Database EvaluatOr

Test plan

Document history

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Version | Date | Author | Changes | Valid from |
| 1.0 | 14-Sep-2016 | Changming Wu | Draft |  |
| 1.1 | 21-Sep-2016 | Changming Wu | Edit |  |

Table of Contents

[1 Introduction 3](#_Toc464122886)

[1.1 Objectives 3](#_Toc464122887)

[1.2 Team Members 3](#_Toc464122888)

[2 Scope 4](#_Toc464122889)

[2.1 Inclusions 4](#_Toc464122890)

[2.2 Exclusions 4](#_Toc464122891)

[3 Risks 5](#_Toc464122892)

[4 Test Approach 5](#_Toc464122893)

[4.1 Unit Test 5](#_Toc464122894)

[4.2 Integration Test 6](#_Toc464122895)

[4.3 System Test 6](#_Toc464122896)

[4.4 Functional Test 6](#_Toc464122897)

[4.4.1 Function 6](#_Toc464122898)

[4.4.2 Test items 6](#_Toc464122899)

[4.4.3 Help Documentation Test 7](#_Toc464122900)

[4.5 Business Test 7](#_Toc464122901)

[4.6 Installation test 7](#_Toc464122902)

[4.6.1 The instruction of installation test 7](#_Toc464122903)

[4.6.2 Installation test methods and standards 7](#_Toc464122904)

[5 Test Environment 8](#_Toc464122905)

[6 Milestones / Deliverables 9](#_Toc464122906)

[6.1 Test Schedule 9](#_Toc464122907)

[6.2 Deliverables 10](#_Toc464122908)

[7 Severity, Priority and Life Cycle of Defect 10](#_Toc464122909)

[7.1 Severity 10](#_Toc464122910)

[7.2 Priority 11](#_Toc464122911)

[7.3 Defect/Bug Life Cycle 11](#_Toc464122912)

[8 Exit criteria 12](#_Toc464122913)

# Introduction

The Test Plan of Database Evaluator has been created to communicate the test approach to team members. It includes the objectives, scope, schedule, risks and approach. This document will clearly identify what the test deliverables will be and what is deemed in and out of scope.

## Objectives

* Client application

Client application is small and light weight application which will be executed by customer on their database system and application generated encrypted dump file which contains all the agreed database and SQL instance parameter values.

* Website

Customer can create their account on the website.

Customer can download Client application from the website.

Customer also can use FTP to upload dump file and download the final report.

* Database Evaluator

Database Evaluator (DBE) can generate the final diagnosed report.

Through testing, to achieve the following objectives:

* Product testing has been implemented to meet the design requirements, including: whether to implement each function point, the business process is correct.
* Operation and operation of the provisions of the stability of the product.
* Bug number and the defect rate is controlled within the acceptable range.

## Team Members

|  |  |
| --- | --- |
| **Resource Name** | **Role** |
| Patrick Cura | C# Developer/ Tester |
| Hardik Kansara | Database Specialist/ Tester |
| Kwinno Pineda | Website Developer/ Tester |
| Changming Wu | Project Manager / Tester Manager |

# Scope

All the defined inclusion scope has in the test plan will be tested during functional testing. All exclusion scope will not be tested.

## Inclusions

* Client application
  + Install
  + Uninstall
  + Execute on user’s system
  + Collect/check the parameters of user’s database
  + Generate the dump files of parameters of customers’ database
* Website
  + User: register
  + User/ Administrator: login in
  + User: forgot password
  + User: download client application form website
  + User: upload the encrypted dump files via website
  + User: view and download the final diagnosed report
  + Administrator: upload the final diagnosed report
* Database Evaluator
  + Quote the encrypted dump files
  + Generate the final diagnosed report

## Exclusions

* This project will not support the all type of operating system except Microsoft Windows.
* This project will not support the all type of database except Microsoft SQL 2008 R2 and Microsoft SQL 2012.

# Risks

The following risks have been identified and the appropriate action identified to mitigate their impact on the project. The impact (or severity) of the risk is based on how the project would be affected if the risk was triggered. The trigger is what milestone or event would cause the risk to become an issue to be dealt with.

* The situation bug fixes
* The achievement of functional modules
* The achievement of the overall system functions
* Write quality code
* Staff experience and familiarity with the software
* Developers, testers agreed on the implementation of the project
* Personnel adjustments lead to the development cycle delay
* Shorten development time lead to some of the test plans cannot be executed

# Test Approach

The project plans to complete the following types of test:

* Unit test
* Integration test
* System test
* Functional test
* Business test
* Installation test

## Unit Test

Unit test is the practice of testing small pieces of code, typically individual functions, alone and isolated. Typically done by the developer sand not by testers, as it requires detailed knowledge of the internal program design and code.

Unit test should be fairly simple to write. A unit test should essentially just give the function that is tested some inputs, and then check what the function outputs is correct. It may require developing test driver modules or test harnesses.

Developers will take duty of the unit test of this project for each phase.

