Project Cost Estimator

Submitted by: Talha Munawar

Supervised by: Dr. Muddassar Azam Sindhu

Department of Computer Science

Quaid-i-Azam University

Islamabad

Session (2013-2017)

Acknowledgment

In the name of ALLAH, Most Beneficent, Most Merciful. First, I thank a lot to almighty Allah for his blessings and grace in completing my project. At the start, it was looking very hard for me but with the passage of time things seems to be simple with the help of almighty Allah. I thank a lot to my Parents because without their support it was impossible for me to reach at this stage. I thanks to my all teachers Dr. M. Afzal Bhatti, Dr. Mudassir Azam Sindhu, Dr. Ghazanfar Farooq, Dr. Shuaib Karim, Sir S.M. Naqi, Dr. Mubasher Mushtaq, Dr. Akmal Khan Khattak, Dr. Khalid Saleem, Dr. Onaiza Maqbool, Dr. Rabeeh Ayaz Abbasi, Dr. Faiza Iqbal, Miss Ifrah Farrukh Khan, Miss Memona Afsheen and for their kind support and cooperation throughout my degree. Special thanks to my project supervisor Dr. Mudassir Azam Sindhu, for their cooperation during implementation of my project. Continuous suggestions of teachers and supervisor enabled me to implement this project in better way. At the end, I thank to my friends Muhammad Umer Khalid, Suleman Tanveer, Ahmed ILyas and fellows who encouraged and motivated me throughout my degree.

Talha Munawar

2013-2017

Abstract

Project Cost Estimator is a web based tool for an organization to estimate the cost, size, effort and duration using previous data of completed projects of the organization. There is major issue in estimation of project as it is a very critical activity. Any wrong estimation can cause a huge loss of the organization. Project cost estimator will help the project manager of the organization to estimate the project in an efficient way.

There are four estimation models are implement in this project. These models are function point, use case point, empirical model and basic cocomo model. Function point and use case point models use historical data of organization for estimation purpose. Project's actual metric is maintain in the organization which contains size of project in KLOC (thousand lines of code), effort in person month, duration in months and cost in rupees. This metric helps to get the productivity factors of effort, duration, size and cost .PCE provides the facility to auto calculate the size of project in many languages and conversion of amount into different currrencies. While empirical model and basic cocomo model use some static equations for estimation.

Table of Contents

List of Tables

List of Figures

Abbreviations

PCE	Project Cost Estimator
KLOC	thousand Line of Cost
LOC	Line of Code
UC	Use Case
CM	Cocomo Model
EE	Empirical Estimation
FP	Function Point
SPMP	Software Project Management Plan
SRS	Software Requirement Specification
SDD	Software Design Description
STD	Software Test Document
SSD	System Sequence Diagram
SQL	Structured Query Language
EIs	Number of external inputs
EOs	Number of external outputs
EQs	Number of external inquiries
ILFs	Number of internal logical files
EIFs	Number of external interface files
VAF	Value adjustment factors
UUCP	Unadjusted Use Case Points
TCF	Technical Complexity Factor
ECF	Environment Complexity Factor
PF	Productivity Factor
UUCW	Unadjusted Use Case Weight
UAW	Unadjusted Actor Weight
TF	Technical Factor
LOC	Lines of code
E	effort
D	development

Chapter 01

Software Project Planer

1.1 INTRODUCTION

The purpose of this document is to provide planning and scheduling for the given project.

1.1.1 Product Overview

The Project Cost Estimator is a web based tool which will estimate the cost, effort, time and size for the optimized completion of project. This product will help the project manager to estimate and scheduling of project. Project manager could estimate metrics using different methods such as function point metric, use case point metric, empirical estimation model and cocomo model. Estimations will make on the basis of previous data of the completed projects.

1.1.2 Project Deliverables

There are two deliverable of projects.

- SPMP & SRS (mid exam)
- Complete document up to final exam
- Complete Project implementation with documentation up to final exam of final semester.

1.2 PROJECT ORGANIZATION

Project organization consists of software process model and roles and responsibilities.

1.2.1 Software Process Model

I shall use water fall as a software model for the completion of project due to the following reasons:

- Project requirements are clear
- Enough time to implement
- Each phase is completed first and then start second

In water fall model every next step depends upon the previous step. If there is any change required then we have to go to the first step again.

1.2.2 Roles and Responsibilities

This project is assigned to a single person so everything related to project will done by me.

1.2.3 Tools and Techniques

Tools and techniques used for this project is following:

- Easy PHP server
- Web browser
- Notepad ++

1.3 PROJECT MANAGEMENT PLAN

There are following tasks will performed for the completion of project.

1.3.1.1 Project description and under standing

This is the first task in which project developer will understand the project and will make an overall description. This task has no any sub tasks. It has no any deliverable because it is only for the understanding of project.

1.3.1.2 Software Project Management Plan (SPMP)

- **Description**
  In this task project planning will done.
- **Deliverable & Milestone**
  - Its deliver able is SPMP document
  - There is no mile stone at this level
- **Resource needed**
  Talha Munawar
- **Dependencies**
  Its dependency is the first task which is project description. For this task planner should have understanding of project.
- **Risk and contingencies.**
  There is no any risk in this task.

1.3.1.3 Software Requirement Specification (SRS)

- **Description**
  This task will define about the software requirement specification
- **Deliverable & Milestone**
  - Its deliverable is SRS document.
  - There is a milestone after the completion of the SPMP and SRS.
- **Resource needed**
  Talha Munawar
- **Dependencies**
  Its dependency is SPMP.
- **Risk and contingencies.**
  There is no any risk in this task.

1.3.1.4 Software Design Description (SDD)

- **Description**
  This task will define about the software design description.
- **Deliverable & Milestone**
  - Its deliverable is SRS document.
  - There is no milestone for this task.

- **Resource needed**
  Talha Munawar
- **Dependencies**
  Its dependency is SRS.
- **Risk and contingencies.**
  There is no any risk in this task.

1.3.1.5 Software Test Documentation (STD)

- **Description**
  This task will define about the software test documentation.
- **Deliverable & Milestone**
  - Its deliverable is whole document.
  - There is a milestone for this task.
- **Resource needed**
  Talha Munawar
- **Dependencies**
  Its dependency is SRS.
- **Risk and contingencies.**
  There is no any risk in this task.

1.3.2 Assignments

There is nothing assign to any other person. Everything will be done by me.

1.3.3 Timetable

Time table for whole project documentation is shown in the Figure 1.1 and Figure 1.2.

1.3.3.1 Plan Chart

Plan chart describes the schedule which we will follow in documentation and implementation. It will help us to complete our work in an efficient way. In this plan chart project is divided into tasks and subtasks with the deadline. There are two plan charts. First one is for the documentation and the second one is for the implementation.

		Name	Duration	Start	Finish	Predecessors
1		Project description and understanding	2.25 days	9/30/16 8:00 AM	10/4/16 10:00 AM	
2		**Software project management plan**	**6 days?**	**10/5/16 8:00 AM**	**10/12/16 5:00 PM**	1
3		Introduction	1 day?	10/5/16 8:00 AM	10/5/16 5:00 PM	
4		Project Organization	2 days?	10/6/16 8:00 AM	10/7/16 5:00 PM	
5		Project Management Plan	3 days?	10/8/16 8:00 AM	10/12/16 5:00 PM	
6		**Software requirement specification**	**22 days?**	**10/13/16 8:00 AM**	**11/11/16 5:00 PM**	2
7		Introduction	1 day?	10/13/16 8:00 AM	10/13/16 5:00 PM	
8		**Specific Requirements**	**15 days?**	**10/14/16 8:00 AM**	**11/3/16 5:00 PM**	
9		External Interface Requirements	1 day?	10/14/16 8:00 AM	10/14/16 5:00 PM	
10		Software produce feature	9 days?	10/15/16 8:00 AM	10/27/16 5:00 PM	
11		Software System Attributes	2 days?	10/28/16 8:00 AM	10/31/16 5:00 PM	
12		Data base requirements	3 days?	11/1/16 8:00 AM	11/3/16 5:00 PM	
13		Additional requirements	2 days?	11/4/16 8:00 AM	11/7/16 5:00 PM	
14		Review SRS and make changes	5 days?	11/5/16 8:00 AM	11/11/16 5:00 PM	
15		Document deliverable(SPMP & SRS)	1 day?	11/18/16 8:00 AM	11/18/16 5:00 PM	6
16		**Software design description**	**15 days?**	**11/29/16 8:00 AM**	**12/19/16 5:00 PM**	6
17		Introduction	1 day?	11/29/16 8:00 AM	11/29/16 5:00 PM	
18		System Architectural Design	4 days?	11/30/16 8:00 AM	12/5/16 5:00 PM	
19		Detailed description of component	6 days?	12/6/16 8:00 AM	12/13/16 5:00 PM	
20		User interface design	3 days?	12/14/16 8:00 AM	12/16/16 5:00 PM	
21		Additional material	1 day?	12/17/16 8:00 AM	12/19/16 5:00 PM	
22		**Software test documentation**	**13 days?**	**12/20/16 8:00 AM**	**1/5/17 5:00 PM**	16
23		Introduction	2 days?	12/20/16 8:00 AM	12/21/16 5:00 PM	
24		Test plan	3 days?	12/22/16 8:00 AM	12/26/16 5:00 PM	
25		Test cases	6 days?	12/27/16 8:00 AM	1/3/17 5:00 PM	
26		Additional material	2 days?	1/4/17 8:00 AM	1/5/17 5:00 PM	
27		Documen deliverable complete	1 day?	1/6/17 8:00 AM	1/6/17 5:00 PM	22

Figure 1.1 *Plan Chart (Documentation)*

	Name	Duration	Start	Finish	Predecessors
1	**Function Point Metric**	14 days?	**2/20/17 8:00 AM**	**3/9/17 5:00 PM**	
2	Algorithem implementation	5 days	2/20/17 8:00 AM	2/24/17 5:00 PM	
3	Algorithem Testing	1 day?	2/25/17 8:00 AM	2/27/17 5:00 PM	
4	FP interface implementatior	3 days	2/28/17 8:00 AM	3/2/17 5:00 PM	
5	using FP to estimate projec	5 days	3/3/17 8:00 AM	3/9/17 5:00 PM	
6	**Use Case Point Metric**	17 days	**3/14/17 8:00 AM**	**4/5/17 5:00 PM**	
7	Algorithem implementation	5 days	3/14/17 8:00 AM	3/20/17 5:00 PM	
8	Algorithem Testing	2 days	3/21/17 8:00 AM	3/22/17 5:00 PM	
9	UCP interface implementati	3 days	3/23/17 8:00 AM	3/27/17 5:00 PM	
10	using UCP to estimate proje	7 days	3/28/17 8:00 AM	4/5/17 5:00 PM	
11	**Empirical Model Metric**	13 days	**4/6/17 8:00 AM**	**4/24/17 5:00 PM**	
12	Algorithem implementation	5 days	4/6/17 8:00 AM	4/12/17 5:00 PM	
13	Algorithem Testing	2 days	4/13/17 8:00 AM	4/14/17 5:00 PM	
14	EE interface implementatior	4 days	4/17/17 8:00 AM	4/20/17 5:00 PM	
15	using EE to estimate projec	2 days	4/21/17 8:00 AM	4/24/17 5:00 PM	
16	**Baasic Cocomo Model Me**	14 days	**4/25/17 8:00 AM**	**5/12/17 5:00 PM**	
17	Algorithem implementation	5 days	4/25/17 8:00 AM	5/1/17 5:00 PM	
18	Algorithem Testing	3 days	5/2/17 8:00 AM	5/4/17 5:00 PM	
19	EE interface implementatior	3 days	5/8/17 8:00 AM	5/10/17 5:00 PM	
20	using EE to estimate projec	2 days	5/11/17 8:00 AM	5/12/17 5:00 PM	
21	**Source code counter**	10 days	**5/15/17 8:00 AM**	**5/26/17 5:00 PM**	
22	Implementation	5 days	5/15/17 8:00 AM	5/19/17 5:00 PM	
23	Testing	5 days	5/22/17 8:00 AM	5/26/17 5:00 PM	
24	**Currency converter**	2 days	**5/29/17 8:00 AM**	**5/30/17 5:00 PM**	
25	Finding api	2 days	5/29/17 8:00 AM	5/30/17 5:00 PM	
26	Implementation	1 day	5/29/17 8:00 AM	5/29/17 5:00 PM	
27	**View Metrics**	8 days?	**5/30/17 8:00 AM**	**6/8/17 5:00 PM**	
28	View FP metric	1 day?	5/30/17 8:00 AM	5/30/17 5:00 PM	
29	View FP metric	1 day?	5/31/17 8:00 AM	5/31/17 5:00 PM	
30	View FP metric	1 day?	6/1/17 8:00 AM	6/1/17 5:00 PM	
31	View FP metric	1 day?	6/2/17 8:00 AM	6/2/17 5:00 PM	
32	View detailed Metric	2 days	6/5/17 8:00 AM	6/6/17 5:00 PM	
33	Delete metric	2 days	6/7/17 8:00 AM	6/8/17 5:00 PM	
34	Internal Module integration	9 days?	6/12/17 8:00 AM	6/22/17 5:00 PM	
35	**Documentation**	3 days	**6/23/17 8:00 AM**	**6/27/17 5:00 PM**	
36	Implementation chapter	3 days	6/23/17 8:00 AM	6/27/17 5:00 PM	
37	**Final integration**	9 days	**6/27/17 8:00 AM**	**7/7/17 5:00 PM**	**1;6;11;16;21;24;27**
38	Integration and Testing	9 days	6/27/17 8:00 AM	7/7/17 5:00 PM	

Figure 1.2 *Plan Chart (Implementation)*

1.3.3.2 Time Line Chart

Time line chart describes the scheduling of tasks.

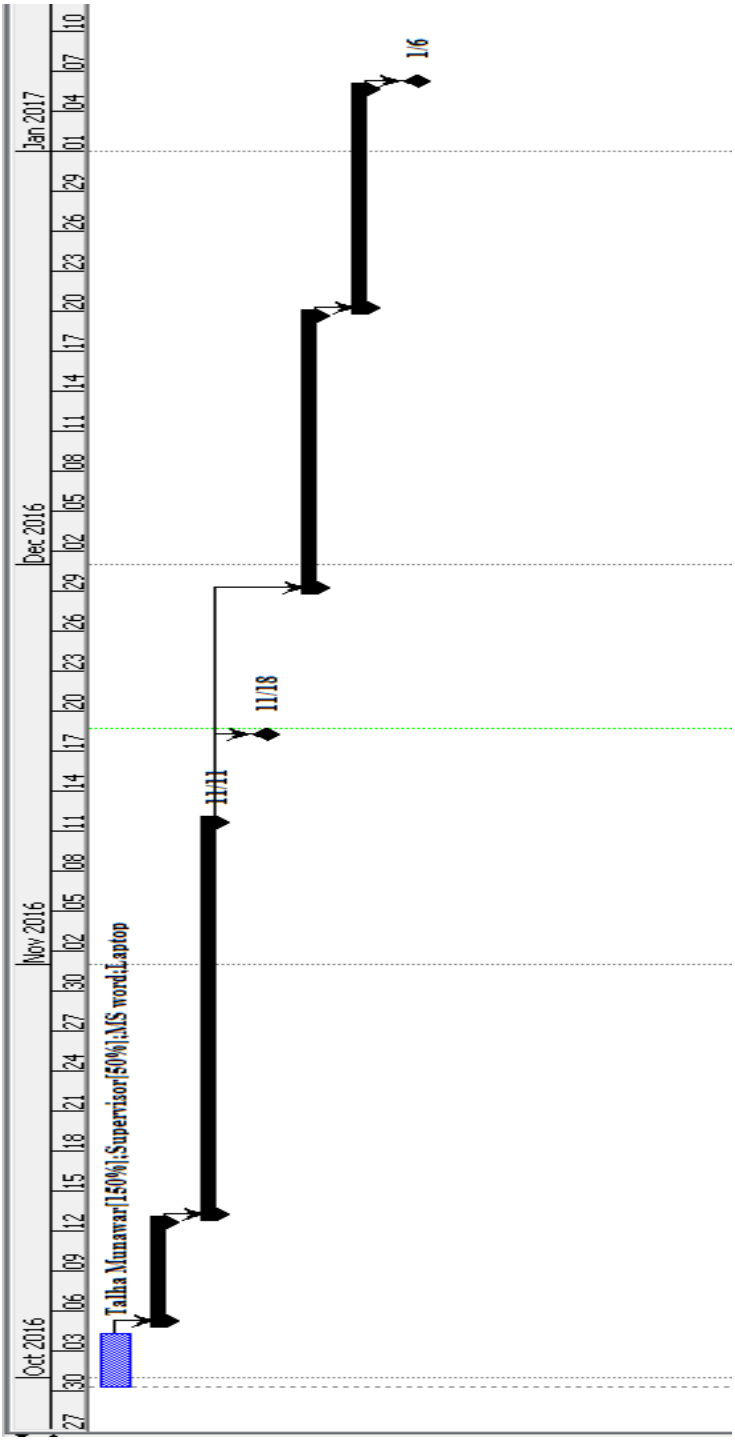


Figure 1. 3 *Time Line Chart (Documentation)*

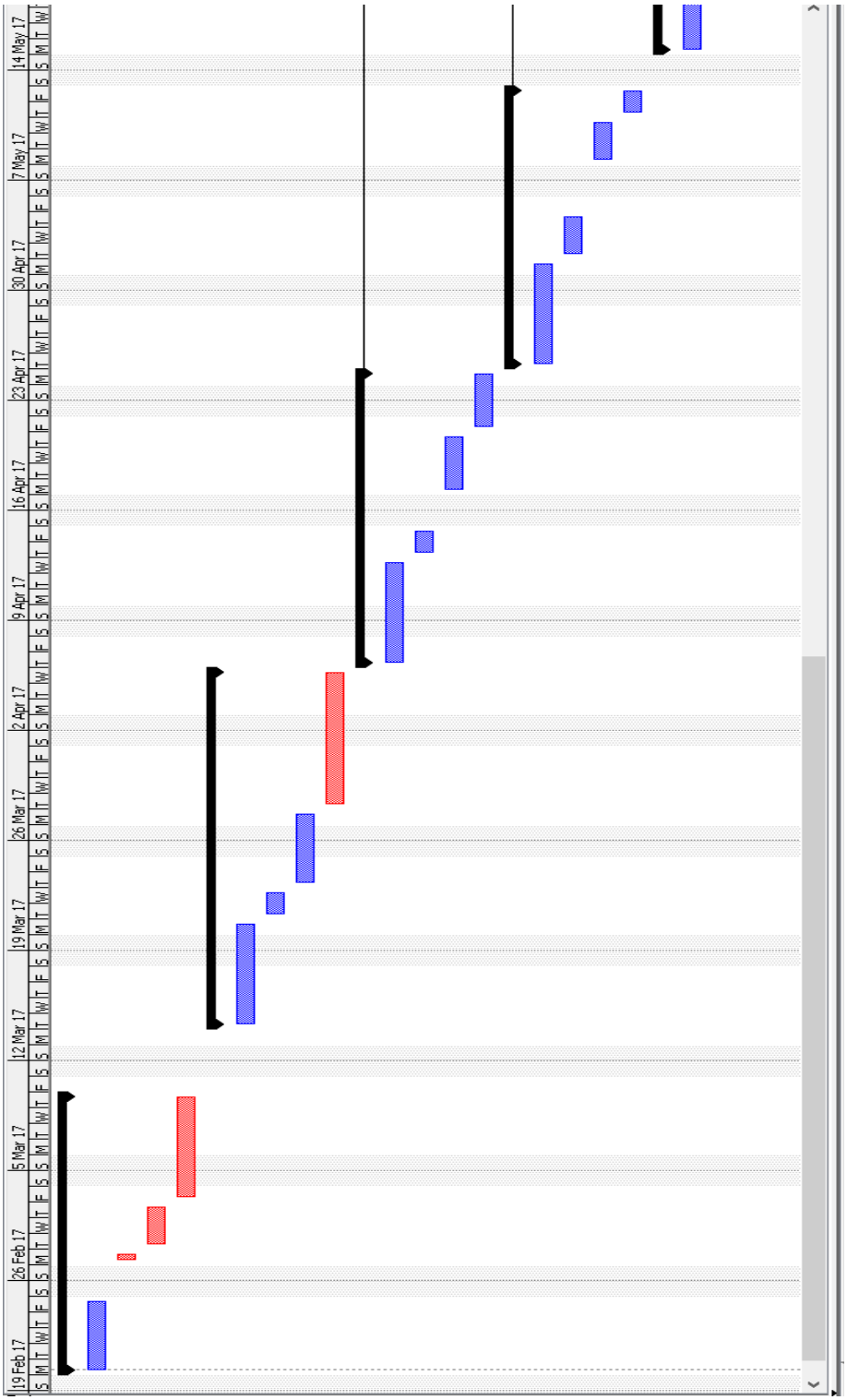


Figure 1. 4 Time Line Chart (Implementation-1)

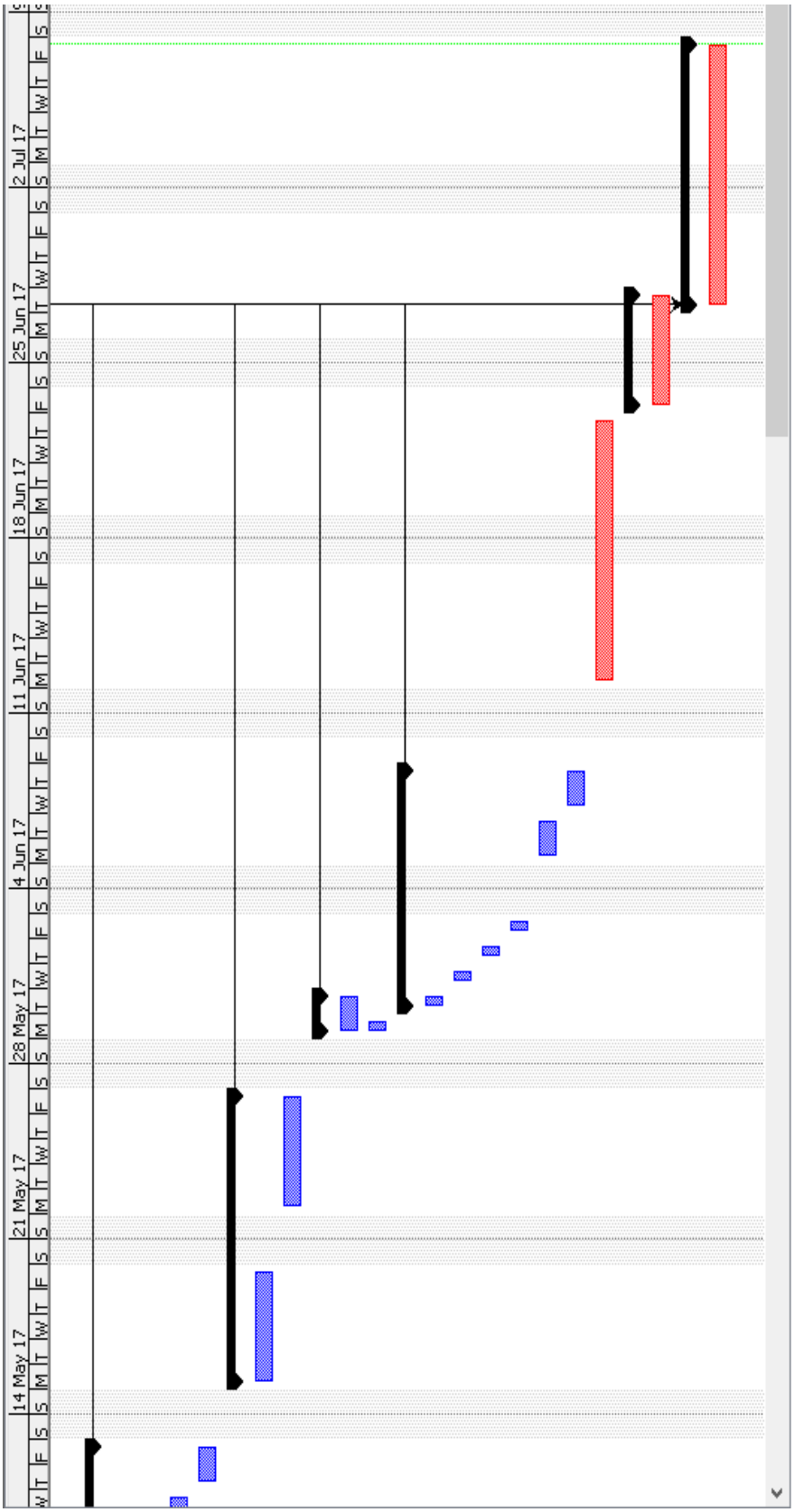


Figure 1. 5 *Time Line Chart (Implementation-2)*

Chapter 02

Software Requirement Specification

2.1 INTRODUCTION

The purpose of this document is to clarify the system requirement specifications.

