Your Company Name

Regression Testing Plan

Date

www.SDLCforms.com

Revision History

Date	Version	Author	Change

www.SDLCforms.com

COPYRIGHT NOTICE
Confidential – ©2015 Documentation Consultants
All rights reserved. These materials are for internal use only. No part of these materials may be reproduced, published in any form or by any means, electronic or mechanical, including photocopy or any information storage or retrieval system, nor may the materials be disclosed to third parties without the written authorization of (Your Company Name).

Table of Contents

www.SDLCforms.com

Note: Text displayed in blue italics is included to provide guidance to the author and should be deleted before publishing the document. In any table, select and delete any blue line text; then click Home→Styles and select "Table Text" to restore the cells to the default value.

1 Introduction

This section provides general information about the systems or applications that require regression testing, e.g.:

- ***Why regression testing is required, e.g., bug fixes, specific enhancements, major or minor upgrade.***
- ***Systems or Applications affected.***
- ***Functional Business Areas affected.***
- ***Testing timeline.***

1.1 Reference Documents

The Regression Plan contains information based on the following documents.

Document	Date or Version Number
Requirements Document	
Design Document	
Specifications Document	
Previous Test Scenarios	
Previous Error Logs or Open change requests	

www.SDLCforms.com

2 What is Regression Testing

2.1 Purpose

The purpose of Regression Testing is to seek to uncover new errors, or regressions, in existing functionality after changes have been made to a system, such as functional enhancements, patches, or configuration changes.

Definition: Regression testing is selective retesting of the system; executed with an objective to ensure the enhancement/bug fixes work and those enhancement/bug fixes have not caused any un-intended effects in the system. The intent of regression testing is to ensure that a change did not introduce new faults.

Common methods of regression testing include rerunning previously run tests and checking whether program behavior has changed and whether previously fixed faults have re-emerged. Regression testing can be used to test a system efficiently by systematically selecting the appropriate minimum set of tests needed to adequately cover a particular change.

2.2 Types of Regression Testing

There are 3 main types of regression testing:

1 Incremental Testing.
   - Test only the code that was added or modified against anticipated results.
   - Repeatedly test in a scheduled set of increments, e.g., after the nightly build.

2 Incremental Regression Testing.
   - Selective retesting of portions of the software that has changed, or is impacted by a change, against a known baseline of results to verify that there are no unintended side effects.

3 Full Regression Testing.
   - Complete retesting of the entire software application against a known baseline of results to verify that there are no unintended side effects anywhere in the application.

www.SDLCforms.com

2.2.1 Functional and Unit Testing

Regression Testing:

- Has traditionally been performed by a software quality assurance team after the development team has completed work. However, defects found at this stage are the most costly to fix. Regression testing can be used not only for testing the correctness of a program, but often also for tracking the quality of its output.
- Can be broadly categorized as functional tests or unit tests.
  - Functional tests exercise the complete program with various inputs. Functional tests may be a scripted series of program inputs, possibly even an automated mechanism for controlling mouse movements.
  - Unit tests exercise individual functions, subroutines, or object methods. Unit tests may be separate functions within the code itself, or driver layer that links to the code without altering the code being tested.

2.3 Regression Test Approach

This section provides information about the regression testing environment that will be used, which should replicate the production environment. Regression testing includes test scripts that are based on the test scenarios used from requirements, design, and specifications documentation.

2.3.1 Guidelines for Patch/Upgrade Releases

The regression guidelines are applicable for both patch and upgrade releases where:

- You are doing a major release of a product, executed all system test cycles and planning a regression test cycle for bug fixes.
- You are doing a minor release of a product having only bug fixes, and you are planning for regression test cycles to take care of those bug fixes.

There can be multiple cycles of regression testing that can be planned for each release. Bug fixes may come in phases or some bug fixes may not work as expected resulting in one more regression cycle.

2.4 Scope of Regression Testing

Summarize and list the functions, features, and applications to be tested. The items listed here should directly relate to the main features / major business functions discussed in the requirements and design (if applicable) documentation.

www.SDLCforms.com

2.5 Test Categories

List the tests that will be performed as part of regression testing.

Note: This table is an example, add categories not listed and remove categories that will not be tested.

