Pontiflex ADLeads

Test Plan

Table of Contents

Document History

Version Number	Date	Contributor	Description
V1.0	05/25/2011	Reaz Patwary	Draft Copy

Roles and Responsibilities

Name	Role	Responsibilities
Reaz Patwary	QA	
Robert Collins	Project Manager	
Roshan Bangera	CTO	

Approval

QA

Project Manager

CTO

Introduction

This document is a Test Plan for the **AppLeads**, produced by Quality Assurance team of **Pontiflex**. This test strategy document describes the appropriate strategies, process, work flows and methodologies used to plan, organize, execute and manage testing of software projects within **Pontiflex.** It also contains various resources required for the successful completion of this project.

Project Overview

Pontiflex is the industry's leading email and social acquisition platform. Our patent pending technology lets you create ads that connect your brand with the right people on websites, social networks and mobile apps.

People sign up when they want to hear more from you. Once they do that, you can continue the conversation through email, Facebook, Twitter and community sites. Best of all, you pay only when people sign up, so you never spend money on ads that might not convert.

- Brand marketers can run ads on top sites, collect the contact info of real people and build effective social acquisition campaigns.
- Direct response marketers can deliver strong call-to-action creative and build responsive email and direct marketing databases.

ADLeads is the lead market lead generation application in the market which will help App developer to make money by signing up new user on their App. It helps App developer to create their own contact database for future references. User sign in to the App and Developer get paid for that. The main theme of Pontiflex is REAC REAL PEOPLE GET REAL RESULT which is a big example of ADLeads application. On the application user can sign in for the appropriate product information they want to receive. After signing in their information will send to the advertiser. By signing up user give privilege to App developer to send the information to the appropriate advertiser. User receive appropriate information from that advertiser and product which they sign up for.

Objective

The primary objective of testing is to deliver a bug free application to its user. To assure that **Pontiflex** Quality Assurance team will perform the quality check for the Application to *meets the full requirements, including quality requirements (AKA: 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.

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.

The secondary objective of testing application 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 deal with appropriately.

Features to be tested

The Pontiflex ADLeads *Test Plan* defines the unit, integration, system, regression, and User Acceptance testing approach. The test scope includes the following:

- Testing of all functional, application performance, Load and Stress, security and use cases requirements listed in the *Use Case* document.

End-to-end testing and User Acceptance Testing will be performed in multi platform and multi browsers. List below will describe the the platforms and browsers and its version

Platform:

1. **Windows (XP, Vista,Windows 7)**
2. **Mac**
3. **Linux (Ubuntu)**

Browser:

1. **Internet Explorer (7.0 and up)**
2. **Mozilla Firefox (3.6 and up)**
3. **Google Chrome**
4. **Safari**

Features not to be tested

It varies project to project so we have to fill this part based on project.

Roles and Responsibilities

Developing Team

Dev team are responsible for developing the application based on use case and business specific documents. They will perform a unit and integration testing before it comes to Quality department. Will fix the bug declare by the QA team.

Quality Assurance Team

QA team is responsible for the whole testing process, work flow, all kind of validation of the application, creating appropriate test scenario, writing test cases, selecting appropriate Functional testing tools, all kinds of risk assignment for the project.

User Experience Team

UE team is responsible for creation appropriate UI, wire frames and visual design for the application.

Definitions

Development Done and Ready for QA

When the developers are done developing all the functionally based on the use case and Business specific document. Find no critical bug After running the unit and integration testing.

QA Done

When All the QA documents (Test Plan, Test cases, Tractability Matrix) is done and the project is tested by following the test methodology written by the Quality assurance team and find no show stopper or critical bug. After Regression testing, thats the time we can say the testing is done and the

project is ready for Deploy.

