Software Test Plan

for

Exam Generator Application

Version 1.0 approved

Prepared by Bryan Smith

Joe LaCava

Scott Arnette

Derek Ouzia

University of Virginia’s College at Wise

5/1/15

Table of Contents

Table of Contents ii

Revision History ii

1. Introduction 1

1.1 Objectives 1

1.2 Testing Strategy 1

1.3 Scope 1

1.4 Reference Material 2

1.5 Definitions and Acronyms 2

2. Test Items 2

2.1 Program Modules 2

2.2 User Procedures 2

2.3 Operator Procedures 2

3. Features 3

3.1 Features to be Tested 3

3.2 Features Not to be Tested 3

4. Approach 3

4.1 Component Testing 3

4.2 Integration Testing 3

4.3 Interface Testing 3

4.4 Regression Testing 4

4.5 Acceptance Testing 4

4.6 Beta Testing 4

5. Pass/Fail Criteria 4

5.1 Suspension Criteria 4

5.2 Resumption Criteria 4

5.3 Approval Criteria 4

6. Testing Process 4

6.1 Test Deliverables 5

6.2 Testing Tasks 5

6.3 Responsibilities 5

6.4 Resources 5

6.5 Schedule 5

7. Environmental Requirements 5

7.1 Hardware 5

7.2 Software 6

7.3 Security 6

7.4 Tools 6

7.5 Publications 6

7.6 Risks and Assumptions 6

8. Change Management Procedures 6

9. Plan Approvals 6

10. Test Procedures 6

11. Requirements Matrix 7

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| Scott Arnette | 5/1/15 | Initial Creation | 1.0 |
|  |  |  |  |

# Introduction

The Introduction section of the Software Test Plan (STP) provides an overview of the project and the product test strategy, a list of testing deliverables, the plan for development and evolution of the STP, reference material, and agency definitions and acronyms used in the STP.

The Software Test Plan (STP) is designed to prescribe the scope, approach, resources, and schedule of all testing activities. The plan must 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.

The project that this test plan exists for is the Exam Generation Application, a Java based program to be used by professors or other instructing staff for the generation of tests based on a bank of various questions Testing of the Exam Generation Application will consist of unit tests and system tests to ensure the full intended functionality of the application is present and no unintended or malicious defects are present.

## Objectives

<Describe, at a high level, the scope, approach, resources, and schedule of the testing activities. Provide a concise summary of the test plan objectives, the products to be delivered, major work activities, major work products, major milestones, required resources, and master high-level schedules, budget, and effort requirements.>

Activities that will occur for the testing of the Exam Generation Application will be unit tests to ensure each module of the application is working as expected. As the units pass testing, they will be integrated and system tests will be performed to ensure full functionality is present. This test plan will deliver summaries of all tests performed, sample data that will be used during testing, and the procedures used during testing. Major milestones include the success of all unit tests and the success of final system tests.

## 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. <This may appear as a specific document (such as a Test Specification), or it may be part of the organization's standard test approach. For each level of testing, there should be a test plan and an appropriate set of deliverables. The test strategy should be clearly defined and the Software Test Plan acts as the high-level test plan. Specific testing activities will have their own test plan. Refer to section 5 of this document for a detailed list of specific test plans.>

Specific test plan components include:

* Purpose for this level of test,
* Items to be tested,
* Features to be tested,
* Features not to be tested,
* Management and technical approach,
* Pass / Fail criteria,
* Individual roles and responsibilities,
* Milestones,
* Schedules, and
* Risk assumptions and constraints.

For the Exam Generation Application, two types of testing will be used. Unit testing will be performed on each component as it is completed. System testing will be performed to ensure the system’s full functionality is present and works as intended. The features that will be tested for the Exam Generation Application will be its ability to generate exams based on various inputs. Additionally, error handling will be tested to ensure that the application can handle input data that is not expected. Each member of the team will perform some unit tests on components that he was not responsible for creating.

## Scope

<Specify the plans for producing both scheduled and unscheduled updates to the Software Test Plan (change management). Methods for distribution of updates shall be specified along with version control and configuration management requirements must be defined.>

Any changes made to the Software Test Plan will be viewable through GitHub (the repository being utilized). Additional scheduled updates are not expected due to the scope of this project; unscheduled updates will be viewable as a change to this document in the repository.

