GROUP TECHNOLOGY & OPERATIONS

Testing Guide (SDLC)

Global ID: DBS_11_G_0033_GR

Scope/Coverage: Group-wide

Issuer: Soo Lee Poh, SDLC Governance Team Lead

Associated Unit: Enterprise Architecture-SRE

Last Review Date: 01 October 2020

Review Frequency: Annually

Reference Policies, Standards, Guides:

No.	Global ID	Name	Document Type
1	DBS_11_S_0027_GR	DBS System Development Life Cycle Standard	Standard
2	DBS_11_S_0008_GR	High Availability and Disaster Recovery Standard	Standard
3	DBS_11_G_0035_GR	Configuration Management Guide	Guide
4	DBS_11_G_0036_GR	Quality Management Guide	Guide

Associated Applications

No.	Application Code	Application Name
		Not applicable

Associated Process Maps/Tools/Templates

No.	Name
1	SDLC Portal
2	Functional and Security Test Scenarios
2	Test Plan
3	Test Specifications
4	Test Defect Log
5	Test Progress Report
6	Test Summary Report
7	Performance Test Summary Report

CONTENTS

1 INTRODUCTION

1.1 Purpose

The purpose of this guide is to provide a set of guidelines that will support the planning, managing, and implementation of testing activities.

It is recommended to use this guide as a supplement after reading the DBS SDLC Standard and Waterfall and Agile Procedures.

1.2 Audience

This guide is created for all Application Teams, Test Managers, Test Leads, Testers, Application Managers, Business Analysts, Business Lead and Key Stakeholders (BU/SU, including Ops), and Delivery Team/Vendor.

1.3 Scope

The scope of this guidelines document covers the recommended practices for the different tests commonly done in the Bank, except Unit Testing. For unit testing, refer to the Code and Build Guide.

2 TEST PROCESS OVERVIEW

2.1 Testing Activities

There are three key activities that comprise the entire testing process:

PLANNING	EXECUTION	CLOSURE
The preparations done prior to the actual testing activities (e.g., scope identification, scheduling, resource allocation)	The implementation of testing activities, including defect management	The activities done to conclude the testing process

2.2 Testing Types

To meet the DBS SDLC Standard, the following tests must be conducted (as applicable):

- System Integration Testing (SIT)
- User Acceptance Testing (UAT)
- Performance Testing
- Security Testing
- Regression Testing

3 PLANNING

3.1 Test Plan Creation

The objective is to plan out the testing activities and other relevant details with the involved stakeholders. At the very least, a Test Plan should cover:

a. Project overview
   - Testing scope
   - Test environment and test data requirements

b. Test Definition
   - Test objective and approach
   - Tests to be conducted

c. Test execution governance
   - Entry and exit criteria
   - Defect management

- Test tools

d. Resource and Schedule

- Resource allocation (e.g., people, environment, tools, etc.)
- Test schedule and deliverables

To document these details, teams can do either of the following:

- Use the Test Plan template – commonly used for Waterfall projects
- Define and integrate test plan tasks under **User Stories** – commonly done in Agile projects

The resulting plan or tasks should be signed off (or agreed upon) by the Project Manager or Scrum Master for SIT, and Business Lead for UAT.

If Vendors are engaged for Projects / Enhancements:

- The Test Plan and/or User Stories must be provided to DBS.
- The DBS Test Manager and/or Project Manager or Scrum Master has reviewed and signed off.

3.2 Test Script Creation

Describe in detail what will be tested and how testing will be carried out. The Test Specification template can be used to document the test scenarios and test cases.

Test Scenario – What is to be tested?

This is a high-level classification of test requirements, which are grouped based on the functionality of a module. Refer to the sample (functional and security) test scenarios for guidance on the essential test coverage. These test scenarios can be uploaded to most of the Bank's supported tools.

Test Case – How to conduct the test?