Category	Description
Functionality	*Functions as described in documentation.*
Security and Access Control	*Provides the proper application and user level security.* *Application-level security, including access to the Data or Business Functions* *System-level Security, including logging into or remote access to the system.*
Data, Database, and Data Integration	*Ensure data accessed, used, and applied is valid and correct. Test databases and the database processes as a subsystem.*
Boundaries	*Fields perform in accordance with the constraints placed on the fields.*
Audit Trail	*Can track user and system activity (adds, changes, and deletes).*
Error Conditions	*Provides specific confirmation and error messages.*
Performance	*Provides or meets required performance guidelines as described in documentation.*
External Interfaces	*Functions timely and correctly with other external systems.*
User Interface	*User interface functions as described in documentation.*
Reporting	*Prints or displays report data as described in documentation.*

www.SDLCforms.com

2.6 Risks, Dependencies, Assumptions and Constraints

Describe any risks, dependencies, assumptions, and constraints that would affect regression testing. Provide any work-around solutions that may apply.

3 Functional Testing

Specify major activities, techniques, and tools to be used to test the functions, features, and applications. Provide information about the major testing tasks and approximate time to run each one.

The following table provides information about functionality that is included in the tests.

Functionality	Description
Included	***Provide a high level outline of the major testing functions planned for the regression testing.***
Excluded	***Provide a high level outline of the tests that have been specifically excluded from regression testing.***

4 Test Environment

The following table provides information about the test environment.

Environment	Description
Hardware	***Provide a description of the hardware that will be used in regression testing.***
Software	***Provide a description of software and applications that will be used in user regression testing.***
Tools	***Provide a description of the testing tools (if any) that will be used in regression testing.***

5 Concluding the Results of Regression Testing

Regression testing generally uses only one build for testing. It is expected that all 100% of those test cases pass using the same build. In situations where the pass % is not 100, look at the previous results of the test case to conclude the expected result:

- If the result of a particular test case was PASS using the previous builds and FAIL in the current build, then regression failed. We need to get a new build and start the testing from scratch after resetting the test cases.

www.SDLCforms.com

- If the result of a particular test case was a FAIL using the previous builds and a PASS in the current build, then it is easy to assume the enhancement/bug fixes worked.
- If the result of a particular test case was a FAIL using the previous builds and a FAIL in the current build and if there are no bug fixes for this particular test case, it may mean that the result of this test case shouldn't be considered for the pass %. This may also mean that such test cases shouldn't be selected for regression.
- If the result of a particular test case is FAIL using the previous builds but works with a documented workaround and
  - If you are satisfied with the workaround then it should be considered as a PASS for both system test cycle and regression test cycle.
  - If you are not satisfied with the workaround then it should be considered as a FAIL for a system test cycle but can be considered as a PASS for the regression test cycle.

6 Test Plan Schedule

Complete the table with the appropriate test schedule information.

Task Description	# of Days	Start Date	End Date

www.SDLCforms.com

7 Testing Matrix

Provide a separate section for each series of testing (from test plan schedule) that is performed for the functions, features, or applications that include the following information:

- ***Assumptions, Pre-conditions, and Risks***
- ***Test instructions with Entrance and Exit Criteria***
- ***Defect Metrics.***

7.1 Test Instructions

Entrance Criteria:

Exit Criteria:

Step	Test Instructions	Expected Result	Pass / Fail	Comments
1.				
2.				
3.				

7.2 Test Completion Summary

Test #	Tester	Date	Description	Pass / Fail
1.				
2.				
3.				
All Tests				

7.3 Associated Defects

Total number of defects opened during testing.	
Number of defects fixed during testing.	
Number of defects to be fixed after system implementation.	
Number of other defects that will not be fixed or will be dropped.	

Note:

Management can track incidents and defects using information in the following tables:

www.SDLCforms.com

Priority

Priority	Definitions
Critical	*Required function is not working and there is no workaround. Unable to continue testing with the current functionality.*
High	*Required function is not working, but there is a workaround. Testing can continue in other areas.*
Medium	*Functionality achieves the intent, but not to the letter of the requirement and/or design.*
Low	*Functionality achieves the intent but includes inconveniences or annoyances.*

Status

Status	Definition
New	*Incident has just been entered.*
Open	*Incident is being reviewed.*
In-Progress	*Incident has been assigned for correction.*
Migrated	*Incident has been coded for correction and is ready for re-test.*
Deferred	*Incident will be addressed in another software version.*
Cancelled	*Incident has been found to be a non-issue.*
Duplicate	*Incident is a duplicate of another incident.*
Closed	*Incident has been re-tested and correction has been confirmed.*

www.SDLCforms.com

8 Appendix

www.SDLCforms.com