

Company Logo	PROCEDURE	SOP-00XXXX
SDLC Process	Software (QA) Testing	Page: 1/12

QRS Software Development Lifecycle

Standard Operating Procedure
Software Quality Assurance Testing

SOP-00XXXX
Revision A
6/19/2016

QRS.
Confidential and Proprietary, All rights reserved

IMPORTANT NOTE: This document contains confidential information...

© 2016 QRS. All rights reserved.

The trademarks identified herein are the trademarks or registered trademarks of QRS or other third party.

Company Logo	PROCEDURE	SOP-00XXXX
SDLC Process	Software (QA) Testing	Page: 2/12

Documentation History

Date	Rev	Author	Comments
6/13/16	A	R G Coles	Initial Release

Approvals

Authors:	Approved by:
Name(s):	Name: Date:
	Name: Date:

Company Logo	PROCEDURE	SOP-00XXXX
SDLC Process	Software (QA) Testing	Page: 3/12

Table of Contents

DOCUMENTATION HISTORY	2
APPROVALS	2
GLOSSARY	4
1. PROCESS SUMMARY	5
1.1 Test Approaches.....	5
1.2 Best Practices.....	5
1.3 Parent Process	5
1.4 Test Activity Procedures	5
1.4.1 QA Testing Role	5
1.4.2 Test Stages	6
1.5 Test Case Designs	7
1.5.1 Design Procedures	7
1.5.2 Test Data	7
2. INPUT.....	8
2.1.1 Test Plans.....	8
2.1.2 Test Cases	8
3. TEST PROCESS	8
3.1.1 Scrum Model	8
3.2 Unit Testing	9
3.3 Functional Testing	9
3.4 Integration Testing.....	9
3.5 System Testing.....	9
4. ANALYZE	10
4.1.1 Test Completion Criteria.....	10
4.1.2 Defects Classification.....	10
4.1.3 Testing Measurements.....	10
4.1.4 Defect Measurements	10
5. OUTPUTS	10
5.1 Unit Testing	10
5.2 Functional Testing	10
5.3 Integration Testing.....	10
5.4 System Testing.....	11
6. USER ACCEPTANCE TESTING.....	11
6.1 Criteria.....	11
6.2 Output	11
7. REFERENCES	12
8. APPENDIX A: DEFECTS CLASSIFICATION.....	12

Company Logo	PROCEDURE	SOP-00XXXX
SDLC Process	Software (QA) Testing	Page: 5/12

1. PROCESS SUMMARY

Link to QRS SDLC Flow chart/process diagram – <http://>

This document provides key testing activities that are tracked and managed during SDLC. They are:

- QA Standard Operating Procedures
- Set of procedures that includes
 - QA processes
 - Roles
 - Responsibilities
- Agile/QA best practices

1.1 Test Approaches

The primary goal of testing is as follows:

- Evaluate the quality of a system throughout the software development process
- Verify systems function according to requirement and design specifications
- Verify system meets business and user needs

1.2 Best Practices

The following are recommended software testing best practices:

- Involve testing team early in a project to review requirements and scope
- Create test case scripts as use cases
 - For automated tests, record scripts as early as possible
 - Don't wait until coding is completed before starting Test Case Scripts

This document applies to the software testing effort that verifies:

- New software development
- Custom Upgrades
- Sustain Support
- Maintenance Support

