Experiment No. 04

Project Planning Document

Software Project Management Lab

Final year, B.Tech IT

Sudhanshu, Rajan ,Amartya, Mahesh
sem - VII

Aug 30, 2022

Veermata Jijabai Technological Institute, Mumbai 19

Estimation Document

LOC based Estimation

The problems of lines of code (LOC)

1. Different languages lead to different lengths of code
2. It is not clear how to count lines of code
3. A report, screen, or GUI generator can generate thousands of lines of code in minutes
4. Depending on the application, the complexity of code is different

Function	**Estimated LOC**
Login/Register Interface	700
Database Management	200
User interface pages	1000
View functions	2000
Database Migrations	1000
Other code	1000
Testing	600
Total estimated LOC	6600

Risk Management Plan

Risk projection, also known as risk estimation, tries to rate each risk in two ways: the likelihood or probability that the risk is real, and the implications of the problems associated with the risk if it happens.

Four risk projection actions are carried out by the project planner, along with other managers and technical staff:

1. Create a scale that reflects the perceived possibility of a risk,
2. Define the risk's repercussions,

3. Estimate the risk's influence on the project and the product, and
4. Take note of the risk projection's overall accuracy to avoid misconceptions.

Risk Id	Risk	Risk Type	Probability	Impact
1	Project Delay	Project	30%	2
2	System Failure	Technical	40%	1
3	User Authentication(misuse of the system)	Technical	20%	2
4	Crossing the stated budget	Project	15%	3

Impact Scale:

Scale	Indication
1	catastrophic
2	critical
3	marginal
4	negligible

Time Management Plan (Schedule)

Defining Task Set for Software Project

A task set is the collection of work tasks, milestones, and deliverables. The grades are allotted as:

Grade	Type
0	Non incidental
1	Minimal
2	Low
3	Moderate

4	Substantial

Task Selector Value Computation:

Adaptation Criteria Grade	**Weight**	**Product**
Size of project 3	1.20	3.6
No. of potential users 4	1.10	4.4
Mission Criticality 4	1.10	4.4
Application endurance 3	1.20	3.6
Stability of Requirements 3	1.20	3.6
Ease of communication 3	0.90	2.7
Maturity of applicable technology 3	0.90	2.7
Performance Constraints 3	0.80	2.4
Embedded/non-embedded characteristics 1	1.20	1.2
Project staffing 3	1.00	3.0
Interoperability 4	1.10	4.4
Re-engineering factors 3	1.20	3.6

Task Network

Project tasks and dependencies are noted diagrammatically intask network according to its functional dependencies.

Task Description
T1 System Design

T2 Detailed Design
T3 Database Implementation
T4 Web Page Design
T5 Coding Input Module
T6 Coding database related module
T7 Coding Output Module
T8 Test Planning
T9 Integration

Timeline chart

Project milestones can be shown in a simple time line chart .While the chart doesn’t look complicated, it provides good amount of information on project progress in a simple and understandable chart

Task	Planned Start	Actual Start	Planned End	Actual End
Scope and objective, requirement gathering, planning	29/7/22	29/7/22	7/8/22	7/8/22
Estimation	12/8/22	15/8/22	26/8/22	26/8/22
Resources and arrange necessary tools and techniques, softwares, etc.	26/8/22	-	4/9/22	-
Risk analysis & management	2/9/22	-	9/9/22	-
Scheduling tasks	9/9/22	-	15/9/22	-
Preparing database and coding it	17/9/22	-	28/9/22	-

Coding for all Modules and front end	29/9/22	-	13/10/22	-
Integrating the modules	16/10/22	-	23/10/22	-
Testing	26/10/22	-	28/10/22	-
Reworking	29/10/22	-	30/10/22	-
Finalizing the project	31/10/22	-	31/10/22	-

Software Quality Assurance Plan

The purpose of the Software Quality Assurance Plan is to define all the techniques, procedures, and methodologies that will be used in the project to assure timely delivery of the software that meets specified requirements within project resources. Software Quality Assurance involves reviewing and auditing the software products and activities to verify that they comply with the applicable procedures and standards and providing the software project and other appropriate managers with the results of these reviews and audits.

An overview of the SQA activities is provided. Note that an SQA plan is developed for a moderate to large project and maybe a separate document is included as an appendix. The software quality control process for this project will consist of the following :

SQA Process Description	Description
SRS Review	The SRS will be reviewed by the project team.
Design Review	Design document will be reviewed by the project team.
Unit Testing	Programmer is responsible for the unit testing of each module.
System Testing	Will be done according to the system test plan, which will be reviewed.

- **Purpose:** The purpose of this Software Quality Assurance Plan (SQAP) is to define the techniques, procedures, and methodologies that will be used for the Bookshop Automation System to assure timely delivery of the software that meets specified requirements within the project resources.
- **Scope:** The use of this plan will help assure the following: (1) That software development, evaluation, and acceptance standards are developed, documented, and followed. (2) That the results of software quality reviews and audits will be given to appropriate management. This provides feedback as to how well the development effort is conforming to various development standards. (3) That test results adhere to acceptable standards.

- **Project Checkpoints (Stage Exits):** Each stage of development will have at least one formal checkpoint called a stage exit. When a stage has been successfully exited, it indicates that all draft deliverables due to date have been completed, all outstanding issues have acceptable action plans, and there is a sound plan for the remainder of the project (detailed for the next stage). The project's designated approvers (signoff authorities) must provide a written position of concur/nonconcur at stage exit. All affected functional areas involved in the project also participate in and can provide input to the stage exit.

SCM Plan

Identifying change:

The identification scheme for software objects must recognize that objects evolve throughout the software process .Evolution graphs for each SCIs are used
For example in Alumni Information System here can be

i. Changes in the people accessing the system simultaneously
ii. Changes in the data attributes that are collected when a alumni is registered.
iii. Changes in the features related to payment, appointment booking and tracking alumni’s data.

Version control:

Version control combines procedures and tools to manage different versions of configuration objects that are created during the software process.

Control change:

An engineering change order (ECO) is generated for each approved change. The ECO describes the change to be made, the constraints that must be respected, and the criteria for review and audit.
Activities involved in this process are as follows:

1. Control ad-hoc change to build a stable software development environment. Changes are committed to the repository
2. The request will be checked based on the technical merit, possible side effects and overall impact on other configuration objects.
3. Manage changes and making configuration items available during the software lifecycle
4. For the above change, an extra server may be arranged as a control measure.

Configurational Auditing:

Ensure proper implementation: A software configuration audit complements the formal technical review by assessing a configuration object.

Activities involved in this task are:

1. Checking that defined processes are being followed and ensuring that the SCM goals are satisfied.
2. To verify compliance with configuration control standards. auditing and reporting the changes made
3. Ensure that traceability is maintained during the process.
4. Ensures that changes made to a baseline comply with the configuration status reports
5. Validation of completeness and consistency

Reporting changes to others.

The changes made in the software are reported to all the participants in the SCM task.

Test Plan

Testing will be carried out according to the scheme presented in the test plan. Initially unit testing will be carried out with the classes. Methods in the classes will be tested by sending a specific input to a method and verifying that the method returns the expected value. After unit testing integration testing was carried out to discover the issues that arise when different modules are interacting with each other. Finally system testing was done to check the overall functionality of the system.

--- End Of Experiment ---