



Certified IT Consultants
Solutions that talk your language

TEST STRATEGY

CARGAS “NGVC”

22 September 2022



Certified IT Consultants



SAP® Certified
Partner Center of Expertise

We deliver results through proven experience and effective solutions to ensure that your IT projects contribute significant value to your organization

Confidential, Property of CIC

TEST STRATEGY

PROJECT IDENTIFICATION

Project Name	CARGAS “NGVC”
Project Number	IM-22-79
Customer Name	CARGAS “NGVC”
Project Type	Implementation
CIC Project Manager	Mr. Ahmed Hamdy
CIC Project Sponsor	Mr. Ibrahim Ahmed
CARGAS Project Manager	Mr. Abdelmoneim El-Farra
CARGAS Project Sponsor	Mr. Hasanain Mohamed

DOCUMENT IDENTIFICATION

Author	Mr. Ahmed Hamdy
Document Location (repository/path/name)	
Version	1
Status	Final
Date	Thursday, September 22, 2022
Document Classification	confidential

REVISION HISTORY

Version	1
Date	Thursday, September 22, 2022
Description	Document Creation

CONTENTS

1. INTRODUCTION	4
1.1. Purpose of Test Plan.....	4
1.2. Definition of Testing	4
1.3. Objectives	4
1.4. Testing Scope	4
1.5. Realize Phase.....	5
1.6. Assumptions	5
2. UNIT TESTING	6
2.1. Description.....	6
2.2. Test Management	7
2.3. Test Documentation	7
2.4. Test Data.....	7
2.5. Test System	7
2.6. Roles & Responsibilities	7
2.7. Entrance Criteria.....	8
2.8. Exit Criteria	8
3. BUSINESS PROCESS (STRING) TESTING	9
3.1. Description.....	9
3.2. Test Management	9
3.3. Test Documentation	10
3.4. Test Data.....	10
3.5. Test System	10
3.6. Roles & Responsibilities	10
3.7. Entrance Criteria.....	11
3.8. Exit Criteria	11
4. SCENARIO (INTEGRATION) TESTING	12
4.1. Description.....	12
4.2. Test Management	13
4.3. Test Documentation	14
4.4. Test Data.....	14
4.5. Test System	14
4.6. Roles & Responsibilities	15
4.7. Entrance Criteria.....	16
4.8. Exit Criteria	16
4.9. Approval of Test Results	17
5. USER ACCEPTANCE TESTING.....	17
5.1. Description.....	17
5.2. Test Management	18
5.3. Test Documentation	18

5.4.	Test Data.....	18
5.5.	Test System	18
5.6.	Roles & Responsibilities	18
5.7.	Entrance Criteria.....	19
5.8.	Exit Criteria	19
6.	TECHNICAL SYSTEM TESTING.....	ERROR! BOOKMARK NOT DEFINED.
6.1.	Description.....	Error! Bookmark not defined.
6.2.	Technical Functionality Tests	Error! Bookmark not defined.
6.3.	Test System	Error! Bookmark not defined.
6.4.	Roles & Responsibilities	Error! Bookmark not defined.
6.5.	Entrance Criteria.....	Error! Bookmark not defined.
6.6.	Exit Criteria	Error! Bookmark not defined.
7.	DEFECT MANAGEMENT	20

1. Introduction

1.1. Purpose of Test Plan

The purpose of functional test plan is to describe the test plan to CARGAS

Functional test plan will describe the types of testing and related activities that will occur during the Realize Phase of the project.

- Unit Testing
- Business Process (String) Testing
- Scenario (Integration) Testing
- User Acceptance Testing

1.2. Definition of Testing

Testing is an activity aimed at evaluating an attribute or capability of a program or system in determining that it meets the business requirements as defined in the business blueprint document.