2.1.1 Product Overview

The PCE is a web base tool which will use to estimate the cost, time, effort, size and scheduling of the project before it started. The basic aim of this project is to help the project manager to estimate the cost, size, duration and effort for making the software. Effort and duration is measured person months and cost is measured in Rupees. This product will help the project manager to estimate the cost, effort in person month, time required and size in KLOC. This product will measure these metrics by using different methods such as Function point metric, Use case point metrics, empirical estimation model and cocomo model. Project manager will enter the actual metrics after the completion of the project and can compare the real metrics with estimated metrics. This will helps the project manager to schedule the project in an efficient way and increase the revenue of the organization.

2.1.2 Purpose

The purpose of PCE (project cost estimator) is to provide the software through which a project manager can estimate the cost, effort, time, size and scheduling for the project before it is started. Before starting a project, the project manager should know that how long it will take, how much effort will be required and how many people will be involved. With correct or close estimation project manager can manage and control the project more efficiently and effectively. This product will help the project manager to make a quick estimate for the cost, time, effort and size to start the new project. As this software will maintain the previous record of metrics which will help the user to compare the estimated metrics with the actual metrics which were uploaded after the completion of the project. This product will provide the facility to the project manager to see the performance of each team and use them in a better way for the next time. It assist the project manager in making decisions that will lead to a successful project completion to optimize cost and resource usage.

2.1.3 Scope

- Estimate time, cost, effort, size and schedule of projects.
- Use metrics of previous projects to estimate the metrics of new project.
- Add actual metrics of completed projects.
- Maintain start date and end date of each project.
- Manage resources in an efficient way.
- Generate reports of previous project metrics.
- Generates graphs to see comparison of different metrics of project.
- Find metrics by using different techniques such as Function point, Use case point, empirical estimation model and cocomo model.
- Count source line code of completed projects and save to actual metric.
- Project manager and software developers are the stakeholders.
- Convert given amount in to different currencies.

2.2 SPECIFIC REQUIREMENTS

It contain the functional requirement of the system. It give detailed description of system and its features.

2.2.1 External Interface Requirements

It contains the following interfaces requirements:

2.2.1.1 User Interfaces

User interface or screens will prepared in such a way that it will provide maximum functionality in the minimum clicks. These interfaces will display the appropriate messages on actions such that user will not found himself lost in the screens. Reduction technique will implement when screens will make because in this technique complex tasks are represents in an easier way. Tunneling techniques are also implements in which a process is completed in a tunnel way. Tailoring technique will also provide to the user such that he/she will be able to view customized information related to his/her interest. It use web browser to see style and functionality for the project cost estimator.

PCE has very easy and user friendly interface. User guide book (project documentation) will be sufficient to guide the users how to use this product without facing any problems or difficulties. PCE should be designed for easy to use, and appropriate error messages for end user inputs. The color scheme will select in such a way that readability effect increases and user will not face any difficulty in reading.

2.2.1.2 Hardware Interfaces

- Keyboard
- Mouse
- Screen

2.2.1.3 Software Interfaces

- Web browser
- MySQL (open source database)
- Easy PHP server
- Windows operating system

2.2.2 Software Product Features

Software product features are given in the following lists:

- Calculate metric
- Calculate function point metric
- Calculate empirical model metric
- Calculate cocomo model metric
- Calculate use case point metric
- Add estimated metric to project
- Re-Estimate metric

- View metrics
- View function point metrics
- View empirical model metrics
- View cocomo model metrics
- View use case point metrics
- View actual metrics
- Update actual metrics
- Delete metrics
- Count lines of code
- Currency converter
- Search metrics
- Filter metrics

2.2.2.1 Use Case Diagram

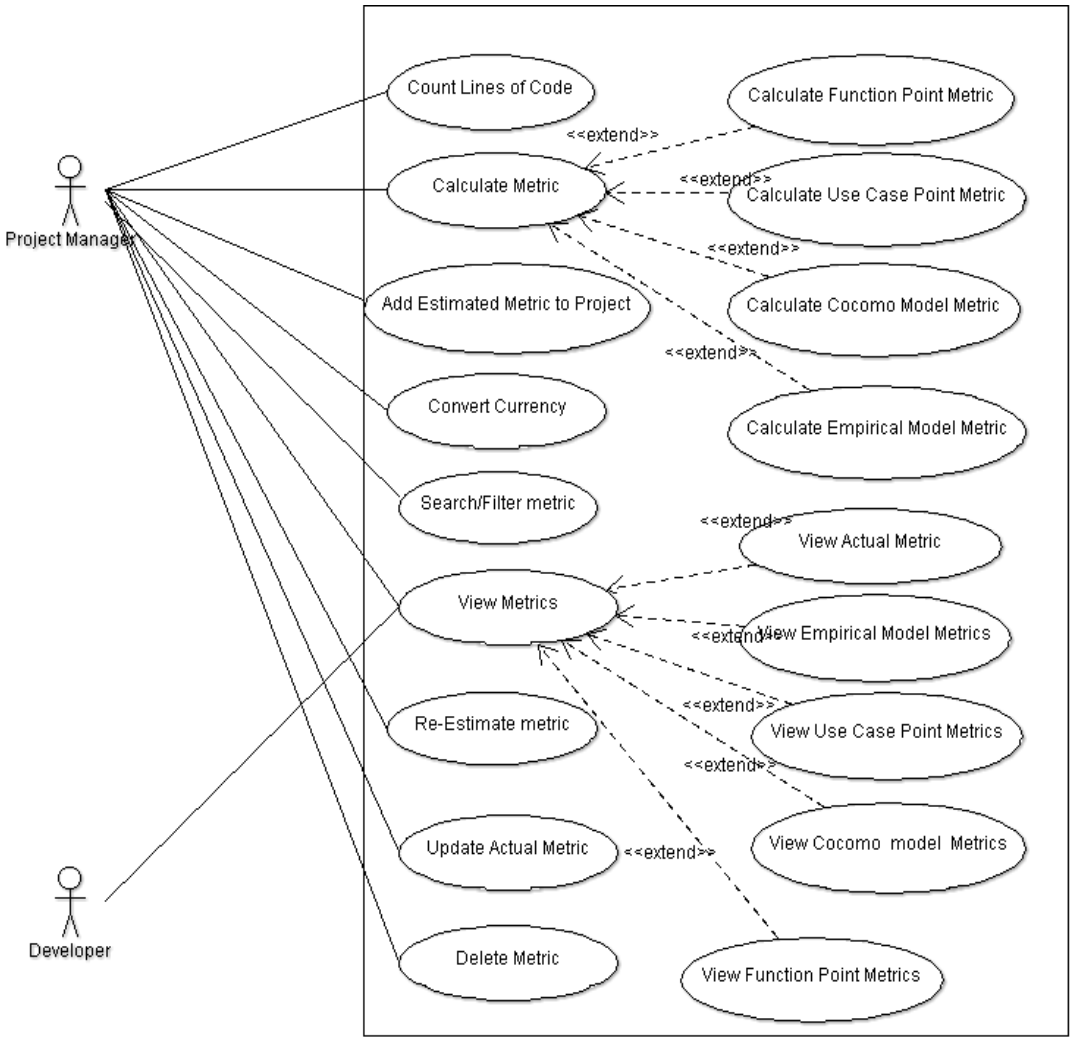


Figure 2. 1 Use Case Diagram

2.2.2.2 Use Cases

[8] Writing use case is an excellent technique to understand and describe the requirements. Use cases are requirements; primarily they are functional requirements that indicate what the system will do.

UC-1: Calculate Metric:

This use case describes the functionality of the metric estimation. There are different types of estimation models which will select for the required estimation. For this purpose user select the model and fill the required inputs in the fields and then press the estimate button then system will display the estimated metric.

Table 2.1 *Calculate Metric*

UC-1: Calculate Metric	
Primary actor	Project Manager
Goal in context	User will be able to calculate metric.
Pre-condition	Project manager should be login and main menu on screen.
Post-condition	User will successfully calculate the metric.
Extension point	Calculate Function Point Metric, step 3. Calculate Use Case Point Metric, step 3. Calculate Empirical Model Metric step 3. Calculate Cocomo Model Metric, step 3.
Main Success Scenario	1. Select “Estimation” button from main menu. 2. Select type of required metric calculation from option. 3. Fill the required fields. 4. Press calculate button. 5. System will display the calculated metric.
Extensions	1a) There is no internet connection. 3a) Required fields are not properly filled. 1. Display message to fill the fields correctly. 4a) There is invalid inputs. 1. Display message to enter the valid inputs.
Special Requirements	None.
Technology	1. Mouse and Keyboard for input 2. Screen for display output.
Frequency	Many times a day or week.

UC-2: Calculate Function Point Metric (extending use case)

This is an extending use case of calculate metric in which user select the model. This use case is corresponding to the function point metric. In this model user fill the required inputs according to the model requirements then press the estimate button to view the estimated metric.

Table 2.2 Calculate Function Point Metric (extending use case)

UC-2: Calculate Function Point Metric (extending use case)	
Primary actor	Project Manager
Goal in context	Function point metric calculation
Pre-condition	Project manager should be login.
Post-condition	Project manager will successfully calculated metric using FP
Trigger	Project manager wants to calculate metric by function point
Main Success Scenario	1) Select Function Point. 2) New screen will appear. 3) Enter input fields with required data. 4) Press Estimate button. 5) System will display the estimated metric.
Extensions	1. 1 Server will not work properly. a) Wait until server will work properly. 2. Page will not load properly. a) Refresh the page. 3. Invalid entry. a) System will display a message for proper input.
Special Requirements	None.
Technology	1. Mouse and Keyboard for input 2. Screen for display output
Frequency	Many times a day.

UC-3: Calculate Empirical Model Metric (extending use case)

This is an extending use case of calculate metric in which user select the model. This use case is corresponding to the empirical model metric. In this model user fill the required inputs according to the model requirements then press the estimate button to view the estimated metric.

Table 2.3 Calculate Empirical Model Metric (extending use case)

UC-3: Calculate Empirical Model Metric (extending use case)	
Primary actor	Project Manager
Goal in context	Empirical model metric calculation
Pre-condition	Project manager should be login.
Post-condition	Empirical metric will calculate successfully.
Trigger	Project manager wants to calculate the empirical metrics.
Main Success Scenario	1. Select the option “Empirical Estimation”. 2. System will display a new screen with input fields. 3. Enter input fields with required data. 4. Press Estimate button to estimate empirical metric.
Extensions	1. Server will not work properly. a) Wait until server will work properly. 2. Page will not load properly. a) Refresh the page. 3. Invalid entry. a) System will display a message for proper input.
Special Requirements	None.
Technology	1. Mouse and Keyboard for input 2. Screen for display output
Frequency	Many times a day.

UC-4: Calculate Cocomo Model Metric (extending use case)

This is an extending use case of calculate metric in which user select the model. This use case is corresponding to the empirical model metric. In this model user fill the required inputs according to the model requirements then press the estimate button to view the estimated metric.

Table 2.4 *Calculate Cocomo Model Metric (extending use case)*

UC-4: Calculate Cocomo Model Metric (extending use case)	
Primary actor	Project Manager
Goal in context	Cocomo model metric calculation
Pre-condition	Project manager should be login.
Post-condition	Metric will calculate successfully.
Trigger	Project manager wants to calculate the Cocomo metrics.
Main Success Scenario	1. Select the option “Cocomo Model”. 2. System will display a new screen with input fields. 3. Fill input fields with required inputs. 4. Press Estimate button for estimation. 5. System will display the metric.
Extensions	1. Server is not working. a) Wait until the server work properly. 2. Screen will not load properly. a) Refresh page. 3. Invalid entry. a) System will display a message for valid entry
Special Requirements	None.
Technology	1. Mouse and Keyboard for input 2. Screen for display output
Frequency	Many times a day.

UC-5: Calculate Use Case Point Metric (extending use case)

This is an extending use case of calculate metric in which user select the model. This use case is corresponding to the use case point metric. In this model user fill the required inputs according to the model requirements then press the estimate button to view the estimated metric.

Table 2.5 Calculate Use Case Point Metric (extending use case)

UC-5: Calculate Use Case Point Metric (extending use case)	
Primary actor	Project Manager
Goal in context	Use case point metric calculation
Pre-condition	Project manager should be login.
Post-condition	Metric will calculate successfully.
Tigger	Project manager wants to calculate the use case point metrics.
Main Success Scenario	1. Select the option “Use Case Point”. 2. System will display new screen with different fields. 3. Fill the input fields with required data 4. Press Estimate button 5. System will calculate and show metric.
Extensions	1. Server is not working properly. a) Wait until server works properly. 2. Page is not properly loaded. a) Refresh the page again 3. Invalid input a) System will display a message for valid inputs.
Special Requirements	None.
Technology	1. Mouse and Keyboard for input 2. Screen for display output
Frequency	Many times a day.

UC-6: Add Estimated Metric to Project

This use case describes the functionality of add estimated metrics to project in which user wants to add estimated metric to already exists project. For this purpose user click add metric button and a list will display and user select the project. After successfully selecting the project user press insert button then metric will added to the project. Estimated metric is only added to pending and in-completed projects. After the completion of project user will not be able to add the estimated metric to the project.

Table 2.6 *Add Estimated Metric to Project*

UC-6:Add Estimated Metric to Project	
Primary actor	Project Manager
Goal in context	Add estimated metric to the project.
Pre-condition	1. Metric should be calculated.
Post-condition	User will add estimated metric to project successfully.
Main Success Scenario	1. Select the button add metric to project. 2. Select the project from the list. 3. Press ok to add the metrics to the selected project. 4. Metric will added successfully to the project.
Extensions	2a. There is no any project in the list. a) System will display a message that there is no any project available. b) First register project and then add metric. 3. Server is not working properly. a) Wait until server will work properly. b) Refresh page.
Special Requirements	None.
Technology	1. Mouse and Keyboard for input 2. Screen for display output
Frequency	Many times a day.

UC-7:Re-Estimate Metric

This use case describes the functionality of re-estimation of the project with different methods. If user wants to re-estimate the projects with different inputs, then he/she will press the re-estimate button and again input screen will appear. User will fill the input fields and press the estimate button and system will show the re-estimated metric to the user. If the user not fill the input fields correctly then on press estimate button system will display the message for enter valid inputs.

Table 2.7 *Re-Estimate Metric*

UC-7:Re-Estimate Metric	
Primary actor	Project Manager
Goal in context	Metric will be re-estimate
Pre-condition	Metric should be calculated and shown on the screen with Re-estimate button.
Post-condition	User will Re- estimate metric successfully.
Main Success Scenario	1. Select the button Re-metric. 2. New screen will display with input fields. 3. Fill the input fields and press calculate button. 4. Estimated metric will display on screen.
Extensions	1. There is internet connection failure. a) Wait until internet available 2. Page is not loaded properly. a) Refresh the page. 3. Invalid inputs. a) System will display a message that enter the valid inputs.
Special Requirements	None.
Technology	1. Mouse and Keyboard for input. 2. Screen for display output.
Frequency	Many times a day.

UC-8: View Metrics

This use case describes the functionality of view metrics. If user wants to view the estimated metrics then he select the option metric from the navigation. On pressing metric button system will display the metric options in which user select the related option. User can view the metrics such as function point metrics, use case point metrics, empirical model metrics and cocomo model metrics.

Table 2.8 *View Metric*

UC-8: View Metrics	
Primary actor	Project Manager
Goal in context	User will see the metric.
Pre-condition	User should be login.
Post-condition	User will view metric successfully.
Extension point	1. View function point metrics, step3. 2. View use case point metrics, step 3. 3. View empirical metrics, step 3. 4. View cocomo metrics, step 3. 5. View actual metrics, step3.
Main Success Scenario	1. User select view metric. 2. System will show options. 3. User will select the option. 4. System will show metric.
Extensions	1. Server is not working properly. a) Wait until server works properly
Special Requirements	None.
Technology	1. Mouse and Keyboard for input 2. Screen for display output
Frequency	Many times a day.

UC-9: View Function Point Metrics (extending use case)

This is an extending use case of view metrics. If user wants to see the function point metrics then he/she will press the view function point metric detail then system will display the function point metrics in the table. If user wants to see the more detail then he can press the view button then more detail about the metric will display.

Table 2.9 View Function Point Metrics (extending use case)

UC-9: View Function Point Metrics (extending use case)	
Primary actor	Project Manager
Goal in context	User will view the Function Point Metric.
Pre-condition	1. User should be login.
Post-condition	User will successfully view the Function Point Metric.
Trigger	Project manager select the Function Point Metrics.
Main Success Scenario	1. Select the option Function Point Metric. 2. System will show Function Point Metric.
Extensions	1. Server is not working properly. a) Wait until server works properly. b) Refresh the page. 2. There is no data about Function Point Metric in data base. a) System will display a message that there is no data to display.
Special Requirements	None.
Technology	1. Keyboard and mouse for input. 2. Screen for display.
Frequency	Many times a day or week.

UC-9: View Empirical Model Metrics (extending use case)

This is an extending use case of view metrics. If user wants to see the empirical model metrics then he/she will press the view empirical model metric detail then system will display the empirical model metrics in the table. If user wants to see the more detail then he can press the view button then more detail about the metric will display.

Table 2.10 *View Empirical Model Metrics (extending use case)*

UC-10: View Empirical Model Metrics (extending use case)	
Primary actor	Project Manager
Goal in context	User will view the empirical metric.
Pre-condition	1. User should be login.
Post-condition	User will successfully view the Empirical Model Metric.
Trigger	Project manager select the Empirical Model Metrics.
Main Success Scenario	1. Select the option Empirical Model Metric. 2. System will show Empirical Model Metric.
Extensions	1. Server is not working properly. a) Wait until server works properly. b) Refresh the page. 2. There is no data about empirical metric in data base. a) System will display a message that there is no data to display.
Special Requirements	None.
Technology	1. Keyboard and mouse for input. 2. Screen for display.
Frequency	Many times a day or week.

UC-9: View Use Case Point Metrics (extending use case)

This is an extending use case of view metrics. If user wants to see the use case point metrics then he/she will press the view use case point metric detail then system will display the use case point metrics in the table. If user wants to see the more detail then he can press the view button then more detail about the metric will display.

Table 2.11 *View Use Case Point Metrics (extending use case)*

UC-11: View Use Case Point Metrics (extending use case)	
Primary actor	Project Manager
Goal in context	User will view the Use Case Point Metric.
Pre-condition	2. User should be login.
Post-condition	User will successfully view the Use Case Point Metric.
Trigger	Project manager select the Use Case Point Metrics.
Main Success Scenario	1. Select the option use case point metric. 2. System will show Use Case Point Metric in new screen.
Extensions	1. Server is not working properly. a) Wait until server works properly. b) Refresh the page. 2. There is no data about Use Case Point Metric in data base. a) System will display a message that there is no data to display.
Special Requirements	None.
Technology	1. Keyboard and mouse for input. 2. Screen for display.
Frequency	Many times a day or week.

UC-9: View Cocomo Model Metrics (extending use case)

This is an extending use case of view metrics. If user wants to see the cocomo model metrics then he/she will press the view empirical model metric detail then system will display the cocomo model metrics in the table. If user wants to see the more detail then he can press the view button then more detail about the metric will display.

Table 2.12 View Cocomo Model Metrics (extending use case)

UC-12: View Cocomo Model Metrics (extending use case)	
Primary actor	Project Manager
Goal in context	User will view the Cocomo Model Metric.
Pre-condition	1. User should be login.
Post-condition	User will successfully view the Cocomo Model Metric.
Trigger	Project manager select the Cocomo Model Metrics.
Main Success Scenario	1. Select the option Cocomo Model. 2. System will show Cocomo Model Metrics.
Extensions	1. Server is not working properly. a) Wait until server works properly. b) Refresh the page. 2. There is no data about cocomo metric in data base. a) System will display a message that there is no data to display.
Special Requirements	None.
Technology	1. Keyboard and mouse for input. 2. Screen for display.
Frequency	Many times a day or week.

UC-9: View Actual Metrics

This use case describes the functionality of view actual metrics. If user wants to view the actual metrics of the projects then he/she will open the project first. When user press the metric button then actual metric will display to the user. This will also show the graphical comparison of the actual metrics with the estimated metrics.

Table 2.13 View Actual Metric (extending use case)

UC-13: View Actual Metric (extending use case)	
Primary actor	Project Manager
Goal in context	User will view the Actual metric.
Pre-condition	1. User should be login. 2. Project will already display on the screen.
Post-condition	User will successfully view the Actual metric.
Trigger	Project manager select the Actual view metrics.
Main Success Scenario	1. Select the button metric in project table. 2. Actual metric will display on screen.
Extensions	1. Server is not working properly. a) Wait until server works properly. b) Refresh the page. 2. There is no data about actual metric in data base. a) System will display a message that there is no data to display.
Special Requirements	None.
Technology	1. Keyboard and mouse for input. 2. Screen for display.
Frequency	Many times a day or week.

UC-14: Update Actual Metric

This use case describes the functionality of update actual metrics. Some of the fields cannot be editable such as duration and effort because these values calculated by the system. User can update the size of project using source code counter, function point and use case points. When user fill the fields then he/she press the update button. If user fill input fields correctly then system will display a message that metric updated successfully and if there is some invalid input then system will display a message that fill the input fields correctly.

Table 2.14 *Update Actual Metric*

UC-14: Update Actual Metric	
Primary actor	Project Manager
Goal in context	User will update project metric.
Pre-condition	1. User should be login. 2. Metric should be displayed on screen. 3.There should be actual project metrics
Post-condition	User will successfully updated metric.
Main Success Scenario	1. Select the metric. 2. Select update. 3. System will show fields to update. 4. Update the fields. 5. Select save. 6. System will display a message that “metric updated successfully”.
Extensions	1. Server is not working properly. a) Wait until server works properly. b) Refresh the page. 2. Invalid update. a) System will display a message for valid inputs.
Special Requirements	None.
Technology	1. Keyboard and mouse for input. 2. Screen for display.
Frequency	Many times a day or week.

UC-15: Delete Metric

This use case describes the functionality of delete metric. For this purpose metric should be visible on screen. If user wants to delete the metric then he/she press the delete button. As it is a critical action so system will confirm the user first. If user confirm the delete action then system will delete the metric and show the message to user that metric has been deleted successfully.

Table 2.15 *Delete Metric*

UC-15: Delete Metric	
Primary actor	Project Manager
Goal in context	User will be able to delete metric.
Pre-condition	Metric should be displayed on screen.
Post-condition	User will successfully delete the metric.
Main Success Scenario	1. Select the metric. 2. Select delete. 3. System will display message for conformation. 4. Select ok to delete. 5. System will display message that “metric deleted successfully”.
Extensions	4a) Server is not working properly. 4b) Wait until sever works properly.
Special Requirements	None.
Technology	1. Mouse and Keyboard for input 2. Screen for display output.
Frequency	Many times a day.

UC-16: Count Lines of Code

This use case describes the functionality of count lines of code. This use case is used when user wants to count the size of project. For this purpose user press count size and upload files. After uploading files user press the count button then system will show the lines of code to the user. If user upload irrelevant file the system will display the message that upload the correct file.

Table 2.16 *Count Lines of Code*

UC-16: Count Lines of Code	
Primary actor	Project Manager
Goal in context	User will count the source line of code.
Pre-condition	1. User should be login.
Post-condition	User will successfully count the source code of project.
Main Success Scenario	1. Select the source code counter. 2. Upload files for source code counting. 3. When all the files are uploaded press count. 4. Source lines of code will display on the screen.
Extensions	1. Server is not working properly. a) Wait until server works properly. b) Refresh the page. 2. Invalid files. a) System will display a message for valid files. 3. Server is not working properly. a) Wait until server works properly
Special Requirements	None.
Technology	1. Keyboard and mouse for input. 2. Screen for display.
Frequency	Many times a day or week.

UC-17: Convert Currency

This use case describes the functionality of currency converter. This use case will use to convert the amount into different currencies. For this purpose user enter the given amount, from currency type and to currency type and then press convert. If user fill the input fields correctly then system will display the converted amount to the user and if there any invalid input then system will display a message that enter the valid inputs.

Table 2.17 *Currency Converter*

UC-17: Convert Currency	
Primary actor	Project Manager
Goal in context	User will convert the given amount of currency.
Pre-condition	1. User should be login.
Post-condition	User will successfully convert the currency.
Main Success Scenario	1. Select the currency converter. 2. Enter the amount to convert. 3. Select from. 4. Select to. 5. Press convert. 6. Amount will be convert and display on screen.
Extensions	1. Server is not working properly. a) Wait until server works properly. b) Refresh the page. 2. Invalid input. a) System will display a message for valid input. 3. From currency is not selected. a) System will display message to select from currency. 4. To currency is not selected. a) System will display message to select to currency. 5. There is no internet connection. a) Wait until internet is available. 6. Too much time taken and error occurred time out. a) Try again or wait for the better internet connection.
Special Requirements	None.
Technology	1. Keyboard and mouse for input. 2. Screen for display.
Frequency	Many times a day or week.

UC-18: Search Metric/Filter Metric

This use case describes the functionality of search metric or filter metric. If user wants to search metric by project name then he/she will enter the project name in the search bar and system will display the results corresponding to the project name. If user wants to filter the metrics by project's status then he/she will enter the status in the filter bar then system will display the related results.