Go/No Go

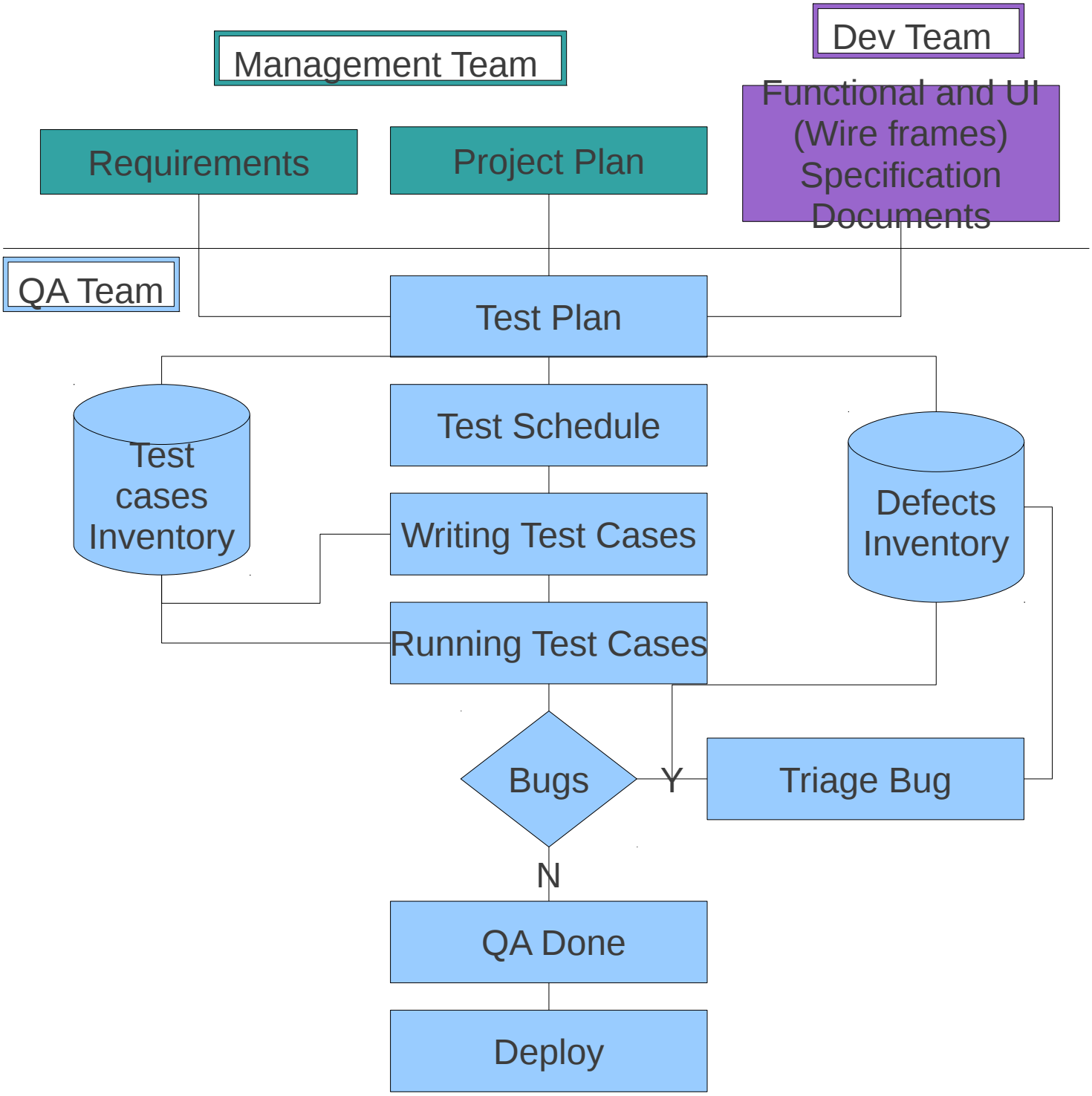
Project manager will decide if the project is ready for user acceptance test after getting the feedback QA done from the QA team.

