Test Plan
Project “Boeing”

Document created by:
Vladyslav Zelenin

Table of Contents

Document Revision History

Date	Version	Description	Author	Reviewer	Approver
22.04	0.1	Test plan was created	V. Zelenin		

1 INTRODUCTION

Customer wants a perfect website, which passed the full cycle of manual testing. Given the specificity of the site it is very important to have the same quality and the site.

The Test Plan has been created to facilitate communication within the team members. This document describe approaches and methodologies that will apply to the unit, integration and system testing of the "https://www.boeing.com/". It includes the objectives, test responsibilities, entry and exit criteria, scope, schedule major milestones, entry and exit criteria and approach. This document has clearly identified what the test deliverables will be, and what is deemed in and out of scope.

2 SCOPE

The document mainly targets the GUI testing and validating data in report output as per Requirements Specifications provided by Client.

2.1 Functions to be tested.

- GUI
- Search and Filters Logic
- Performance

2.2 Functions not to be tested.

Not other than mentioned above in section 2.1

3 QUALITY OBJECTIVES

A primary objective of testing is to: assure that the system meets the full requirements, including quality requirements (functional and non-functional requirements) and fit metrics for each quality requirement and satisfies the use case scenarios and maintain the quality of the product. At the end of the project development cycle, the user should find that the project has met or exceeded all of their expectations as detailed in the requirements.

3.1 Primary Objectives

Any changes, additions, or deletions to the requirements document, Functional Specification, or Design Specification will be documented and tested at the highest level of quality allowed within the remaining time of the project and within the ability of the test team.

3.2 Secondary Objectives

The secondary objectives of testing will be to: identify and expose all issues and associated risks, communicate all known issues to the project team, and ensure that all issues are addressed in an appropriate matter before release. As an objective, this requires careful and methodical testing of the application to first ensure all areas of the system are scrutinized and, consequently, all issues (bugs) found are dealt with appropriately.

4 TEST APPROACH

The approach, that used, is Analytical therefore, in accordance to requirements-based strategy, where an analysis of the requirements specification forms the basis for planning, estimating and designing tests. Test cases will be created during exploratory testing. All test types are determined in Test Strategy.Team also must used experience-

based testing and error guessing utilize testers' skills and intuition, along with their experience with similar applications or technologies.

The project is using an agile approach, with weekly iterations. At the end of each week the requirements identified for that iteration will be delivered to the team and will be tested.

4.1 Test Automation

Automated unit tests are part of the development process, and UI smoke-tests from CHL01 must be also automated during which performance data must be captured.

5 ROLES AND RESPONSIBILITIES

Role	Staff Member	Responsibilities
Project Manager		1. Acts as a primary contact for development and QA team. 2. Responsible for Project schedule and the overall success of the project.
QA Lead		1. Participation in the project plan creation/update process. 2.Planning and organization of test process for the release. 3.Coordinate with QA analysts/engineers on any issues/problems encountered during testing. 4.Report progress on work assignments to the PM

QA		1. Understand requirements 2. Writing and executing Test cases 3. Preparing RTM 4. Reviewing Test cases, RTM 5. Defect reporting and tracking 6. Retesting and regression testing 7. Bug Review meeting 8. Preparation of Test Data 9. Coordinate with QA Lead for any issues or problems encountered during test preparation/execution/defect handling.

6 ENTRY AND EXIT CRITERIA

6.1 Entry Criteria

- All test hardware platforms must have been successfully installed, configured, and functioning properly.
- All the necessary documentation, design, and requirements information should be available that will allow testers to operate the system and judge the correct behavior.
- All the standard software tools including the testing tools must have been successfully installed and functioning properly.
- Proper test data is available.
- The test environment such as, lab, hardware, software, and system administration support should be ready.
- QA resources have completely understood the requirements
- QA resources have sound knowledge of functionality
- Reviewed test scenarios, test cases and RTM

6.2 Exit Criteria

- A certain level of requirements coverage has been achieved.
- No high priority or severe bugs are left outstanding.

- All high-risk areas have been fully tested, with only minor residual risks left outstanding.
- Cost – when the budget has been spent.
- The schedule has been achieved.

7 SUSPENSION CRITERIA AND RESUMPTION REQUIREMENTS

7.1 Suspension criteria

- The build contains many serious defects which seriously or limit testing progress.
- Significant change in requirements suggested by client
- Software/Hardware problems
- Assigned resources are not available when needed by test team.

7.2 Resumption criteria

Resumption will only occur when the problem(s) that caused the caused the suspension have been resolved.

8 TEST STRATEGY

8.1 QA role in test process

- Understanding Requirements:

- Requirement specifications will be sent by client.
- Understanding of requirements will be done by QA

- Preparing Test Cases:

QA will be preparing test cases based on the exploratory testing. This will cover all scenarios for requirements.