Table 2.18 *Search Metric/ Filter Metrics*

UC-18: Search Metric/Filter Metric	
Primary actor	Project Manager
Goal in context	User will search/Filter the metrics.
Pre-condition	1. User should be login. 2. View metric screen appeared on the screen.
Post-condition	User will search/filter metrics successfully.
Main Success Scenario	1. User selects the search/filter bar. 2. User inputs the required data. 3. Search/Filter result display on screen.
Extensions	2. Server is not working properly. a) Wait until server works properly
Special Requirements	None.
Technology	1. Mouse and Keyboard for input 2. Screen for display output
Frequency	Many times a day.

2.2.2.3 System Sequence Diagrams

[8] A system sequence diagram (SSD) is a picture that shows, for a particular scenario of a use case, the events that external actors generate, their order, and inter-system events. All systems are treated as a black box; the emphasis of the diagram is events that cross the system boundary from actors to systems.

System sequence diagram for PCE are given below:

Calculate Metric:

Figure show the system sequence diagram for calculate metric.

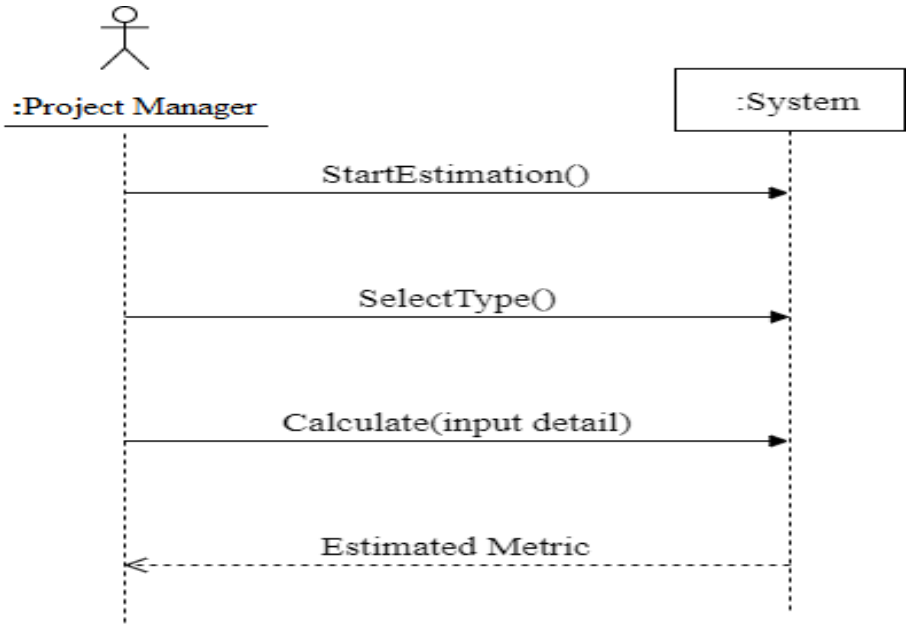


Figure 2.2 *SSD for Calculate Metric*

View Actual Metric:

Figure 2.3 show the system sequence diagram for view actual metric.

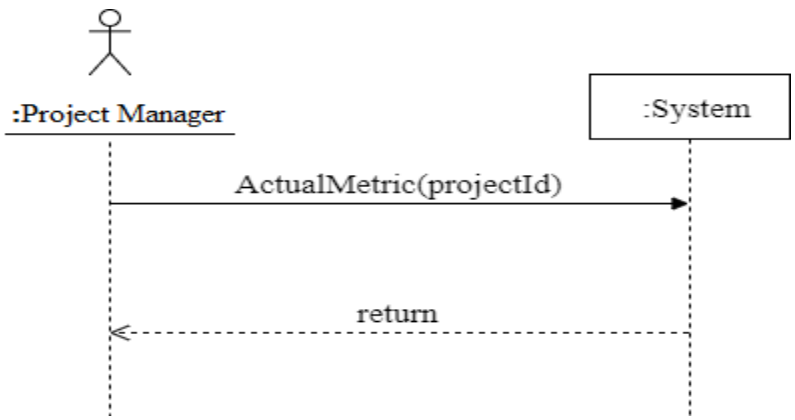


Figure 2.3 *SSD for View Actual Metric*

Add Estimated Metric to Project:

Figure 2.4 show the system sequence diagram for add estimated metric to project.

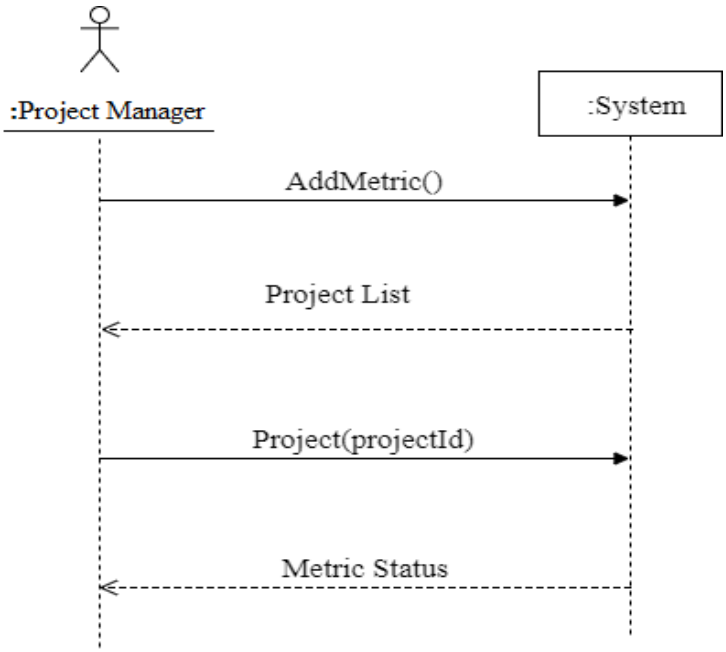


Figure 2.4 *SSD for Add Estimated Metric to Project*

Re-Estimate Metric:

Figure 2.5 show the system sequence diagram for re-estimate metric.

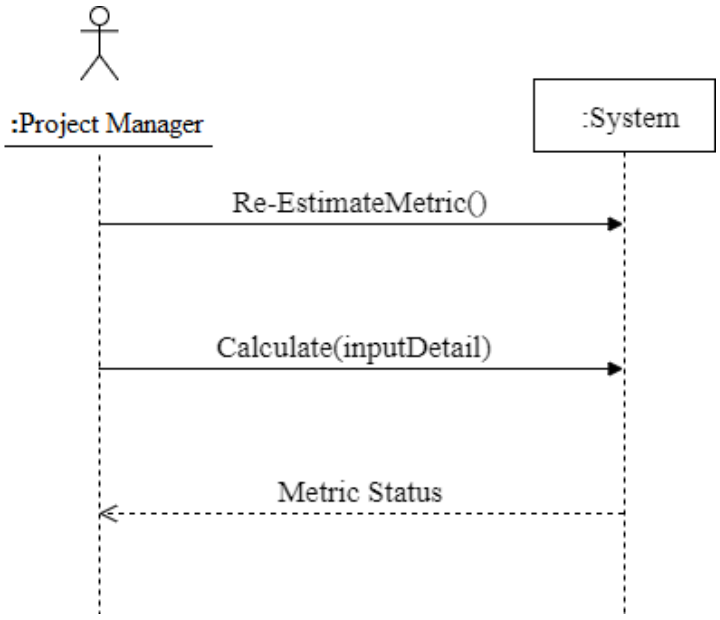


Figure 2.5 *SSD for Re-Estimate Metric*

View Metric:

Figure 2.6 show the system sequence diagram for view metric.

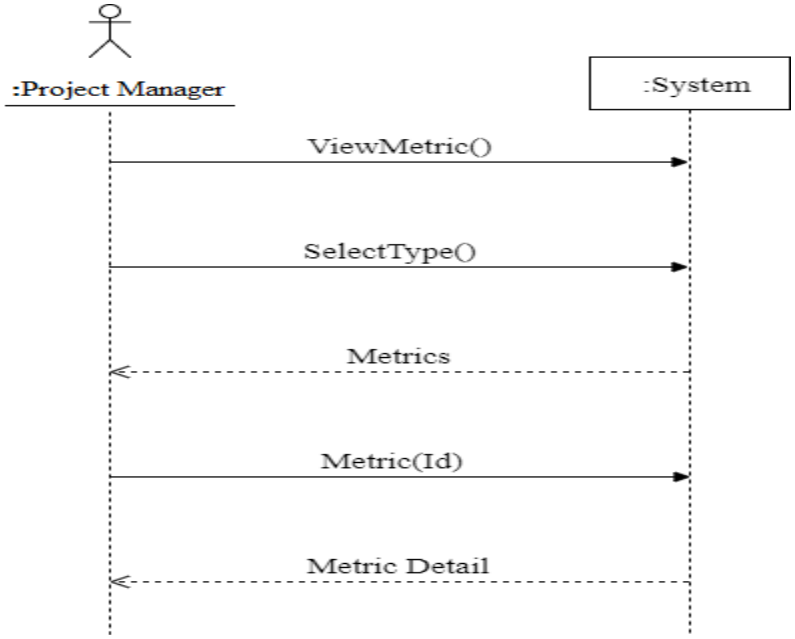


Figure 2.6 *SSD for View Metric*

Update Actual Metric:

Figure 2.7 show the system sequence diagram for update actual metric.

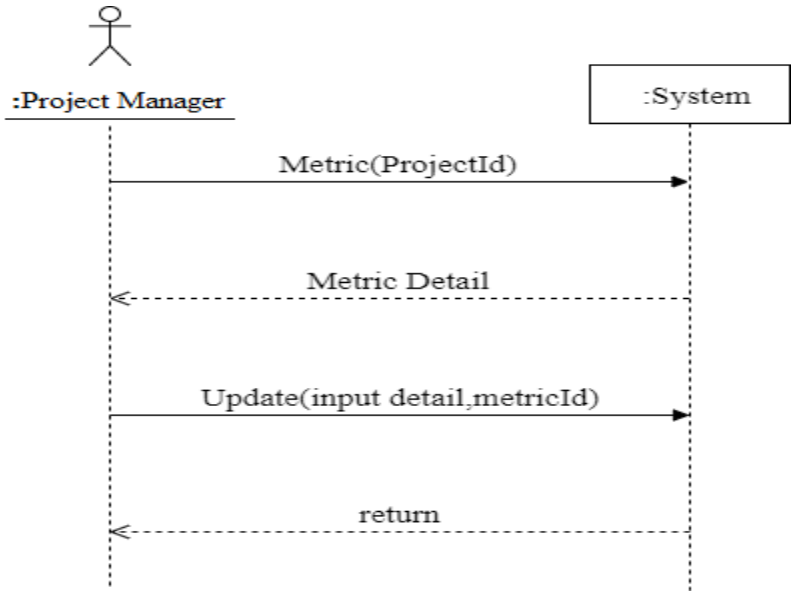


Figure 2.7 *SSD for Update Actual Metric*

Delete Metric:

Figure 2.8 show the system sequence diagram for delete metric.

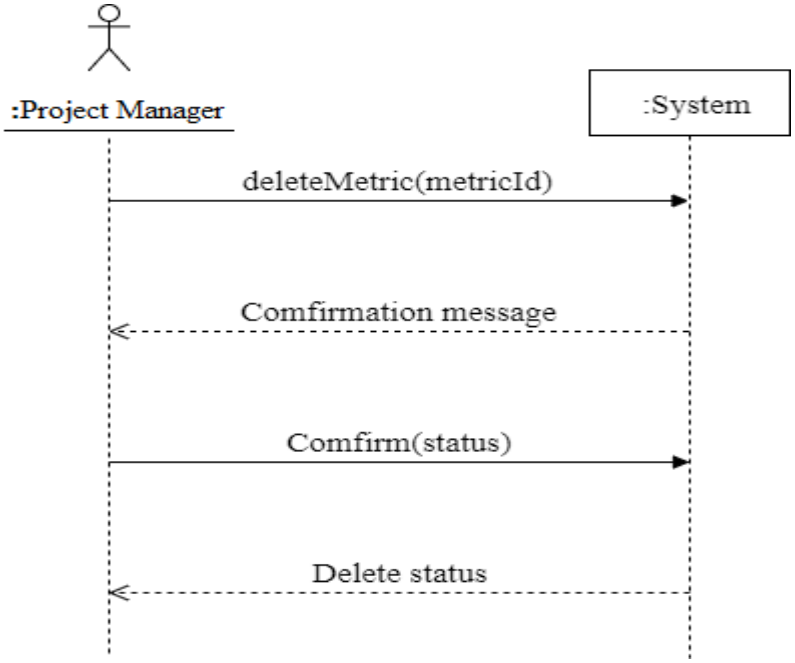


Figure 2.8 *SSD for Delete Metric*

Count Lines of Count:

Figure 2.9 show the system sequence diagram for count lines of code

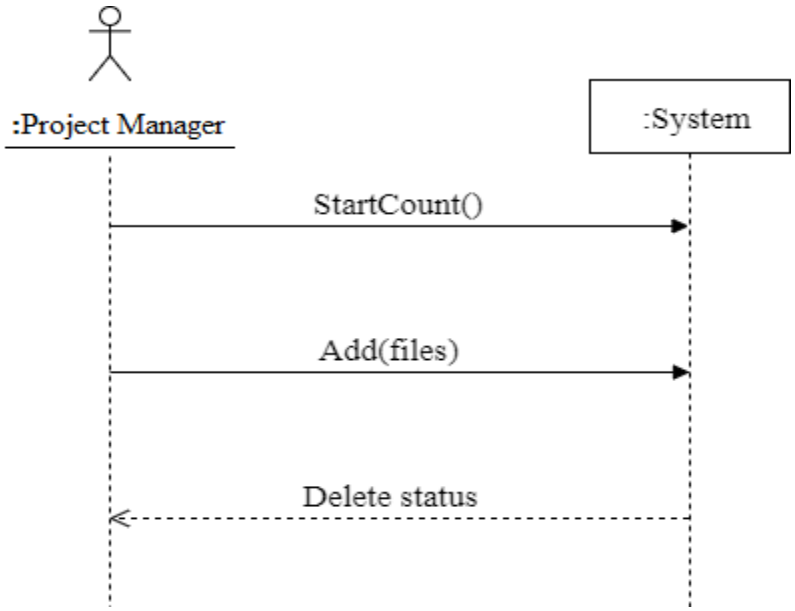


Figure 2.9 *SSD for Count Lines of Code*

Convert Currency:

Figure 2.10 show the system sequence diagram for convert currency.

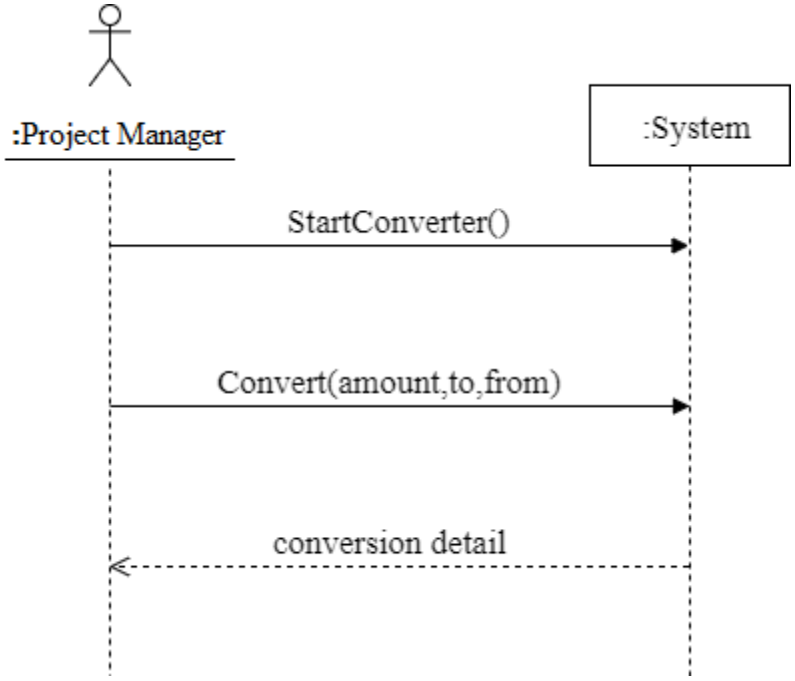


Figure 2.10 *SSD for Currency Converter*

Search/Filter Metric:

Figure 2.11 show the system sequence diagram for search/filter metric.

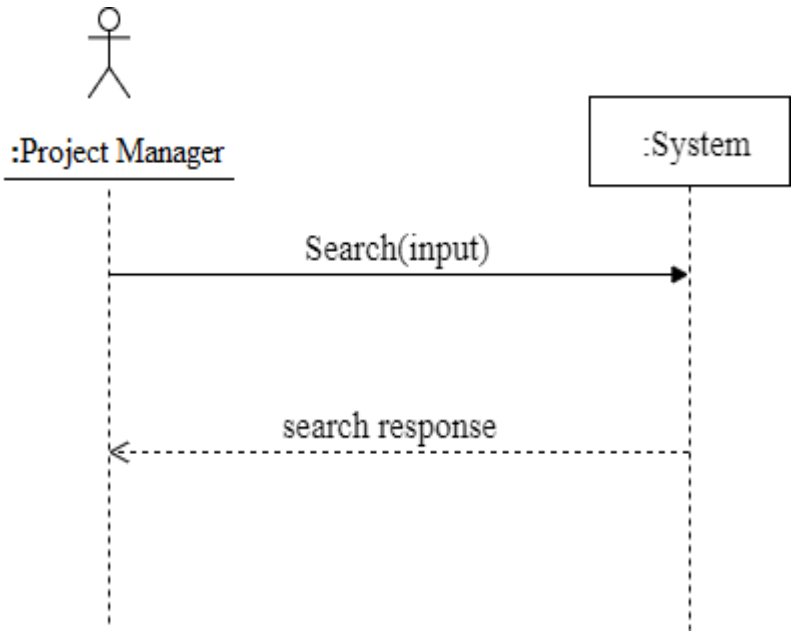


Figure 2.11 *SSD for Search/ Filter Metrics*

2.2.3 Software System Attributes

Software system attributes consists of the reliability, availability, security, maintainability, portability and performance.

2.2.3.1 Reliability

The probability of failure is zero. The system shall never crash, other than as the result of an operating system error. If there will any error occur then it will display an appropriate messages such that user will not feel any ambiguity while using this product. So over all reliability of system is approximately is more than 90-95%.

2.2.3.2 Availability

For availability it requires the following conditions:

- Internet connection
- Running server
- Database connectivity

Once it will in running condition then it will available 24 hours a day under above conditions.

2.2.3.3 Security

- Secure database will used which required password before connection.
- Password send to the database will be in encrypted form.
- User should be login before performing any tasks.
- Each type of user will have limited access to modules. For example project manager will be able to register the project while software developer will not be able to add project

2.2.3.4 Maintainability

If there will any fault detects then it will easy to correct fault and modified because source code will be written in a structured way. Code will be well commented such that it will help the software developer to understand the code easily and will be able to make changes or upgrade. As software maintenance is a very critical operation so it requires a structured and well commented code.

2.2.3.5 Portability

The PCE is the web based so it ensures the portability. It will require web browser to run. So it could run on any operating system and required no any special software except web browser.

2.2.3.6 Performance

As PCE is a web based application so its performance could affect by the internet speed. If there will no problem internet then it will display results of query from user sides quickly. Usually results will show in millisecond or maximum 3 to 4 seconds. New screen will come quickly when user perform any action.

2.2.4 **Database Requirements**

MySQL database will use for this system to store all baselines of project. The project manager will calculate a metrics and can compare this metrics with the previous stored metrics and make final decision for allocating the resources, time. Each team member will able to update progress at the end of day. So at the end of project, project manager will add the final metrics of the project. Project manager will also able to generate reports regarding the project metrics and will able to see these reports in a graphical view for comparison purpose.

Database design diagram is shown in figure (Figure 2.12) in which there are nine tables. Values will store in these tables related to their metric type with project id and metric id. Project will implement in some languages so language table will maintain Lines of code per function point and use case point for estimation in future. Actual metric table will store the data related to the actual metric of project which can be compared with the estimated metrics. There are four tables related to the estimated metrics using different models such as Function Point, Use Case Point, Empirical Model and Cocomo Model. Estimated metrics According to their type will store in these tables and can compare with the actual metric to check the deviation in estimation. Project size table only stored the static values of project size in such a way that Basic cocomo model required project size for estimation such as small, medium, large etc. Each type will estimate with different equations. So this project size metric is maintain by the organization according to their need.

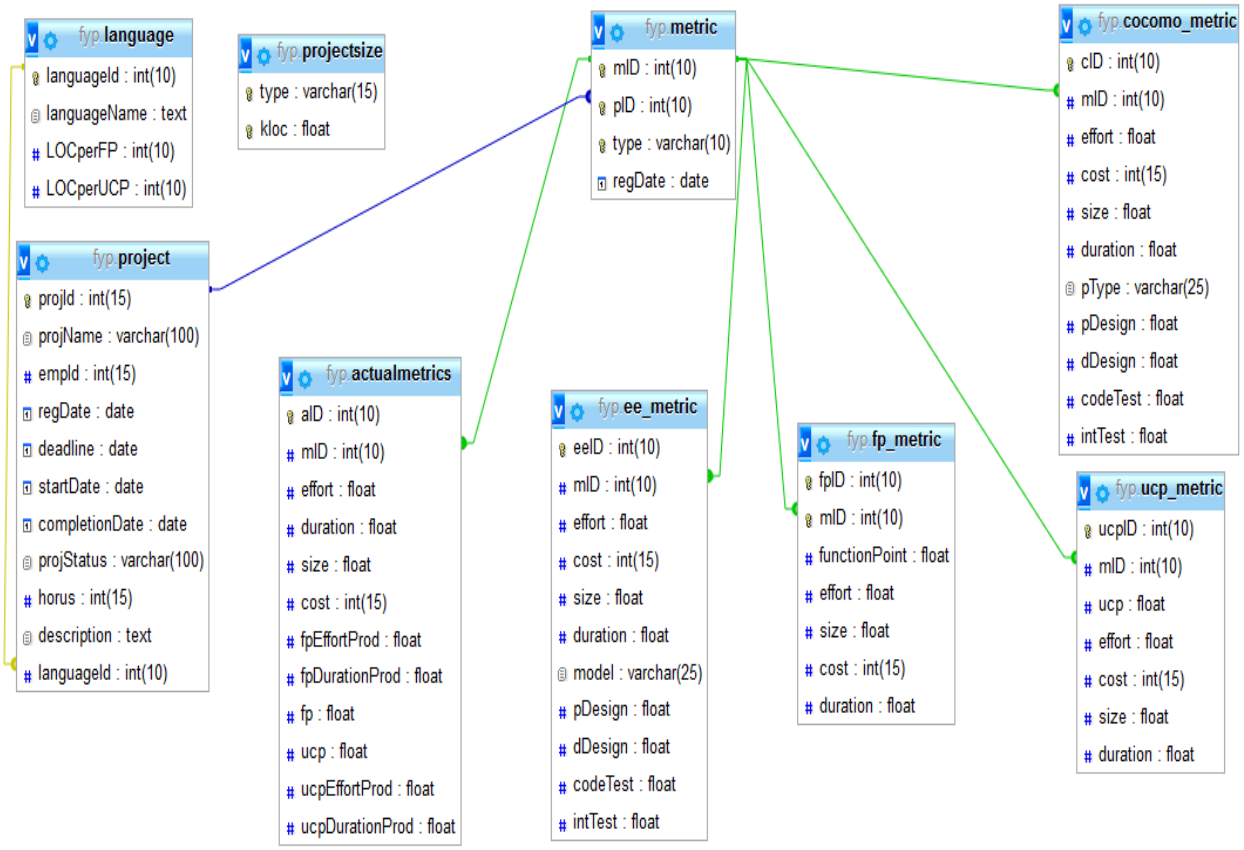


Figure 2.12 *Database Design*

Chapter 03

Software Design Description

3.1 INTRODUCTION

The software design document provides the design detail of PCE and tracks the necessary information required to effectively define architecture and system design Product Overview. This document will help the developer to understand the inner workings of the PCM.

3.1.1 Design Overview

The software design document provides design details of PCE. It includes the following diagrams which will explain the design overview of the system.