## Integration Test

Testing of integrated modules to verify combined functionality after integration. Typically done by developers or by testers.

Integration test is about to test how parts of the system work together – the integration of the parts. Integration test is similar to unit test, but there is one big difference.

Integration test is often slower than unit test because of the added complexity. It also might need some set up or configuration, such as the setting up of a test database. For example, a database and application is working together correctly, and that calls for an integration test. As a result, when validating integration test results, you could for example validate a database related test by querying the database to check the database state is correct.

Integration tests can usually be written with the same tools as unit tests.

Developers will take duty of the unit test of this project for each phase.

## System Test

System test is tested as per the requirements that is based on overall requirements specifications/system analysis document, covers all combined parts of a system to test all requirements.

System test is the first level in which the complete application is tested as a whole. The goal at this level is to evaluate whether the system has complied with all of the outlined requirements. System test is undertaken by independent testers who have not played a role in developing the program. This test is performed in an environment that closely mirrors production.

System Test is very important because it verifies that the application meets the technical, functional, and business requirements that were set by the customer.

## Functional Test

### Function

The system can be realized in accordance with the design requirements of each module, data should be complete, interface should be artistic and operate easily.

Specific reference in this document to test the focus and order parts.

### Test items

* Alphanumeric data entry is correctly echoed, and entered into the system?
* Are graphics mode data items (such as a slider) working properly?
* Are able to identify illegal data?
* Could the data input message be understandable or not?

### Help Documentation Test

* Whether the document is an accurate description of how to use various usage patterns?
* Describe the interaction of the order is accurate?
* Are examples accurate?
* Terms, description of the menu system, and the response are consistent with the actual program?
* Can be easily positioned in the document guide?
* Is able to easily use the document to exclude errors?
* Content and index documents if accurate and complete?
* Design documentation (layout, indents, and graphics) whether the information easy to understand?
* Error message displayed to the user if there is a more detailed explanation of the document?
* If using hyperlinks, hyperlinks if accurate and complete?

## Business Test

After the completion of functional test to do the business test, business test focuses on business process and data flow from a module of software correctness in the process of flow to another module.

## Installation test

### The instruction of installation test

In addition to the embedded software, installation is the first step to realize its function software products. If the software cannot be installed correctly, it will not be executed, so the installation test is particularly important.

### Installation test methods and standards

* Automatic installation or manual configuration installation, testing various combinations of installation, and verify the various groups
* Together correctness, the ultimate goal is for all combinations can be installed successfully.
* After installation exit, confirm that the application can properly up and running.
* Uninstall test and installation test is equally important, if the system provides automatic uninstall tool, you need to test whether the system is to delete all the files after uninstalling all, whether the registration information about the registry is also removed.
* At least to be installed on a laptop in the test, because there are many products will have problems in your notebook, especially system-level products. (Conditional case)
* After the installation is complete, you can then simply use the execution after the unloading operation, some systems after use will change, become unavailable uninstalled.
* Installation time is reasonable;
* For the client-server model of application system, you can install the client, and then install the server side, to test whether there will be problems.
* Examine whether the installation of the system's impact on other applications, in particular Windows operating system, such problems often occur.

# Test Environment

* Operating system
* Microsoft Windows 7 and 10
* Windows Server 2008 R2
* Windows Server 2012

Client application and database evaluator must be run on windows environment only and used by customer.

* Database type and version
* SQL Server 2008 R2
* SQL Server 2012
* Internet
* Other software

# Milestones / Deliverables

## Test Schedule

|  |  |  |  |
| --- | --- | --- | --- |
| **Task Name** | **Start** | **Finish** | **Comments** |
| **Test Plan** | 19/09/16 | 21/09/16 |  |
| **Unit test** |  |  |  |
| Website: Exploration Phase - Iteration One | 13/08/16 | 13/08/16 |  |
| Website: Exploration Phase - Iteration Two | 27/08/16 | 27/08/16 |  |
| Website: Exploration Phase - Iteration Three | 03/09/16 | 03/09/16 |  |
| Application: Exploration Phase - Iteration One | 13/08/16 | 13/08/16 |  |
| Application: Exploration Phase - Iteration Two | 27/08/16 | 27/08/16 |  |
| Application: Exploration Phase - Iteration Three | 03/09/16 | 03/09/16 |  |
| **Integration test** |  |  |  |
| Website: Engineering Phase - Iteration One | 10/09/16 | 10/09/16 |  |
| Website: Engineering Phase - Iteration Two | 17/09/16 | 17/09/16 |  |
| Website: Engineering Phase - Iteration Three | 24/09/16 | 24/09/16 |  |
| Application: Engineering Phase - Iteration One | 10/09/16 | 10/09/16 |  |
| Application: Engineering Phase - Iteration Two | 17/09/16 | 17/09/16 |  |
| Application: Engineering Phase - Iteration Three | 24/09/16 | 24/09/16 |  |
| **System test** |  |  |  |
| System plan | 19/09/16 | 21/09/16 |  |
| Test case | 22/09/16 | 24/09/16 |  |
| Iteration One | 26/09/16 | 27/09/16 |  |
| Iteration Two | 29/09/16 | 30/09/16 |  |
| Iteration Three | 01/10/16 | 03/10/16 |  |
| Iteration Four | 05/10/16 | 06/10/16 |  |
| Test report | 09/10/16 | 09/10/16 |  |

* Base on the situation of system test, project team will add more iteration for system test.