1.3 Parent Process

QRS SDLC

1.4 Test Activity Procedures

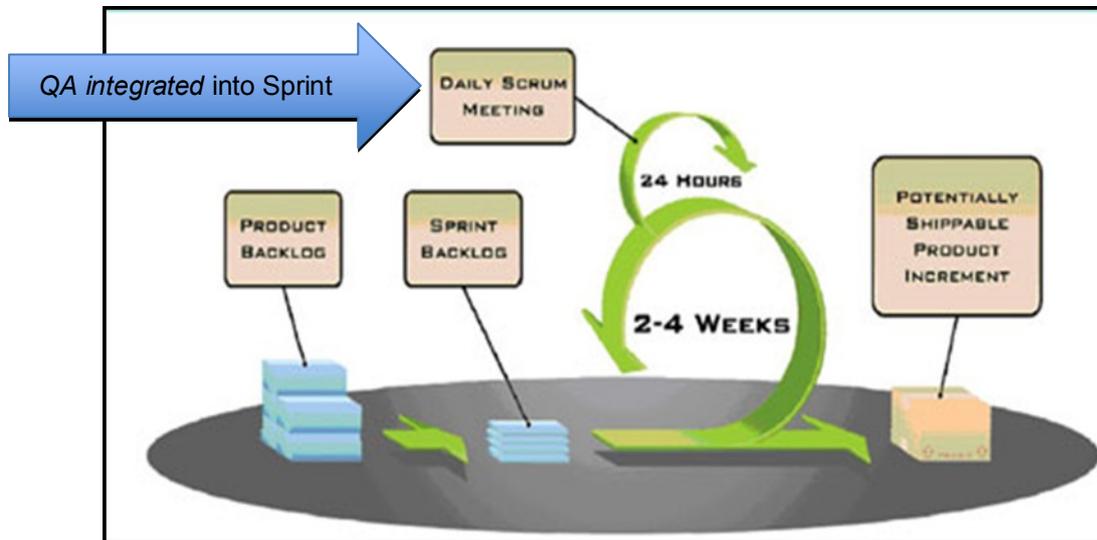
Two of the most prominent methods used today in software development are Agile and Waterfall. At QRSTrust, there are five levels of tests using the Agile System. Customer delivery projects will be discussed in a separate section at the end of this document.

Before testing of any kind can begin, it is necessary to explain the roles of QA.

1.4.1 QA Testing Role

- QA integrated into every team – a member of the QA team is now an active participant in sprint planning and a sprint
- QA engineers determine how best to implement that test (manual or automated, which tools, etc.)

Company Logo	PROCEDURE	SOP-00XXXX
SDLC Process	Software (QA) Testing	Page: 6/12



Now that the role of QA has been defined, we'll discuss the five test stages that are integrated into the Agile Scrum System. These stages are: Unit, Functional, Integration, Performance (includes Stress, Volume and Resource), and System).

1.4.2 Test Stages

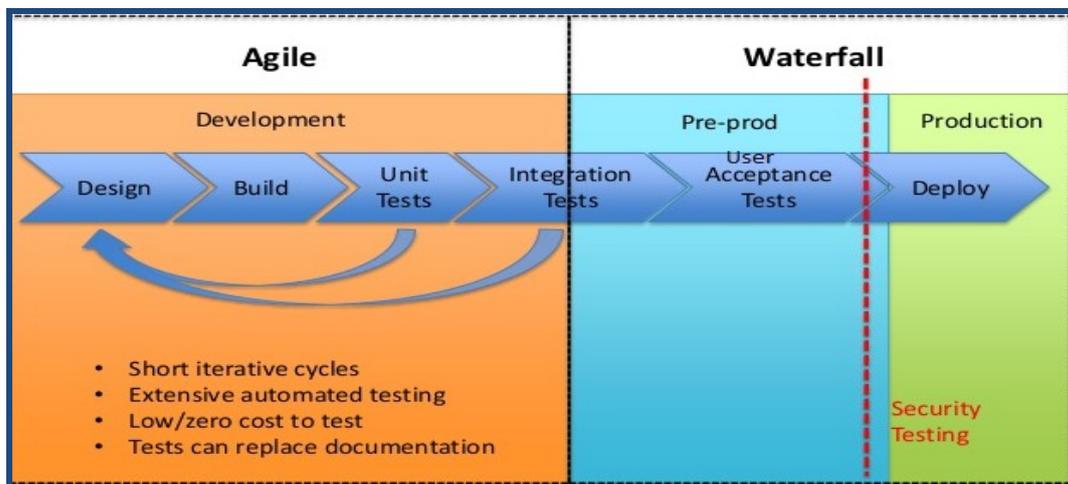
Agile testing is a continuous process, and has many parts. Below is a short list of reasons for these test stages.

- Test activities must be integrated in the development process. This means that testing is not a separate phase, rather a continuous activity of an agile team.
- All team members must be prepared to perform test activities. Although the team should contain professional QA Engineers, this does not mean that all test activities can be carried out by QA Engineers alone.