- Preparing Test Matrix:

QA will be preparing test matrix which maps test cases to respective requirement. This will ensure the coverage for requirements.

- Reviewing test cases and matrix:

- Peer review will be conducted for test cases and test matrix by QA Lead
- Any comments or suggestions on test cases and test coverage will be provided by reviewer respective Author of Test Case and Test Matrix
- Suggestions or improvements will be re-worked by author and will be send for approval
- Re-worked improvements will be reviewed and approved by reviewer

- Creating Test Data:

Test data will be created by respective QA on client's developments/test site based on scenarios and Test cases.

- Executing Test Cases:

- Test cases will be executed by respective QA on client's development/test site based on designed scenarios, test cases and Test data.
- Test result (Actual Result, Pass/Fail) will updated in test case document

Defect Logging and Reporting:

QA will be logging the defect/bugs in Word document, found during execution of test cases. After this, QA will inform respective developer about the defect/bugs.

- Retesting and Regression Testing:

Retesting for fixed bugs will be done by respective QA once it is resolved by respective developer and bug/defect status will be updated accordingly. In certain cases, regression testing will be done if required.

- Deployment/Delivery:

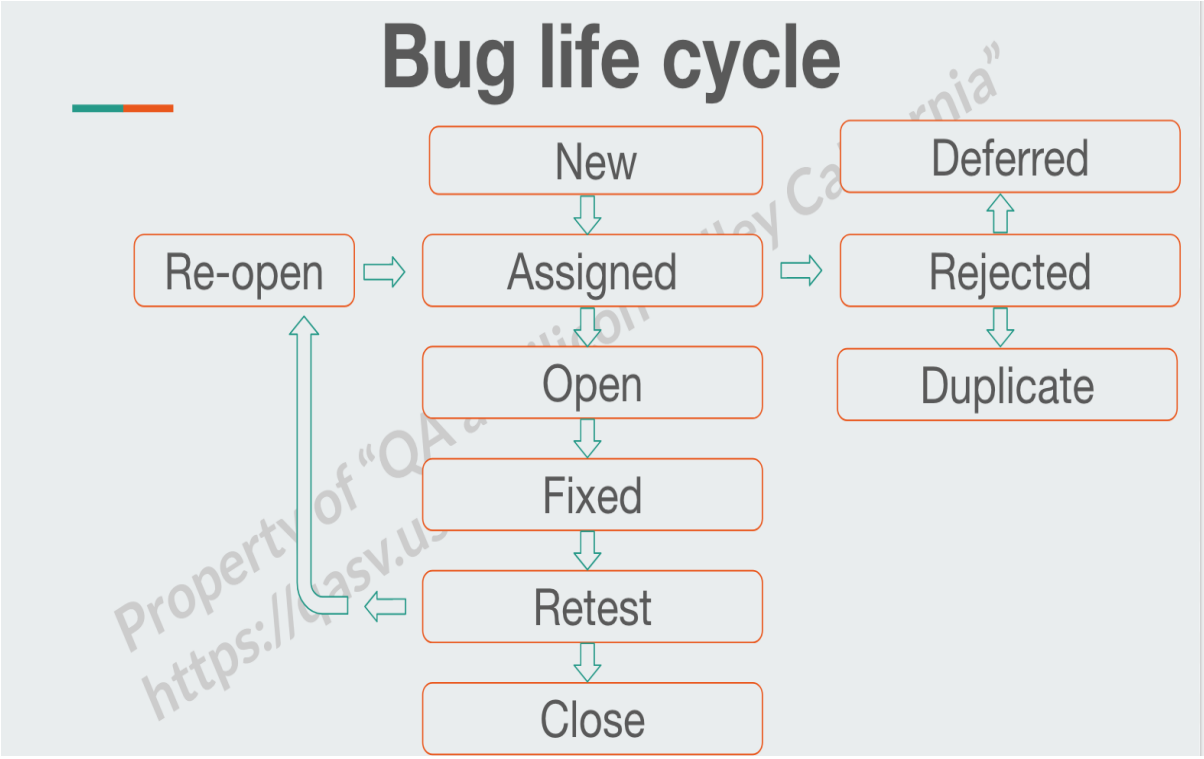
- Once all bugs/defect reported after complete testing is fixed and no other bugs are found,

report will be deployed to client’s test site by PM.

- Once round of testing will be done by QA on client's test site if required Report will be delivered along with sample output by email to respective lead and Report group.
- QA will be submitting the filled hard copy of delivery slip to respective developer.
- Once lead gets the hard copy of delivery slip filled by QA and developer, he will send the report delivery email to client.