1.3. Objectives

The objectives of testing are:

- Ensure that the system meets all the business requirements determined to be in scope
- Ensure that the system meets technical requirements and meets service levels for application response time, throughput, and infrastructure performance at typical production loads

1.4. Testing Scope

The scope of testing includes business and technical requirements defined in the approved Business Blueprint document. The Business Blueprint requirements include the following elements:

- Enterprise Organizational
- Master Data Scope
- SAP Solution – Configuration
 - Business Scenario
 - Business Processes
 - Process Steps

- Transactions
- Testing of SAP Solution – Configuration
- Custom Development Objects (RICEFW):
 - Reports
 - Interfaces
 - Conversions
 - Enhancements
 - Forms
- Security Roles & Profiles

1.5. Realize Phase

The project will conduct testing in multiple iterations that starts with unit testing and continues with integration testing of business scenarios, data, and security, and ends with user acceptance testing. Testing iterations should be completed sequentially, allowing for testing of new configuration and development, as well as those tests completed during previous cycles.

Testing is designed to validate business requirements to provide traceability to the Business Blueprint requirements to build the SAP solution. The following illustration represents the dependencies between the types of testing that will be executed during the Realize Phase of the project.

As part of testing, it is important to simulate daily, weekly, and monthly business events and activities (e.g. daily batch processes, generation of key reports, and execution of financial monthly close) during business process (string) testing and scenario (integration) testing in the quality assurance environment

1.6. Assumptions

The following assumptions to be made during the development of functional test plan.

- Business requirements acceptance and approval is completed before Realize Phase begins
- Unit Testing is complete before Integration Testing commences
- Project Management Plans are in place before Integration Testing commences – governs changes to the system, design decisions, documentation, etc. Specifically:
 - Scope Management Plan
 - Integration Change Control Process
 - Issue Management Process
 - Risk Management Plan

2. Unit Testing

2.1. Description

Unit Testing validates that individual functions are configured and/or developed to appropriately translate technical and functional requirements. This would include testing of individual configuration transactions, development objects, and manual business process steps associated with business transactions.

- Testing of configurable transactions
 - Test ERP transactions and related business process steps to achieve a defined business outcome with a module.
 - Test manual transactions or steps performed as part of the configurable transaction
- Testing of development objects. These tests will further include:
 - Testing of the code within the development object. These tests will be based on the Technical Specifications documents. This activity will be owned and conducted by the Technical Team.
 - Testing the functional aspects of the development object. These tests will be based on the Functional Specifications documents & Business Process Procedures and will be conducted by the Functional Teams with the assistance of the Technical Team.

. Unit testing is the lowest level of testing where the business transaction or development object is tested and evaluated for errors.

Unit testing is the first test that is completed during configuration and is focused towards the program's inner functions, rather than towards the integration. Both positive and negative testing should be performed for all critical functionality.

- Positive Testing – validates that test functions correctly by inputting a known value that is correct and verifies that the data/view return is what is expected
- Negative Testing – validates that the test fails by inputting a value that we know is incorrect and verify that the component or test case fails. This allows us to understand and identify failures and that the target application is operating correctly by displaying the appropriate warning message.

Unit testing begins during baseline configuration (test cycle/iteration 1) the project team will work on configuring transactions and related business process steps that can be configured in the SAP Solution without CARGAS development objects (e.g., programming or enhancements).

Test cases developed during the Realize Phase should be used to unit test configurable transactions and related business process steps of an SAP module.

As development objects are completed, unit testing will be performed to ensure functional specifications are met and integration of custom development objects with configurable transactions and process steps operate successfully.

2.2. Test Management

Because baseline configuration unit testing concentrates on testing single business transactions; there is normally no need to develop detailed test documentation. For simple transactions, testing will be done straightforward during configuration.

However, some configuration transactions, development objects, and business process steps associated with business transactions are very complex involving multiple screens, functions and variations to run. These types of complex transactions (i.e. Sales order) will be documented and tested using a test case. Additionally, test data and test case information documented within the functional specification of CARGAS development objects will be used to review code of technical objects and test functional aspects of the development object.

2.3. Test Documentation

A test case will be used to execute testing process:

- Develop test case using test case template
 - Test case contains the detail steps, step-by-step, and criteria for completing a test (functional and performance) to support testing.
- After the creation of the test case, the actual testing is performed by a project team member
- The results of each test case are recorded manually
- Test defects are tracked and monitored manually.