	Stage of Test	Deliverable Verified	
QRS	Unit Testing	Code for a unit	Agile SDLC
	Functional Testing	<ul style="list-style-type: none"> ➤ Functionality of a component within a system ➤ Usability of the software ➤ Data validation and data storage 	
	Integration Testing	<ul style="list-style-type: none"> ➤ Code for related components or sub-systems ➤ Interactions among the components ➤ Conformance of interface requirements ➤ Security validation between the components ➤ Data integrity between the components 	
	Stress, Volume, Performance, and Resource Usage Testing	<ul style="list-style-type: none"> ➤ Validate data rates & volume ➤ Achieve performance that mimics realistic business usage, loads and timing ➤ Validate stability of System under "overload", extreme and abnormal conditions ➤ Verify resource consumption does not exceed required level, system not particularly 	

Company Logo	PROCEDURE	SOP-00XXXX
SDLC Process	Software (QA) Testing	Page: 7/12

		sensitive to certain input values ➤ Validate system can meet acceptable service levels	
	System Testing	Completed system as defined in requirements and design specifications	
Customer	User Acceptance Testing	Completed system as defined by business objectives	Waterfall



Another part of the QA process is to create test cases which includes test data or use cases to see how well the software performs during each step of the process.

1.5 Test Case Designs

Test cases should be derived from the use case and feature design. If the use cases do not address tests such as boundary, equivalence partitioning, etc., additional test cases should be created to test these conditions.

The following are Test Case Naming Conventions used at QRS:

- Test Case Parent – TC1, TC2

1.5.1 Design Procedures

All test scripts should be written as detailed as possible. The test lead should work with both the Project Manager and Product Owner to determine the level of detail in test scripts.

1.5.2 Test Data

In order to test a software application you need to enter some data for testing most of the features. The Test Manager or Analyst should work with the appropriate person(s) to determine the test data needs. Test data should be included in the test cases or scripts.

Company Logo	PROCEDURE	SOP-00XXXX
SDLC Process	Software (QA) Testing	Page: 8/12

2. INPUT

2.1.1 Test Plans

A test plan is a document detailing the objectives, target market, internal beta team, and processes for a specific beta test for a software or hardware product. The plan typically contains a detailed understanding of the eventual workflow.

A test plan lets you specify what you want to test and how to run those tests. A test plan can be applied to a specific iteration of your project. You can have just one default test suite for your test cases, or you can create a test suite hierarchy.

2.1.2 Test Cases

A test case is a set of conditions under which a tester will determine whether an application, software system or one of its features is working as it was originally established for it to do.

3. TEST PROCESS

QRS uses two lifecycle processes: Agile and Waterfall. For the remaining document, we will focus on Agile only. This section talks about the five test stages listed above in section 1.5.2. User Acceptance Testing will be discussed in Section 6 as its own entity.

3.1.1 Scrum Model

The standard test operating procedures cover the following test cycles in the SDLC:

- Preparation and Sprint Planning
- Execution where Quality is built-in from the first sprint
 - Unit/Functional Testing
 - Integration Testing
 - System Testing
- Completion (Product Release) or User Acceptance Testing

Type of Testing	Who	When	Automation	Tool
Unit	Developer	Coding	Always	NUnit, JUnit
Integration	Tester	Sprint Build/Test	Always	SoapUI
System	Tester	Sprint Build/Test	Possible	Selenium, Jmeter
Acceptance	Client/Users	Deployment/Delivery	Possible	

Company Logo	PROCEDURE	SOP-00XXXX
SDLC Process	Software (QA) Testing	Page: 9/12

3.2 Unit Testing

Unit testing focuses on early examination of software such as units of code, modules, subroutines, and procedures. It evaluates functionality that may not be visible during integration, system, performance, and acceptance testing. Unit testing reduces defects by evaluating a system during development rather than at its completion. These procedures are:

- Keep unit tests small and fast.
 - The entire test suite should be executed before every code check in.
- Unit tests should be fully automated and non-interactive.
 - The test suite is normally executed on a regular basis and must be fully automated to be useful.
 - If the results require manual inspection, the tests are not proper unit tests.
 - Configure the development environment so that single tests and test suites can be a single command or a one-button click.

Structural testing is conducted to verify that paths and conditions follow detailed design specifications.

3.3 Functional Testing

Functional testing is a quality assurance (QA) process and a type of black-box testing that bases its test cases on the specifications of the software component under test. Functions are tested by feeding them input and examining the output, and internal program structure is rarely considered (not like in white-box testing). Functional testing usually describes *what* the system does.

Functional testing typically involves six steps

1. The identification of functions that the software is expected to perform
2. The creation of input data based on the function's specifications
3. The determination of output based on the function's specifications

3.4 Integration Testing

Continuous integration is checking in code in small increments and testing it against the large code base.