Deploy

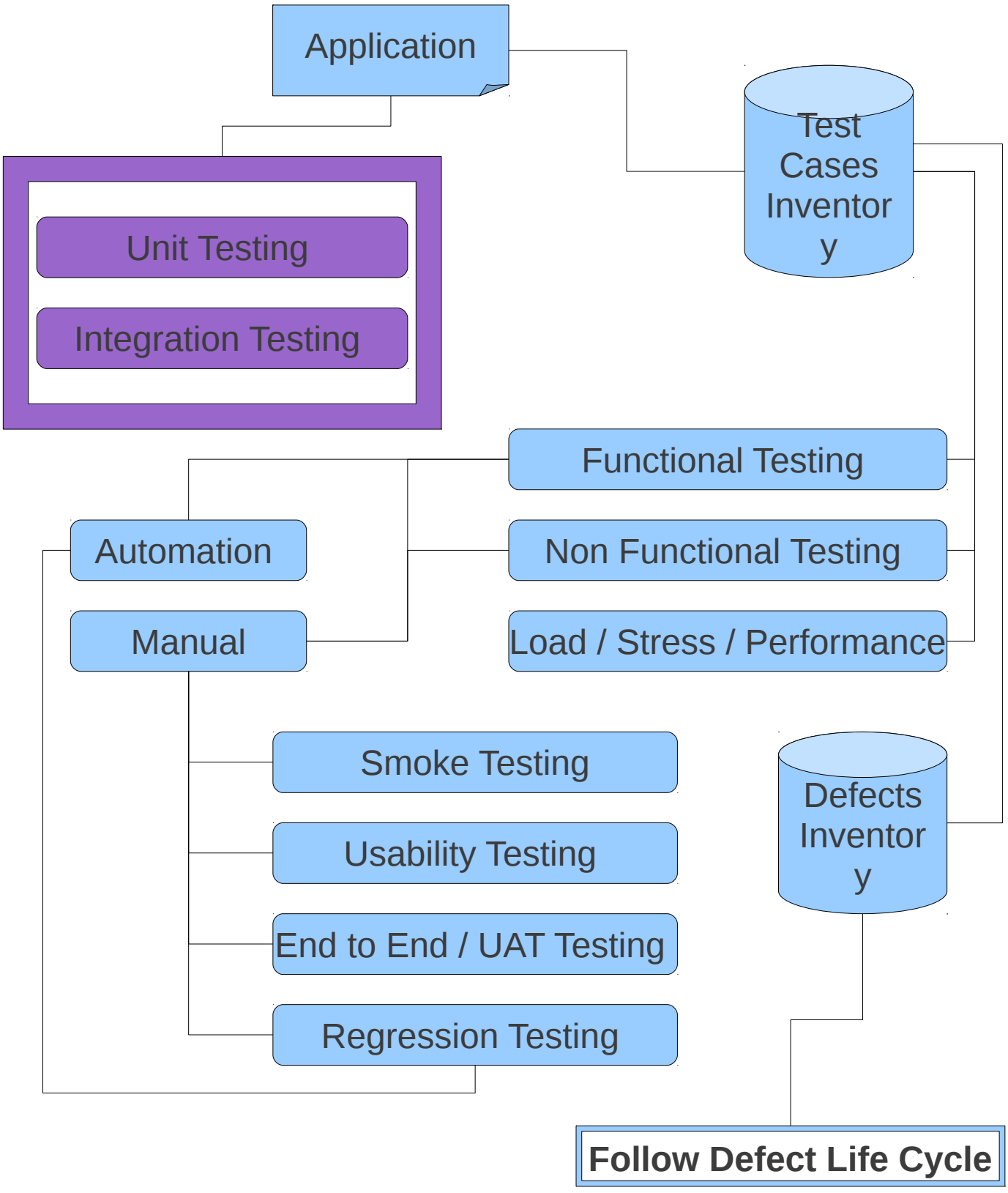
Project manager will decide if the project is ready to go live.