## Deliverables

|  |  |
| --- | --- |
| **Deliverable** | **Date / Milestone** |
| Test Plan | 21/09/16 |
| Test Case(system test case) | 24/09/16 |
| Test report |  |
| Website: Unit test - Iteration One | 13/08/16 |
| Website: Unit test - Iteration Two | 27/08/16 |
| Website: Unit test - Iteration Three | 03/09/16 |
| Application: Unit test - Iteration One | 13/08/16 |
| Application: Unit test - Iteration Two | 27/08/16 |
| Application: Unit test - Iteration Three | 03/09/16 |
| Integration test |  |
| Website: Integration test - Iteration One | 10/09/16 |
| Website: Integration test - Iteration Two | 17/09/16 |
| Website: Integration test - Iteration Three | 24/09/16 |
| Application: Integration test - Iteration One | 10/09/16 |
| Application: Integration test - Iteration Two | 17/09/16 |
| Application: Integration test - Iteration Three | 24/09/16 |
| System test report | 09/10/16 |

# Severity, Priority and Life Cycle of Defect

|  |  |  |
| --- | --- | --- |
| Severity |  |  |
| **ID** | **Level** | **Definition** |
| A | Blocker | The system cannot be executed, crashes or a serious shortage of resources, the application module cannot start or abnormal exit, unable to test, resulting in system instability. |
| B | Critical | The impact of system functions or operations, the main function there are serious flaws, but will not affect the system stability. |
| C | Major: | The user does not implement the basic functions of the implementation, there is a reasonable way to correct, the user can perform basic operations. |
| D | Minor | The operator is not convenient to use, but does not affect the function of the realization. Can improve the function. |
| E | Cosmetic | The ease of use and the proposed features |
|  |  |  |
| Priority |  |  |
| **ID** | **Level** | **Definition** |
| A | Immediate | "Immediately resolve" that the problem must be resolved immediately, otherwise the system simply cannot meet the scheduled needs. |
| B | Urgent | Urgent need to solve, said the repair of the problem is very important, very urgent, related to the system's main function module can be normal. |
| C | High | "A high degree of attention", that there is time to be resolved immediately, or the system deviation from the larger demand or scheduled functions cannot be normal. |
| D | Normal | Normal processing into the personal plan to solve, said the problem does not affect the realization of demand, but affect other aspects of use. |
| E | Low | Low priority, that is, the problem must be confirmed before the release of the system can be resolved or confirmed to be resolved. |

## Defect/Bug Life Cycle

|  |  |
| --- | --- |
| **Status** | **Definition** |
| New | When a new defect is logged and posted for the first time. It is assigned a status NEW. |
| Assigned | Once the bug is posted by the tester, the lead of the tester approves the bug and assigns the bug to developer team. |
| Open | The developer starts analyzing and works on the defect fix |
| Fixed | When developer makes necessary code change and verifies the change, he or she can make bug status as "Fixed." |
| Pending retest | Once the defect is fixed the developer gives particular code for retesting the code to the tester. Since the testing remains pending from the testers end, the status assigned is "pending request." |
| Retest | Tester does the retesting of the code at this stage to check whether the defect is fixed by the developer or not and change the status to "Re-test." |
| Verified | The tester re-tests the bug after it got fixed by the developer. If there is no bugdetected in the software, then the bug is fixed and the status assigned is "verified." |
| Reopen | If the bug persists even after the developer has fixed the bug, the tester changes the status to "reopened". Once again the bug goes through the life cycle. |
| Closed | If the bug is no longer exits then tester assign the status "Closed." |
| Duplicate | If the defect is repeated twice or the defect corresponds the same concept of the bug, the status is changed to "duplicate." |
| Rejected | If the developer feels the defect is not a genuine defect than it changes the defect to "rejected." |
| Deferred | If the present bug is not of a prime priority and if it is expected to get fixed in the next release, then status "Deferred" is assigned to such bugs. |
| Not a bug | If it does not affect the functionality of the application then the status assigned to a bug is "Not a bug". |

# 

# Exit criteria

* The functionality and performance of product has met the requirements of the product requirements specification (requirement analysis document).
* The error found in the system testing has been modified and the defect repair rate levels up to standard.
* The defect rate (Defect Rate): defect ratio after a full unit test cases found in less than 5%.
* Defect density (Defect Density): per unit length (typically 1000-bit line) code, or FP (Function Point) number of defects found in less than 5%.
* Test Coverage (Test Coverage): the degree of test code or use case is covered 100%.