Integration testing is incremental. It begins with small groups of related units and continues until all units are incorporated into the intended system. Integration testing verifies that all related components such as functions, screens, or scripts interact as specified in approved requirements and design specifications. Integration testing also verifies the interaction between components and their interfaces. Input to Execute Integration Test: Integration Test Plan that should cover:

- The strategy to use when testing the integrated modules and how the tests will be conducted
- What will be tested e.g. software features, Sprint backlog and planning

3.5 System Testing

The purpose of System testing is to evaluate the system's compliance with the specified requirements. It is performed after Integration Testing but before User Acceptance testing. Its objective is to demonstrate that a system meets its documented requirements. Below is a list of potential types of testing that may occur:

- Functional
- Performance/Load/Stress
- Usability

Company Logo	PROCEDURE	SOP-00XXXX
SDLC Process	Software (QA) Testing	Page: 10/12

4. ANALYZE

4.1.1 Test Completion Criteria

Testing is considered complete when all of the following is achieved:

- Acceptance criteria has been met
- All test scripts have been successfully executed

4.1.2 Defects Classification

All issues/defects can be classified as a type of severity, whether it's Critical, major, minor, or cosmetic. These issues can be broken down further and are explained in Appendix A.

4.1.3 Testing Measurements

- Number of test cases
- Number of untested test cases

4.1.4 Defect Measurements

- Number of defects
- Number of closed defects

5. OUTPUTS

The following is a list of different outputs expected from the different tests performed throughout the QA process. Again, User Acceptance testing can be found in Section 6 below.

5.1 Unit Testing

Unit testing is complete when the following criteria are met:

- Unit test is conducted and issues resolved (Development)
- Test results and deliverables are provided to the team

5.2 Functional Testing

Functional testing is complete when the following criteria are met:

- Features and capabilities work the way the Product Owner intended
- User acceptance tests validate the aggregate behavior of many user stories
- The system meets both usability and functionality requirements

5.3 Integration Testing

Integration testing is complete when the following criteria are met:

- Outstanding defects logged in the defect tracking tool
- All integration test cases scripts have been successfully executed
- Test results and deliverables provided to the test team

Company Logo	PROCEDURE	SOP-00XXXX
SDLC Process	Software (QA) Testing	Page: 11/12

5.4 System Testing

System testing is complete when the following criteria are met:

- Test cases executed for the scope as defined in the test plan
- All critical and major defects resolved or postponed to the next build

6. USER ACCEPTANCE TESTING

6.1 Criteria

Once all the criteria for testing have been met, then User Acceptance testing can begin. As with any type of testing, User Acceptance testing has its own criteria and is complete when the following are met:

- All critical and major incidents resolved or postponed
- Test results and deliverables are provided

Criteria for entering and existing UAT are below:

Entrance Requirements		
Entrance Criteria	Compliant	Not Compliant
UAT plan approved		
All UAT team members are identified and roles defined		
Delivered target hardware/software installed and configured, including network connectivity and all backup/restore functions		

Exit Requirements		
Exit Criteria	Compliant	Not Compliant
All UAT test cases included in the acceptance test plan have been executed		
Joint evaluation of the defect issues discovered during UAT is complete		

6.2 Output

The purpose of the acceptance testing is to validate that all elements of the system are:

- Fully and properly integrated
- Overall end-to-end system functionality and performance is achieved

These tests are conducted on a platform as close to the real production environment as possible.

Below is a list of various testing done during user acceptance testing:

- Data and Database Integrity Testing, including testing of converted data and all associated ID documents and parameter files (using production values)
- Functional Testing of the delivered System software and hardware
- User Interface/Usability Testing

Company Logo	PROCEDURE	SOP-00XXXX
SDLC Process	Software (QA) Testing	Page: 12/12

7. REFERENCES

Miscellaneous References for Definitions, etc.

<http://whatis.techtarget.com/>

<http://www.softwaretestinghelp.com/software-test-metrics-and-measurements/>

8. APPENDIX A: DEFECTS CLASSIFICATION

Below is a list of all issues/defects categorized in accordance to its Source definitions.

