SAMS Project

Software Maintenance Plan

Thomas Travers
Grantham University
CS406

SAMS Project

Thomas Travers

Software Maintenance Plan

I.Description of Software Maintenance

Software maintenance is modifying a software system or component after delivery to correct faults, improve performance, add new capabilities, or adapt to a changed environment. (IEEE Standard 610.12-1991)

Software maintenance is very important for keeping a project fully functional in an ever evolving world and changing environment. Some software systems are supported decades after release making them legacy software. These software systems must undergo changes to make them compatible with newer hardware, operating systems, and associated components.

II.Factors that Mandate Change

We will mandate change when we need

- *Bug Fixes*
- *There is a change in operating environment*
- *Change in government policies and regulations*
- *Change in business procedures*
- *Changes to prevent future problems*

III.Maintenance Process Models

The IEEE process model would be best suited for this type of project. Here are the phases of the model...
Problem Identification / Classification, Analysis, Design, Implementation, Regression / System Testing / Acceptance Testing, and Delivery. This process model can fit in nicely with our agile techniques and allow us to continuously maintain our project in an efficient manner ensuring quality code along the way.

IV.Types of Software Maintenance

This project will employ Corrective maintenance, adaptive maintenance, perfective maintenance, and emergency maintenance. All of these types of maintenance are crucial to providing a smooth and functional experience to all users of the finished product as well as protect their user data.

V.Software Reverse Engineering

Software reverse engineering is a process that converts the code to recover the design, specification, and a problem statement (Kung, 2013). This could be useful in examining 3rd party components and designing better integration tools to work with them. The better we understand them, the better the system as a whole can be.

VI.Software reengineering

Software reengineering is a process that restructures a software system or component to improve certain aspects of the software (Kung, 2013). As stated in the previous section, this will be useful in developing better integration with third party components of the system.

Sources:

610.12 (H)-1990 - IEEE Standard Glossary of Software Engineering Terminology (HyperCard Stack). (n.d.). Retrieved August 03, 2020, from https://standards.ieee.org/standard/610_12(H)-1990.html

Kung, David. (2013). Object-Oriented Software Engineering: An Agile Unified Methodology (1st ed). New York, NY: McGraw-Hill Education
ISBN: 9781264056149