3.1.1.1 Domain Model

A domain model is visual representation of real world objects in a domain of interest. [8]They are also related to conceptual entity relationship models, which are capable of showing purely conceptual views of domains, but that have been widely re-interpreted as data models for database design. Domain model for the project cost estimator is given in the following figure (Figure 3.1)

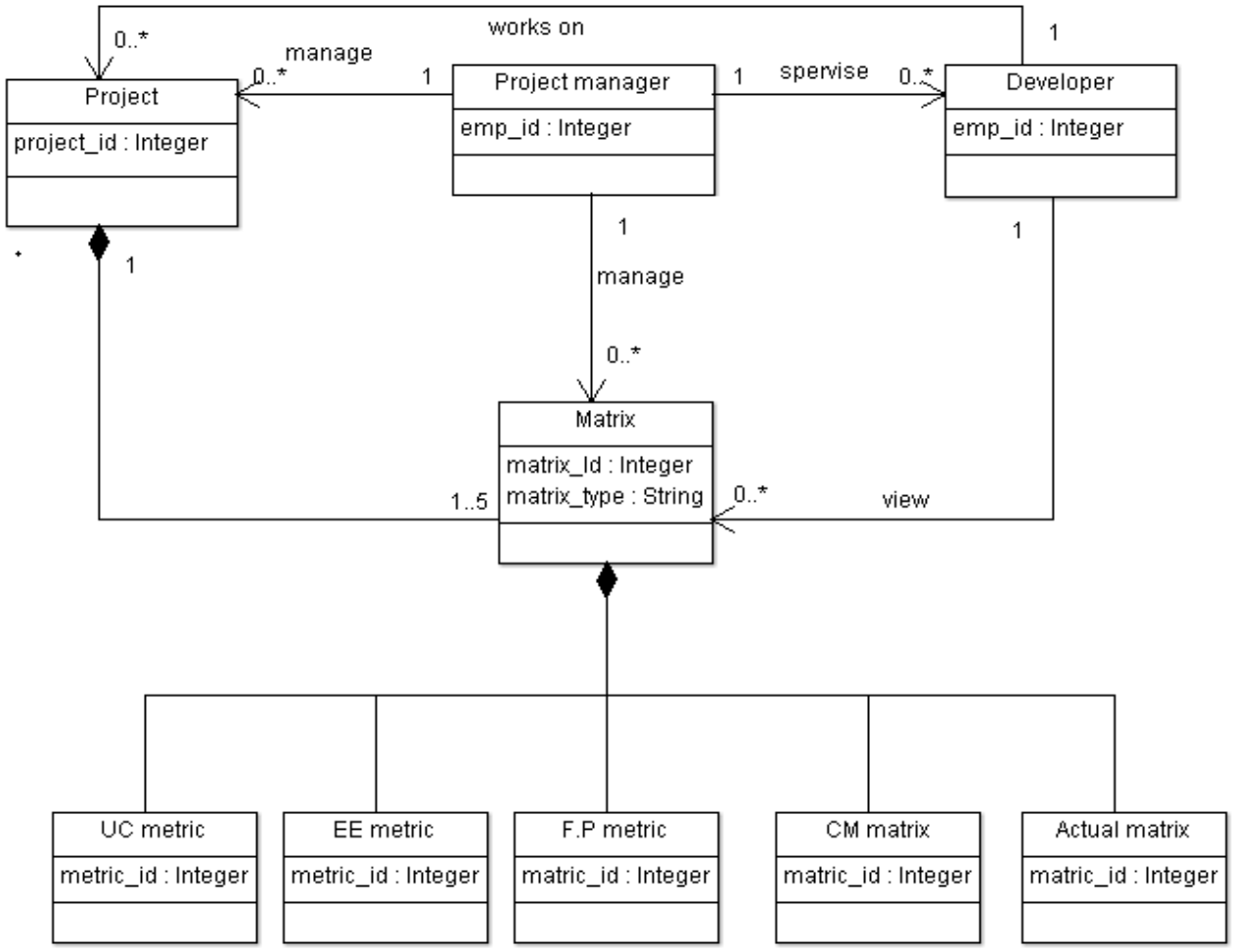


Figure 3.1 *Domain Model*

3.1.1.2 Activity Diagrams

An activity is a state of doing something. The activity diagram describes the sequencing of activities. Activity diagram depicts the dynamic behavior of a system or part of a system through the flow of control between actions that the system performs. It is similar to a flowchart except that an activity diagram can show concurrent flows.

Activity diagrams for PCE are given below:

Calculate Metric:

Figure 3.2 shows the activity diagram for calculate metric.

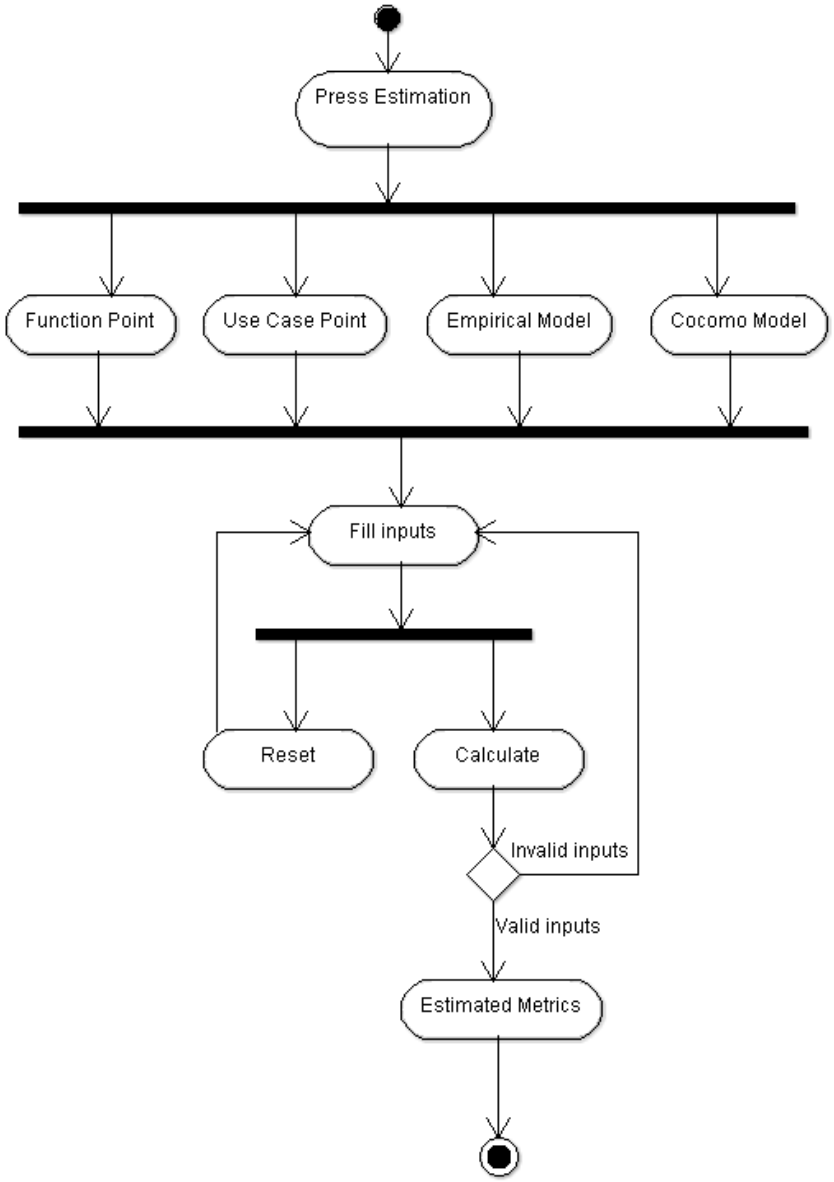


Figure 3.2 *Activity Diagram for Calculate Metric*

View Metric:

Figure 3.3 shows the activity diagram for view metric.

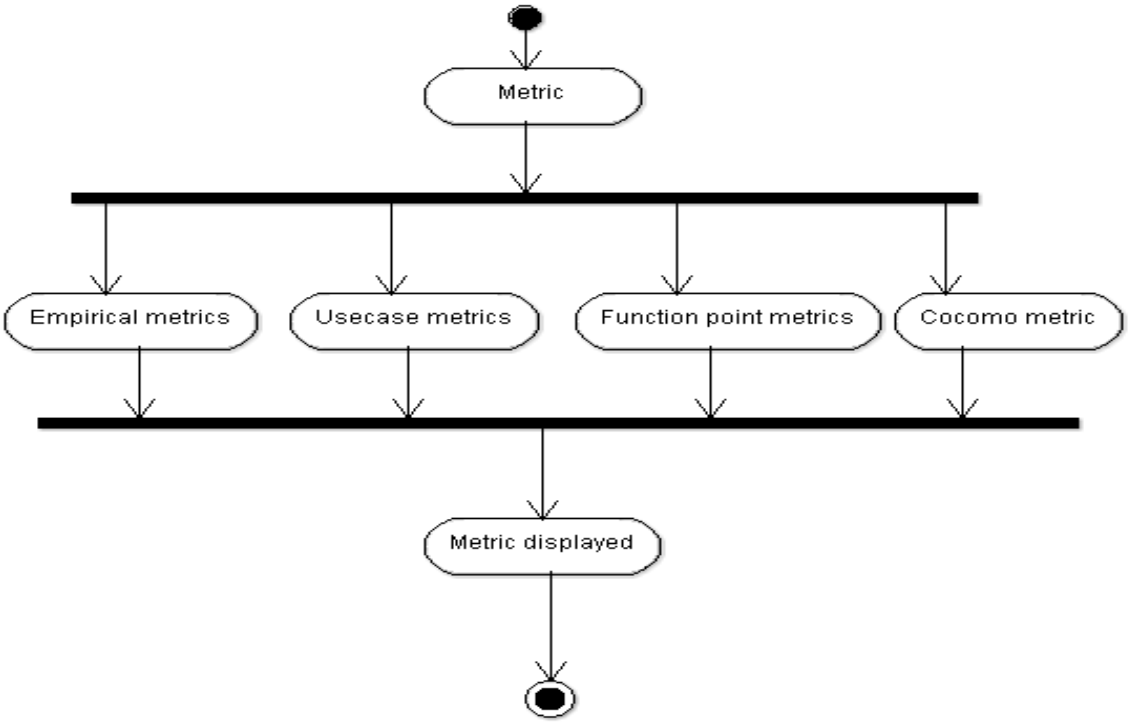


Figure 3.3 *Activity Diagram View Metric*

Update Actual Metric:

Figure 3.4 shows the activity diagram for update actual metric

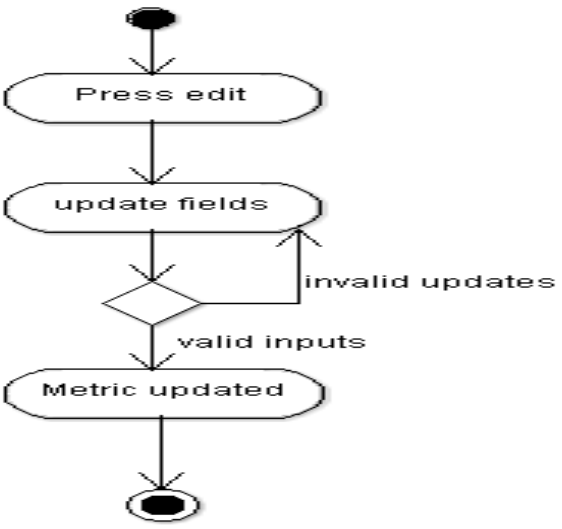


Figure 3.4 *Activity Diagram for Update Actual Metric*

Re-Estimate Metric:

Figure 3.5 shows the activity diagram for re-estimate metric.

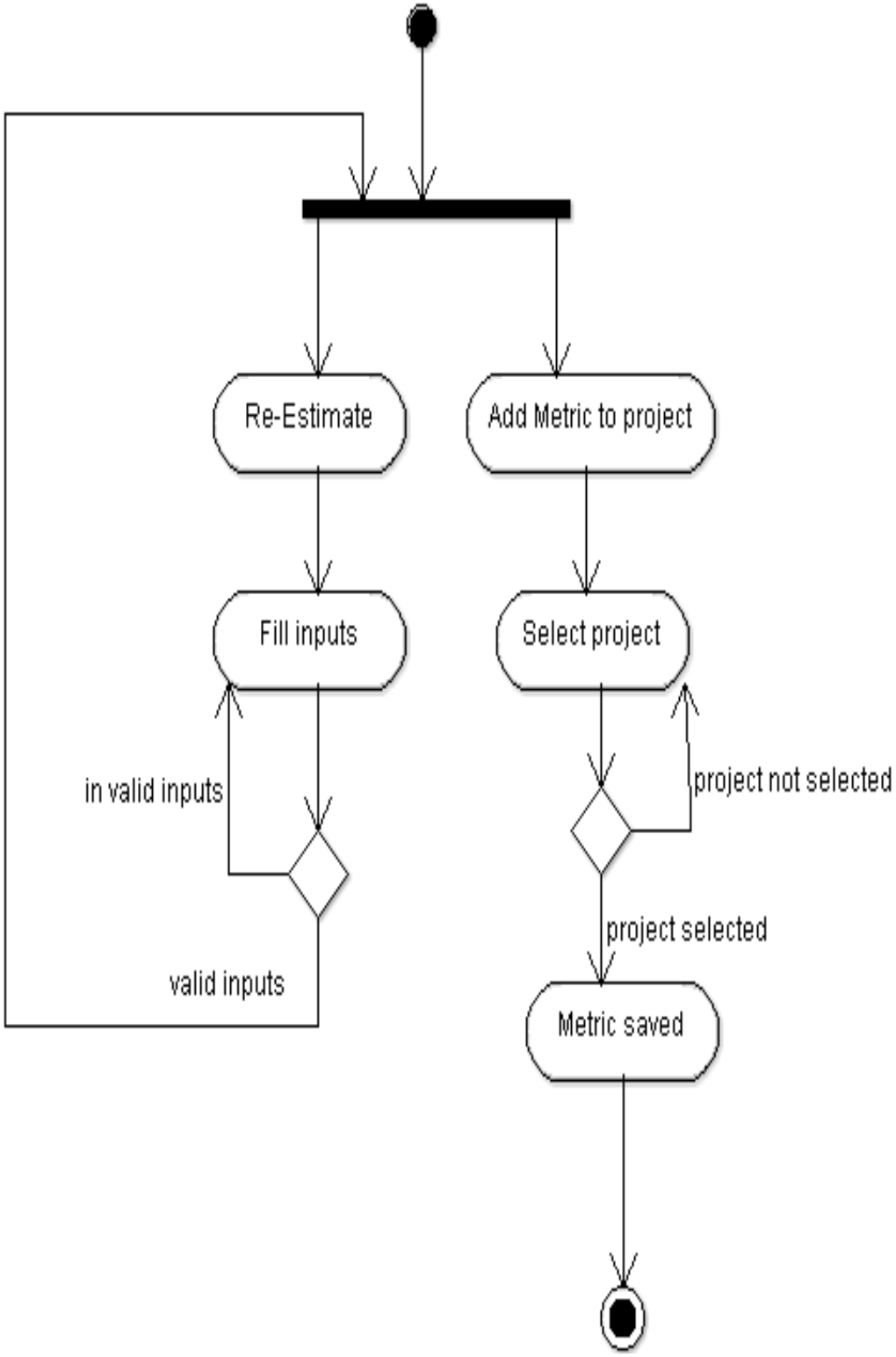


Figure 3.5 *Activity Diagram Re-Estimate Metric*

Search/Filter Metric:

Figure 3.6 shows the activity diagram for search/filter metric.

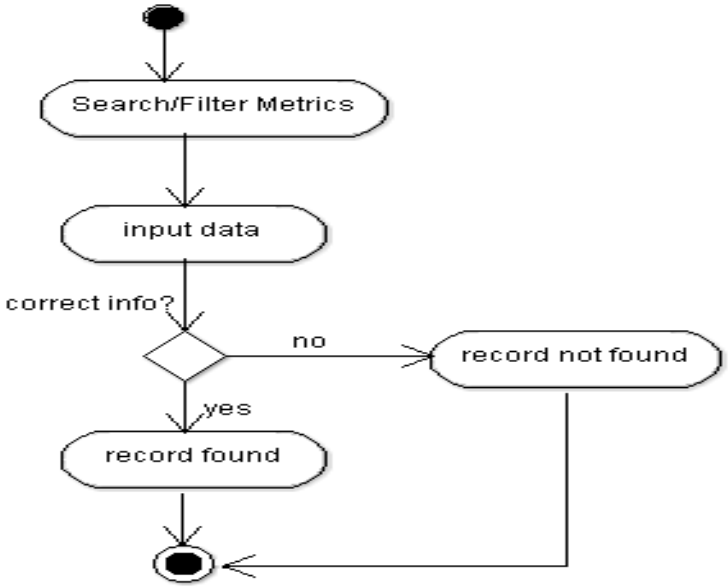


Figure 3.6 *Activity Diagram for Search/ Filter Metrics*

Convert Currency:

Figure 3.7 shows the activity diagram for convert currency.

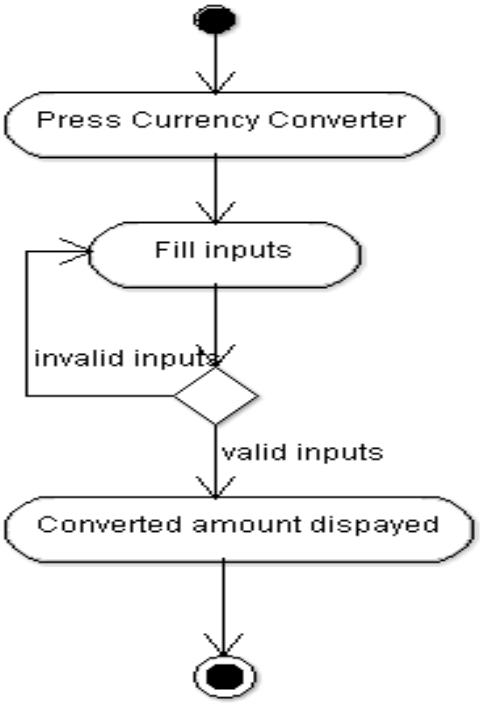


Figure 3.7 *Activity Diagram for Currency Converter*

Count Lines of Code:

Figure 3.8 shows the activity diagram for count lines of code.

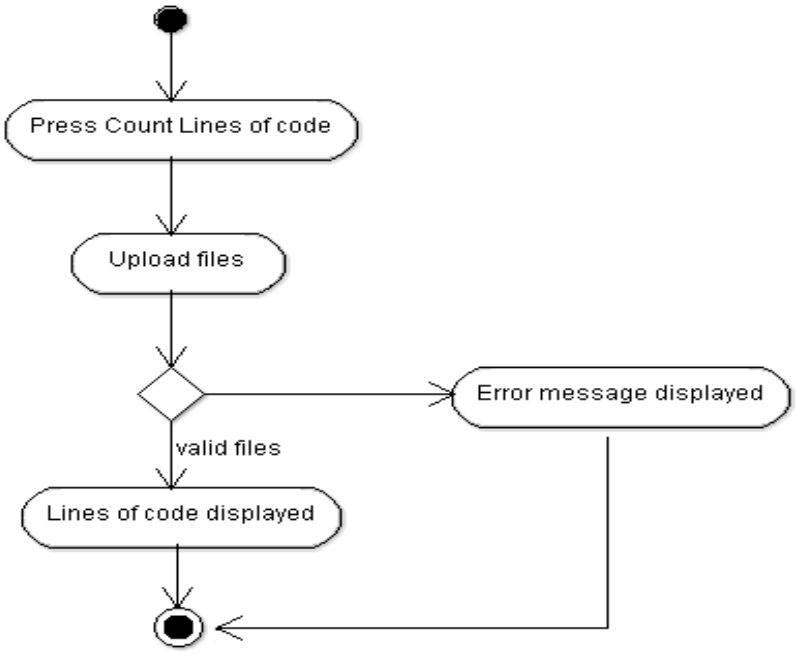


Figure 3.8 *Activity Diagram Code Counter*

Delete Metric:

Figure 3.9 shows the activity diagram for delete metric.

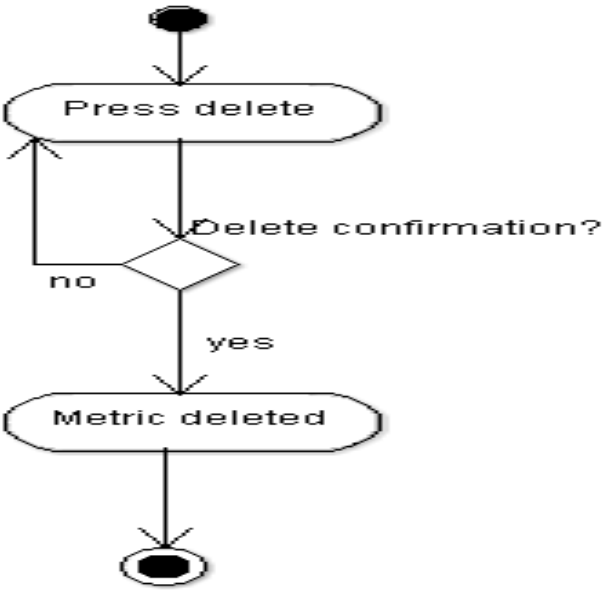


Figure 3.9 *Activity Diagram Delete Metric*

3.2 SYSTEM ARCHITECTURAL DESIGN

[2] A system architecture is a conceptual model that defines the structure, behaviour, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviours of the system.

3.2.1 Chosen System Architecture

I chose three tier architecture. Three tier architecture is a client server architecture in which first layer is presentation layer, second layer is business logic or application layer and the third layer is data layer which consist of database. Architecture diagram for Project Cost Estimator is given in Figure 3.10.

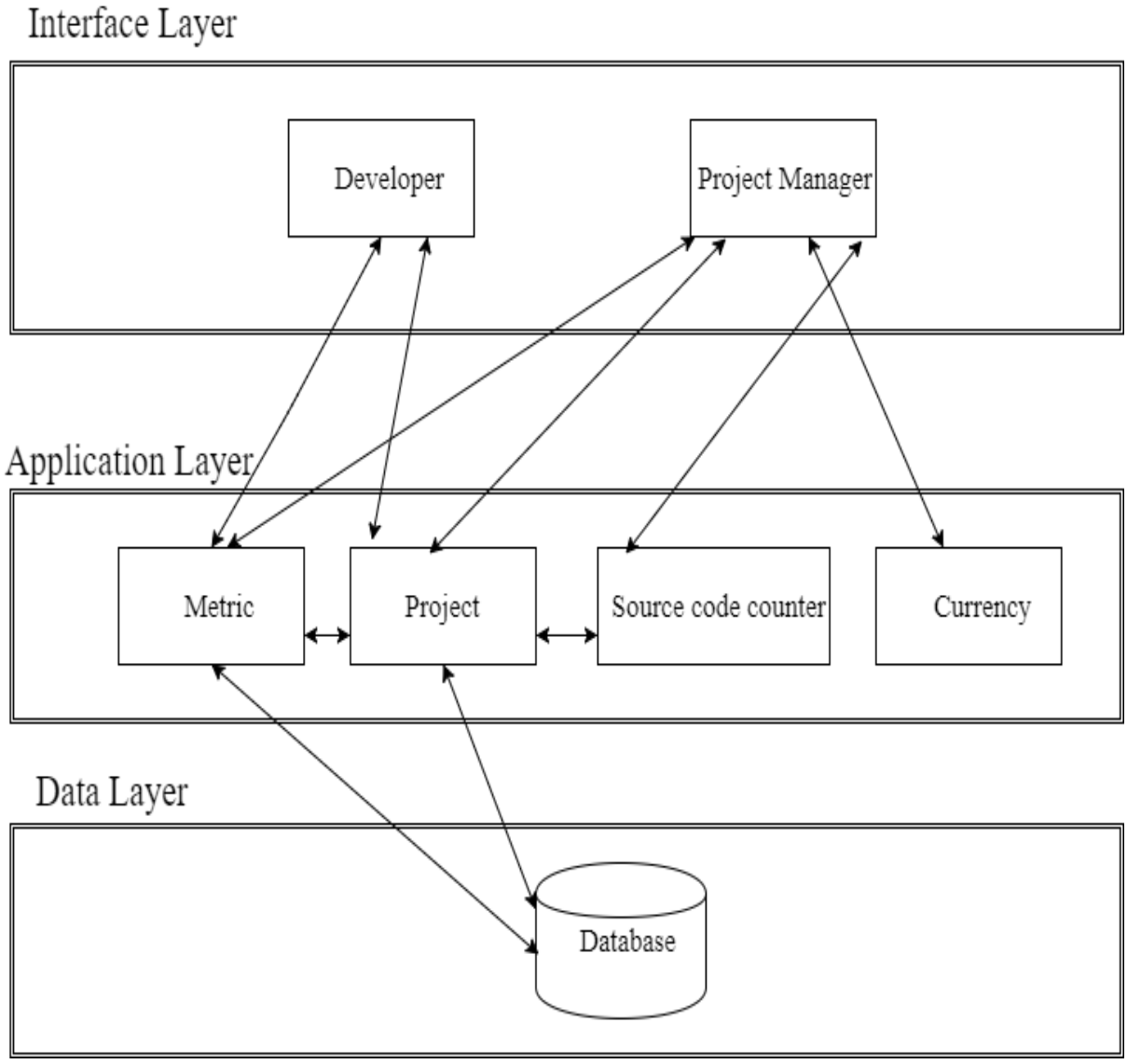


Figure 3.10 *System Architecture Diagram*

3.2.2 System Interface Description

System interface description addresses the composition and interaction of Systems. System interface description diagram in given in Figure 3.11.