These are low-level actions derived from test scenarios. When designing test cases, make sure to:

a. Capture the following:

- Test conditions
- Steps to execute
- Expected results
- Traceability to the requirements (including any requirement changes) being tested

b. Cover all of the functional and non-functional requirements mentioned in the requirement specifications.

Test cases should be comprehensive in its coverage with valid and invalid data value to cover both negative and positive testing. Examples of conditions for negative testing are embedded single quote test data, required data entry, field type entry, and field size test.

c. Define the following details for the test data:

- Source and type of test data
- Expiration date (e.g., after a specific date due to the policy to clear masked production data)
- Range of the test data value

d. Parameterize the test cases with scenarios that require multiple sets of test data.

It is recommended to start planning for test cases—even acceptance test cases—as early as possible (i.e., while preparing the business case, needs assessment, and requirements analysis). Early preparation of test scripts helps to include script details that may otherwise be forgotten and provide ideas to re-engineer the way things are done.

Aside from early preparation of scripts, it is also advised to plan for additional cycles of acceptance testing for contingency.

If Vendors are engaged for Projects / Enhancements:

- Test cases with traceability to requirements must be provided to DBS.
- The DBS Business Analyst and/or Test Manager or Scrum Master have reviewed and agreed on the test coverage.

3.3 Test Data Management

For data request, Development Teams can contact their respective Application / Data Owner. Use the D'Concierge to raise request for Data Restoration (Production to UAT).

There are two types of test data:

1. **Generated Test Data** – created specifically for testing new or enhanced applications as per the testing requirement.

   This test data can be created by:

   - The front-end application (manual)
   - Running automated scripts via data creation or automation tools
   - Bulk or file uploading of test data (e.g., using a file upload in XML format)
   - Back-end uploading of test data (although this may not be entirely reliable due to the complexity of the application test database structures and data table dependencies)

2. **Live Test Data** – a copy of the existing production data is being used to test the new or enhanced application as per the testing requirement

   This test data can be created by:

   - Migrating or converting test data from the previous version of an application under test, or from a source application used in the past to the target future application
   - Moving live data from the actual production environment, or from a DR environment, using tools

   When using live test data, ensure the following:

   a. Sensitive customer information are masked.

   b. The test data is moved into the test environment, and is in sync with the interfaces present in the test environment.

   c. Batch processes are in place to get the initial state of data to its desired state. These should be aligned with the test execution plan for efficient output.

   d. Data profiling and data subsetting are performed in case there is a high volume of production data (i.e., for performance testing requirements).

e. Live test data is purged upon completion of testing, or on the expiration date of the data request.

3.4 Test Environment Management

Test Environment should be planned and set up as per the test requirements and scope during the test schedule. Teams can follow below activities as guide:

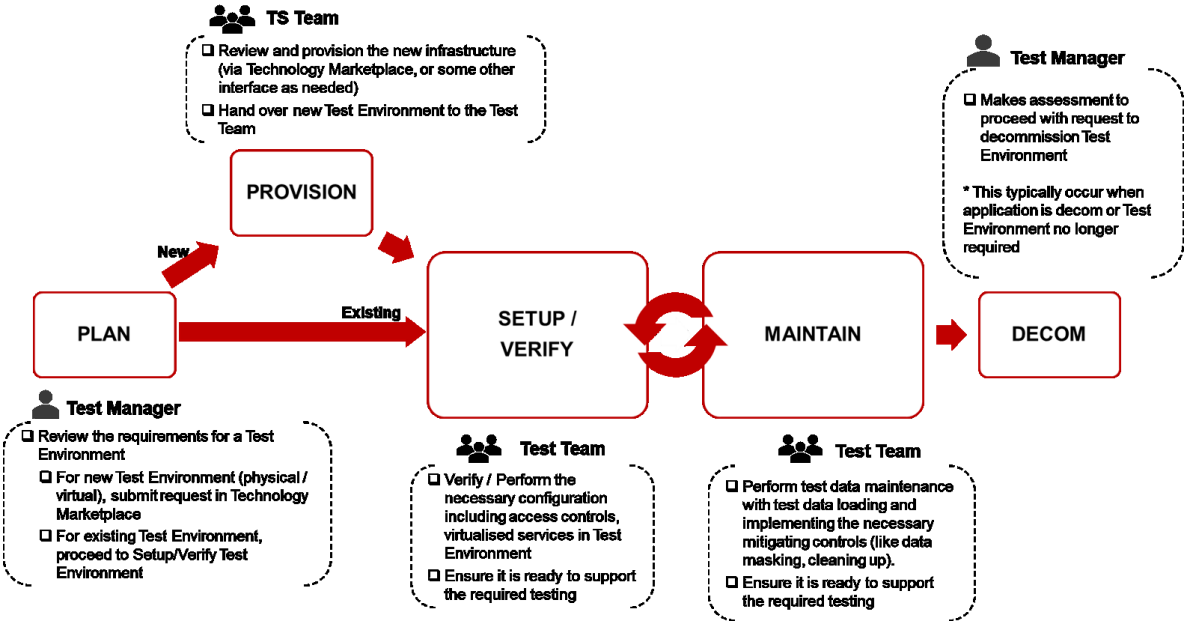

Figure 1 - Test Environment Management Process

RACI Matrix

Activity	Test Manager	Infra Team	Test Team	Notes
PLAN				
Review the requirements for a Test Environment	R	C	C	Performed during testing planning.
For new Test Environment (physical / virtual), submit request in Technology Marketplace	R	I	I	
For existing Test Environment, proceed to Setup / Verify Test Environment	A		I	
PROVISION				
Review and provision the required infrastructure (via Technology Marketplace, or some other interface as needed)	C	R	C	

Activity	Test Manager	Infra Team	Test Team	Notes
Hand over the new Test Environment to the Test Team	R		I	Test Manager informed when Test Environment is provisioned
SETUP / VERIFY				
Verify / Perform the necessary configuration including access controls, virtualised services in Test Environment and ensure it is ready to support the required testing	A	C	R	Use BOLT SV Genie to request for virtualised services
MAINTAIN				
Perform test data maintenance with test data loading and implementing the necessary mitigating controls (like data masking, cleaning up) and ensure it is ready to support the required testing	A		R	Use D'Concierge to request for data restoration
DECOM				
Makes assessment to proceed with request to decommission Test Environment	R	C	C	This typically occur when the application is decommissioned or when Test Environment no longer required
R = **R**esponsible **A** = **A**ccountable **C** = **C**onsulted **I** = **I**nformed				

3.5 Test Entry and Exit Criteria

Entry criteria are factors that must be present to start testing, and exit criteria are factors that must be present to declare that the testing activity is complete.

Below are the recommended entry and exit criteria for each test type. Teams, however, can add, amend, or remove entries as they deem fit.

3.5.1 System Integration Testing

Entry Criteria

1. Unit testing has been conducted and passed. The Unit Test Summary Report has been delivered and reviewed by the Technical Lead.
2. The System Requirement Specifications is signed off, and a walkthrough has been done.
3. The latest version of the code has been deployed to the SIT environment.
4. SIT test cases have been reviewed and approved.
5. SIT Testers have been trained on system functionality and the tools required for testing.
6. Test environments have been provisioned and are available. Testers have access to the required test environment and tools.

7. Test data are populated in the SIT environment.
8. The SIT Test Plan has been reviewed and signed off.
9. Sanity or Smoke testing has been completed and passed.

Exit Criteria

1. All SIT test cases have been executed at least once (100% execution coverage), with updated test results.
2. At least 80% of the executed SIT test cases have passed.
3. All SL1 and SL2 defects from SIT have been addressed and resolved.
4. There is an agreed plan to resolve SL3 and SL4 defects.
5. The SIT Summary Report has been reviewed and signed off by the Project Manager and Technical Lead.
6. The latest version of the code has been deployed to the UAT environment.