2.4. Test Data

Fabricated or CARGAS specific master data will be manually entered as required for unit testing and used by other teams where appropriate.

2.5. Test System

Unit testing will be performed in the project development environment (DEV) with manually entered test data to execute test cases.

2.6. Roles & Responsibilities

The **CIC consultant** will conduct the unit testing of configuration and execute test cases per the unit test plan.

The developer along with the consultant will conduct the unit testing of development objects (if available).

- **CIC Functional Consultant** will resolve configuration issues
- **CIC Technical Consultant** will resolve any issues with development objects
- **Security & Administration Team Member** will resolve Security-related issues
- **Team Leads** will ensure the creation, completeness, quality of unit test cases and testing, managing test resources
- **Project Manager** is responsible for the overall planning, tracking and reporting of Unit Testing status and results.

2.7. Entrance Criteria

Business Requirements are finalized and frozen

- Business Blueprint Requirements are documented and approved.
- Business requirements document and custom development object listing is finalized and frozen

Development Test Environment is built, accessible, and ready for unit test

- Project team security profiles and roles are evoked.
- Configuration documentation requirements and transport procedure is defined and communicated to the project team

2.8. Exit Criteria

Unit test cases executed and defects resolved.

- Unit test cases have been executed.
- Unit test cases and test results have been reviewed and approved by functional leads and/or technical leads.
- All significant defects (business-critical or high integration impact) have been resolved and retested.

Documentation updated for changes made during Unit Testing.

- Any documentation updates because of changes made during the unit testing have been completed and approved.

3. Business Process (String) Testing

3.1. Description

Business Process (String) Testing validates the full operability of interconnected functions, methods or objects within the functional areas of an SAP Solution (e.g., Sales).

- Includes a set of logically related activities or business process steps to achieve defined business processes.
- Includes cross functional business processes (e.g., Sales and Finance).

Business process (string) testing is not meant to be a full-blown integration test but rather a more comprehensive test within a module or some limited testing between some modules as needed (e.g. create an order, deliver it, bill it and apply cash application).

Test cases are updated as part of baseline configuration and used to perform business process (string) testing of custom development objects, configurable transactions and related business process steps for the SAP Solution that have been unit tested.

Business process procedures (BPP) are developed in the QA environment during business process (string) string testing and continue throughout scenario testing.

BPP's will be utilized during end user acceptance testing and used to assist with end user training.

3.2. Test Management

The project teams will use manual testing techniques to perform business process (string) testing.

Unit testing will continue to be performed for new and/or changed configuration and development objects using test cases.

3.3. Test Documentation

A test case will be used to execute the testing process:

- Develop test case using test case template
 - Test case contains the detail steps, step-by-step, and criteria for completing a test (functional and performance) to support manual testing.
- After the creation of the test case, the actual testing is performed by a project team member
- The results of each test case are recorded.
- Test defects are tracked and monitored manually.

3.4. Test Data

Most test data will be created manually for simple and low-volume data conversion activities to load test data to execute test cases and test scripts, which will be used by other teams where appropriate.

Manual Entry - Appropriate for low record count and ability to easily manage data entry accuracy by qualified project team members.

3.5. Test System

Business process (string) testing is performed in the project quality assurance (QA) environment.

NOTE: If a QA environment is not available, consider partitioning the DEV environment with a QA “like” environment for test automation.

3.6. Roles & Responsibilities

The CIC consultant will conduct the unit testing of new configuration and execute test cases and test scripts per the business process (string) test plan.

The developer will conduct the unit testing of new development objects (if available) and integrate previously unit tested development objects into business process (string) testing.

- CIC Functional Consultant will resolve configuration issues

- CIC Technical Consultant will resolve any issues with development objects
- Security & Administration Team Member will resolve Security-related issues
- Team Leads will ensure the creation, completeness, quality of unit test cases, test scripts, and testing, managing test resources
- Project Manager is responsible for the overall planning, tracking and reporting of testing cycle status and results.