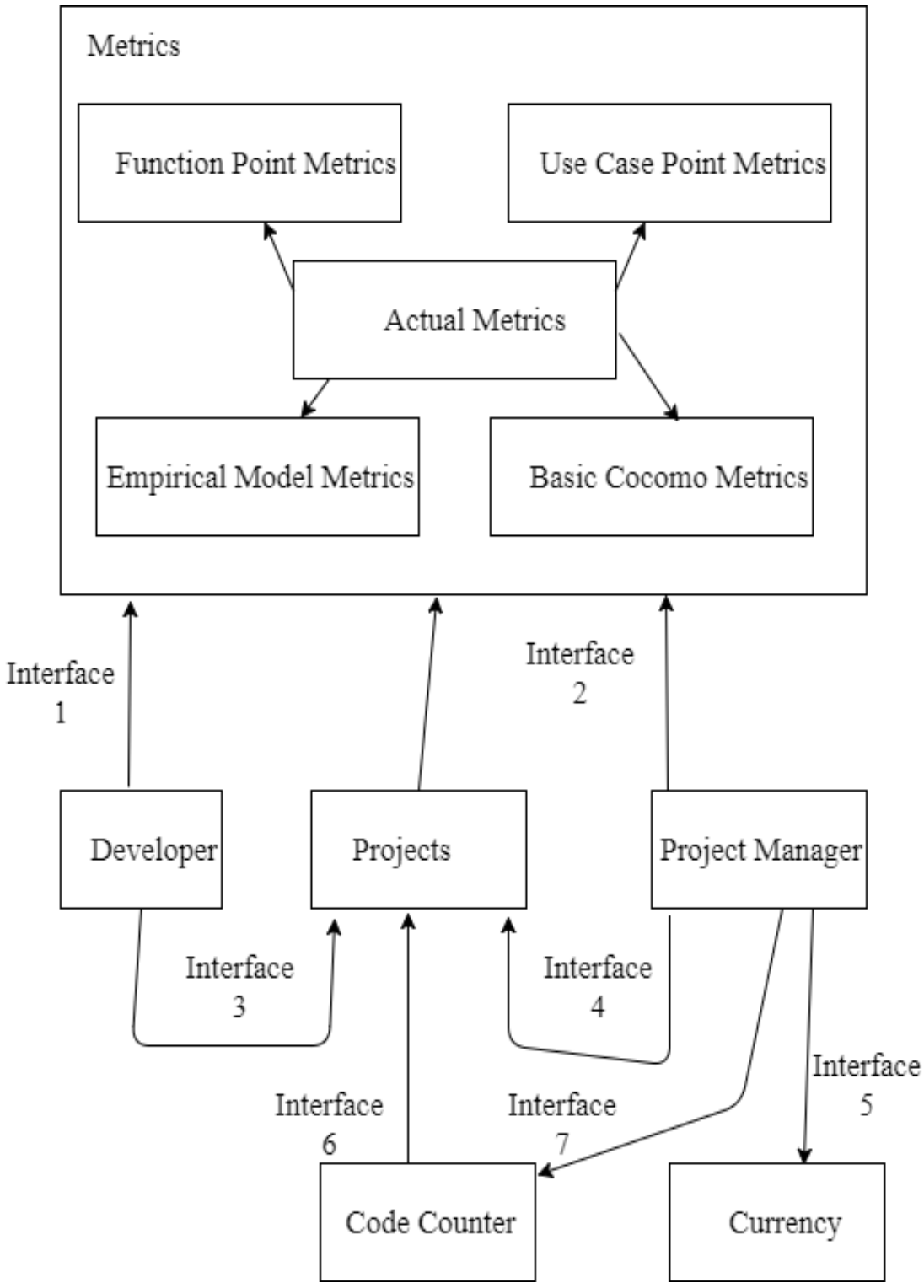


Figure 3.11 *System Interface Description Diagram*

3.3 DETAILED DESCRIPTION OF COMPONENTS

Detailed description of components describes the different components of PCE, their interaction and dependencies. Detailed description of components is given in diagram Figure3.12.

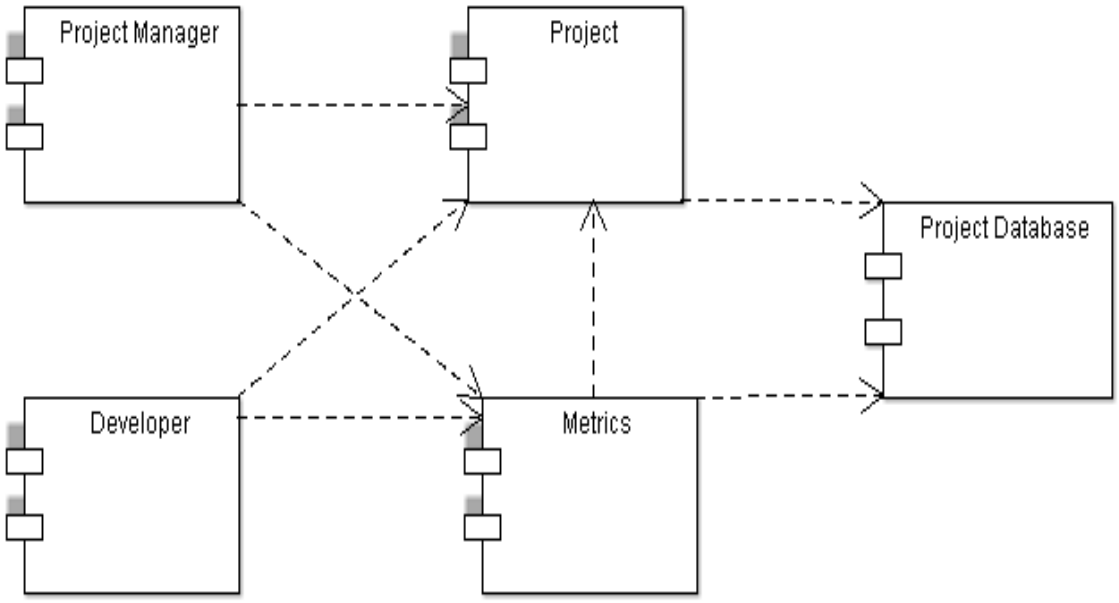


Figure 3.12 *Component Diagram*

3.3.1 Project Manager

Project manager will be able to calculate metrics and add them to the project. Project manager will view metrics, update metrics and delete metrics.

3.3.2 Developer

Developer will only able to view metrics and works on project.

3.3.3 Metrics

The component metrics consists of many classes such as FP metrics, UC metrics, OO metrics, LOC metrics and actual metrics.

3.3.4 Project

Project component consists of multiple classes such as tasks and sub tasks. Project has multiple metrics. Project can have only one actual metrics and can be multiple estimated metrics.

Chapter 04

Software Algorithms and Equations

4.1 INTRODUCTION

This chapter consists of description of algorithms and equations which are used in the estimation of project. There are different models which are used such as Function point, Use case point, Empirical model and Cocomo model.

4.1.1 Function Point

This model is taken from the book [8] **Software Engineering A Practitioner's Approach Seventh edition** written by **Roger S. Pressman.**

The function point (FP) metric can be used effectively as a means for measuring the functionality delivered by a system. Using historical data, the FP metric can then be used to estimate cost, size, effort and duration.

Function points are derived using an empirical relationship based on countable (direct) measures of software's information domain and qualitative assessments of software complexity. Information domain values are defined in the following manner:

Number of external inputs (EIs)

Each external input originates from a user or is transmitted from another application and provides distinct application-oriented data or control information.

Number of external outputs (EOs)

Each external output is derived data within the application that provides information to the user. In this context external output refers to reports, screens, error messages, etc.

Number of external inquiries (EQs)

An external inquiry is defined as an online input that results in the generation of some immediate software response in the form of an online output.

Number of internal logical files (ILFs)

Each internal logical file is a logical grouping of data that resides within the application's boundary and is maintained via external inputs.

Number of external interface files (EIFs)

Each external interface file is a logical grouping of data that resides external to the application but provides information that may be of use to the application.

Once these data have been collected, the table in Figure 4.1 is completed and a complexity value is associated with each count. Organizations that use function point methods develop criteria for determining whether a particular entry is simple, average, or complex. Nonetheless, the determination of complexity is somewhat subjective.

To compute function points (FP), the following relationship is used:

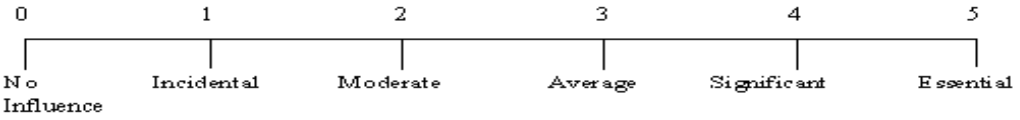
FP =count total x [0.65 +0.01 x $\sum$ (Fi)]

Where count total is the sum of all FP entries obtained from Figure 4.1.

Information Domain Value	Count		Weighting factor Simple	Average	Complex		
External Inputs (EIs)		×	3	4	6	=	
External Outputs (EOs)		×	4	5	7	=	
External Inquiries (EQs)		×	3	4	6	=	
Internal Logical Files (ILFs)		×	7	10	15	=	
External Interface Files (EIFs)		×	5	7	10	=	
Count total							

Figure 4.1 Function Point Inputs

Each of the following questions is answered using a scale that ranges from 0 (not important or applicable) to 5 (absolutely essential).



The Fi (i =1 to 14) are value adjustment factors (VAF) based on responses to the following questions:

1. Does the system require reliable backup and recovery?

2. Are specialized data communications required to transfer information to or from the application?

3. Are there distributed processing functions?

4. Is performance critical?

5. Will the system run in an existing, heavily utilized operational environment?

6. Does the system require online data entry?

7. Does the online data entry require the input transaction to be built over multiple screens or operations?

8. Are the ILFs updated online?

9. Are the inputs, outputs, files, or inquiries complex?

10. Is the internal processing complex?

11. Is the code designed to be reusable?

12. Are conversion and installation included in the design?

13. Is the system designed for multiple installations in different organizations?

14. Is the application designed to facilitate change and ease of use by the user?

4.1.2 Use Case Point

This model is taken from the research paper named [3] **Project Estimation with Use Case Points** written by **Roy K. Clemmons**.

The Use Case Point (UCP) method provides the ability to estimate person month a software requires from its use case. The UCP method analyzes the use case actors, scenarios, and various technical and environmental factors and abstracts them into an equation.

The UCP equation is composed of three variables:

1. Unadjusted Use Case Points (UUCP).
2. The Technical Complexity Factor (TCF).
3. The Environment Complexity Factor (ECF).

UCP = UUCP * TCF * ECF

Additionally, when productivity is included as a coefficient that expresses time, the equation can be used to estimate the number of person month needed to complete a project. Here is the complete equation with a Productivity Factor (PF) included:

UCP = UUCP * TCF * ECF * PF

The necessary steps to generate the estimate based on the UCP method are the following:

1. Determine and compute the UUCPs.
2. Determine and compute the TCFs.
3. Determine and compute the ECFs.
4. Determine the PF.
5. Compute the estimated number of months.

UUCPs:

UUCPs are computed based on two computations:

1. The Unadjusted Use Case Weight (UUCW) based on the total number of activities (or steps) contained in all the use case scenarios.

2. The Unadjusted Actor Weight (UAW) based on the combined complexity of all the actors in all the use cases.

UUCP=UUCW+UAW

UUCW:

The UUCW is derived from the number of use cases in three categories:

- Simple
- Average
- Complex.

Table 4.1 Use Case Classification

Use Case Category	Category	Weight	Number of Use Case	Result
Simple	Simple user interface. Touches only a single database entity. Its success Scenario as three steps or less. Its implementation involves less than five classes.	5	X	5X
Average	More interface design. Touches two or more database entities. Between four and seven steps. Its implementation involves between five and 10 classes.	10	Y	10Y
Complex	Complex user interface or processing. Touches three or more database entities. More than seven steps. Its implementation involves more than 10 classes.	15	Z	15Z
			Total UUCW=	**5X+10Y+15Z**

UAW:

In a similar manner, the Actor Types are classified as

- Simple
- Average
- Complex

Table 4.2 Actor Classification

Actor Type	Category	Weight	Number Actors	Result
Simple	The actor represents another system with a defined application programming interface	1	X	X
Average	The actor represents another system Interacting through a protocol, like Transmission Control Protocol/Internet Protocol.	2	Y	2Y
Complex	The actor is a person interacting via an interface	3	Z	3Z
			Total UAW=	**X+2Y+3Z**

TCFs

Thirteen standard technical factors exist to estimate the impact on productivity that various technical issues have on a project. Each factor is weighted according to its relative impact.

TCF = 0.6 + (.01 * Technical Total Factor)

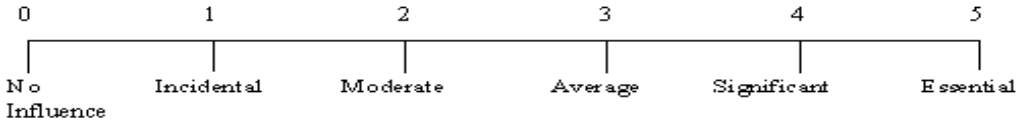


Table 4.3 UCP Technical Factor

Technical Factor	Description	Weight	Perceived complexity	Calculated factor
T1	Distributed System	2		
T2	Performance	1		
T3	End User Efficiency	1		
T4	Complex Internal Processing	1		
T5	Reusability	1		
T6	Easy to Install	0.5		
T7	Easy to use	0.5		
T8	Portability	2		
T9	Easy to Change	1		
T10	Concurrency	1		
T11	Special Security Features	1		
T12	Provides Direct Access for Third Parties	1		
T13	Special User Training Facilities Are Required	1		
			Technical Total Factor	

ECFs:

The ECF provides a concession for the development team's experience. More experienced teams will have a greater impact on the UCP computation than less experienced teams.

ECF = 1.4 + (-0.03 * Environmental Total Factor)

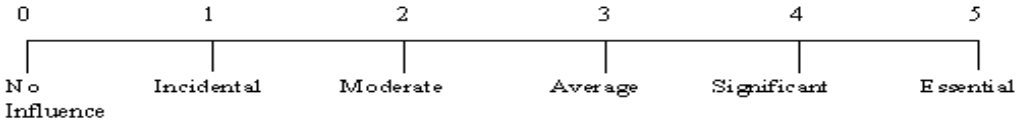


Table 4.4 UCP Environmental Factors

Environmental Factor	Description	Weight	Perceived complexity	Calculated factor
E1	Familiarity with UML*	**1.5**		
E2	Part-Time Workers	**-1**		
E3	Analyst Capability	**0.5**		
E4	Application Experience	**0.5**		
E5	Object-Oriented Experience	**1**		
E6	Motivation	**1**		
E7	Difficult Programming Language	**-1**		
E8	Stable Requirements	**2**		
			Environment total factor	

PF:

The PF is the ratio of development person month needed per use case point. Statistics from past projects provide the data to estimate the initial PF.

Estimated Months:

The total estimated number of months for the project is determined by multiplying the UCP by the PF.

Total Estimate = UCP * PF

4.1.3 Empirical Model Estimation

A typical estimation model is derived using regression analysis on data collected from past software projects. This model is taken from the book [8] **Software Engineering A Practitioner's Approach Seventh edition** written by **Roger S. Pressman.** The overall structure of such models takes the form

$$\mathbf{E = A + B \times (KLOC)^C}$$

Where A, B and C are empirically derived constants, E is effort in person-months, and the estimation variable (either LOC or FP). Empirical model equations are given in Figure 4.2.

KLOC for upper equation will measure using function point technique.

$E = 5.2 \times (\text{KLOC})^{0.91}$	Walston-Felix model
$E = 5.5 + 0.73 \times (\text{KLOC})^{1.16}$	Bailey-Basili model
$E = 3.2 \times (\text{KLOC})^{1.05}$	Boehm simple model
$E = 5.288 \times (\text{KLOC})^{1.047}$	Doty model for KLOC > 9

Figure 4.2 *Empirical Model Equations*

A quick examination of these models indicates that each will yield a different result for the same values of LOC or FP.

4.1.4 Basic Cocomo Model Estimation

This model is taken from the webpage having reference of the book **Software Engineering A Practitioner's Approach Fourth edition** written by **Roger S. Pressman.**

There are four steps that the user needs to follow, which are:

Step 1: The user will estimate the size of project using function point technique. This model outputs the function points.

Step 2: The user will select the language and then function points are converted into lines of code using LOC of that language per function point. This step will outputs the lines of code.

Step 3: This is the final step of the basic COCOMO model. The end user has to select one of the three (3) types of modes, which are

- Organic
- Semi-detached
- Embedded.

Organic Mode: Relatively small, simple software projects in which a small teams with good application experience work to a set of less than rigid requirement. The equation for the Effort (E) and Development time (D) for this model are:

- **E = 2.4 * (KLOC)^1.05**
- **D = 2.5 * (E)^0.38**

Semi-Detached Mode: An intermediate (in size and complexity) software project in which teams with mixed experience levels must meet a mix of rigid and less than rigid requirements. The equation for the Effort (E) and Development time (D) for this model are:

- **E = 3.0 * (KLOC)^1.12**
- **D = 2.5 * (E)^0.35**

Embedded Mode: A software project that must be developed within a set of tight hardware, software and operational constraints. The equation for the Effort (E) and Development time (D) for this model are:

- **E = 3.6 * (KLOC)^1.20**
- **D = 2.5 * (E)^0.32**

4.1.5 Estimation Process Cycle:

This cycle shows the estimation process using historical data of the organization.

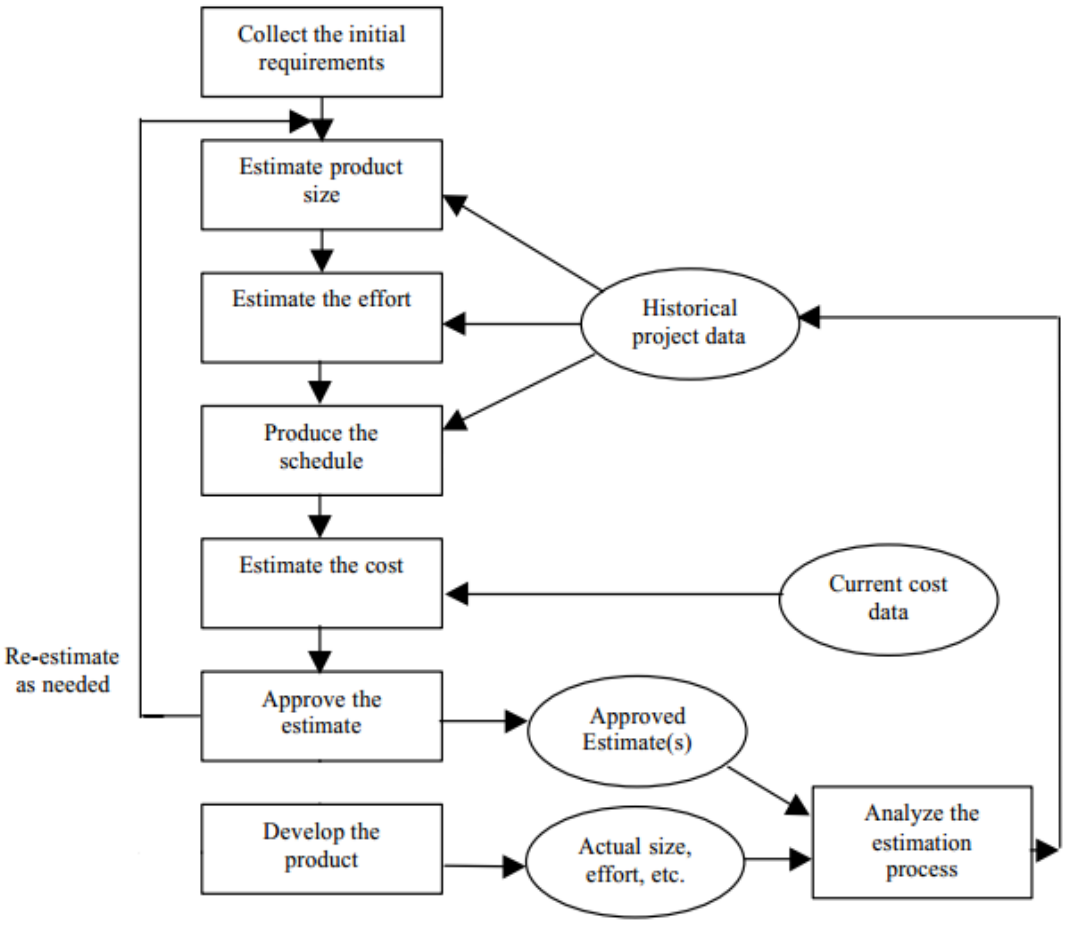


Figure 4.3 Estimation Process Cycle

Chapter 05

Software Implementation Document

5.1 INTRODUCTION

This document describes the project implementation for developing the PCE. In this documents there is a list of tools, language and other programs used in the development of the PCE. It consists of tools and languages with their versions and a brief description about them. The project implements in PHP 5, MySQL 5.6.15 and standard HTML.

5.2 Language Selection

The project implements in the following languages:

- **PHP 5**
  - Used for server scripting
- **MySQL 5.6.15**
  - Use for database
- **HTML/CSS/Bootstrap Library**
  - Used for designing of web pages
- **JavaScript/JQuery**
  - Use for scripting and validation.
- **Ajax**
  - Use for the client-side browser to communicate with the server without having to perform a page refresh.
- **Node.js**
  - Used for count source line of code. It compiles the .coffee files and convert them into the .js (javascript) files and run through cmd.

5.3 Api's

- **Google Charts**
  - [5]Google charts library is used to draw the different graphs such as metric comparison graph.
- **Google Finance API**
  - Google Finance API is used in PCE for currency conversion.
- **SLOC Master**
  - [4]SLOC Master is an open source program is used in PCE to count the source lines of code.
  - This program is implemented in .coffee and when compiles using node.js then it converts into the .js format.

5.4 Tools Selection

Tools which are used for the implementation of PCE are given in the following list.

- Notepad++
- EasyPHP Server
- Web Browser

5.5 Application Screen Shots

Application screen shots helps to understand the system, its design, input fields and output fields. These screenshots helps to understand the application views. These are the real screenshots of the application PCE.

Basic Cocomo Model Input Screen-1:

Basic Cocomo model's screenshot-1 is given in Figure 5.1. This is the upper half part of the screen in which there is a navigation bar at left side. Contrast color scheme is used for the navigation area and the rest of the screen to easily identify the things. For this screen at first row there are three input fields. First field provide the option to select language in which project is to be done. Next option is select project type and after selecting the project type user enter the current person per month rate. Then user fill the remaining inputs with their relevant weights. This is the upper half screenshot for basic cocomo model estimation.

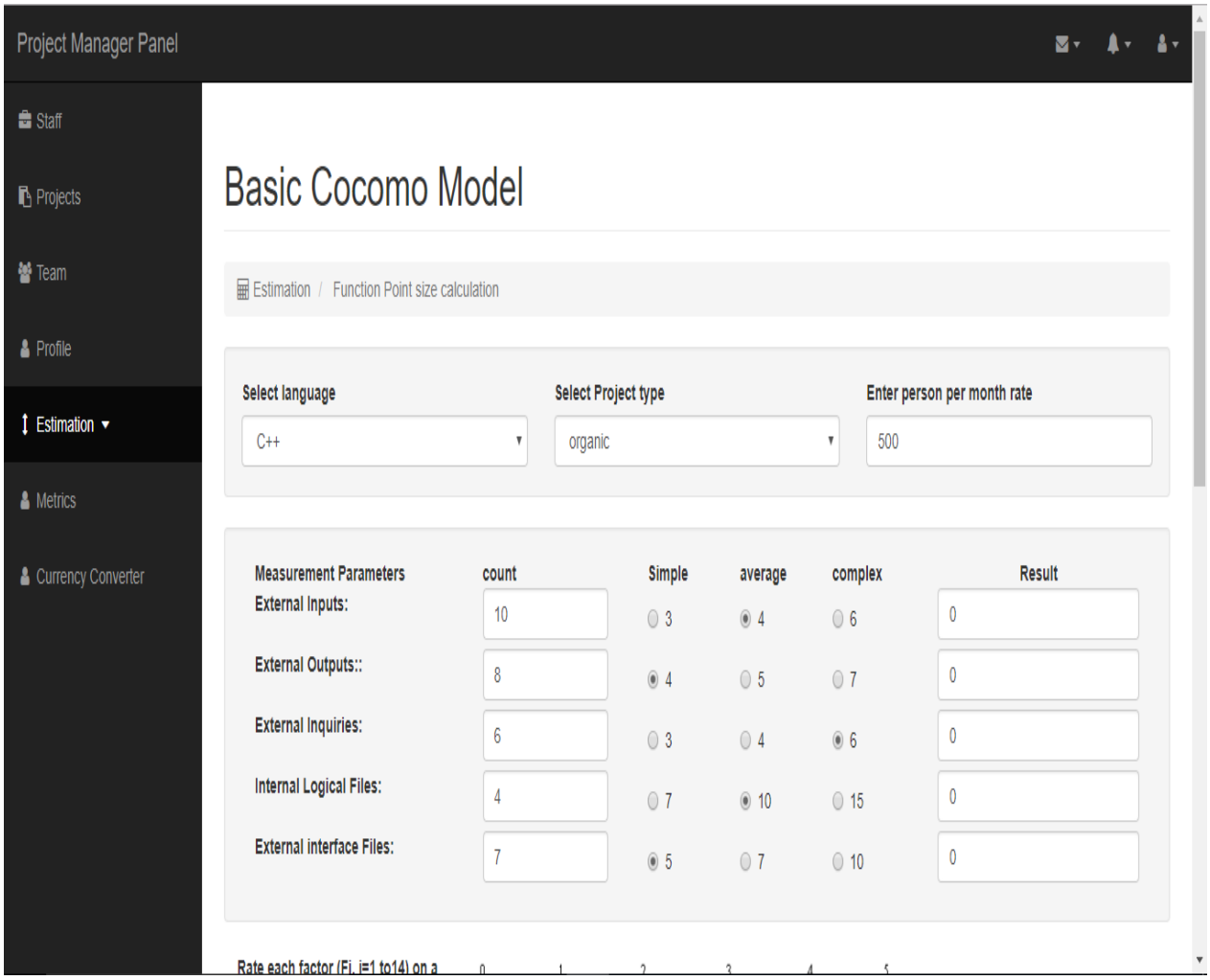


Figure 5.1 Basic Cocomo Model Input Screen-1

Basic Cocomo Model Input Screen-2:

Basic Cocomo model's screenshot-2 is given in Figure 5.2.