Test Methodology

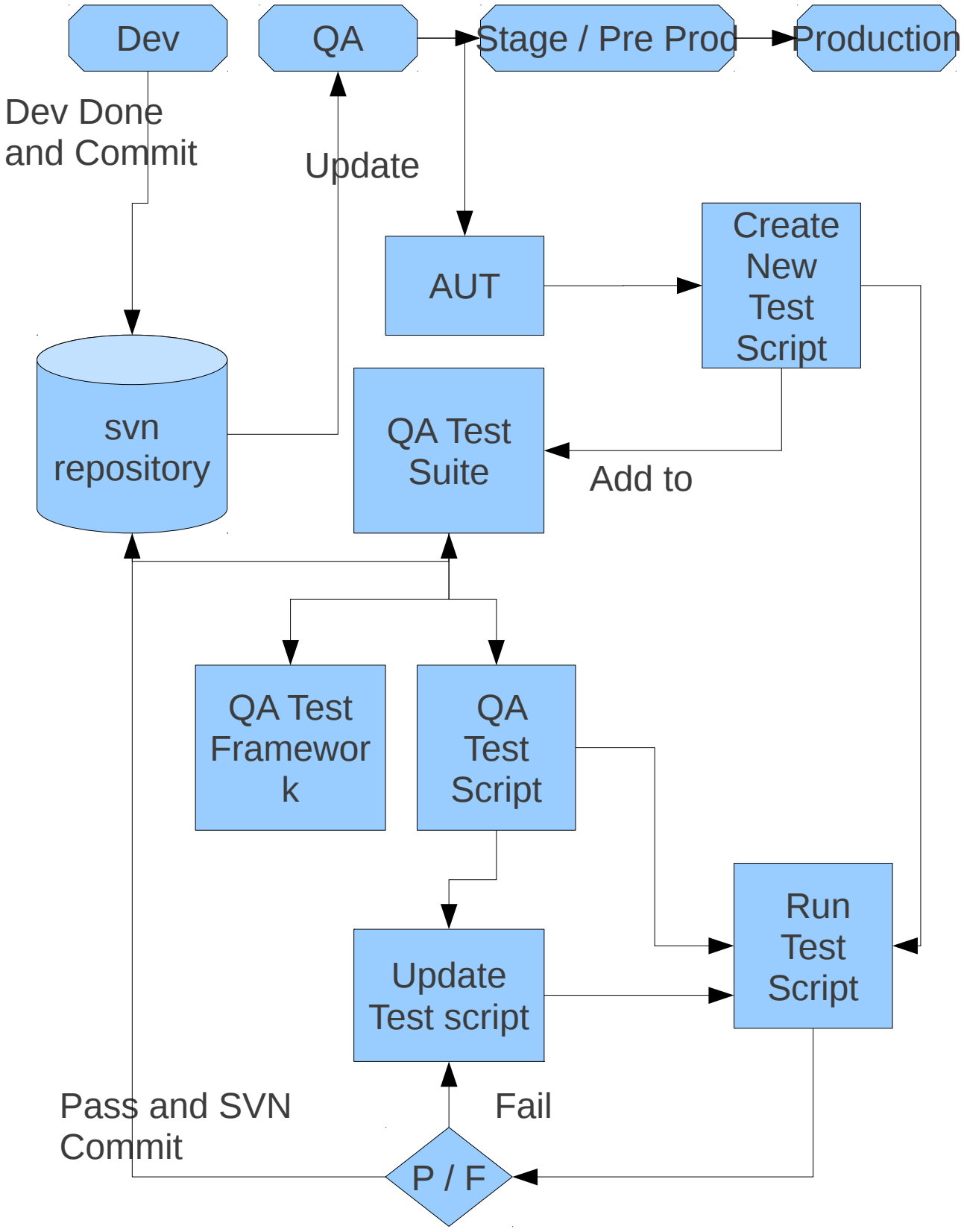
Testing Process Management



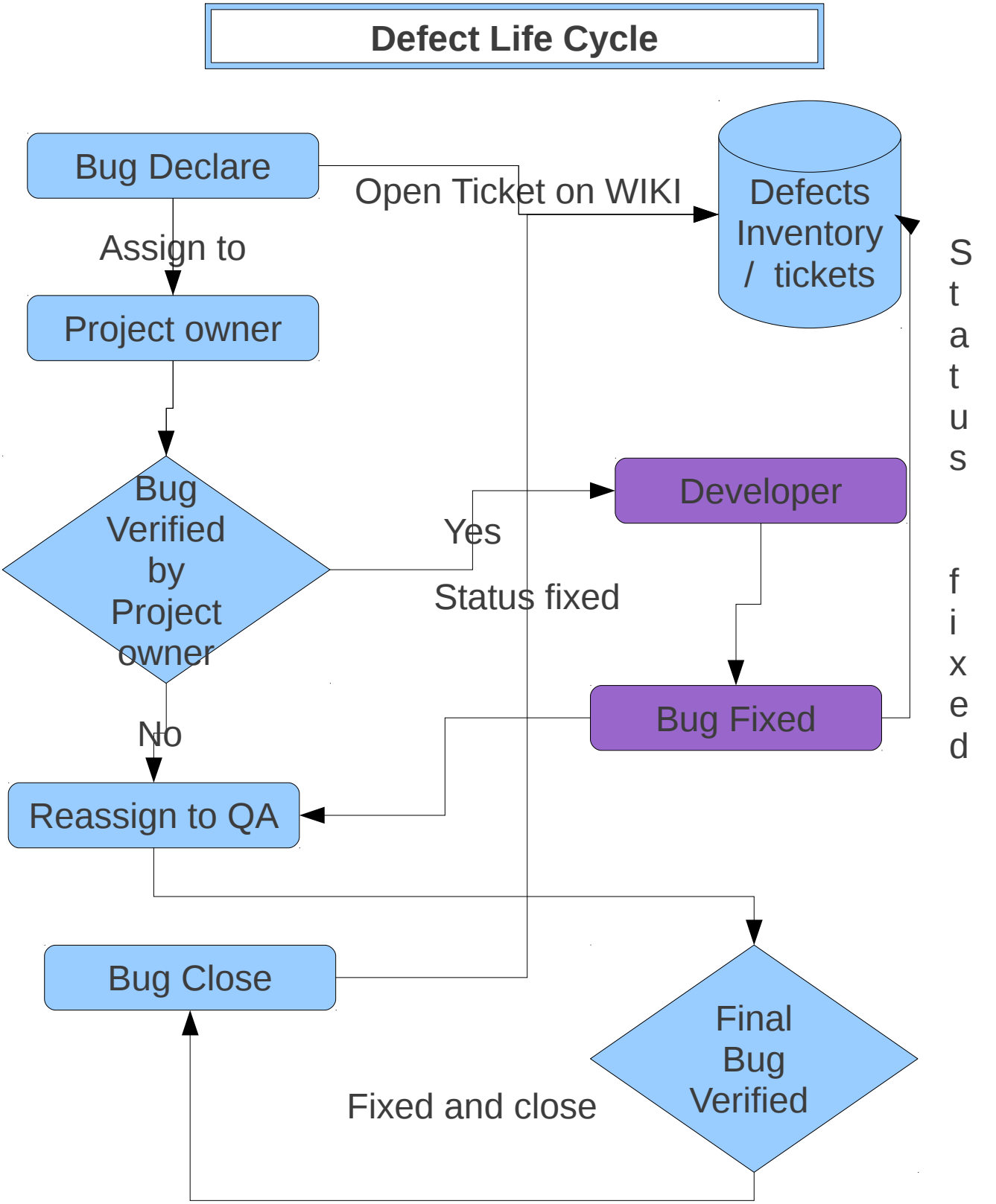
Process Flow Diagram



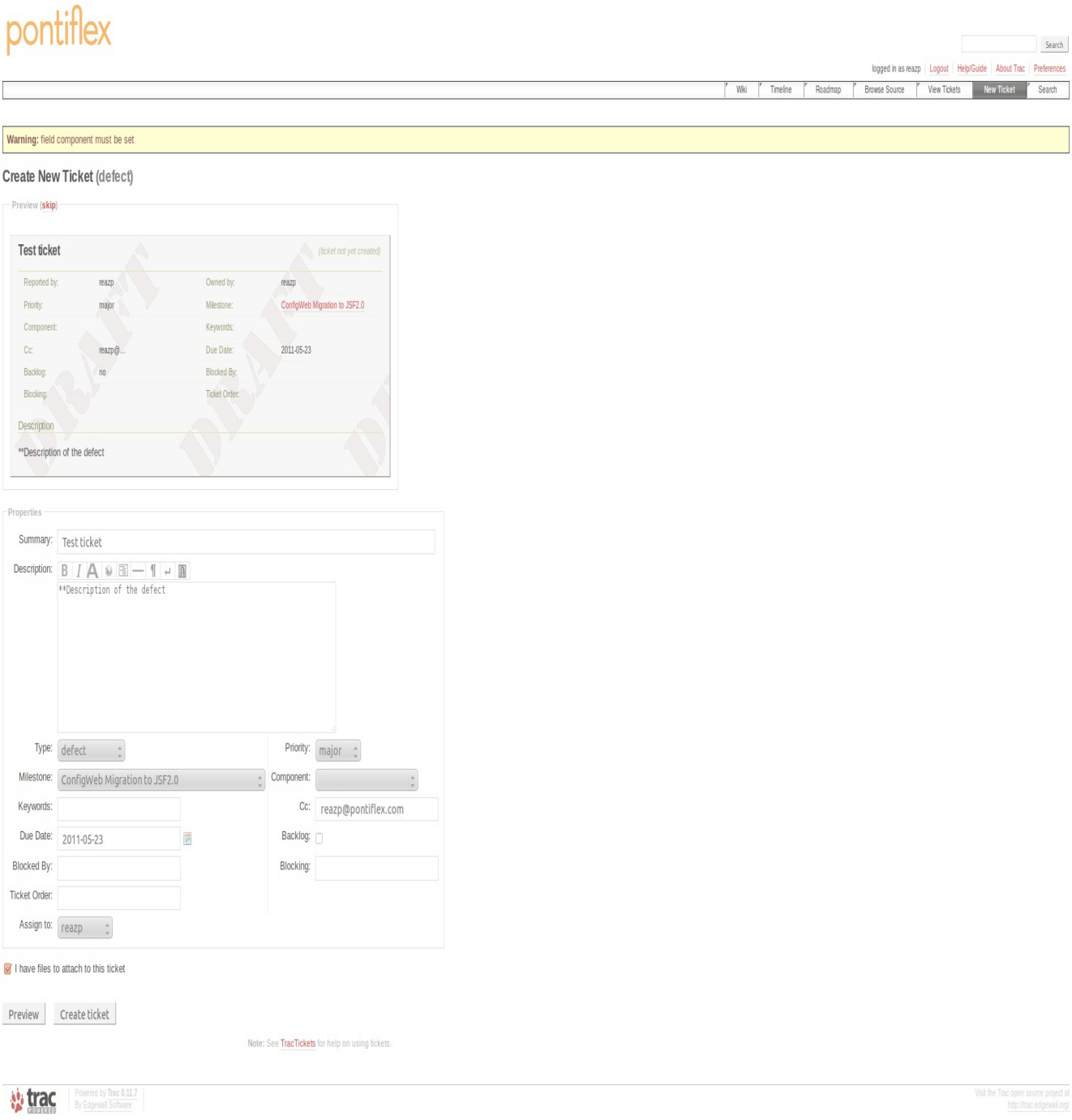
Test Automation Framework Architecture



Defect flow diagram



Sample Defect Creation



Test scenario creation

Test scenario can be created by following the Design Document and Wire Frames. QA tester will perform this task to implement functional and non- functional test cases. In this step whole QA team will be working closely with the Development team to find out critical test scenario, Dependency of a test and Tractability matrix for the test which will provide the cross reference of any test with requirements.

Test Execution

The whole test case inventory will be divided into Functional and non functional test cases. The whole test cases will tested manually. Every test will be assigned to a tester and he will be the owner of that test. While running this test if tester find any bug, he will declare a bug on wiki (Pontiflex tracking system) with appropriate screen-shot. The bug will follow the defect life cycle until it get closed. While doing this manual testing, functional test will be automated through functional testing tool. Pontiflex is using Selenium to automate functional test. All the service test will be automate through this tools as well. Before sign off the testing the automated tester will run the whole test suite to complete the regression testing.

