ERPNext Software Testing

PROJECT MANAGEMENT PLAN

Aradhana Nandhikeswaran, 2861686

1. INTRODUCTION
   1. purpose

This document provides an overview of the testing plan for the ERPNext software system, which is available on GitHub. ERPNext is an all-in-one software system that covers various functions, including accounting, warehouse management, and customer relations management. The system is built on the Frappe Framework, which uses Python and JavaScript. To ensure the testing project's effectiveness, it is important to limit its scope, especially when dealing with complex software like ERPNext. This document will explain the project's structure, assumptions, limitations, and schedule.

* 1. CONSTRAINTS

To ensure a comprehensive testing of software, several factors including available resources, intended purpose of the software, and development practices are taken into consideration. However, the process is often constrained by various assumptions and limitations that restrict the extent to which it can be conducted.

In this project, it is assumed that the team has knowledge of software development and is familiar with software testing tools and concepts. The testing will be conducted on an 8GB laptop which may restrict the type of testing tools that can be used.

Additionally, the project is limited to using open-source testing tools, which may restrict the level of testing that can be achieved.

1. SCOPE MANAGEMENT

The objective of this project is to assess the software for any errors, deficiencies, or unfulfilled requirements by conducting a comprehensive testing process. The scope of the testing includes various aspects of the software, such as ease of installation and usability, graphical interfaces, and each of the ERPNext services (excluding customized services) to ensure that all requirements are met.

Additionally, the testing will evaluate the software's adherence to Object Oriented Programming concepts and software principles to identify any programming gaps. Furthermore, the security of the software will be assessed by testing APIs and verifying its resilience against threats such as Cross-Site Scripting.

1. WORK BREAKDOWN STRUCTURE

The project will be conducted in the following phases:

1. Testing requirements specification
2. Planning

In this test functions to test, test design methods and test schedules will be planned. The requirements will also be redefined according to the plan designed.

1. Selection of testing tools and methods
2. Test case specification and testing.

Testing will be done via the black box methodology, which will involve behavioral testing. Therefore, the test cases will be defined based on user experience and the general principles of software.

1. Reporting
2. SCHEDULE

| **Milestones** | **Estimated Completion Timeframe** |
| --- | --- |
| Requirements specification and Planning | 03/27/2023 |
| Tools and methods testing and Test case specification | 03/27/2023 |
| Testing | 03/27/2023-04/03/2023 |
| Reporting | 05/01/2023 |

1. **DEPENDENCIES**

* The development of a test plan requires the specification of requirements. It will be impossible to create a productive plan without clearly stated needs and objectives.
* Resource planning and test environment setup need the development of a test plan. It is impossible to locate the required resources and build up a suitable testing environment without a clearly defined test plan.
* Test Execution requires Resource Planning as a prerequisite. Effective testing cannot be done without sufficient resources.
* Test Execution requires the setup of the test environment. For accurate and trustworthy test results, a testing environment that has been correctly designed is necessary.
* Reporting is a requirement for test execution. Analysis and documentation of test results are required for stakeholders.

1. RISK MANAGEMENT and mitigation

* Inadequate resources
* Constant changing of requirements.
* Unfamiliarity with technologies

The following steps are taken to overcome the risks:

To ensure that the available resources are completely optimized, popular tools and techniques will be favored above other ones. This will also guarantee that technological unfamiliarity is reduced by their documentation.

To guarantee that the criteria stay flexible but fall inside a given scope, test cases will be designed with restricted scopes.