CS 4750

Recreation
OUTLET

Software Quality Assurance Plan

Version: (1.0) **Date: (02/13/2014)**

<u>**Document History and Distribution**</u>

1. Revision History

Revision #	Revision Date	Description of Change	Author
1.0	02/13/2014	Initial Release – Prototype Test Plan	Aaron Sorensen

2. Distribution

Recipient Name	Recipient Organization	Distribution Method

TABLE OF CONTENTS

1. INTRODUCTION

(NOTE 1: THE SOFTWARE TEST PLAN GUIDELINES WERE DERIVED AND DEVELOPED FROM IEEE STANDARD FOR SOFTWARE TEST DOCUMENTATION (829-1998)).

The Software Test Plan (STP) is designed to prescribe the scope, approach, resources, and schedule of all testing activities. The plan will identify the items to be tested, the features to be tested, the types of testing to be performed, the personnel responsible for testing, the resources and schedule required to complete testing, and the risks associated with the plan.

1.1 Objectives

Testing will grow to cover all implemented features of the system. Team members will be assigned test cases between milestones to periodically check the system. These members will test the system and record their results into a test report to be released to the team for reference. These results must be done before the following milestone so they may be presented.

1.2 Testing Strategy

Testing is the process of analyzing a software item to detect the differences between existing and required conditions and to evaluate the features of the software item. Team members who are assigned to testing will complete all test cases and record their results preceding the following milestone. Testing will be done on the most current build of the system with the most current list of test cases. As features are implemented new test cases will need to be prepared.

1.3 Scope

Testing will be performed at several points in the life cycle as the product is constructed. Testing is a very 'dependent' activity. As a result, test planning is a continuing activity performed throughout the system development life cycle. The test plans must be revised for each level of product testing.

1.5 Definitions and Acronyms

po – purchase order

pl – product line

2. TEST ITEMS

2.1 Program Modules

Developer is expected to perform basic testing on all features they implement. New features should be tested against all use case requirements.

2.2 User Procedures

It is required that all user documentation that is reviewed should be revised to ensure it is correct, complete, and comprehensive.

2.3 Operator Procedures

Testing procedures will be implemented further in the lifetime of the project to ensure that the application can be run and supported in a production environment.

3. FEATURES TO BE TESTED

The following features and combinations of software features are to be tested:

Create Invoice

Create Item Transfer

Create Item

Create Department

Create Category

Create Subcategory

Create Product Line

Create Purchase Order

Create Receiving Log

Create Invoice

Create Vendor

Edit Vendor

Search Vendors

Print Label

Login

4. FEATURES NOT TO BE TESTED

Features not listed above are done so for not being implemented, being partially implemented, or being implemented without test cases. These are not to be tested in this revision until such a time comes when they are implemented and test cases are created.

5. APPROACH

Here we present the recommended approach to the testing of the software applications. The previous section on Test Requirements described *what* will be tested; this describes *how* it will be tested.

The main considerations for the test strategy are the techniques to be used and the criterion for knowing when the testing is completed.

In addition to the considerations provided for each test below, testing should only be executed using known, controlled databases, in secured environments.

The following test strategy is generic in nature and is meant to apply to the requirements listed in the test cases.

5.1 Component Testing

Developers will ensure that the program logic is complete and correct to ensure that the component works as designed will be done periodically by as features are implemented.

5.2 Integration Testing

Testing conducted in which software elements and hardware elements are combined and tested until the entire system has been integrated will be done when we move to the warehouse. This will ensure that design objectives are met and will ensure that the software, as a complete entity, complies with

operational requirements.

5.3 Conversion Testing

Database members may need to further work on conversion testing to ensure that all data elements and historical data is converted from their old system format to our new system format to ensure seamless integration.

5.4 Job Stream Testing

As team members pull and commit to SVN, testing may be done to ensure that the application continues to operate in our production environment.

5.5 Interface Testing

This testing will ensure that the application operates efficiently and effectively outside the application boundary with all interface systems.

5.6 Recovery Testing

When main features are implemented testing will be done to ensure that application restart and backup and recovery facilities operate as designed.

5.7 Performance Testing

Testing will be done at the warehouse to ensure that that the application performs to customer expectations (response time, availability, portability, and scalability).

5.8 Regression Testing

As features are implemented and new test cases are created, testing will consistently be done using previous test cases of other systems to ensure that that applied changes to the application have not adversely affected previously tested functionality.

5.9 Acceptance Testing

Each meeting with the client testing will be conducted to determine whether or not the system satisfies the acceptance criteria and to enable the client to determine whether or not to accept the system. Acceptance testing will

ensure that customer requirements' objectives are met and that all components are correctly included in a customer package.

5.10 Beta Testing

Upon immediate release, testing done by the customer, using a pre-release version of the product to verify and validate that the system meets business functional requirements and to detect application faults, failures, and defects.

6. PASS / FAIL CRITERIA

Here we specify the criteria to be used to determine whether each item has passed or failed testing.

6.1 Suspension Criteria

- **System crashes**
- **Error received during test**
- **Test has undesired result**
- **Feature currently being worked on**

6.2 Resumption Criteria

- **Build updated to fix crash, error, or undesired result.**
- **Developer has committed build (if it was being worked on)**

6.3 Approval Criteria

- **90% of the test cases pass**
- **All test cases dealing with critical functionality must pass**
- **All medium and high severity defects must be fixed**
- **Test coverage must be at least 90%**

7. TESTING PROCESS

Here we identify the methods and criteria used in performing test activities. We define the specific methods and procedures for each type of test and define the detailed criteria for evaluating test results.

7.1 Test Deliverables

A testing report will be created for each software tester showing the result of each test case. Any bugs found will be sent to a bug report to be resolved.

7.2 Testing Tasks

Testing must be done on a machine capable of running the most current build of the software. Pull the most recent build and walk through each test case. Record results and any bugs found until finished. When a bug is identified by the tester it should be committed to the bug log. When finished tester must post results.

7.3 Responsibilities

Group members will be assigned responsibilities for managing, designing, preparing, executing, witnessing, checking, and resolving test activities between milestones and will change week to week.

7.4 Schedule

Testing responsibilities will be assigned between milestones and must be completed and all documentation submitted on the Thursday before the following milestone. Testing documentation will be submitted in the read ahead materials before the milestone.

8. ENVIRONMENTAL REQUIREMENTS

Here we specify both the necessary and desired properties of the test environment including the physical characteristics, communications, mode of usage, and testing supplies. We identify special test tools needed and other testing needs.

8.1 Hardware

- **Stable computer**
- **Stable internet connection**

8.2 Software

- **SVN**
- **Visual Studio 2012**

8.5 Publications

- **Test cases**

8.6 Risks and Assumptions

- **Testing relies on current test cases and features. If a feature is broken testing is halted until it is fixed.**
- **Test case may not get implemented for new feature.**
- **Hard to determine if a feature is currently being worked on.**
- **We assume the current build is a stable enough to compile and run.**

9. CHANGE MANAGEMENT PROCEDURES

As testing advances all team members may review and update the test plan. Any individuals who update must record their name and changes at the beginning of the document.

10. PLAN APPROVALS

Frank Eddy and Dr. Fry may both approve the current test plans when they are updated the following milestone.