potential limitations
4. Document limitations
5. Perform volume testing

Creating Test Scripts

Data entry procedures required

Use of software packages

Sequencing of events

Stop procedures

To develop, use, and maintain test scripts, testers should perform the following five steps:

1. Determine testing levels.

2. Develop test scripts.

3. Execute test scripts.

4. Analyze the results.

5. Maintain test scripts.

Payroll Application Example

Determining Testing Levels

There are five levels of testing for scripts, as follows:

Unit scripting

Pseudo-concurrency scripting

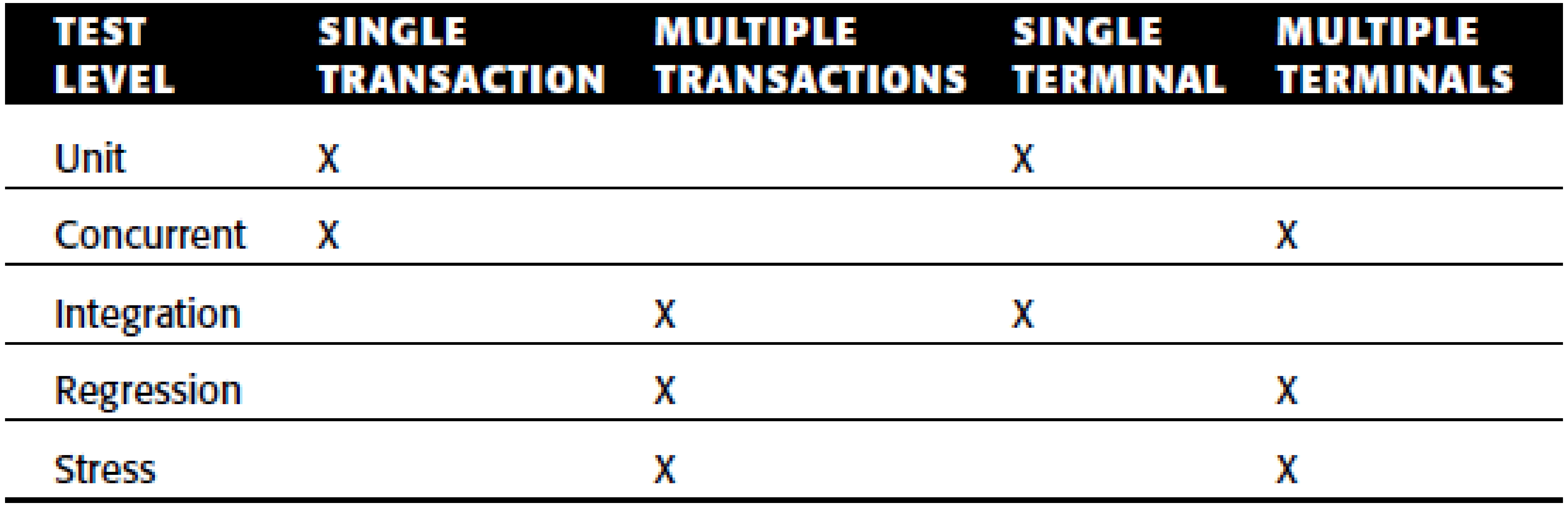
Integration scripting

Regression scripting

Stress/performance scripting

Developing Test Scripts

Table 10-1 Script Development Strategies



TEST LEVEL	SINGLE TRANSACTION	MULTIPLE TRANSACTIONS	SINGLE TERMINAL	MULTIPLE TERMINALS
Unit	X		X	
Concurrent	X			X
Integration		X	X	
Regression		X		X
Stress		X		X

Payroll Application Example

Executing Test Scripts

- Environmental setup
- Program libraries
- File states/contents
- Date and time
- Multiple terminal arrival modes
- Serial (cross-terminal) dependencies
- Processing options
- Stall detection
- Synchronization of different types of input data
- Volume of inputs
- Arrival rate of input

Payroll Application Example

Analyzing the Results

- System components
- Terminal outputs (screens)
- File contents
- Environment variables, such as
- Status of logs
- Performance data (stress results)
- Onscreen outputs
- Order of outputs processing
- Compliance of screens to specifications
- Ability to process actions
- Ability to browse through data

Payroll Application Example

Maintaining Test Scripts

- Identifiers for each script
- Purpose of scripts
- Program/units tested by this script
- Version of development data that was used to prepare script
- Test cases included in script

Task 2: Execute Tests

Manual, regression, and functional testing (reliability)

Functional and regression testing (coupling)

Compliance testing

- Authorization
- Performance
- Security

Functional testing

File integrity

Audit trail

Correctness

Recovery testing (continuity of testing)

Stress testing (service level)

Testing complies with methodology

Manual support testing (ease of use)

Inspections (maintainability)

Disaster testing (portability)

Operations testing (ease of operations)

Task 3: Record Test Results

Testers must document the results of testing so that they know what was and was not achieved. The following attributes should be developed for each test case:

- Condition
- Criteria
- Effect
- Cause

A well-developed problem statement includes each of these attributes. When one or more of these attributes is missing, questions almost always arise, such as:

Condition. What is the problem?

Criteria. Why is the current state inadequate?

Effect. How significant is it?

Cause. What could have caused the problem?

Task 3: Record Test Results

Documenting the Deviation

The statement of condition should document as many of the following attributes as appropriate for the problem:

- Activities involved
- Procedures used to perform work
- Outputs/deliverables
- Inputs
- Users/customers served
- Deficiencies noted

Documenting the Effect

Task 3: Record Test Results

Documenting the Cause

Most findings involve one or more of the following causes:

Nonconformity with standards, procedures, or guidelines

Nonconformity with published instructions, directives, policies, or procedures from a higher authority

Nonconformity with business practices generally accepted as sound

Employment of inefficient or uneconomical practices

Output

Validation testing has the following three outputs:

- The test transactions to validate the software system
- The results from executing those transactions
- Variances from the expected results

Reference

1. https://centricconsulting.com/client-stories/world-class-software-testing-organization/
2. https://www.edureka.co/blog/software-testing-models/

THANK YOU