## Reference Material

<Provide a complete list of all documents and other sources referenced in the Software Test Plan. Reference to the following documents (when they exist) is required for the high-level test plan:

* Project authorization,
* Project plan,
* Quality assurance plan,
* Configuration management plan,
* Organization policies and procedures, and Relevant standards.>

This Software Test Plan may reference the Software Design Document and the Software Requirements Sheet of the Exam Generation Application.

## Definitions and Acronyms

(Specify definitions of all terms and agency acronyms required to properly interpret the Software Test Plan.)

STP: Software Test Plan

EGA: Exam Generation Application

# Test Items

<Specify the test items included in the plan. Supply references to the following item documentation:

* Requirements specification,
* Design specification,
* Users guide,
* Operations guide,
* Installation guide,
* Features (availability, response time),
* Defect removal procedures, and
* Verification and validation plans.>

## Program Modules

<Outline testing to be performed by the developer for each module being built.>

## User Procedures

<Describe the testing to be performed on all user documentation to ensure that it is correct, complete, and comprehensive.>

## Operator Procedures

<Describe the testing procedures to ensure that the application can be run and supported in a production environment (include Help Desk procedures).>

# Features

## Features to be Tested

<Identify all software features and combinations of software features to be tested. Identify the test design specifications associated with each feature and each combination of features.>

Features of the EGA that will be tested include its ability to generate various exams based on different inputs (different inputs will result in different outputted exams). Additionally, the application’s ability to handle incorrect inputs will be tested.

## Features Not to be Tested

<Identify all features and specific combinations of features that will not be tested along with the reasons.>

As of creation, all intended functionality of the Exam Generation Application will be tested through unit and system testing.

# Approach

<Describe the overall approaches to testing. The approach should be described in sufficient detail to permit identification of the major testing tasks and estimation of the time required to do each task. Identify the types of testing to be performed along with the methods and criteria to be used in performing test activities. Describe the specific methods and procedures for each type of testing. Define the detailed criteria for evaluating the test results.>

<For each level of testing there should be a test plan and the appropriate set of deliverables. Identify the inputs required for each type of test. Specify the source of the input. Also, identify the outputs from each type of testing and specify the purpose and format for each test output. Specify the minimum degree of comprehensiveness desired. Identify the techniques that will be used to judge the comprehensiveness of the testing effort. Specify any additional completion criteria (e.g., error frequency). The techniques to be used to trace requirements should also be specified.>

## Component Testing

<Testing conducted to verify the implementation of the design for one software element (e.g., unit, module) or a collection of software elements. Sometimes this is called unit testing. The purpose of component testing is to ensure that the program logic is complete and correct and ensuring that the component works as designed.>

Component testing will ensure that all modules accept inputs correctly, perform their intended function and create appropriate outputs. Component testing will be performed to ensure all units are acceptable and complete before integration is performed.

## Integration Testing

<Testing conducted in which software elements, hardware elements, or both are combined and tested until the entire system has been integrated. The purpose of integration testing is to ensure that design objectives are met and ensures that the software, as a complete entity, complies with operational requirements. Integration testing is also called System Testing.>

Integration testing will be performed once all modules are created and component tests completed. The objective during integration testing is to ensure that all functionality is working as intended. Testing will include using appropriate inputs to ensure correct outputs are generated, using incorrect inputs to ensure errors are thrown and the end-user notified, and ensuring the performance of the entire system is within satisfactory boundaries.

## Interface Testing

<Testing done to ensure that the application operates efficiently and effectively outside the application boundary with all interface systems.>

As the Exam Generation Application is a stand-alone system with no external interfaces, no interface testing will be performed.

## Regression Testing

<Testing done to ensure that that applied changes to the application have not adversely affected previously tested functionality.>

As only one integration occurs, no in depth regression testing will occur as all functionality will be maintained in the system test.

## Acceptance Testing

<Testing conducted to determine whether or not a system satisfies the acceptance criteria and to enable the customer to determine whether or not to accept the system. Acceptance testing ensures that customer requirements' objectives are met and that all components are correctly included in a customer package.>

The performance of acceptance testing will be to ensure that the correct output files are being generated from the Exam Generation Application. This will be similar to system testing being performed.

## Beta Testing

<Testing, done by the customer, using a pre-release version of the product to verify and validate that the system meets business functional requirements. The purpose of beta testing is to detect application faults, failures, and defects.>

For the Exam Generation Application, no outside beta testing will be performed.