3.7. Entrance Criteria

Unit testing for baseline is completed

- Documentation updates because of the changes made during the unit testing have been completed

Quality assurance (QA) environment is built, accessible, and ready for unit test

- Project team security profiles and roles are in evoked.

Dedicated test lab or area is established to perform business process (string) testing on standard CARGAS issued desktops and laptops

- Utilize security profile and roles for business process (string) testing
- Schedule daily testing status meetings with testers to review daily test plan, dependencies, successor tests, and outcome.

3.8. Exit Criteria

Business process (string) testing test cases and test scripts have been executed and defects resolved.

- All test cases have been documented and test scripts have been recorded as required.
- All test cases and test scripts have been executed.
- Test case, test script, and test results have been reviewed by functional leads and/or technical leads.
- All significant defects (business-critical or high integration impact) have been resolved and retested.

Business process (string) test results reported

- BI-Weekly test results report – prepared and reviewed,

Documentation updated for changes made during business process (string) testing.

- Any documentation updates because of the changes made during the testing have been completed

4. Scenario (Integration) Testing

4.1. Description

Scenario (Integration) Testing – validates a set of business processes that define a business scenario in a comprehensive and self-contained manner on a macro level.

- Integration testing is recommended to be done in multiple iterations.
- The initial iteration of integration testing concentrates on testing all important business processes within the SAP components of the implemented solution, starting with touch point scenarios and ending with end-to-end scenarios.
- The final iteration of integration testing focuses on cross-functional business scenarios with non-SAP systems and applications, custom development objects, converted data, and solution security.

Scenario (Integration) testing will be designed at the business scenario level. Since business scenarios are collections of business process and process steps, Integration Tests will include a collection of unit tests. Integration tests may also include other transactions including manual transactions, custom transactions, security steps, etc. as defined in the Business Blueprint document.

Business process procedures (BPP) are developed in the QA environment during business process (string) string testing and continue throughout scenario (integration) testing. BPP's will be utilized during end user acceptance testing and used to assist with end user training.

Scenario (Integration) testing is recommended to be done in multiple iterations.

- The first iteration of scenario (integration) testing concentrates on testing all important business processes inside the SAP system, starting with business processes and ending with end-to-end-scenarios across functional teams. Test cases and test scripts should be updated to incorporate business processes and end-to-end business scenarios, which include security profiles and roles, manual data entry and data conversion testing.
- The n iteration of scenario (integration) testing will cover the full range of business scenarios with some variation and include the business critical, custom-developed objects, and reports. Security roles & profiles will be used in each iteration. However, security-related defects will not stop progress of testing a business scenario. Data conversions will be in scope to the extent that conversion programs are available and ready. However, only a portion of the full data will be converted.

- Final iteration of scenario (integration) testing is an evolutionary process that will be driven from previous testing efforts. The final iteration of integration testing focuses on the cross-functional integration points, end-to-end business processes, and critical cross-enterprise scenarios with touch points to external components and legacy applications, including testing of all custom development objects, security profiles and roles, regression testing of changes to existing production systems, and data conversions. The final iteration of integration testing is accomplished through the execution of predefined support for business flows, or scenarios, that emulate how the system will run your business. These business flows, using migrated data from the preexisting systems, are performed in a multifaceted computing environment comprising SAP software, third-party software, system interfaces, and various hardware and software components. Security roles and profiles will be used for all testing activities. Testing of a business scenario will be stopped for any defects (security or otherwise) encountered. All data conversions will be in scope and an effort will be made to load all the data that is ultimately to be loaded into the Production environment.

During scenario (integration) testing the project team will perform mock builds to develop and practice cutover activities during scenario (integration) testing iterations using the QA environment to prepare and simulate production build of the SAP Solution. The QA environment will be refreshed between the iterations of scenario (integration) testing, Testing Business Activities and Practicing Cutover Build in the QA Environment.

Security profiles and roles will be used during scenario (integration) testing iterations. This is required for the following reasons.