3.5.2 User Acceptance Testing

Entry Criteria

1. SIT has been conducted and passed. The SIT Summary Report has been delivered and reviewed by the Project Manager and Technical Lead.
2. The latest version of the code has been deployed to the UAT environment.
3. UAT test cases have been reviewed and approved.
4. UAT Testers have been trained on system functionality and tools required for testing.
5. Test environments have been provisioned and are available. Testers have access to the required test environment and tools.
6. Test data are populated in the UAT environment.
7. The UAT Test Plan has been reviewed and signed off.
8. Sanity or Smoke testing has been completed and passed.
9. Regression test cases have been identified and reviewed.

Exit Criteria

1. All regression test cases have been executed with 100% passed. No defects resulted from regression testing.
2. All UAT test cases have been executed at least once (100% execution coverage), with updated test results.
3. At least 90% of the executed UAT test cases have passed.
4. All SL1 and SL2 defects from UAT have been addressed and resolved.
5. There is an agreed plan to resolve SL3 and SL4 defects.
6. The UAT Summary Report has been reviewed and signed off by the Project Manager and Business Lead.
7. A formal UAT sign-off has been obtained from the business.

3.5.3 Performance Testing

Entry Criteria

1. Non-functional requirements (NFRs) are baselined and documented.
2. Performance test scope has been assessed based on defined NFRs and change impact.
3. Performance test activities have been planned and included in the overall Test Plan.
4. Test data/scripts have been determined and prepared.
5. Test environments have been provisioned and are available. Testers have access to the required test environment and tools.

Exit Criteria

1. All scripts have been completed and are successfully running with multiple users for multiple iterations.
2. The planned Performance Testing has been completed, and results are recorded and stored in a central repository.
3. Performance test issues are tracked and addressed accordingly.
4. The Performance Test results have been approved.

4 EXECUTION

4.1 Testing Types

4.1.1 System Integration Testing

SIT is conducted to validate all software modules are functionally correct, and to verify there is proper end-to-end execution and integration between the different systems within the solution.

4.1.2 User Acceptance Testing

UAT is conducted to verify that the system or application meets the user's requirements / Definition of Done. This test enables users, customers, or other authorized entities to determine whether they can accept the system. For UAT, utilize live or "near live" data, environments, and user scenarios; focusing on typical product usage scenarios, and not extreme conditions.

4.1.3 Security Testing

The goal of Security Testing is to verify that a system's security features are implemented according to the security requirements and design specifications. It also aims to validate that source codes do not contain any code-related security issues during implementation.

To meet these objectives, teams must conduct Security Testing through:

1. Security Functional Testing – to ensure the security requirements and controls are in place.

   Test Scenarios / Cases and Test Results must be provided to substantiate that required testing has been performed successfully.

2. The use of Static Application Security Testing (SAST) and Dynamic Application Security Testing (DAST) tools – to uncover code security issues

SAST is designed to check an application's source code for security vulnerabilities. It is conducted on regular cadences (e.g. upon successful build; or on a nightly schedule) during Code and Build.

DAST is designed to detect security vulnerabilities of an application in its running state (e.g., web applications). It is conducted on a functionally stable build (e.g., after the first iteration of SIT).

***NOTE**: To determine if a SAST and DAST tool is compatible with the application, refer to the Fortify Playbook.*

3. Application Security Assessment (ASA)

   Refer to the Security Assessment Standard and ISS Pentest Operation SharePoint for more details on scope, criteria and findings resolution.

4. Automated Security Testing in Quarantine Environment for Mobile Applications

   A quarantine environment has been set up to conduct security check on mobile applications (via automated scripts) before deployment. The checkpoints include:

   SSL Cert Pinning - To prevent Man-in-the-Middle attacks or sniffing of HTTPS traffic (applicable to Android and iOS devices)

   Code Reverse Engineering - To verify if code obfuscation is enabled for code reverse engineering prevention (applicable to Android devices only)

5. System Security Acceptance Testing (SSAT) for Critical Information Infrastructure (CII) Applications.

   If SSAT is required as determined in the Security Risk Planning (SRP) Form, Application Team must

   - Engage ISS Pentest Team to perform SSAT
   - Obtain approval from ISS Pentest Team on the SSAT

NOTE:

- Define mitigation plan to fix **ALL Critical**, **High**, and **Medium** security issues. Refer to Fortify or Open Source Software Quality Gate Process, if required. Submit the Application Security Assessment Approval prior to production deployment.