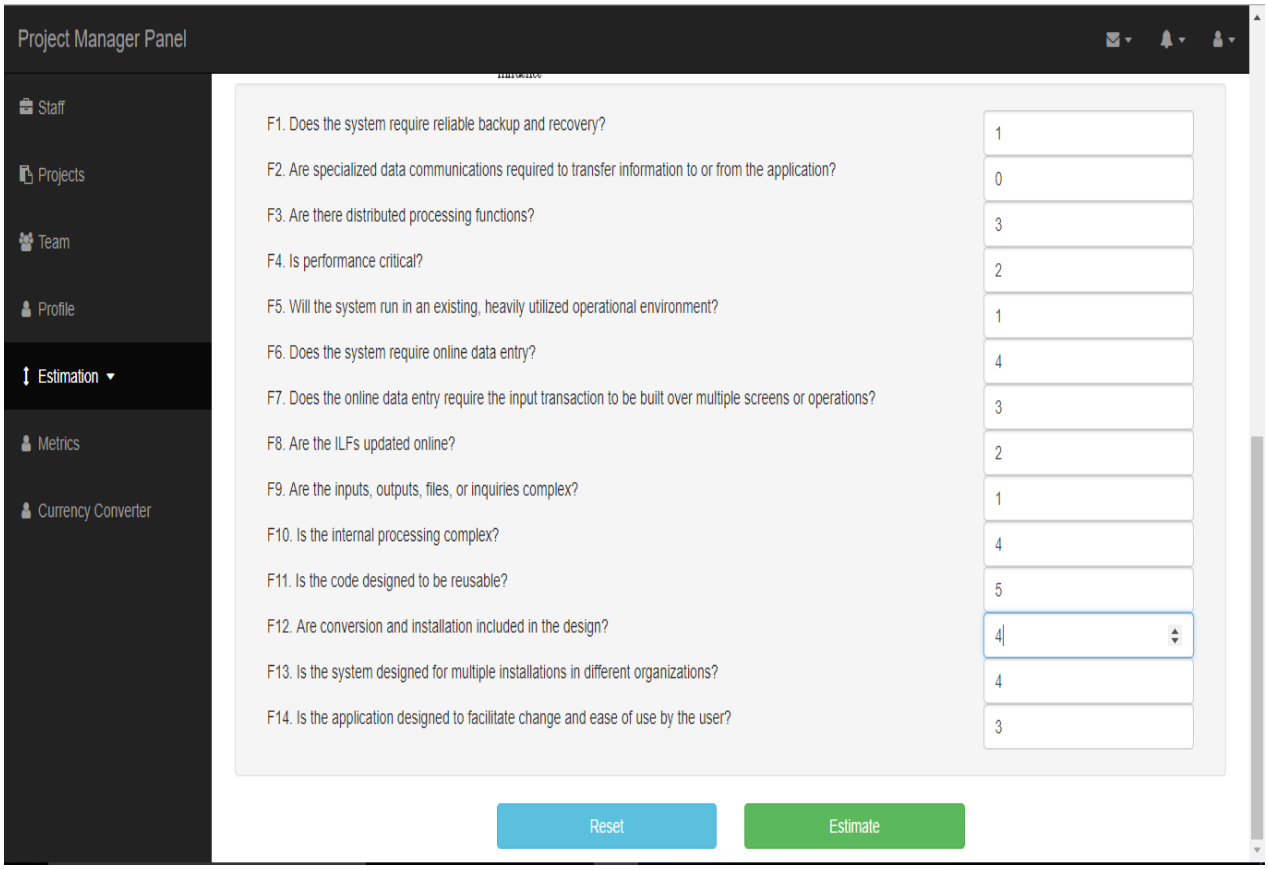


Figure 5.2 Basic Cocomo Model Input Screen-2

Use Case Point Input Screen-1:

Use case point's screenshot-1 is given in Figure 5.3.

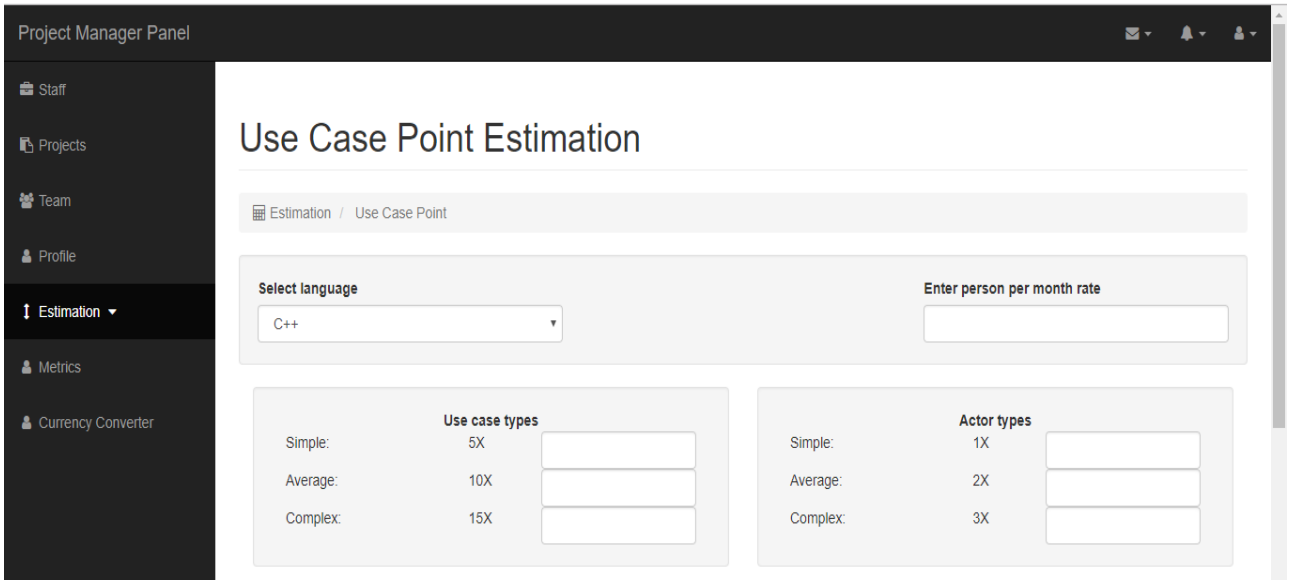


Figure 5.3 Use Case Point Estimation Input Screen-1

Use Case Point Input Screen-2:

Use case point's screenshot-2 is given in Figure 5.4.

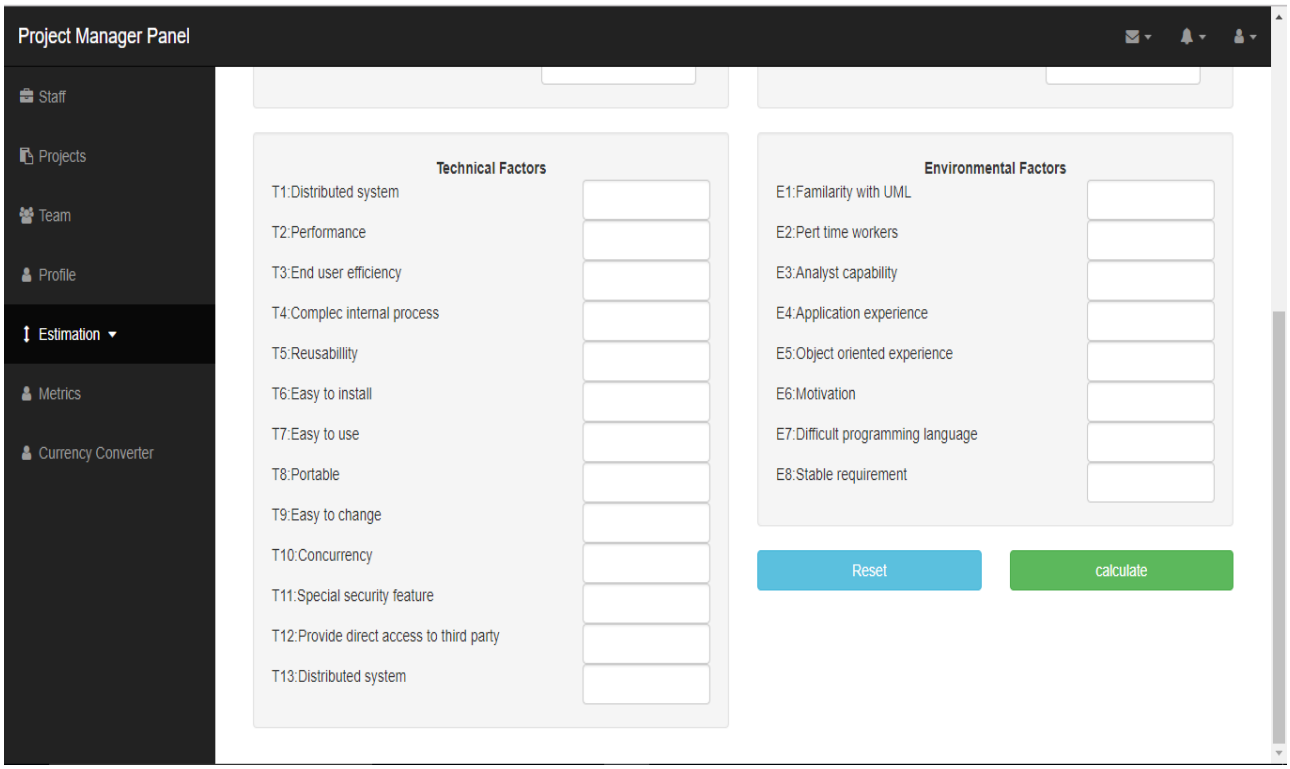


Figure 5.4 Use Case Point Estimation Input Screen-2

Function Point Input Screen-1:

Function point's screenshot-1 is given in Figure 5.5.

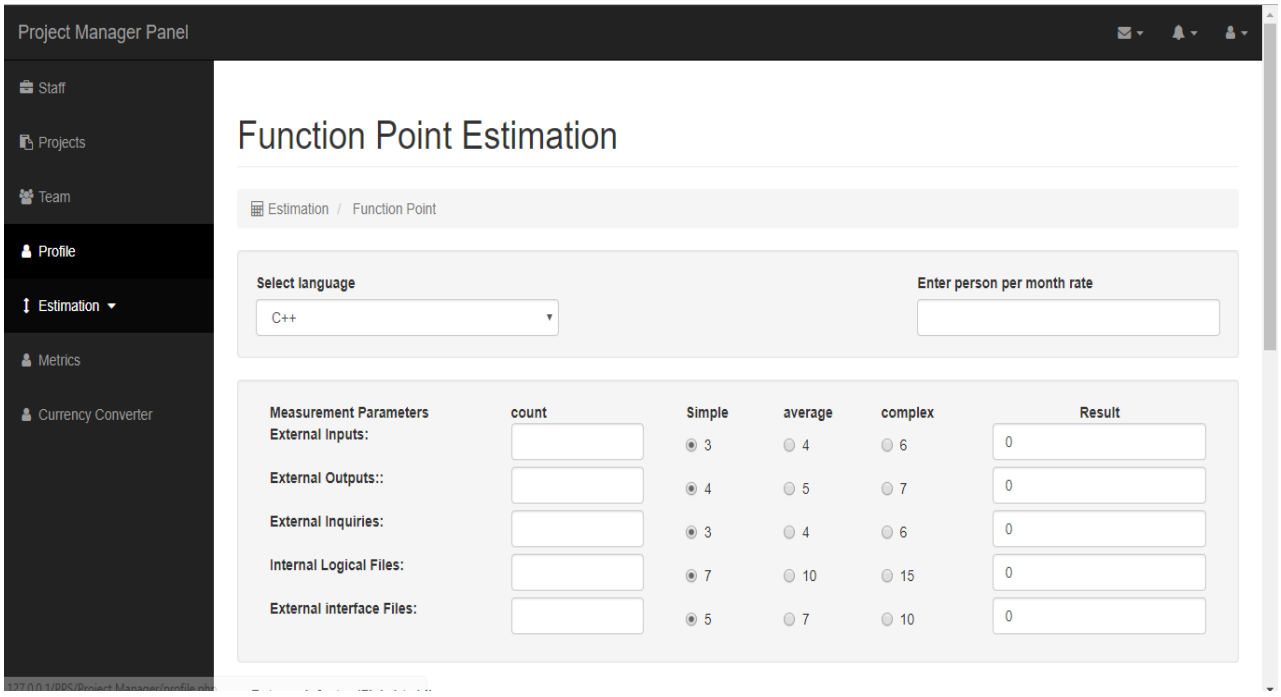


Figure 5.5 Function Point Screen Input Screen-1

Function Point Input Screen-2:

Function point's screenshot-2 is given in Figure 5.6.

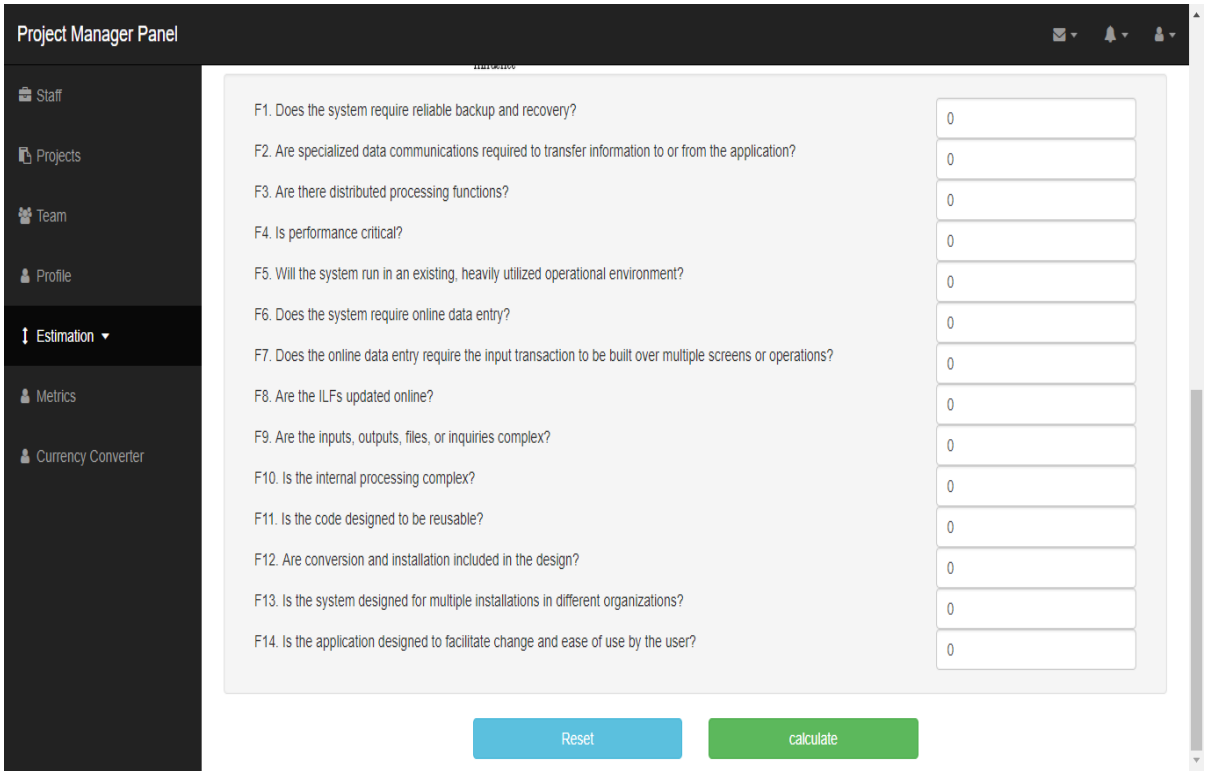


Figure 5.6 *Function Point Input Screen-2*

Function Point and Use Case Point output Screen:

Function point and use case point's output screenshot is given in Figure 5.7.

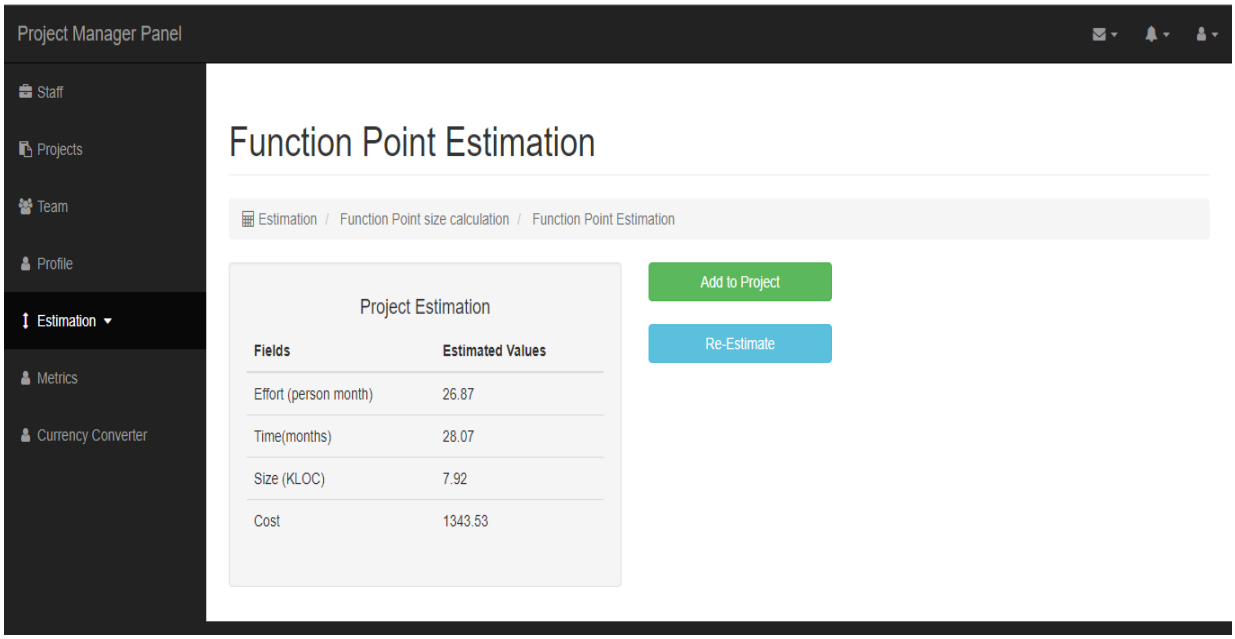


Figure 5.7 *Estimation Output Screen of Function Point and Use Case Point*

Basic Cocomo Model and Empirical model Output Screen:

Basic Cocomo Model and Empirical estimation output screenshot is given in Figure 5.8.

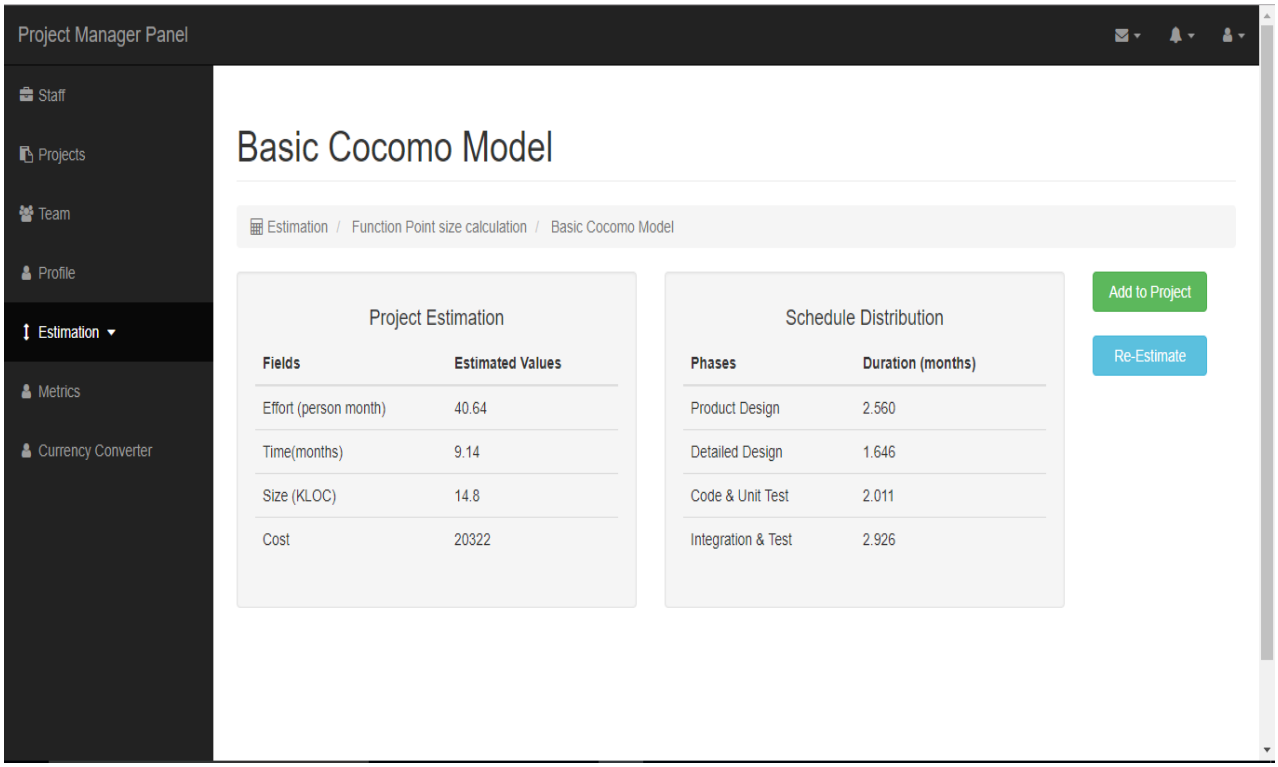


Figure 5.8 *Estimation Output Screen of Basic Cocomo Model and Empirical Model Estimation*

Add Metric to Project Screenshot:

Add metric to project screenshot is given in Figure 5.9.

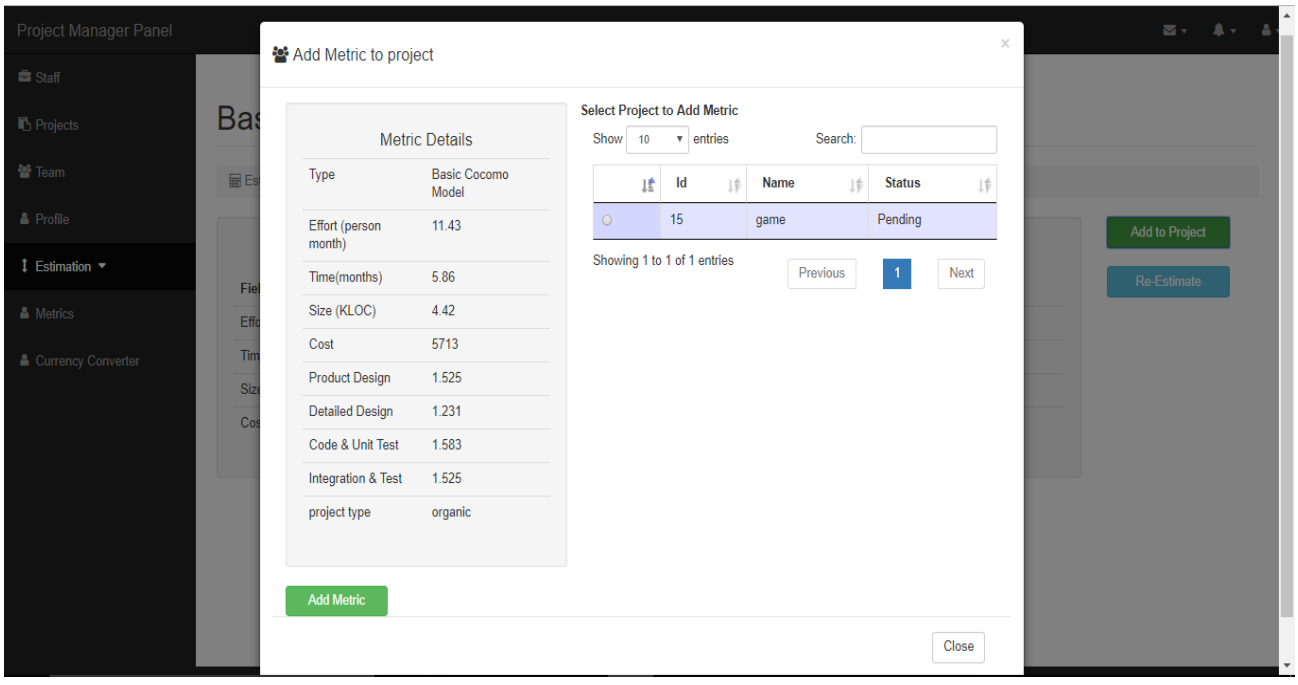


Figure 5.9 *Add Metric to Project Screen*

Add Metric to Project Response Screenshot:

Add metric to project response screenshot is given in Figure 5.10.