Smoke Tests

The objective is to determine if further testing is possible. These test cases should emphasize breadth more than depth. All components should be touched, and every major feature should be tested briefly by the Smoke Test. If any Level 2 test case fails, the build is returned to developers un-tested.

Functional Testing:

The objective of function test is to measure the quality of the functional (business) components of the system. Tests verify that the system behaves correctly from the user / business perspective and functions according to the requirements, models, storyboards, or any other design paradigm used to specify the application. The function test must determine if each component or business event: performs in accordance to the specifications, responds correctly to all conditions that may be presented by incoming events / data, moves data correctly from one business event to the next (including data stores), and that business events are initiated in the order required to meet the business objectives of the system.

Non Functional Testing

In order to ensure your system is ready to go live it is necessary to go beyond just functional testing. Non-Functional Testing is designed to evaluate the readiness of your system according to several criteria not covered by functional testing. These criteria include:

Performance: We test a system's performance to ensure that it provides acceptable response times and
user load capabilities. Performance testing will: Test applications against predicted future volumes identify break points to assess the applications ability to cope with a given number of users undertake performance based load testing to assess applications performance when being accessed by large numbers of users simultaneously test over an extended period of time to ensure, for example, that an application is not running out of memory

Disaster Recovery: We will ensure that you are prepared in case of a disaster. Disaster recovery plans are checked and back-up systems are tested. Proving the effectiveness of your back-up systems will prevent damage to your business' reputation and minimize the costs involved if the main system malfunctions.

Security: We will test the security of your system and assess its vulnerability to hacking, etc

Usability Testing

The purpose of usability testing is to ensure that the new components and features will function in a manner that is acceptable to the user.

Development will typically create a non-functioning prototype of the UI components to evaluate the proposed design. Usability testing can be coordinated by testing, but actual testing must be performed by non-testers (**as close to end-users as possible).** Testing will review the findings and provide the project team with its evaluation of the impact these changes will have on the testing process and to the project as a whole.

Regression Testing

Regression testing is the process of running all existing test cases and verifying that all test cases pass. The purpose of regression testing is to detect unexpected faults—especially those faults that occur because a developer did not fully understand the internal code correlations when he or she modified or extended code that previously functioned correctly. Regression testing is the only reliable way to ensure that modifications did not introduce new errors into code or to check whether modifications successfully eliminated existing errors. Every time code is modified or used in a new

environment, regression testing should be used to check the code's integrity. As in this stage we will have to test whole application from start to end, the best idea is to use Automated Functional testing tool which will help to run the whole test suite quick and easy. The whole test suit will be schedule to run at night (during automated nightly builds) to ensure that errors are detected and fixed as soon as possible in the morning.

Final release Testing

Assuming critical bugs are resolved during previous iterations testing- Throughout the Final Release test cycle, bug fixes will be focused on minor and trivial bugs . Testing will continue its process of verifying the stability of the application through regression testing (existing known bugs, as well as existing test cases).

The overall purpose of testing is to ensure the APPLeads application current release has a stable build and ready for user acceptance testing

User Acceptance testing:

Once the application is ready to be released the crucial step is User Acceptance Testing. In this step a group representing a cross section of end users tests the application. This is the last step of testing on a real user environment.

The overall purpose of testing is to ensure the APPLeads application performs at an acceptable level for the customer. This document outlines the detailed plan for user acceptance testing of this application.

Bug Reporting

Pontiflex has its own central project management system name WIKI which keep track of all the current project, project specification documents, Dev and QA related all information and all defects. When a QA person find any bug, his responsibility is to go to that system and create a new ticket for that defect. On that ticket he will attach the persons related to that specific project with a screenshot. The system will automatically send an email to that person about the bug. While declaring the bug QA person has to give the status, a brief description and select the priority level of that bug. Pontiflex has five different priority level which described below on Bug Priority Definition section. After receiving the email from WIKI system, developer verified the bug to make sure its a bug ans start working with it and change the status as assigned to him. After fixing the bug he update the ticket description as fixed and send it to QA for re-verification. QA person verify the fix and close the bug when the bug has been fixed and update the status as fixed.

