



California State University, Sacramento
College of Engineering and Computer Science
CSC 131-05: Computer Software Engineering

SOFTWARE EVOLUTION DOCUMENT

(Maintenance Phase - Waterfall Methodology)

for

PROJECT BRAVO

Prepared By: Sparsh Saraiya



Table Of Contents

1. Introduction & Objectives

1.1 Purpose, Scope, & Objective

2. Maintenance Activities

2.1 Corrective Maintenance

2.2 Adaptive Maintenance

2.3 Perfective Maintenance

3. Future Enhancements and Improvements

3.1 Proposed Feature Upgrades

3.2 Usability and Performance Improvements



CSC 131-05: Computer Software Engineering

Project Bravo: Software Evolution Document

Chapter 1: Introduction

1.1 Purpose, Scope, & Objective:

The goal of the Software Evolution phase is to show how the Expense Tracker app can be sustained and progressed after its original development and release. This phase guarantees the application continues to function as it was designed, is updated to function with new technologies, and evolves to meet the ongoing needs of future end-users. Although the project discussed in this paper is for academic purposes, it demonstrates principles that apply to modern software maintenance and evolution. The scope of this phase covers the ongoing maintenance and evolution of the software once the first release has occurred. This includes bug-fixing as issues become apparent after deployment, updating to run in new environments, and adding improvements that improve the current features in regards to usability and performance. The primary aim of software evolution is to preserve the stability of the system, correct defects discovered during the user testing phase, and meet improvement projects from either user input or new technology. The purpose of attempting to achieve these aims is to preserve the Toll Tracker's stability, efficient applications, and meet users' expectations well past the initial version. **This document matches the Maintenance phase of the Waterfall model**, following deployment and validation

Chapter 2: Testing Methods and Approach

2.1 Corrective Maintenance:

Corrective maintenance is the process of discovering and fixing problems or bugs that occur after the software has been released to production. All stages of testing for the Expense Tracker application were completed, but after release some users in the future may experience minor bugs after frequent use, such as calculation bugs, display errors, and minor GUI bugs. Corrective maintenance ensures that these bugs are resolved in a timely fashion to ensure system correctness and reliability.

2.2 Adaptive Maintenance:

Adaptive maintenance is the process of modifying the Expense Tracker so it continues to operate correctly in new or changing environments. As technology evolves, the Expense Tracker may require updates to support new operating systems, database versions, or new security standards. For instance, in the future, an adaptation may include upgrading the database to support increased data volume or reconfiguring the Expense Tracker to operate on another desktop platform. Adaptive maintenance ensures that the system continues to operate effectively and properly as technology advances.

2.3 Perfective Maintenance:



CSC 131-05: Computer Software Engineering

Project Bravo: Software Evolution Document

Perfective maintenance consists of improving features of the current capabilities of, or adding new features to, the software based on comments, feedback, and usage observations of users. The improvements to the software functionality will come from end-users making future suggestions or requests over an extended period. For example, the end-users may want enhanced financial reporting and analysis functionality, the ability to track budget goals, and the ability to create customized reports. Improvements from perfective maintenance will also include enhancements to the user interface of the software to improve usability, and it will include updates to improve overall performance with the software processing accounting transactions faster. Perfective maintenance has a focus on improving usage and efficiency of the software, so the software is able to continue to support the changing needs of users. The development team is also hopeful to include future improvements by adding AI driven features and improvement for system security.

Chapter 3: Future Enhancements and Improvements

3.1 Proposed Feature Upgrades:

In order to upgrade the Expense Tracker, there will be additional improvements that will improve the application's functionality and optimize the user experience. The development team plans to incorporate AI enhanced features which make the application work in a more customized manner for each end-user by being able to automatically categorize expenses, predict spending habits from user history, and provide better financial suggestions for users' spending styles. Another suggested improvement involves the improvements of the system's security. Future changes would involve stronger encryption of user data, multi-factor authentication, and role-based access control to further secure the end-users data. There would also need to be new features created such as cloud synchronicity that would allow a user to access their data from more than one device securely. The team would also like to add options to export data, such generating PDF reports, to allow users to analyze their financial data outside of the Expense Tracker application. All of these function upgrades would improve the functionality, flexibility, and overall utility of the Expense Tracker application for the long haul.

3.2 Usability and Performance Improvements:

Along with enhancements and new features, future Expense Tracker versions must look to advance general usability and performance. The user interface could also be refined to become more visually appealing and intuitive. The user interface layout and navigation designs could also be simplified to help users access key features quickly while maintaining a professional and consistent design so that new users and returning users would find the system easily usable. Performance improvements could also improve future application releases. The development team hopes to optimize the system's data processing and storage methods to ensure faster load times and to ensure smoother operation when managing larger amounts of financial data. These improved software reconstructions would make the application more responsive and more reliable even when the applied number of users grows and as records are stored. The application should continue to prioritize usability and added performance to ensure the Expense Tracker maintains a seamless experience that promotes user efficiency and satisfaction to ensure the application continues to be effective and relevant.