# Pass/Fail Criteria

<Specify the criteria to be used to determine whether each item has passed or failed testing.>

## Suspension Criteria

<Specify the criteria used to suspend all or a portion of the testing activity on test items associated with the plan.>

If a test item fails to complete its specified test for any reason, it will be suspended for correction. A bug report will be created and the developer assign to investigate the issue.

## Resumption Criteria

<Specify the conditions that need to be met to resume testing activities after suspension. Specify the test items that must be repeated when testing is resumed.>

Once a developer has marked a bug as in progress and then has “fixed” the bug, the test item will be moved to resumption status.

## Approval Criteria

<Specify the conditions that need to be met to approve test results. Define the formal testing approval process.>

Only if all tests have been passed will an item be approved and has successfully passed testing.

# Testing Process

<Identify the methods and criteria used in performing test activities. Define the specific methods and procedures for each type of test. Define the detailed criteria for evaluating test results.>

## Test Deliverables

<Identify the deliverable documents from the test process. Test input and output data should be identified as deliverables. Testing report logs, test incident reports, test summary reports, and metrics' reports must be considered testing deliverables.>

## Testing Tasks

<Identify the set of tasks necessary to prepare for and perform testing activities. Identify all intertask dependencies and any specific skills required.>

## Responsibilities

<Identify the groups responsible for managing, designing, preparing, executing, witnessing, checking, and resolving test activities. These groups may include the developers, testers, operations staff, technical support staff, data administration staff, and the user staff.>

## Resources

<Identify the resources allocated for the performance of testing tasks. Identify the organizational elements or individuals responsible for performing testing activities. Assign specific responsibilities. Specify resources by category. If automated tools are to be used in testing, specify the source of the tools, availability, and the usage requirements.>

## Schedule

<Identify the high level schedule for each testing task. Establish specific milestones for initiating and completing each type of test activity, for the development of a comprehensive plan, for the receipt of each test input, and for the delivery of test output. Estimate the time required to do each test activity.>

<When planning and scheduling testing activities, it must be recognized that the testing process is iterative based on the testing task dependencies.>

# Environmental Requirements

<Specify both the necessary and desired properties of the test environment including the physical characteristics, communications, mode of usage, and testing supplies. Also provide the levels of security required to perform test activities. Identify special test tools needed and other testing needs (space, machine time, and stationary supplies. Identify the source of all needs that is not currently available to the test group.>

## Hardware

<Identify the computer hardware and network requirements needed to complete test activities.>

## Software

<Identify the software requirements needed to complete testing activities.>

## Security

<Identify the testing environment security and asset protection requirements.>

## Tools

<Identify the special software tools, techniques, and methodologies employed in the testing efforts. The purpose and use of each tool shall be described. Plans for the acquisition, training, support, and qualification for each tool or technique.>

## Publications

<Identify the documents and publications that are required to support testing activities.>

## Risks and Assumptions

<Identify significant constraints on testing such as test item availability, test resource availability, and time constraints. Identify the risks and assumptions associated with testing tasks including schedule, resources, approach and documentation. Specify a contingency plan for each risk factor.>

# Change Management Procedures

<Identify the software test plan change management process. Define the change initiation, change review, and change authorization process.>

# Plan Approvals

<Identify the plan approvers. List the name, signature and date of plan approval.>

# Test Procedures

<Fill out the following table for each test procedure.>

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Procedure Number:** | |  | | |
| **Date Tested:** | |  | | |
| **Test Performed By:** | |  | | |
| **Project Name:** | |  | | |
| **Software Version:** | |  | | |
| **Related Requirements:** | |  | | |
| # | Test Step Description | | Expected Result | Passed |
|  |  | |  |  |
|  |  | |  |  |
|  |  | |  |  |
|  |  | |  |  |
|  |  | |  |  |
|  |  | |  |  |
|  |  | |  |  |
|  |  | |  |  |
|  |  | |  |  |
|  |  | |  |  |
|  |  | |  |  |
|  |  | |  |  |
|  |  | |  |  |
|  |  | |  |  |
|  |  | |  |  |
|  |  | |  |  |
|  |  | |  |  |
|  |  | |  |  |
|  |  | |  |  |
|  |  | |  |  |
|  |  | |  |  |
|  |  | |  |  |
|  |  | |  |  |
|  |  | |  |  |
|  |  | |  |  |

# Requirements Matrix

*<Provide a cross reference that traces the test procedures to the requirements in your SRS document. Use a tabular format to show which tests satisfy each of the functional requirements from the SRS. Refer to the functional requirements by the numbers/codes that you gave them in the SRS.>*