- Integration Testing seeks to test end-to-end business processes and attempts to simulate 'real-life' business events. Consequently, it is important to use security profiles to ensure that the hand-offs between departments that are inherent in business processes, occur seamlessly.
- Integration Testing also uses "production" data that is converted into the test environment. It is important to restrict access to sensitive data (i.e., salary information, personal information, bank information, etc.)

4.2. Test Management

The project teams will use manual testing techniques to perform scenario (integration) testing.

Test cases will be used to perform manual tests.

Scenario (integration) testing status reporting will be included in the BI-Weekly project status report.

4.3. Test Documentation

A test case will be used to execute manual testing. Manual testing process using a test case:

- Develop test case using test case template
 - Test case contains the detail steps, step-by-step, and criteria for completing a test (functional and performance) to support manual testing.
- After the creation of the test case, the actual testing is performed manually by a project team member
- The results of each test case are recorded manually
- Test defects are tracked and monitored manually.

CARGAS functional lead and/or technical lead will review and approve all test cases.

4.4. Test Data

Test data will be derived from the following processes or tools:

Manual Entry - Appropriate for low record count and ability to easily manage data entry accuracy by qualified project team members. Source of data must be document to enable test case steps to compete

- data entry. This type of data migration is utilized by functional teams to perform unit testing, business process (string) testing, and integration testing cycles.
- Legacy System Migration Workbench (LSMW) provides a recording function that allows the generation of a "data migration object" in an entry or change transaction. Functional and technical specifications, mapping documents, and test scripts are required. This type of data conversion is utilized by functional teams to perform integration testing cycles/iterations.
- Custom development objects developed for data load programs (API) used for data transfer. Functional and technical specifications, mapping documents, and test scripts are required. This type of data conversion is utilized by functional teams to perform integration testing cycles/iterations.

4.5. Test System

The different environments that would be used through this testing cycle/iteration are described below:

- QA Environment – As mentioned earlier, the scenario (integration) testing iterations will be conducted in a controlled QA environment. To ensure that testing is valid, an Integrated Change Control

Process will be used to govern all changes to the system during scenario (integration) testing iterations. Typically, no changes to the system will be implemented directly in the QA environment. Any changes required (to fix defects or incorporate approved requirements) will be sourced in the DEV environment and migrated to the QA environment using Solution Manager after successful unit testing. Also, no new functionality will be introduced into the QA environment (typically) in the middle of scenario (integration) test iteration. Any exceptions should be approved by the Project Management.

- DEV Environment – Any changes required to be made to the system during the scenario (integration) testing iteration (to fix defects or approved requirements) will be first made in the DEV environment and unit tested before it is migrated to the QA environment.

The basis/infrastructure team will be responsible for the creation and maintenance of these environments as well as enforcing necessary controls on the QA environment.

4.6. Roles & Responsibilities

- **CARGAS Project Team Members** identify, develop, and update initial test cases and test scripts to perform tests per the scenario (integration) test plan.
- **Subject Matter Experts** perform all necessary scenario (integration) testing per the integration test plan using business process procedures to execute test cases and/or test scripts.
- **CIC Functional Consultant** will resolve configuration issues
- **CIC Technical Consultant** will resolve any issues with development objects
- **Business Users or Subject Matter Experts** (SMEs) confirm business process and UI functionality (if applicable)
- **Legacy System SME's** support data conversion and migration activities. Validates and confirms data loads and verifies data accuracy.
- **Basis/infrastructure team** will be responsible for the creation and maintenance of the project environments as well as enforcing necessary controls on the QA environment.
- **Security & Administration Team Member** will resolve Security-related issues

- **Team Leads** will ensure the creation, completeness, quality of test cases/scripts and testing, managing test resources, and sign off on test results.
- **Project Manager** is responsible for the overall planning, tracking and reporting of testing status and results.

4.7. Entrance Criteria

Unit testing for final configuration is completed

- Documentation updates because of changes made during the testing have been completed and approved
- Integrated change control process and procedure is established and in place
- Configuration is frozen

Data conversion and migration loads are available and ready for testing

Quality assurance (QA) environment has been refreshed, accessible, and ready for scenario integration testing

- Project team security profiles and roles are in evoked.

