MySpace – A socio-academic site for students

Software Project Management Plan Document



### Instructor: Dr. Santosh Singh Rathore

### Group 3 Members: -

Abhisekh Yadav 2020BCS-003

### Aditya Kumar Singh 2020BCS-004

### Anamika Mallick 2020BCS-075

Table of Contents

[1 SIZE ESTIMATION 3](#_Toc97501205)

[1.1 Function Point Metric 3](#_Toc97501206)

[1.1.1 UFP (Unadjusted Function Point) Computation 3](#_Toc97501207)

[1.1.2 Refining Parameters 4](#_Toc97501208)

[1.1.3 Refine UFP based on complexity of the Overall Project 4](#_Toc97501209)

[2 EFFORT AND TIME ESTIMATION 6](#_Toc97501210)

[2.1 COCOMO Model 6](#_Toc97501211)

[2.1.1 Estimation of Source Lines of Code (SLOC) 6](#_Toc97501212)

[2.1.2 Estimation of Development Effort 6](#_Toc97501213)

[2.1.3 Estimation of Development Time 6](#_Toc97501214)

[3 PROJECT SCHEDULE BREAKDOWN 7](#_Toc97501215)

[3.1 Activity Network 7](#_Toc97501216)

[3.1.1 Critical Path Method (CPM) 8](#_Toc97501217)

[3.1.2 Tabular Representation of Activity Network with CPM parameters 8](#_Toc97501218)

[3.2 PERT Chart 9](#_Toc97501219)

[4 RISK MANAGEMENT PLAN 10](#_Toc97501220)

[4.1 Project Risks 10](#_Toc97501221)

[4.2 Technical Risks 10](#_Toc97501222)

[4.3 Business Risks 10](#_Toc97501223)

# SIZE ESTIMATION

The project size is the most fundamental parameter for a Software Project Management Plan Document. All other parameters like effort, duration, cost, etc., can be determined with precision if project size is estimated with very low margin of error.

For the purpose of our report, we are using the Function Point(FP) metric to estimate the project size.

## Function Point Metric

Function Point metric proposes that the size of a project is directly dependent upon the number of different high-level functions or features it supports.

As the number of high-level features supported by the project increases, so does the project size, as the module supporting the added feature also gets added to the project size. This technique helps in estimating the project size directly form the problem specification.

### UFP (Unadjusted Function Point) Computation

**External Inputs(EI)**

1. Google account details (during Sing Up)
2. Google e-mail id (during Log In)
3. Google e-mail id (during Password Reset)
4. Ask question and Post answers in Q&A section
5. Title and Body in the Technical Article section
6. Upvote and Downvote input for both Q&A and Technical Article section

Total number of EI = 6

**External Outputs**

1. Confirmation Message: Account Created Successfully
2. Confirmation Message: Successfully Logged In
3. Confirmation Message: Successfully Logged Out
4. Student Dashboard
5. Links for Resources
6. Displaying question and answers in Q&A section
7. Displaying articles in Technical Article section
8. Links of class schedules
9. Upcoming Event Details
10. Error Message: Account could not be Created
11. Error Message: Unable to Login

Total number of EO = 11

**External Inquiries(EQ)**

1. Request Resources using Search Bar
2. Request Resources to Download
3. Request to show answers to a specific question
4. Request to view schedule
5. Request to view upcoming event details

Total number of EQ = 5

**Internal Logical Files(ILF)**

1. List of Registered students with their personal details
2. List of Question
3. List of Answers
4. List of Articles

Total number of ILF = 4

**External Interface Files(EIF)**

1. Google API for accessing Google’s OAuth 2.0 server for Google Sign Up functionality
2. Google Maps API

Total number of EIF = 2

**Calculation**

= 6\*4 + 11 \*5 + 6\*4 + 4\*10 + 2\*10 = 163

### Refining Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **Type** | **Simple** | **Average** | **Complex** |
| Inputs(EI) | 3 | 4 | 6 |
| Outputs(EO) | 4 | 5 | 7 |
| Inquiries(EQ) | 3 | 4 | 6 |
| Files(ILF) | 7 | 10 | 15 |
| Interfaces(EIF) | 5 | 7 | 10 |

Table: Refined Weights of Parameters based on Complexity

**External Inputs(EI) :** 4 Simple + 2 Average

**External Outputs(EO) :** 9 Simple + 2 Average

**External Inquiries(EQ) :** 4 Simple + 1 Average + 1 Complex

**External Logical Files(ILF) :** 4 Simple

**External Interface(EIF) :** 1 Simple + 1 Average

**Refined UFP** = (4\*3 + 2\* 4 ) + (9\*4 + 2 \* 5 ) + ( 4 \*3 + 1\*4 + 1\*6) + (4\*7) + (1\*5 + 1\*7) = 128

### Refine UFP based on complexity of the Overall Project

Several factors that can impact the overall project size are considered to refine the computed UFP such as high transaction rates, response time requirements, scope for reuse, etc.