8.2 Bug life cycle:

All the issues found while testing will be logged into JIRA.



8.3 Testing types

Black box testing:

It is some time called behavioral testing or Partition testing. This kind of testing focuses on the functional requirements of the software. It

enables one to derive sets of input conditions that that will fully exercise all functional requirements for a program.

GUI Testing:

GUI testing will includes testing the UI part of report. It covers users Report format, look and feel, error messages, spelling mistakes, GUI guideline violations.

Integration Testing:

Integration testing is systematic technique force on structing the program structure while conducting test to uncover errors associated with interacting. In Report, integration testing includes the testing Report from respective location(s).

Functional Testing:

Functional testing is carried out in order to find out unexpected behavior of the report. The characteristic of functional testing are to provide correctness, reliability, testability and accuracy of the report output/data.

System Testing:

System testing of software is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements.

Performance Testing:

- Check the optimal time the page is loaded
- Check the operation of the system under load

User acceptance testing:

The purpose behind user acceptance testing is to conform that system is developed according to the specified user requirements and is ready for operational use. Acceptance testing is carried out at two levels - Alpha and Beta Testing. User acceptance testing (UAT) will be done at the Client.

Alpha testing:

The alpha test is conducted at the developer's site by client.

8.4 Bug Severity and Priority Definition

Bug Severity and Priority fields are both very important for categorizing bugs and prioritizing if and when the bugs will be fixed. The bug Severity and Priority levels will be defined as outlined in the following tables below. Testing will assign a severity level to all bugs. The Test Lead will be responsible to see that a correct severity level is assigned to each bug.The QA Lead, Development Lead and Project Manager will participate in bug review

meetings to assign the priority of all currently active bugs. This meeting will be known as

"Bug Triage Meetings". The QA Lead is responsible for setting up these meetings on a routine basis to address the current set of new and existing but unresolved bugs.

Severity List

The tester entering a bug into GForge is also responsible for entering the bug Severity.

Severity ID	Severity	Severity Description
1	Critical	The module/product crashes or the bug causes non-recoverable conditions. System crashes, GP Faults, or database or file corruption, or potential data loss, program hangs requiring reboot are all examples of a Sev. 1 bug.

2	High	Major system component unusable due to failure or incorrect functionality. Sev. 2 bugs cause serious problems such as a lack of functionality, or insufficient or unclear error messages that can have a major impact to the user, prevents other areas of the app from being tested, etc. Sev. 2 bugs can have a work around, but the work around is inconvenient or difficult.
3	Medium	Incorrect functionality of component or process. There is a simple work around for the bug if it is Sev. 3.
4	Minor	Documentation errors or signed off severity 3 bugs.

Priority List

Priority	Priority Level	Priority Description
1	Must Fix	This bug must be fixed immediately; the product cannot ship with this bug.
2	Should Fix	These are important problems that should be fixed as soon as possible. It would be an embarrassment to the company if this bug shipped.
3	Fix When Have Time	The problem should be fixed within the time available. If the bug does not delay shipping date, then fix it.
4	Low Priority	It is not important (at this time) that these bugs be addressed. Fix these bugs after all other bugs have been fixed. Enhancements/ Good to have features incorporated- just are out of the current scope.

9 RESOURCE AND ENVIRONMENT NEEDS

9.1 Testing Tools

Process	Tool
Test case creation	Microsoft Word, Microsoft Excel, JIRA
Test case tracking	JIRA, Confluence
Test case execution	Manual, Selenium IDE

Test case management	Microsoft Excel, JIRA, Confluence
Defect management	Microsoft Word, JIRA, Confluence
Test reporting	JIRA
API Testing	Postman
Performance Testing	Lighthouse, GT Metrix, SpeedLab
Automation Testing	Selenium IDE, XPath (for locators searching)

9.2 Configuration Management

- Documents CM: SVN
- Code CM: Git

9.3 Test Environment

- Support level 1 (browser):
  - Windows 10 PRO: Google Chrome (111.0.5563.148 (64-bit))

- Support level 2 (device):
  - LAPTOP Dell-H26RA96
- Support level 3:
  - Windows 10: Google Chrome (110)

10 TEST SCHEDULE

Task Name	Start	Finish	Effort	Comments

APPROVALS:

	Project Manager	QA Lead
Name	Sergey Efremov	
Signature		

TERMS/ACRONYMS

The below terms are used as examples, please add/remove any terms relevant to the document.

TERM/ACRONYM	DEFINITION
API	Application Program Interface
GUI	Graphical user interface
PM	Project manager
UAT	User acceptance testing
CM	Configuration Management
QA	Quality Assurance
RTM	Requirements Traceability Matrix