Goals and Deliverables

Goals and deliverables of the test plan of the “ADLeads” are as follows –

Goals

- ✓ To accomplish all tasks described in this test plan.
- ✓ To install a measurable, improvable, repeatable, and manageable test process.
- ✓ To verify the functionality and content of the current version of the application.
- ✓ To reduce the frequency of error associated with manual testing.
- ✓ To find and successfully track 100% of defects present along the user path defined in this plan.

Deliverables

- ✓ Test Planning Stage
- ✓ Test Schedule
- ✓ Test Execution and Defect Tracking Stage
- ✓ Test Result Report
- ✓ Project wrap up on QA environment and declare QA done.

Requirements Traceability Matrix

A Requirements Traceability Matrix (RTM) which is used to link the test scenarios to the requirements and use cases is a required part of the Test Plan documentation for all projects. Requirements traceability is defined as the ability to describe and follow the life of a requirement, in both a forward and backward direction (i.e. from its origins, through its development and specification, to its subsequent deployment and use, and through periods of ongoing refinement and iteration in any of these phases).

Testing Tool

Tracking Tool

Pontiflex has its own tracking system name Wiki. Test team will declare or create bug with appropriate screen shot and assign them the project owner. The Bug will follow the bug life cycle until the bug has been close by the bug declared person.

Test Environment

Hardware

ThinkPad T420 - 2 Yr Depot Topseller Warranty
Processor: Intel Core i7-2620M Processor (2.70GHz, 4MB L3, 1333MHz) with Intel HD Graphics 3000
OS: Windows 7 Home Premium 64
OS language: Windows 7 Home Premium 64 US English
Display type: 14.0" HD (1600x900) LED Backlit Anti-Glare Display, Mobile Broadband Ready
Graphics: Intel HD Graphics 3000
Total memory: 2 GB PC3-10600 DDR3 SDRAM 1333MHz SODIMM Memory (1 DIMM) at 1066MHz (purchase extra 2GB DDR3 chip from www.newegg.com)
Keyboard: Keyboard US English
Camera: 720p HD Camera
Hard drive: 320 GB Hard Disk Drive, 7200rpm
Optical device: DVD Recordable
Expansion: Express Card Slot & 4 in 1 Card Reader
Battery: 6 cell Li-Ion Battery - 55+
Power cord: Country Pack North America with Line cord & 65W AC adapter
Bluetooth: Bluetooth 3.0
PCI adapter: IEEE 1394a port
WiFi adapters: Intel centrino wireless-n 1000
Broadband: Integrated Mobile Broadband - Upgradable
Language pack: Language Pack US English

Software

OS

- Windows
- Ubuntu
- Apple OS x

Browser

- Firefox
- Internet Explorer
- Google Chrome
- Safari

Test Automation tool

- Selenium
- Java
- Junit (framework)

IDE

- IntelliJ

Bug Priority Definition

Bug Priority field is very important for categorizing bugs and prioritizing if and when the bugs will be fixed. The bug Priority levels will be defined as outlined in the following tables below. Testing will assign a Priority level to all bugs. The Test Lead will be responsible to see that a correct Priority level is assigned to each bug.

Priority List

Wiki	Priority Level	Priority Description
Blocker	Must Fix	This bug must be fixed immediately; the product cannot deploy with this bug.
Critical	Should Fix	These are important problems that should be fixed as soon as possible.
Major	Fix When Have Time	The problem should be fixed within the time available.
Minor	Low Priority	It is not important (at this time) that these bugs be addressed. Fix these bugs after all other bugs have been fixed.
Trivial	Trivial	Enhancements/ Good to have features incorporated- just are out of the current scope.

Terms/Acronyms

TERM/ACRONYM	DEFINITION
IO	Insertion Order
Lead	
QA	Quality Assurance
TBD	To Be Determined