Library Management System(LMS) Configuration/Change Management Report

1   Introduction

Changes are inevitable during the development phase of a project. Changing circumstances, unexpected obstacles and requests can cause these changes. The most important part of this situation is how it affects the project.

The advantages and disadvantages of the changes should be compared to the project. This comparison should decide how the project will proceed.

It is important how the changes will affect time and cost. Some changes will have a positive impact, while others will have a negative impact. The project team should avoid these negative effects as much as possible.

2   Where Changes Originate

To realize a change, it is necessary to know where the change originated. The steps to be followed later may vary. The sources of change can be classified as planned development, unexpected problems, or enhancements.

Ideally, a change is due to planned software development. When making these changes, you should also look at time and money effects. As long as it is not demanded, even if it is beneficial, all the changes will consume resources and disrupt the flow.

3   Our Configuration / Change Management

All the changes made so far in our project have been made consciously. It is for development purposes, not for any errors. These changes are listed below.

Interface Change	Unlike the examples we prepared in the GUI prototype, the demo we published is slightly different. This is why the display looks nicer and the account type selection is done on the Log In screen. We used different visuals in different parts of the interface. The layout remained the same, but we made this change so that it does not get bored the user, and it looks nicer.
Adding a Librarian Actor	We added the librarian actor because we did not want to give full authority to the library staff. In this way, the authority of the library staff will be limited. Not every employee can be an administrator.
Adding Check Information Use-Case	The user had to return the book in the first version of the program in order to get information about the book the user bought (delay time, debt, date of receipt, etc.). To avoid this, Check Information use case, which can be used for each book, is defined.

Our policy to make changes;

- The changes do not affect the fulfillment of the requirements

- Try to keep the negative effect at a minimum level in terms of time and cost

- Do not make radical changes on the system

- Appropriate for the purpose of the system

- Does not negatively affect ease of use

- Does not negatively affect performance

- Not cause fault

4   References

http://www.drdobbs.com/software-change-management/184415707

Software Engineering-Sommerville-9th Edition, Chapter 25