Master test plan checklist

**Client:** *[Client name]*

**Project:** *[Project Name]*

# Document history

|  |  |  |
| --- | --- | --- |
| **Version** | **Date** | **Author** |
| *Draft 1.0* |  |  |

# Testing schedule/Resources/Status

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Milestone/Test phase** | **Duration (days)** | **Resources** | **Date(s)** | **Status** |
| Unit testing |  |  |  |  |
| Functional testing |  |  |  |  |
| Load testing |  |  |  |  |
| Volume testing |  |  |  |  |
| Acceptance testing |  |  |  |  |
| Usability testing |  |  |  |  |

# High level test objectives

*[Adapt as needed.]*

* *To ensure that the [work product] satisfies all project requirements.*
* *To ensure that all components of the [work product] function according to design.*
* *To ensure that all use case scenarios can be executed successfully.*
* *To ensure that the [work product] can perform under the anticipated user load.*
* *To determine if the application is intuitive and easy to use, and if it presents the users with the intended user experience.*

# Milestones/test phases

*[The items below should be covered in a Master Test Plan. Format will depend on subject matter and test needs for each project or product.]*

## Unit testing

*Unit tests are basic tests at the module level to ensure that a given function works. Unit testing is generally performed by the programmer or developer while building a system.*

* Environment
* URL
* Access Instructions
* User ID/password
* Components (for each component provide a numbered test plan with the following info):
  + Test set up and preconditions
  + Step by step test instructions (including sample input data)
  + Expected outcome
  + Actual outcome

## Functional testing

*Functional tests are conducted to ensure the system behaves according to the functional requirements. Use cases are developed to validate that all functions outlined in the requirements are present in the system and work as intended. This testing should not be performed by the people who built the system.*

* Environment
* URL
* Access Instructions
* User ID/password
* Function Points (for each function point a numbered test plan with the following info):
* Test set up and preconditions
* Step by step test instructions (including sample input data)
* Expected outcome
* Actual outcome
* Be sure in all cases to check browser compliance, error checking, data population, business logic and full functionality)

## Load testing

*Load tests put the application under heavy loads, such as testing of a Web site under a range of loads to determine at what point the system's response time degrades or fails. Automated testing tools are used to conduct this type of test.*

* Environment
* URL
* Access Instructions
* User ID/password
* Methodology
* Tools
* Load assumptions/targets
* Outcome

## Volume testing

*Volume tests subject a system to a high volume of data in order to determine how many transactions can be supported by the system. Automated testing tools are used to conduct this type of test.*

* Environment
* URL
* Access Instructions
* User ID/password
* Methodology
* Tools
* Stress assumptions/targets
* Outcome

## Acceptance testing

*Testing by the client or sponsor to confirm that the system meets all requirements and is ready for operational use.*

* Environment
* URL
* Access Instructions (additional instructions needed for client)
* User ID/password
* Use cases (for each use case provide a numbered test case with the following info):
* Test set up and preconditions
* Step by step test instructions (including sample input data)
* Expected outcome
* Actual outcome

## Usability testing

*Testing the ease with which users can learn and use the system.*

* Environment
* URL
* Access Instructions
* Test subjects
* Test Instructions (for testers)
* Test results questionnaire