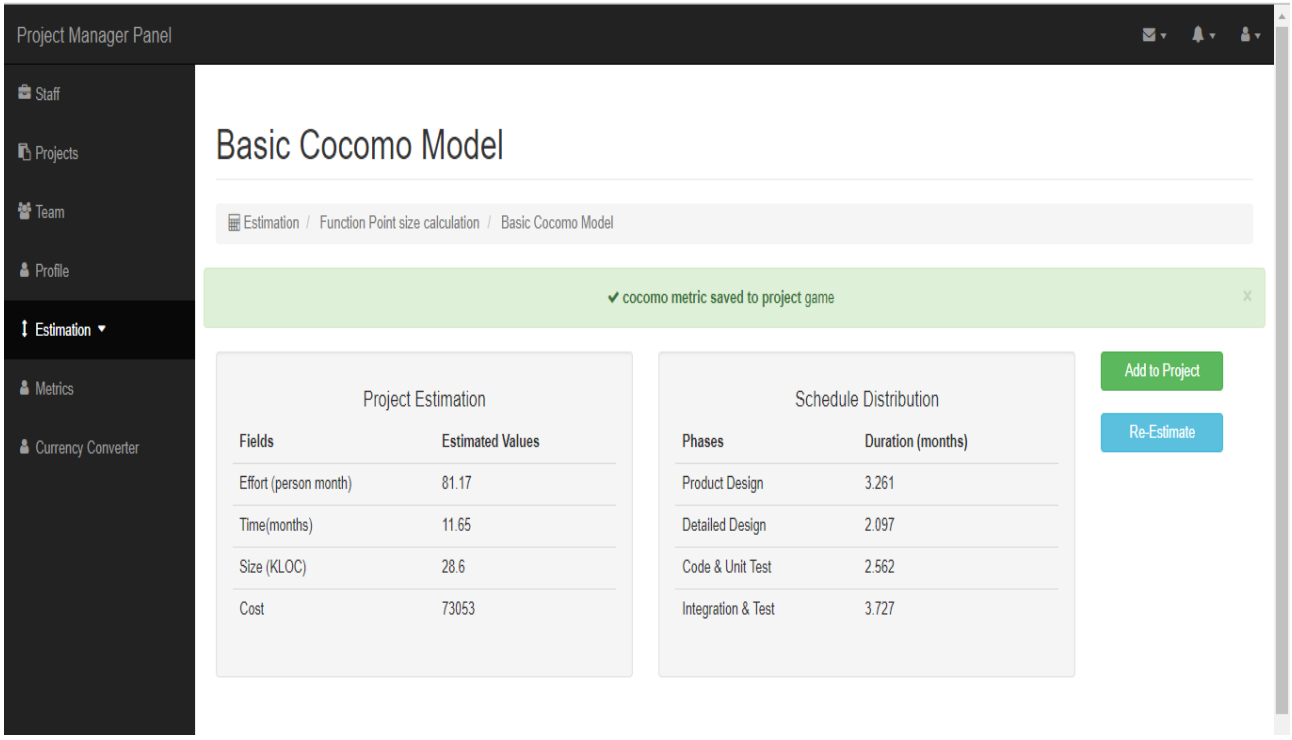


Figure 5.10 Add Metric to Project Response Screen

Update Actual Metric Input Screenshot:

Update actual metric screenshot is given in Figure 5.11.

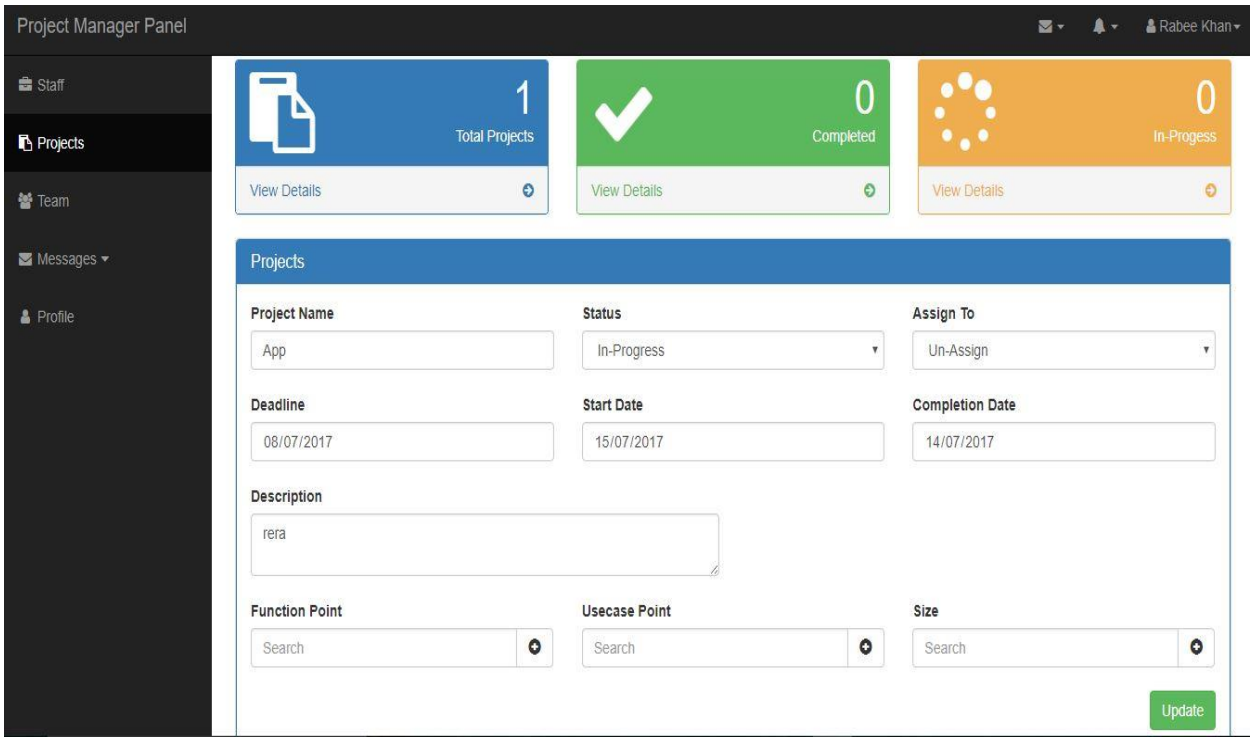


Figure 5.11 Update Actual Metric Input Screen

Update Actual Metric Option Screenshot:

Update actual metric option screenshot is given in Figure 5.12.

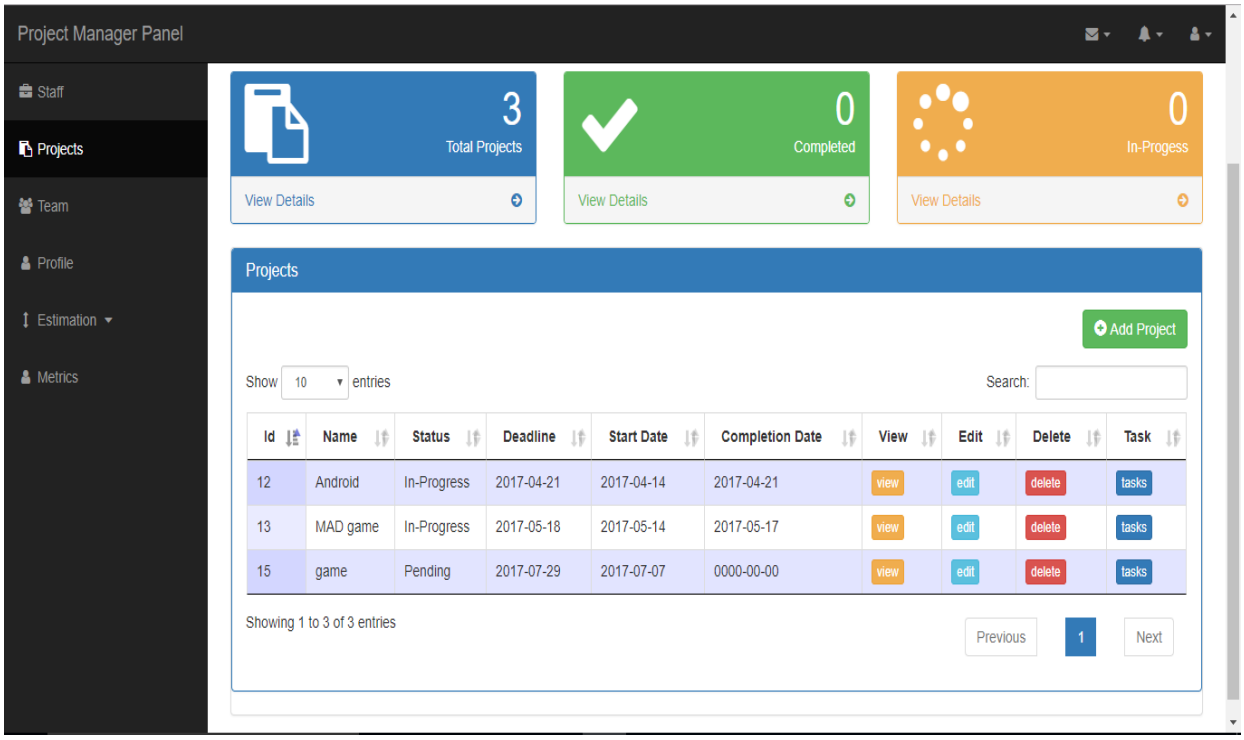


Figure 5.12 *Update Actual Metric Option Screen*

View Function Point Metric Screenshot:

View function point metric screenshot is given in Figure 5.13.

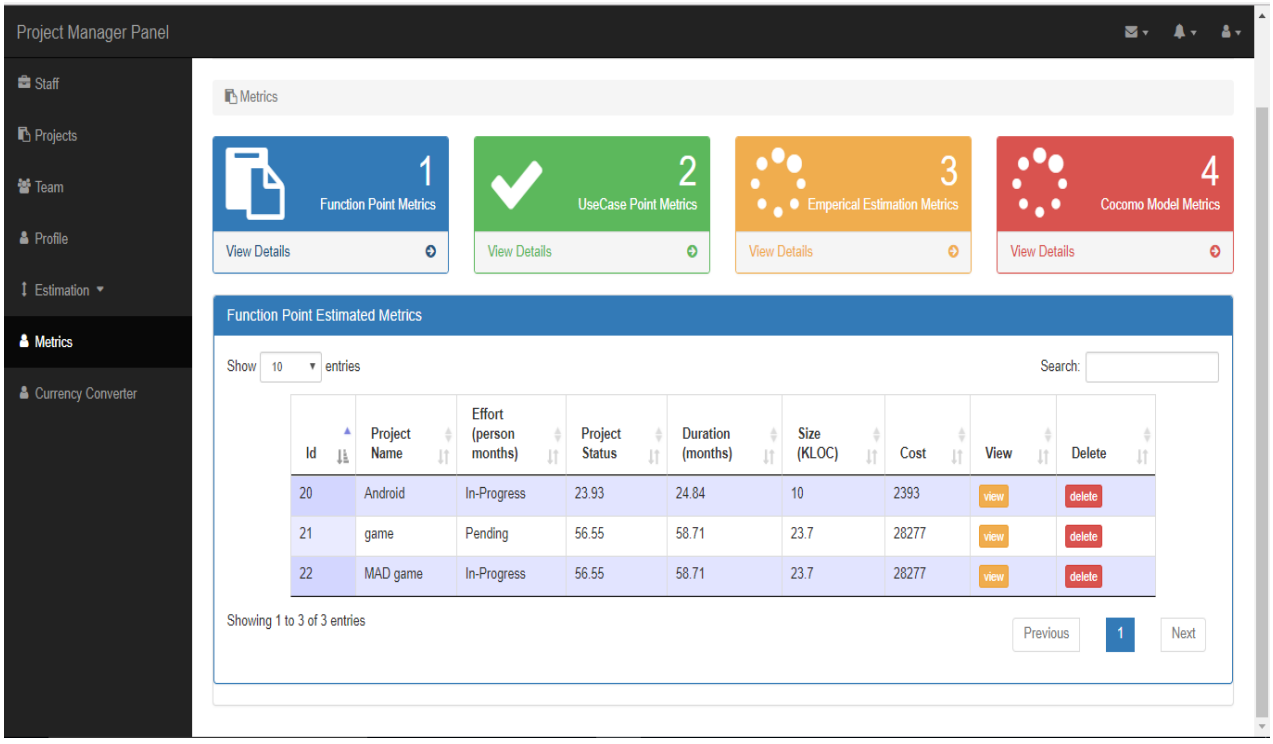


Figure 5.13 *View Function Point Metric Screen*

View Function Point Metric Detail Screenshot:

View function point metric detail screenshot is given in Figure 5.13.

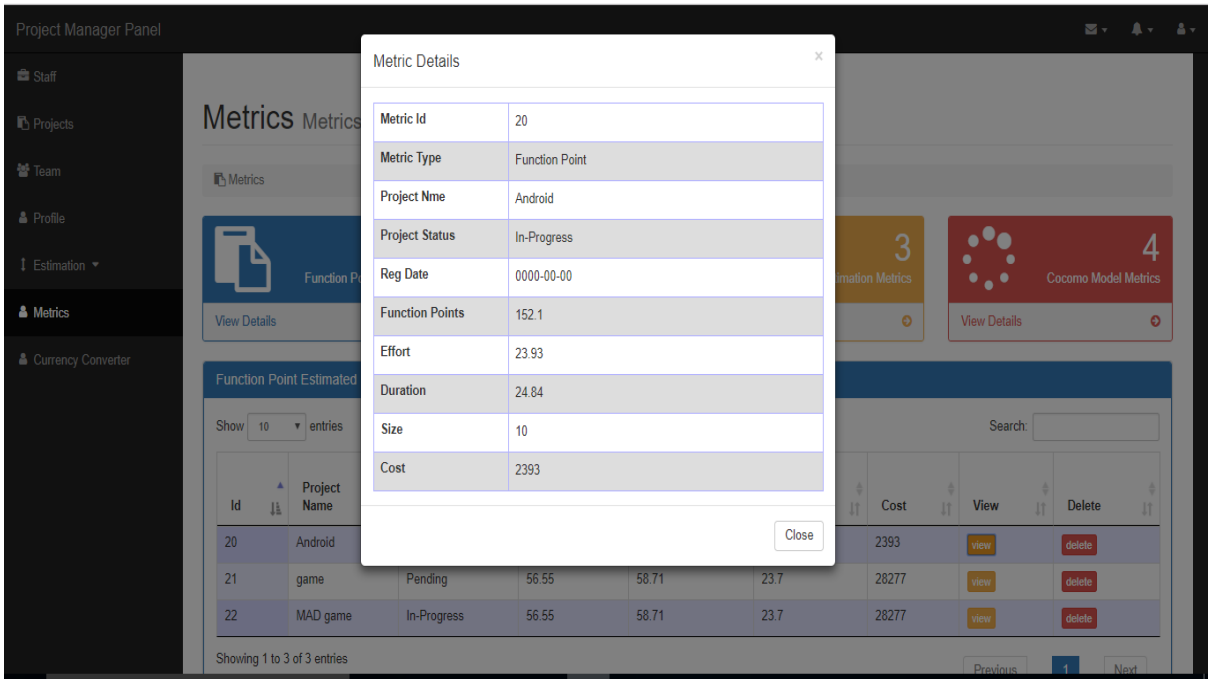


Figure 5.14 View Function Point Detail Screen

View Use Case Point Metric Screenshot:

View use case point metric screenshot is given in Figure 5.15.

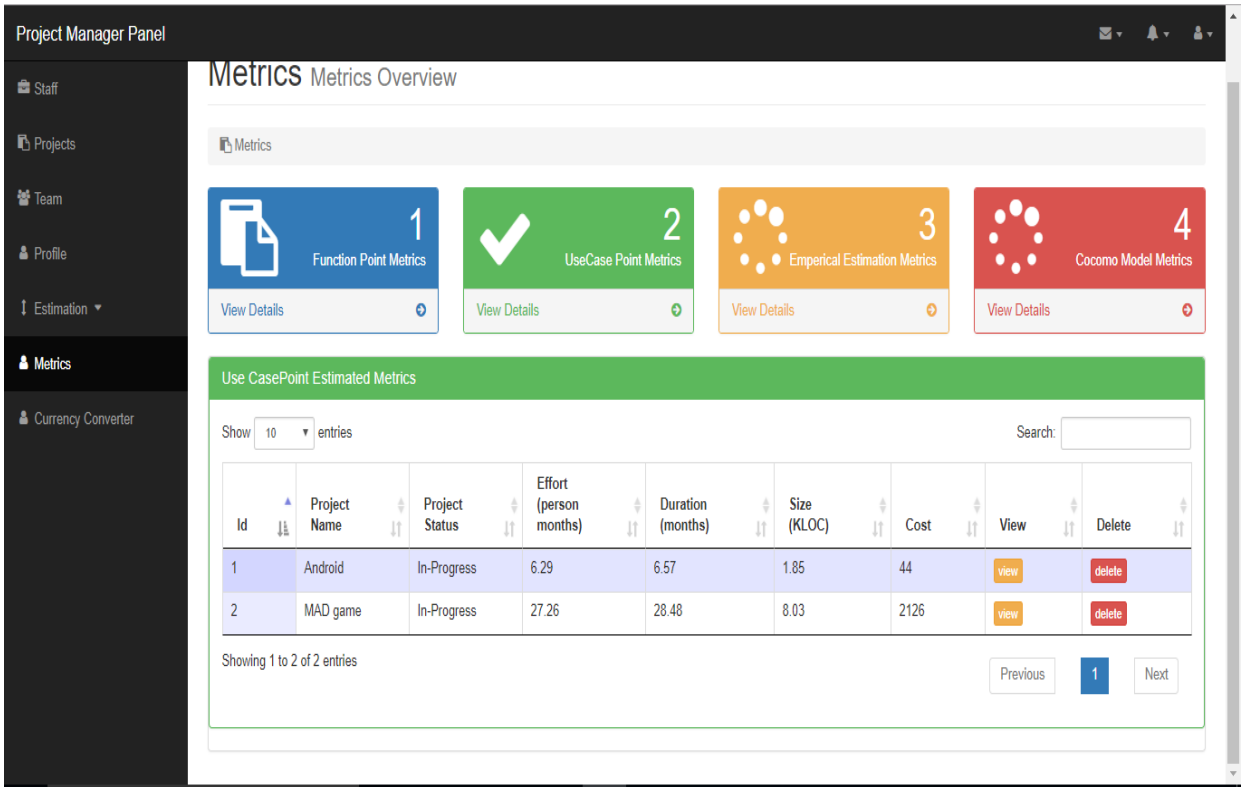


Figure 5.15 View Use Case Point Metric

View Use Case Point Metric Detail Screenshot:

View use case point metric detail screenshot is given in Figure 5.16.

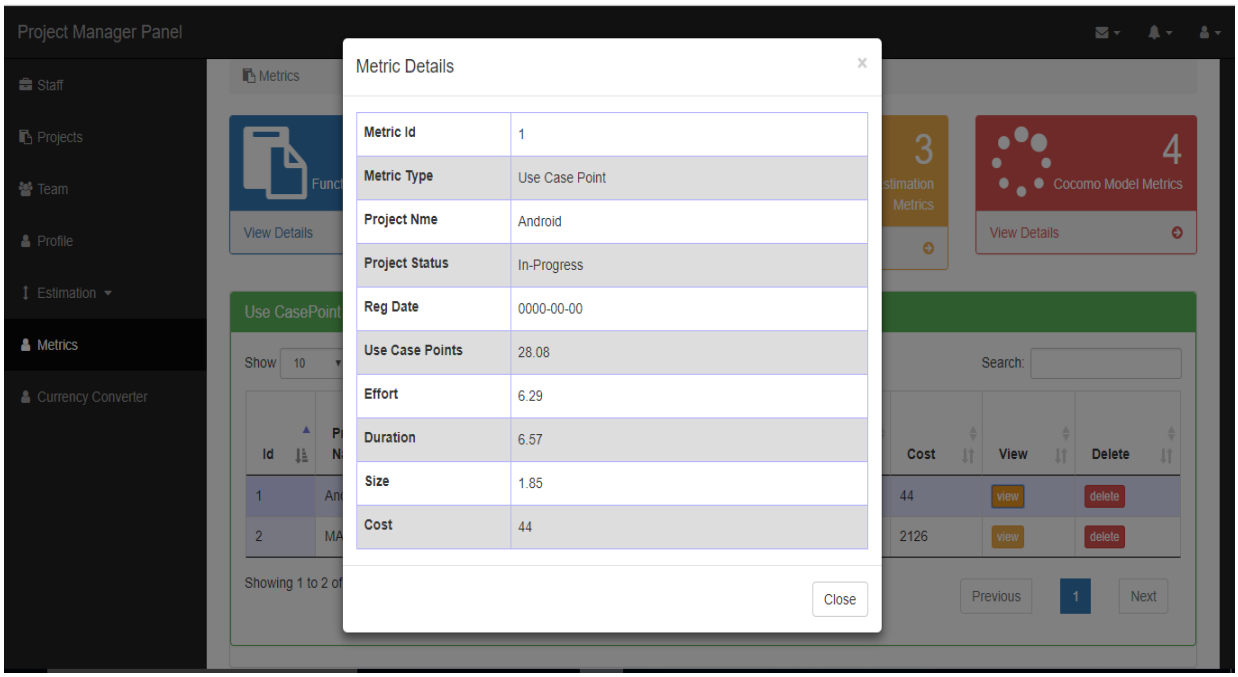


Figure 5.16 View Use Case Point Metric Detail Screen

View Empirical Model Estimation Metric Screenshot:

View empirical model estimation metric screenshot is given in Figure 5.17.

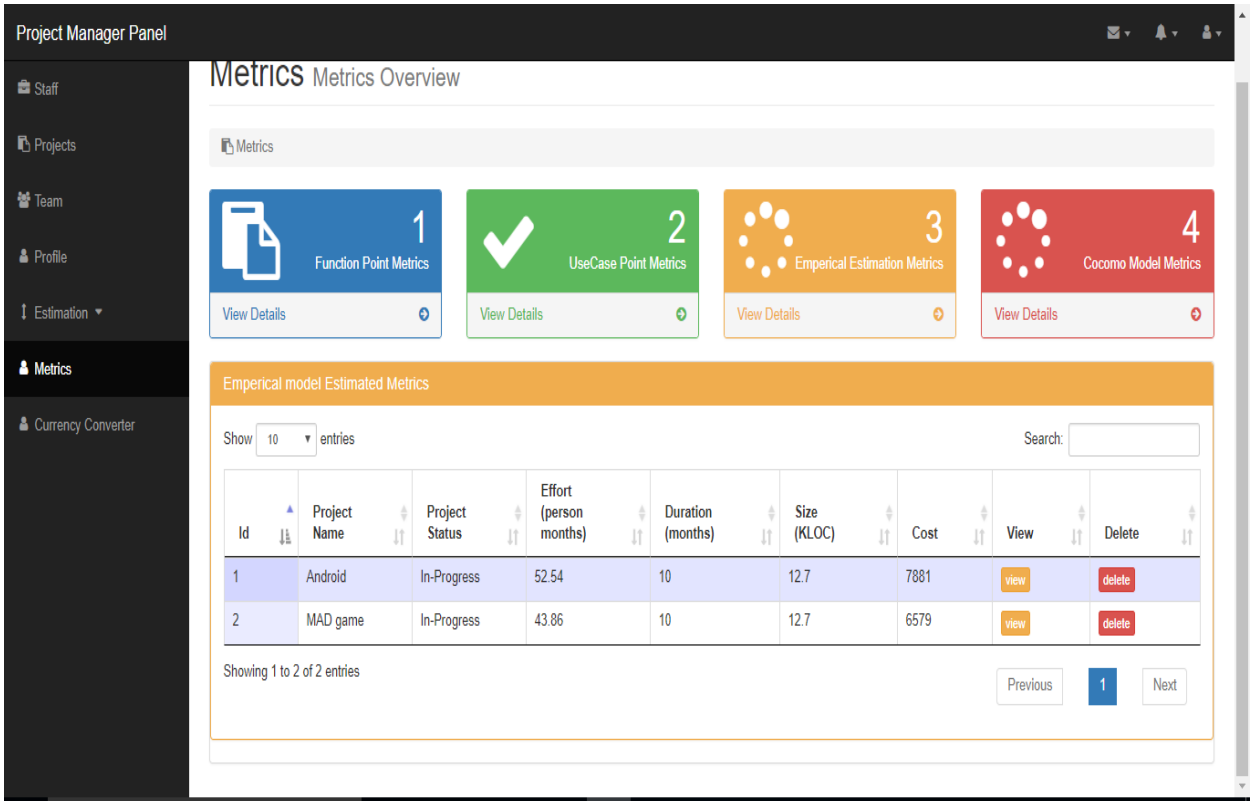


Figure 5.17 View Empirical Model Estimation Metric

View Empirical Model Metric Detail Screenshot:

View empirical model metric detail screenshot is given in Figure 5.18.