Term	Description										
Issue Severity	<table border="1"> <tr> <td>S1 Critical</td> <td> <ul style="list-style-type: none"> ➤ Unrecoverable data loss ➤ Major functionality is not working </td> </tr> <tr> <td>S2 Major</td> <td> <ul style="list-style-type: none"> ➤ Major functionality is not working with identified work-around ➤ Secondary functionality is not working with no work-around </td> </tr> <tr> <td>S3 Minor</td> <td> <ul style="list-style-type: none"> ➤ Secondary functionality is not working with identified work-around ➤ Seldom-used functionality issues with identified work-around </td> </tr> <tr> <td>S4 Cosmetic</td> <td> <ul style="list-style-type: none"> ➤ Inconveniences ➤ Minor annoyances </td> </tr> </table>	S1 Critical	<ul style="list-style-type: none"> ➤ Unrecoverable data loss ➤ Major functionality is not working 	S2 Major	<ul style="list-style-type: none"> ➤ Major functionality is not working with identified work-around ➤ Secondary functionality is not working with no work-around 	S3 Minor	<ul style="list-style-type: none"> ➤ Secondary functionality is not working with identified work-around ➤ Seldom-used functionality issues with identified work-around 	S4 Cosmetic	<ul style="list-style-type: none"> ➤ Inconveniences ➤ Minor annoyances 		
S1 Critical	<ul style="list-style-type: none"> ➤ Unrecoverable data loss ➤ Major functionality is not working 										
S2 Major	<ul style="list-style-type: none"> ➤ Major functionality is not working with identified work-around ➤ Secondary functionality is not working with no work-around 										
S3 Minor	<ul style="list-style-type: none"> ➤ Secondary functionality is not working with identified work-around ➤ Seldom-used functionality issues with identified work-around 										
S4 Cosmetic	<ul style="list-style-type: none"> ➤ Inconveniences ➤ Minor annoyances 										
Issue Type	<table border="1"> <tr> <td>Change Request (CR)</td> <td> <ul style="list-style-type: none"> ➤ Customer's request to change an existing signed off requirement or additional requirement in addition to signed off system requirements document and functional specification ➤ May have a significant impact on the program scope </td> </tr> <tr> <td>Enhancement</td> <td> <ul style="list-style-type: none"> ➤ Usually it institutes the improvement to existing functionality and in most cases has a smaller scope and program's impact that change request ➤ Enhancement may not affect the signed off requirements and specs and is subject to be reviewed by Program multifunctional team </td> </tr> <tr> <td>Clarification</td> <td> <ul style="list-style-type: none"> ➤ Elaboration or request for elaboration of existing requirement and/or functionality ➤ May be initiated by Customer </td> </tr> <tr> <td>Defect</td> <td> <ul style="list-style-type: none"> ➤ Any deviation from signed off requirements and/or system malfunctioning </td> </tr> <tr> <td>Action item</td> <td> <ul style="list-style-type: none"> ➤ Defines particular, concrete action to be taken by assignee </td> </tr> </table>	Change Request (CR)	<ul style="list-style-type: none"> ➤ Customer's request to change an existing signed off requirement or additional requirement in addition to signed off system requirements document and functional specification ➤ May have a significant impact on the program scope 	Enhancement	<ul style="list-style-type: none"> ➤ Usually it institutes the improvement to existing functionality and in most cases has a smaller scope and program's impact that change request ➤ Enhancement may not affect the signed off requirements and specs and is subject to be reviewed by Program multifunctional team 	Clarification	<ul style="list-style-type: none"> ➤ Elaboration or request for elaboration of existing requirement and/or functionality ➤ May be initiated by Customer 	Defect	<ul style="list-style-type: none"> ➤ Any deviation from signed off requirements and/or system malfunctioning 	Action item	<ul style="list-style-type: none"> ➤ Defines particular, concrete action to be taken by assignee
Change Request (CR)	<ul style="list-style-type: none"> ➤ Customer's request to change an existing signed off requirement or additional requirement in addition to signed off system requirements document and functional specification ➤ May have a significant impact on the program scope 										
Enhancement	<ul style="list-style-type: none"> ➤ Usually it institutes the improvement to existing functionality and in most cases has a smaller scope and program's impact that change request ➤ Enhancement may not affect the signed off requirements and specs and is subject to be reviewed by Program multifunctional team 										
Clarification	<ul style="list-style-type: none"> ➤ Elaboration or request for elaboration of existing requirement and/or functionality ➤ May be initiated by Customer 										
Defect	<ul style="list-style-type: none"> ➤ Any deviation from signed off requirements and/or system malfunctioning 										
Action item	<ul style="list-style-type: none"> ➤ Defines particular, concrete action to be taken by assignee 										