4.1.4 Performance Testing

Performance Testing is conducted to ensure the system or application meets the defined non-functional requirements (NFRs).

It is recommended to conduct Performance Testing towards the end of UAT. An initial test is conducted based on the baselined NFRs, and then subsequent tests thereafter for the following conditions to ensure the system's quality attributes are not impacted.

1. If there is a revision to non-functional requirements of the application or system.
2. Where changes have been made to the application's core processing.
3. When there are infrastructure changes to the application servers.
4. For Cat 1, 2. And 3 applications, refer to the High Availability and Disaster Recovery Standard for the required frequency to conduct.

At the end of performance test, the overall results are analysed and compiled in the Performance Test Summary Report. This report details all the test cycles, types, defects, and performance results.

4.1.5 Regression Testing

Regression testing ensures the system or application still performs correctly after any changes to:

- Application code for new or enhanced features
- Application code for defect resolution including production hotfixes
- Data that has been migrated (e.g. one-off data import)
- Infrastructure and/or application configuration (e.g. network, web and application server configuration files)
- Infrastructure, including Operating System upgrades and Security Patches

Development Teams may identify the set of test cases to be used for Regression Testing based on the following:

- Test cases that are testing the **Core** feature of the system. Core feature refers to system functionality that when malfunction will result in Production Incident Severity Level 1, 2 and 3. Refer to the T&O Severity Level Guideline.
- **Common** problems that always surface during testing or production issues. Development Teams could utilize defect tracking system in identifying areas of the system that has statistically more errors.

Test cases that have been selected for Regression Testing will have to be reviewed and updated upon each production release. This is to ensure the validity of the test cases.

Regression Testing is to be conducted during or at the end of UAT.

***If ASA is applicable**, another round of Regression Testing is to be done at the end of ASA if there are any changes introduced. For the ASA exit criteria, refer to Security Assessment Standards.*

Regression Testing is deemed completed when all regression test cases have been executed with 100% passed i.e. there are no outstanding defects resulted from regression testing.

4.2 Progress and Defect Management

To properly manage the testing progress and issues, the Test Manager or Scrum Master should ensure:

1. Test cases are executed and results are recorded in a test management tool, or in the Test Specifications template.
2. Defects are logged based on agreed severity, managed accordingly, and resolved. The Test Defect Log or a test management tool may be used. Refer to the table below for defect severity definitions.

Severity	Testing Impact	Business Impact
SL 1 - Critical / Severe Impact	The defect prevents the user from executing the task. There is no workaround, and testing cannot continue. The problem is of major impact and is highly visible to business operations.	Critical business operation is down. Full system and/or critical / core services are down; thus, impacting the Bank's ability to meet service level commitments.

Severity	Testing Impact	Business Impact
SL 2 - High / Moderate Impact	The defect prevents the user from executing the task, but there is a workaround and testing can continue. A large percentage of test cases is affected. The problem is of high impact and is highly visible to business operations.	A significant portion of the business operation is down. Partial system and/or critical / core services are down; thus, impacting the Bank's ability to meet service level commitments.
SL 3 - Medium / Minimal Impact	The defect does not prevent the user from executing the task. The tested function does not perform as expected, but testing can continue. A small percentage of test cases is affected, and/or the problem has limited visibility. The system may remain operational, however in a degraded manner.	A unit or component failure does not impact the Bank's ability to meet service level commitments.
SL 4 - Low / No Impact	The defect is related to aesthetic changes (e.g., view layout, field name changes, spelling errors).	A unit or component failure does not impact the Bank's ability to meet service level commitments.