Dedicated test lab or area is established to perform testing on standard CARGAS issued desktops and laptops

- Utilize security profile and roles for testing
- Schedule daily testing status meetings with testers to review daily test plan, dependencies, successor tests, and outcome.

4.8. Exit Criteria

Scenario (Integration) Testing iteration test cases and/or test scripts have been executed and defects resolved.

- All Scenario (Integration) Testing iteration test cases and/or test scripts have been documented as required.
- Scenario (Integration) Testing iteration test cases and/or test scripts have been executed.
- Scenario (Integration) Testing iteration test cases and/or test scripts results have been reviewed and approved by functional leads and/or technical leads.
- All significant defects (business-critical or high integration impact) have been resolved and retested.

Data conversions successfully converted and completed.

Scenario (Integration) Testing iteration results reported and approved

- BI-Weekly Scenario (Integration) Testing iteration results report - prepared, reviewed, and approved

System meets the business application requirements as defined in the Business Blueprint Document

4.9. Approval of Test Results

Scenario (integration) testing test cases and/or test scripts may not be completely executed with a “Passed” status within the timeframe available. Consequently, scenario (integration) testing would be considered complete under the following circumstances:

- All scenario (integration) testing test cases and/or test scripts have been executed at least once.
- All defects that have been categorized as “Show-stoppers” (Severity 1 and 2) have been resolved and retested successfully.

In addition to documentation of test results for each scenario (integration) testing test cases and/or test scripts, an overall Integration test results report will be produced at the end of each test iteration. This report will include the following sections.

- A list of all the test scripts used for this stage.
- A list of all the defects reported during this stage with related status.
- A report of the compliance with the Entrance Gates and Exit Gates agreed upon for the scenario (integration) testing iteration including possible exceptions.
- Any unplanned events / decisions that occurred during this testing iteration that would constitute a deviation from the approved testing strategy and approach or this document.

5. User Acceptance Testing

5.1. Description

User Acceptance Testing (UAT) - users test the complete, end-to-end business processes to verifying that the implemented solution performs the intended functions and satisfies the business requirements.

UAT is the last test cycle of a SAP solution implementation and is essential part of gaining end user acceptance of the system. This cycle occurs at the end of Realize Phase of the implementation, subsequent to scenarios testing cycles to ensure that the system has been tested thoroughly by the project team and is ready to be released to the end user community. The emphasis of UAT to demonstrate that the system functions as designed.

5.2. Test Management

The project team members will assist key users in performing manual testing techniques for UAT.

UAT status reporting will be included in the project status report.

5.3. Test Documentation

Manual test cases executed throughout scenario testing will be utilized for UAT.

5.4. Test Data

Sample production data will be used for UAT.

5.5. Test System

UAT will be performed in the QA environment

5.6. Roles & Responsibilities

- **Key users** perform all necessary UAT per the plan using business process procedures to execute test cases and/or test scripts and sign off on test results.
- **CIC Functional Consultant** will resolve configuration issues
- **CIC Technical Consultants** will resolve any issues with development objects
- **Basis/infrastructure team** will be responsible for the creation and maintenance of the project environments as well as enforcing necessary controls on the QA environment.

- **Security & Administration Team Member** will resolve Security-related issues
- **Team Leads** will ensure the creation, completeness, quality of test cases/scripts and testing,
- **Project Manager** is responsible for the overall planning, tracking and reporting of testing status and results.

5.7. Entrance Criteria

Scenario testing is completed

- Documentation updates because of changes made during the testing have been completed and approved
 - Configuration is frozen
 - All configuration issues have been closed, and documented
 - Business Process Procedures completed
 - Configuration Documentation completed
 - Integrated change control process and procedure is established and in place
- Quality assurance (QA) environment has been refreshed, accessible, and ready for UAT
- Project team security profiles and roles are in evoked.

