**1. Introduction**

**1.1. Statement of Purpose**

This document serves as the draft test approach for the software development project steganography. It sets the scope of various tests to be conducted, the activities to be completed, the general resources required, the methods and processes to be used to test before the product is delivered to the customer.

**1.2. Overview of the document**

Section 1 describes the purpose, overview, and test types of the test plan document. Section 2 describes scope and objectives of the test plan, acceptance testing, system testing, integration testing and unit testing. Section 3 describes the test schedule followed for the project. Section 4 describes the human resources and hardware resources used for the project. Section 5 describes the recording procedures during the testing phase, test report forms and the results of testing.

**1.3. Test types**

**1.3.1. User acceptance testing**

This testing is based on the business requirements of the project. As we are following V-shape waterfall model for steganographic project, there will be two phases validation phase and verification phase. Both the phases will be running parallel, testing will be done by tester. User acceptance testing will be done once the business requirement specification phase is done. It enables the customer to determine whether to accept the system or not.

**1.3.2. System testing**

Once the system specification phase is completed, we get a basic working model for which the system testing is performed, the software is put in place into the working environment and a “system test” is applied. It ensures that system meets all functional and business requirements. In this phase the system interacts with actual user input such as an Image, encrypted key.

**1.3.3. Integration testing**

This is a black box testing in which individual software modules are combined and tested as group and it is performed after system testing. Testing of more than one unit together to determine if they function correctly. This testing is conducted after the architecture design phase. This testing is done by developers/designers and testers in collaboration.

**1.3.5. Unit testing**

This is a white box testing which is conducted to find incorrect code in each module. This type of testing is generally done by the programmers. This is the most micro scale of testing. In developing this project V-shape waterfall model is used, unit testing is done once the coding is done so that the errors can be detected. This testing ensures that the component is working according to the detailed design/build specifications of the module. This testing is also known as component module/program testing.

**1.4. Testing Process**



**[2]**

**2. Scope and objectives**

As mentioned in the requirements document, Steganography project performs all the required operations according to the customer requirements. Outlined below are the test types that will be performed on the software. The entire test plans and conditions will be developed from the requirements document and project design document.

**2.1. User acceptance testing**

The requirements of a specification are reviewed. It determines whether a system satisfies its acceptance criteria and business requirements or not. This testing is also known as beta testing, application testing or end user testing.

**Entrance criteria**- Business requirement specification phase has to be completed and we should have whole requirements according to customer.

**Exit criteria**- Real business users enable whether to accept the system or not.

**2.2. System testing**

This test intends to prove that the functionality delivered by the team is as specified by the team in system specification document. It also assesses the quality of the software and ensures that the software will successfully work according to the intended requirements by the customer. The testing strategy for the validation of the system as a whole will also utilize the “black box” method. All possible user input must be examined and any deficiencies addressed.

**Entrance Criteria** – All modules and classes are implemented; user acceptance testing is performed. The system is put into actual working environment.

**Exit Criteria** – All major defects and errors from system testing must be fixed and reported.

**2.3. Integration testing** This test proves that all areas of the system interface with each other correctly and that there are no gaps in the data flow. Final integration test proves that system works as an integrated unit when all the fixes are complete. The actual testing method used for this phase will be the “Black box” method.

**Entrance Criteria** – High level design phase has to be completed and system testing to be conducted on at least one component.

**Exit Criteria** – All important errors from Integration tests must be fixed and tested.

**2.4. Unit testing**

The software implementation process is reviewed for every module. Every component is subjected to undergo unit testing where white box testing, also known as “basic path testing” is done. If necessary, the testing schedule will correspond to the completion to a system module. The objective of these reviews is to ensure correctness and to test the functional integrity within each individual component. Issues to consider are matching of parameters and their data types, arguments, relative attributes.

**Entrance Criteria** – At least one component should be coded and there should be at least 100 lines of code for the testing. A meeting is conducted after the testing is done to make a schedule of the further process and also the bug fixing.

**Exit Criteria** – All errors identified should be noted and if possible fixed.

**3. Test schedule**

For the Testing phase, there should be a proper working schedule. The following schedule will apply

Requirements document testing: 4 days

System testing: 5 days

System Integration testing: 9 days

Unit testing: 17 days

**4. Resources**

**4.1. Human**

The Tech crew team consists of

Vasantha Gundeti

Divya Thalla

Krishna Bakka

Surya Nadella

Krishna Bakka and Vasantha Gundeti will coordinate and work as the development team and co-ordinate the unit testing phase of the project.

Surya Nadella and Divya Thalla will combine into a team and coordinate the Integration testing and System testing phases of the project.

Formal reviews and also other test phases to be coordinated by Surya Nadella and Divya Thalla.

**4.2. Hardware**

The project development environment will be as follows:

**Hardware** will include a personal (desktop or notebook) computer with the following specifications:

* Dual core processor.
* Memory of 1 GB RAM.
* Hard Disk of 40 GB
* a CD-RW Drive

**Software** will include:

* Windows7
* MS Project
* Eclipse

**5. Recording procedures**

Once after the system tests, Integration tests and unit tests are performed, defects and errors will be recorded as they are detected. All the errors will be categorized accordingly and will be noted. If there is a small error, then the testing team will fix the defects. If the error is too complex, then those errors are handed over to the development team during the team meetings.

**5.1. Test Report Forms**

A Test Report is documented at the end of every testing. The Report generated will be thoroughly reviewed in the team meetings. A sample Test report that records some errors during the testing is given in the System Test Cases 5.2.1.

**5.2. User Acceptance test results**

All the requirements of the customer are identified and customer should be able to perform any test based on their business processes. If customer is not satisfied with the requirements, all the errors have to be corrected and requirements have to be taken down to correct them.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test case | Input | Expected output | Actual output | Error message |
| Not encryption is taking place | dog.png, 1234567891011121, Hello | dog.png(stegoImage) | Just Plain Image | Message not encrypted |
| If key is not provided for encryption | dog.png," ", hello | Error message | Error message | No key provided |
| If key is not provided for decryption | dog.png, " " | Error message | error message | No key provided |
| If wrong key is provided | dog.png, 123456ljjffhngnm,"hello" | dog.png,@##$\* | dog.png, hello | none |
| If no message is provided for encryption | dog.png, 1234567891011121," " | dog.png, hello | dog.png | Please provide message |
| If no image is provided for encryption | " ",1234567891011121,"hello" | dog.png, hello | none | Please provide image |
| If no image is provided for decryption | " ",1234567891011121,"jeepers" | dog.png, hello | none | Please provide image |
| If key less than 16 bit | Dog.png,fdh3435263,"hello" | Dog.png, hello | none | Invalid key |
| If key greater than 16 bit | Dog.png,fdh3435263472472475,"hello" | Dog.png, hello | none | Invalid key |

**5.3. System test results**

All the bugs identified while applying each of the following test cases during System test along with category of the error and action taken to correct the error will be documented.

**5.3.1** The table describes about the test cases during system test along with the category of error and action taken to correct the error.

**5.4. Integration test results**

All the errors recorded during Integration test phase are taken down and documented.

**5.5. Unit test results**

All the errors recorded during unit tests of each component along with the action taken to correct the error will be documented.

**6. References**

[1]. Pressman Roger S., “Software Engineering – A Practitioner’s Approach 7th edition”.

[2]. SommervilleI.,“Software Engineering 9th edition”.

[3]. Stringfellow, Dr.Catherine.