3. A defect impact analysis is performed to plan for appropriate retests and/or regression test.
4. Progress is reported on a regular basis at the agreed interval – daily or weekly. The Test Progress Report template can be used for reference.

Proper and regular updates on the test progress ("status scorecard") should be communicated to managers, and testers. Communication of any detected defect must be constant throughout the testing to avoid the 'big bang' effect, when all issues are reported only upon completion of the testing.

Before concluding the Execution activity, teams should validate the exit criteria are satisfied to confirm the completion of the tests done.

If Vendors are engaged for Projects / Enhancements:

- Documented test results, including defects, must be provided to DBS.
- A progress report must be provided regularly to DBS.

5 CLOSURE

To finalise the testing process, a Test Summary Report must be provided. This report covers the following:

- Test scope
- Test execution results
- Defect summary and metrics
- Exit criteria validation
- Conclusion and recommendations

All Test Summary Reports must be signed off by the relevant stakeholders.

If Vendors are engaged for Projects / Enhancements:

- The Test Summary Report must be provided to DBS.
- The DBS Project Manager or Scrum Master and Technical Lead have reviewed and signed off the SIT Summary Report.
- The DBS Project Manager or Scrum Master, Technical Lead and Business Lead have reviewed and signed off the UAT Summary Report.

APPENDIX 1 ADDITIONAL REFERENCES

1. D'Concierge (For Data Extraction and Sharing requests)
2. Fortify Playbook
3. Playbook on Open Source Software (OSS) Risk Management Tool
4. T&O Severity Level Guideline
5. Security Assessment Standard
6. ISS Pentest Operation - Pentest and System Security Acceptance Testing request
7. Security By Design Playbook - CSA SBD framework and DBS SBD 2.0 Requirements

APPENDIX 2 VERSION HISTORY

Version	Date of Issue	Summary of Key Changes
1.0	10 Mar 2017	Initial version.
1.1	26 Jun 2017	• Updated Security Testing. • Updated Performance Testing. • Removed Appendix for Standard Test Date Requirement Template. • Removed Test Environment Booking Form and Contact Template.
2.0	30 Sept 2017	• Reorganized flow and revised content for ease of navigation and readability. • Revised content to include applicability to Agile-related projects. • Added data migration as part of Regression Testing condition.
2.1	29 Mar 2018	• Added reference to sample functional and security test scenarios • Removed Pre-Go Live Metrics (PGLM) Report – sunset wef Jan 2018 (as T&O Project Executive Committee [PEC] has been cancelled.
2.2	29 Jun 2018	• Added Entry and Exit Criteria for SIT / UAT / Performance Testing in Planning. • Defect severity definitions included under Progress and Defect Management section. • Added Automated Security Testing in Quarantine Environment for Mobile Applications.
2.3	15 Apr 2019	• Change Associated Unit from ITSS to Enterprise Architecture-SRE.
2.4	01 Jun 2019	• Update for clarity on Performance Testing requirements.
2.5	01 Jul 2019	• Updates for Cyber Security Agency Security By Design framework: ○ Added Independent Security Acceptance Testing under Execution-Security Testing for Critical Information Infrastructure (CII) Applications.
2.6	30 Sep 2019	• Updated Planning - Test Data Management, Request for Data Restoration - Production to UAT to be raised in D'Concierge instead of Online Data Shop, as at 24-May-2019.
2.7	28 Feb 2020	• Update to include Fortify playbook as recommended from TW Regulator check on Source Code Review Process.
2.8	01 Jun 2020	• DBS CSA SBD 2.0 Implementation for CII Applications.

Version	Date of Issue	Summary of Key Changes

2.9	01 Oct 2020	• Updated 3.4 Test Environment Management - Add Test Environment Management Cycle with RACI Matrix.
		• Updated 3.5.3 Performance Testing - Entry Criteria.
		• Updated 4.1.4 Performance Testing:
		  ○ The reminder of retaining the assessment on whether performance testing is required.
		  ○ To outline the performance testing type applicability.
		• Review frequency changed to annual basis to align with SDLC Standard.