Dedicated test lab or area is established to perform testing on standard CARGAS issued desktops and laptops

- Utilize security profile and roles for UAT
- Schedule daily testing status meetings with testers to review daily test plan, dependencies, successor tests, and outcome.

5.8. Exit Criteria

UAT test cases and/or test scripts have been executed and defects resolved.

- All UAT test cases and/or test scripts have been documented as required.
- UAT test cases and/or test scripts have been executed.
- UAT test cases and/or test scripts results have been reviewed and approved
- All significant defects (business-critical or high integration impact) have been resolved and retested.

UAT results reported and approved

- BI-Weekly UAT results report - prepared, reviewed, and approved

System meets the business application requirements as defined in the Business Blueprint Document

6. Defect Management

A defect is a test problem or error that must be corrected. Defects will be written for SAP Solutions, and manual steps that are part of the testing scope of the program or project. A defect is formally documented in a program or tool for managing test defects. Defects can become project issues if the solution does not meet the business or technical requirements of CARGAS.

The tester assigned to test case or test script is responsible for entering the defect into the Defect Repository by specifying the following:

- Defect ID
- Status
- Severity is the prioritization of the defect:
 - Severity 1 (1-Critical) – Serious errors that prevents or stops testing of a function or serious data type error (i.e., system locks-up).
 - Severity 2 (2-High) – Serious or incorrect functionality errors, incorrect data, or significant Load problems that may make the application unusable (i.e., login takes over 5 minutes; query takes 10 minutes, etc.).
 - Severity 3 (3-Medium) – Defects should not prevent or hinder the functionality or Load of the system (i.e., an incomplete phone number string is returned).
 - Severity 4 (4-Low) – Defects that do not prevent or hinder functionality of the system, which is normally confined to the User Interface (i.e., missed spelled word).

Note: When tester identifies a potential Severity One (1) or Two (2) defects, the tester will contact the test lead/project manager immediately to implement the appropriate corrective action

- Summary (brief description of defect)
- Project (Cycle of Testing)
- Project Area (i.e., Functional/Technical Team).
- Description (detailed description of the defect)
- R&D Comments (update the history associated with a defect)
- Attachments (i.e., pictures)
- Detected on Date
- Close Date
- Defect Type use an appropriate value to define the tracking status of a defect.
 - Code: Defect is within the code.
 - Data: Bad data or data errors
 - Enhancement: Used with a Change Control Procedure process to request change
 - Other: Use only once you receive approval from the Test Lead
 - Performance: Based on a Performance problem
 - Specification: Requirements are not correct
 - UI/Cosmetic: UI error (i.e., misspelled label)
 - Usability: Unable to perform a task
 - Defect Status - Use the appropriate value to define the tracking status of a defect.

- Closed: The identified test case with a defect passes all applicable regression tests
- Duplicate-Closed: The Test Lead sets this status if a test step/process is determined to be no longer valid.
- Fixed: Developer corrected and unit tested defect and it is ready for retest.
- Deferred: Further investigation is required to identify if this is a true defect
- Fixed Pending Build: Defect is fixed and the testing SRF is pending the update.
- New: Defect was entered into the system has not been validated by the Test Lead for assignment
- Open: Defect is not assigned to anyone to fix
- Re-open: Defect is found after it was closed

The next step is to pass the defect through the Developer Fix Process, where the developer or consultant who executed the configuration works to resolve defect:

The development/configuration team is responsible for updating the defect information in the Defect repository by specifying the following:

- Assigned To (developer's name)
- Status
- Assigned Date
- Closing Date
- R&D Comments (what action was taking to correct a defect)

The Integration Lead or Test Lead or project manager is responsible for periodically checking Defect Repository to validate when a defect is fixed and ready for retest or to escalate when a Severity 1 or 2 are impacting the test schedule. Once a defect is successfully tested (and relevant Regression testing has taken place), the Tester will update the status in Defect Repository.

Acceptance of the TEST STRATEGY

Customer Project Manager

Name	Mr. Abdelmoneim Elfarra
Date	
Signature	

CIC Project Manager

Name	Mr. Ahmed Hamdy
Date	
Signature	