|  |  |
| --- | --- |
| Requirement for reliable backup and recovery | 6 |
| Requirement for data communication | 3 |
| Extent of distributed processing | 2 |
| Performance requirements | 4 |
| Expected operational environment | 4 |
| Extent of online data entries | 6 |
| Extent of multi-screen or multi-operation online data input | 0 |
| Extent of online updating of master files | 5 |
| Extent of complex inputs, outputs, online queries and files | 5 |
| Extent of complex data processing | 2 |
| Extent that currently developed code can be designed for reuse | 4 |
| Extent of conversion and installation included in the design | 0 |
| Extent of multiple installations in an organisation and variety of customer  organisations | 0 |
| Extent of change and focus on ease of use | 5 |
| Degree of Influence (DI) | 46 |

**Technical Complexity Factor (TCF)**  = 0.65 + 0.01 \* DI = 0.65 + 0.46 = 1.11

**Function Point(FP)**  = **UFP \* TCF** = 128 \* 1.11 = 142.08 ≈ 142

# EFFORT AND TIME ESTIMATION

## COCOMO Model

COCOMO (COnstructive COst MOdel) proposed by Boehm in 1981 is a heuristic estimation technique. It is a single variable model which divides the software product development into three categories:

* Organic – Application Programs
* Semidetached – Utility Programs
* Embedded – System Programs and real-time system programs

Based on this categorisation and characteristics of our project (small team working on well understood application), this project belongs to **organic** category.

Considering the scale of the project we are using **Basic COCOMO** model for quick and rough estimation of project parameters.

### Estimation of Source Lines of Code (SLOC)

Since, in this web project we’ll be using mostly JavaScript (75-85%), HTML5(7-10%) and CSS3 (7-10%). So we are considering 1 FP ≈ 40 LOC. Hence,

**SLOC = FP \*40** = 142 \* 40 = 5680 LOC = 5.680 KLOC

### Estimation of Development Effort

**Development Effort = 2.4 \* (KLOC)1.05**

= 2.4 \* (5.680)1.05

= 14.868 Person-Month

### Estimation of Development Time

**Development Time = 2.5 \* (Effort)0.38**

= 2.5 \* (14.868)0.38

= 6.97 Months

# PROJECT SCHEDULE BREAKDOWN

The estimated development time using **basic COCOMO** model is approximately 7 months. However, considering the project submission deadline and the fact that the **basic COCOMO** model does not incorporate the reuse of existing frameworks, the project schedule is made for approximately 3 months duration.

## Activity Network

A screenshot of a computer

Description automatically generated with low confidence An activity network is used to graphically show different activities involved in building a project, their estimated durations and interdependencies.

Fig: Activity Network Chart

Here the critical path is shown is purple colour.

### Critical Path Method (CPM)

**Minimum Time(MT) =** 77 days

A screenshot of a video game

Description automatically generated with medium confidence The (Earliest Start, Earliest Finish) and (Latest Start, Latest Finish) is shown in the following chart with corresponding **Slack Time** (which is LS - ES).

Fig: Activity Network Chart with (ES,EF) (LS, LF) and ST

### Tabular Representation of Activity Network with CPM parameters

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Tasks** | **ES** | **EF** | **LS** | **LF** | **ST** |
| Feasibility Study | 0 | 6 | 0 | 6 | 0 |
| Requirement Analysis and Specification | 6 | 16 | 6 | 16 | 0 |
| Project Management Plan | 16 | 22 | 16 | 22 | 0 |
| Designing GUI | 22 | 30 | 29 | 37 | 7 |
| Database Design | 22 | 32 | 22 | 32 | 0 |
| Code GUI | 30 | 50 | 37 | 57 | 7 |
| Code Database | 32 | 57 | 32 | 57 | 0 |
| User Documentation | 22 | 28 | 71 | 77 | 49 |
| Integration and Testing | 57 | 77 | 57 | 77 | 0 |

## PERT Chart

PERT(Project Evaluation and Review Technique) charts are used to represent the statistical variations in the project estimates assuming these to be a normal distribution.

PERT allows for some randomness in task completion times and therefore provides the capability to determine the probability for achieving project milestones based on the probability of completing each task along the path to that milestone. Each task is annotated with three estimates:

**Optimistic (O):** The best possible case task completion time.

**Most likely estimate (M):** Most likely task completion time.

A screenshot of a computer

Description automatically generated with low confidence **Worst case (W):** The worst possible case task completion time.

# RISK MANAGEMENT PLAN

## Project Risks

Since all the tools and frameworks used in this project are open source so there is no risk related to budgetary. The risk of schedule slippage would be mitigated by constant communication among team members. In case we are not meeting the project deadline, functionality(or functionalities) with low priority will not be implemented and website will be up with limited functionality only.

## Technical Risks

This is a well understood application and all the framework and technologies are open-source. So, there are no technical risks associated with this project.

## Business Risks

There is no risk associated with losing budgetary commitments. Upon completion of the project if the user base is not growing as expected then student-survey will be taken and changes will be made accordingly.