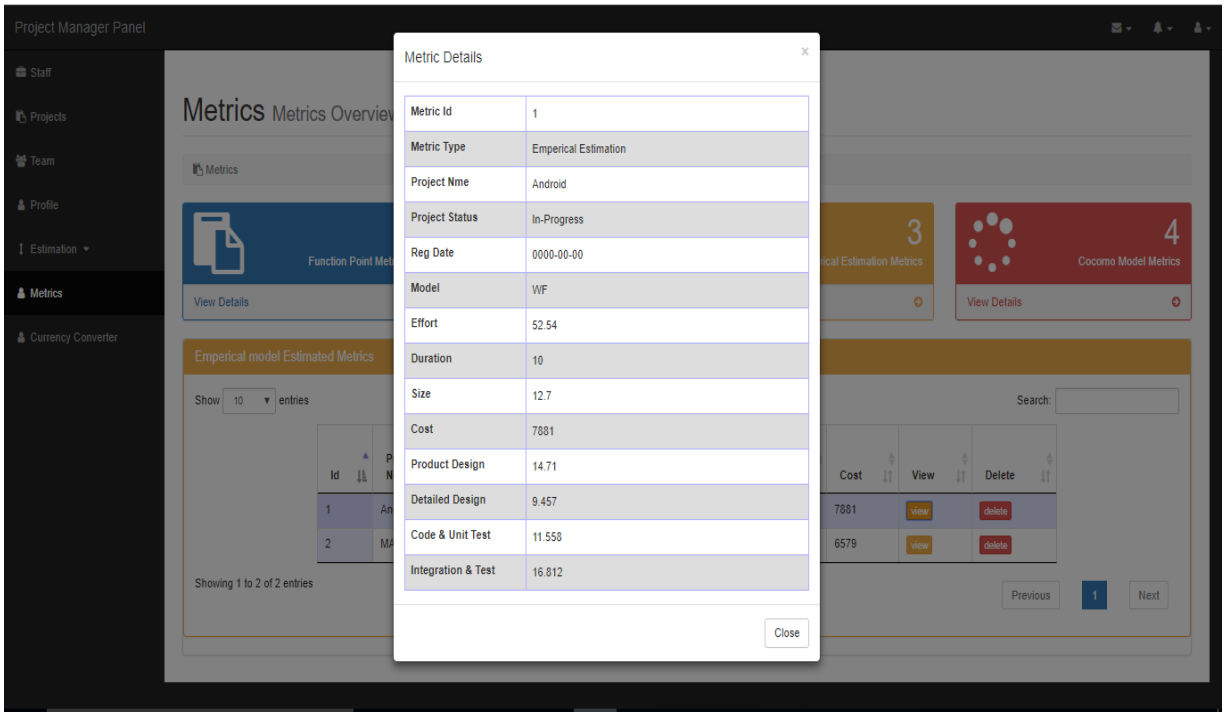


Figure 5.18 View Empirical Model Estimation Detail

View Cocomo Model Metric Screenshot:

View cocomo model metric screenshot is given in Figure 5.19.

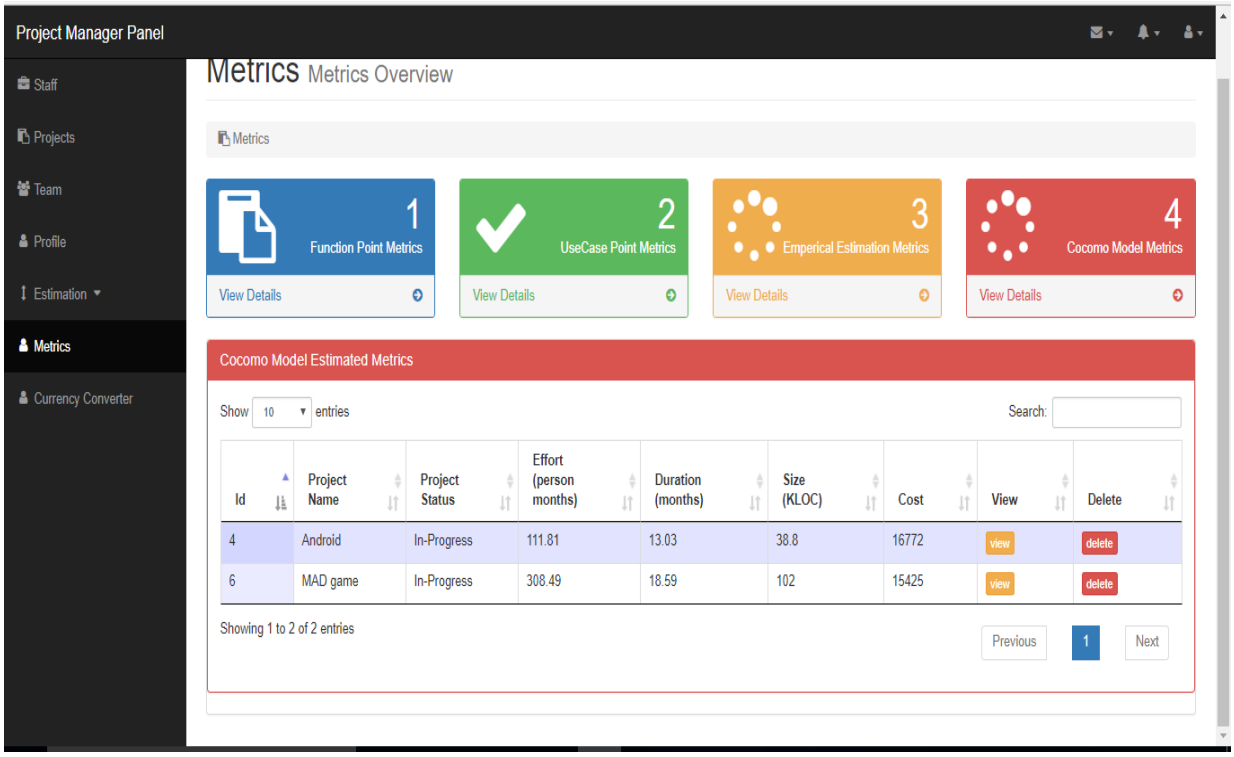


Figure 5.19 View Cocomo Model Metric

View Cocomo Model Metric Detail Screenshot:

View cocomo model metric screenshot is given in Figure 5.20.

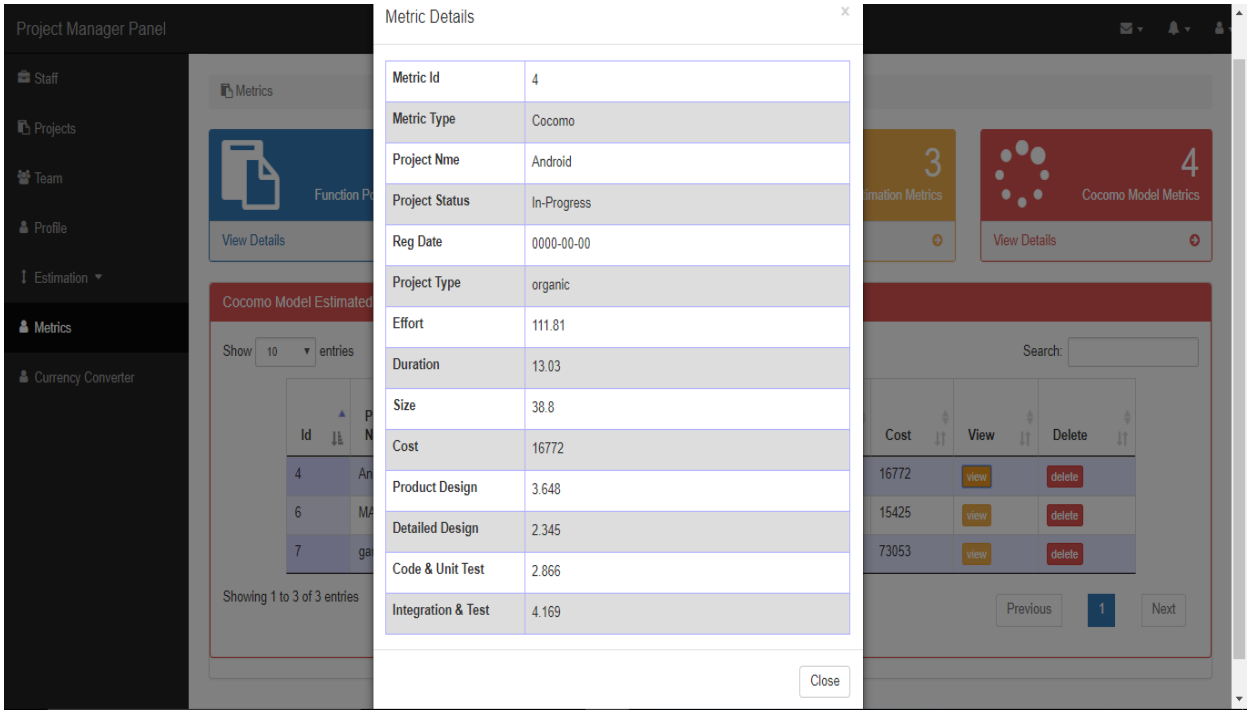


Figure 5.20 *View Cocomo Model Metric Detail Screen*

View Actual Metric of In complete Project Screenshot:

View actual metric of incomplete project screenshot is given in Figure 5.21.

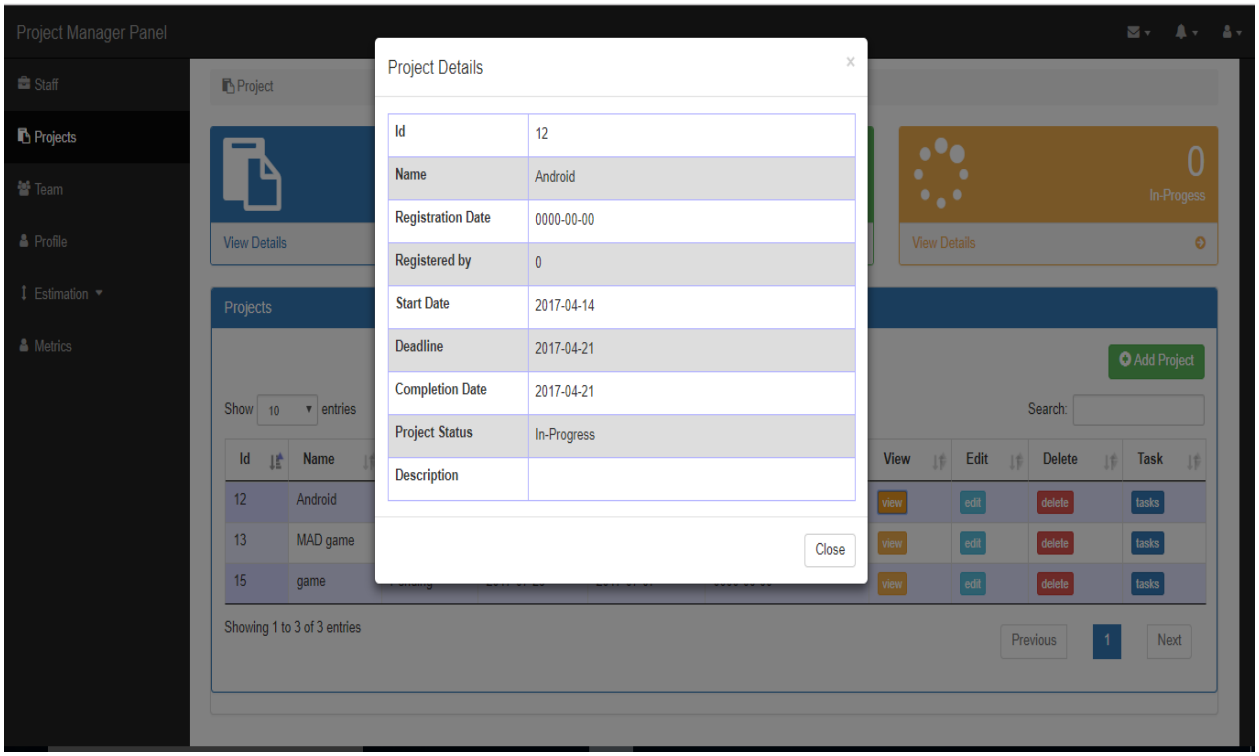


Figure 5. 21 *View Actual Metric of Incomplete Metric*

Currency Converter Screenshot:

Currency converter screenshot is given in Figure 5.22.

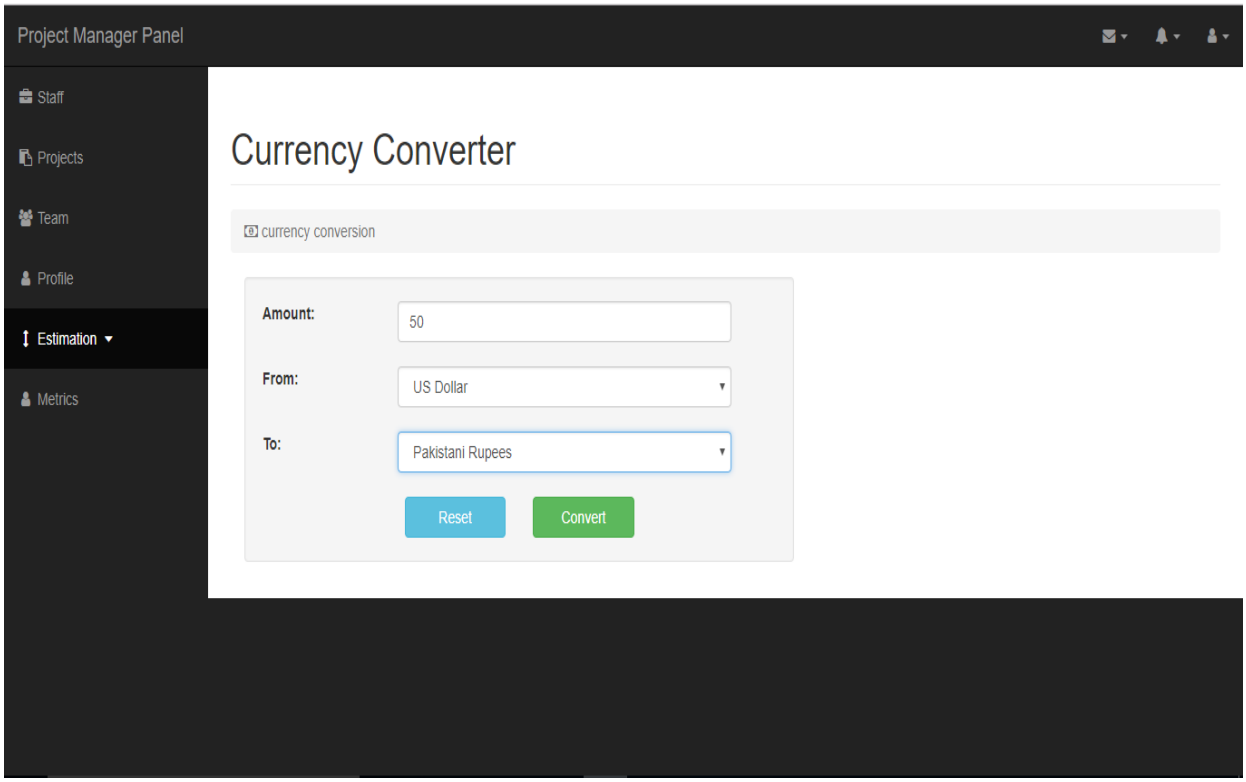


Figure 5.22 *Currency Converter Screen*

Chapter 06

Software Test Document

6.1 INTRODUCTION

The purpose of this document is to provide a structural way to test the software. Documentation for software testing helps in estimating the testing effort required. Testing is the process of evaluating a system with multiple inputs and scenarios to find whether it satisfies the specified requirements or not. This document will help the tester to find the errors when he is performing the testing of software.

6.1.1 System Overview

System overview focusing on the structural aspects of testing, provides an overview of the system in terms of the components that are tested during the unit test.

6.1.2 Test Approach

A test approach is the test strategy implementation of a project, defines how testing would be carried out. I will use acceptance test approach for testing the PCE. Acceptance test is a test conducted to determine if the requirements of a specification are met.[1] The main purpose of this test is to evaluate the system's compliance with the business requirements and verify if it is has met the required criteria for delivery to end users.

6.2 Test Plan

A test plan outlines the strategy that will be used to test an application, the resources that will be used, and the test environment in which testing will be performed and the time which will spend on testing.

6.2.1 Features to be Tested

- Calculate by Function Point
- Calculate by Empirical Estimation
- Calculate by Cocomo Model
- Calculate by Use Case Point
- Add Metric to Project
- View Metric
- View FP Metrics
- View Empirical Metrics
- View UC Metrics
- View Cocomo Metrics
- View Actual Metric
- Update Metric
- Delete Metric
- Count Line of Code
- Currency Converter

6.2.2 Testing Tools and Environment

During testing phase I need a computer, web browser, server and internet connection.

6.3 TEST CASES

These are the test case which will be test.

6.3.1 TC-1 Update Metric

Table 6.1 Test Case for Update Metric

TC-1 Update Metric	
Actor	Project manager
Purpose	Project manager wants to add the e metrics
Steps	1. User open the main menu. 2. Click the project button. 3. Select the project from the display list of projects. 4. Click on edit button of that project. 5. Fill the input fields. 6. Click add button.
Inputs	1. User fill the inputs correctly. 2. Unchecked radio button will use their default values.
	1. User fill the input fields with incorrect values.
Expected result	1. Metrics will added to the project successfully. 2. There will be message display on screen that your metric is added successfully.
	1. Metrics will not added to the project. 2. There will display a message that please fill the input fields correctly.
Actual result	As expected
	As expected
Pass/Fail	Pass
	Pass

6.3.2 TC-2 Calculate Function Point

Table 6.2 Test Case for Calculate Function Point Metric

TC-2 Calculate Function Point Metric	
Actor	Project manager
Purpose	Project manager wants to calculate FP metrics
Steps	1. User open the main menu screen. 2. User select calculate metrics button from main menu. 3. User select FP metrics. 4. Fill the required information. 5. Press calculate button.
Inputs	1. User fill correct information in the required fields. 2. Calculate button.
	1. User fill incorrect information in the required fields. 2. Click calculate button.
Expected result	1. Calculated metrics will show on screen
	1. There will a pop up message which shows a message that “please fill the correct information”
Actual result	As expected
	As expected
Pass/Fail	Pass
	Pass

6.3.3 TC-3 Calculate Use Case Point Metric

Table 6.3 Test Case for Calculate Use Case Metric

TC-3 Calculate Use Case Point Metric	
Actor	Project manager
Purpose	Project manager wants to calculate UC metrics
Steps	1. User open the main menu screen. 2. User select calculate metrics button from main menu. 3. User select UC metrics option. 4. Fill the required information. Press calculate button.
Inputs	1. User fill correct information in the required fields. 2. Calculate button.
	1. User fill incorrect information in the required fields. 2. Click calculate button.
Expected result	1. Calculated metrics will show on screen
	1. There will a pop up message which shows a message that “please fill the correct information
Actual result	As expected
	As expected
Pass/Fail	Pass
	Pass

6.3.4 TC-4 Calculate Empirical Metric

Table 6.4 Test Case for Empirical Model Metric

TC-4 Calculate Empirical Metric	
Actor	Project manager
Purpose	Project manager wants to calculate EE metrics
Steps	1. User open the main menu screen. 2. User select calculate metrics button from main menu. 3. User select EE metrics option. 4. Fill the required information. 5. Press calculate button.
Inputs	1. User fill correct information in the required fields. 2. Calculate button.
	1. User fill incorrect information in the required fields. 2. Click calculate button.
Expected result	1. Calculated metrics will show on screen
	1. There will a pop up message which shows a message that “please fill the correct information”
Actual result	As expected
	As expected
Pass/Fail	Pass
	Pass

6.3.5 TC-5 Calculate Cocomo Model Metric

Table 6.5 Test Case for Calculate Cocomo Model Metric

TC-5 Calculate Cocomo Model Metric	
Actor	Project manager
Purpose	Project manager wants to calculate Cocomo Model metrics
Steps	1. User open the main menu screen. 2. User select calculate metrics button from main menu. 3. User select Cocomo Model metrics option. 4. Fill the required information. 5. Press calculate button
Inputs	1. User fill correct information in the required fields. 2. Click calculate button.
	1. User fill incorrect information in the required fields. 2. Click calculate button.
Expected result	1. Calculated metrics will show on screen
	1. There will a new screen appear which shows a message that “please fill the correct information”
Actual result	As expected
	As expected
Pass/Fail	Pass
	Pass

6.3.6 TC-2 Delete Metrics

Table 6.6 Test Case for Delete Metric

TC-6 Delete Metrics	
Actor	Project manager
Purpose	Project manager wants to delete project metrics
Steps	1. User select metric. 2. Press delete button. 3. Press confirm.
Inputs	1. User delete metric id 5.
	1. User delete metric id 7.
Expected result	1. Metric id=5 deleted
	1. Metric id=7 deleted
Actual result	As expected
	As expected
Pass/Fail	Pass
	Pass

6.3.7 TC-7 Currency Converter

Table 6.7 Test Case for Currency Converter

TC-7 Currency Converter	
Actor	Project manager
Purpose	Project manager wants to convert given amount of currency from one currency to another.
Steps	1. User open the main menu screen. 1. Select the currency converter. 2. Fill input fields 3. Press convert.
Inputs	1. User fill correct information in the required fields. 2. Click update button.
	1. User fill incorrect information in the required fields. 2. Click update button.
Expected result	1. Converted amount is displayed on the screen with conversion rate.
	1. System will display a message to fill correct information.
Actual result	As expected
	As expected
Pass/Fail	Pass
	Pass

6.3.8 TC-8 Count Lines of Code

Table 6.8 Test Case for Count Lines of Code

TC-8 Count Lines of Code	
Actor	Project manager
Purpose	Project manager wants count the source lines of given code of project.
Steps	1. User open the main menu screen. 2. Select project. 3. Select edit project. 4. Select size counter. 5. Upload files. 6. Press count. 7. System will display the number of lines of code.
Inputs	1. User upload correct files. 2. Press count button.
	1. User upload wrong files. 2. Press count button.
Expected result	1. System will display lines of code.
	1. System will display a message that upload correct files.
Actual result	As expected
	As expected
Pass/Fail	Pass
	Pass

6.3.9 TC-9 Add Estimated Metric to Project

Table 6.9 Add Estimated Metric to Project

TC-9 Add Estimated Metric to Project	
Actor	Project manager
Purpose	Project manager wants add estimated metric to project.
Steps	1. Press button “Add Metric to Project” 2. Select project from list. 3. Press insert
Inputs	1. User select project id=6.
	1. User select project id=9
Expected result	1. Metric added to project id=6
	1. Metric added to project id=9
Actual result	As expected
	As expected
Pass/Fail	Pass
	Pass

6.3.10 TC-10 View Function Point Metric

Table 6.10 View Function Point Metrics

TC-10 View Function Point Metric	
Actor	Project manager
Purpose	Project manager wants to view function point metrics
Steps	1. Select Metric 2. Select View function point metric
Inputs	1. User select function point metric.
	1. User select function point metric
Expected result	1. Function point metric displayed
	1. Function point metric displayed
Actual result	As expected
	As expected
Pass/Fail	Pass
	Pass

6.3.11 **TC-11 View Use Case Point Metric**

Table 6.11 View Use Case Point Metrics

TC-11 View Use Case Point Metric	
Actor	Project manager
Purpose	Project manager wants to view use case point metrics
Steps	1. Select Metric 2. Select View use case point metric
Inputs	1. User select use case point metric.
	1. User select use case point metric
Expected result	1. Use case point metric displayed
	1. Use case point metric displayed
Actual result	As expected
	As expected
Pass/Fail	Pass
	Pass

6.3.12 **TC-12 View Empirical Model Metric**

Table 6.12 View Empirical Model Metrics

TC-12 View Empirical Model Metric	
Actor	Project manager
Purpose	Project manager wants to view empirical model metrics
Steps	1. Select Metric 2. Select View empirical model metric
Inputs	1. User select empirical model metric.
	1. User select empirical model metric
Expected result	1. Empirical model metric displayed
	1. Empirical model metric displayed
Actual result	As expected
	As expected
Pass/Fail	Pass
	Pass

6.3.13 **TC-13 View Cocomo Model Metric**

Table 6.13 View Cocomo Model Metrics

TC-13 View Cocomo Model Metrics	
Actor	Project manager
Purpose	Project manager wants to view cocomo model metrics
Steps	1. Select Metric 2. Select View cocomo model metric
Inputs	1. User select cocomo model metric.
	1. User select cocomo model metric
Expected result	1. Cocomo model metric displayed
	1. Cocomo model metric displayed
Actual result	As expected
	As expected
Pass/Fail	Pass
	Pass

6.3.14 **TC-14 View Actual Metric**

Table 6.14 View Actual Metrics

TC-14 View Actual Metric	
Actor	Project manager
Purpose	Project manager wants to view actual metrics
Steps	1. Select project 2. Select metric button.
Inputs	1. User select metric of project=6.
	1. User select metric of project id=3
Expected result	1. Actual metric of project id=6 displayed
	1. Actual metric of project id=3 displayed
Actual result	As expected
	As expected
Pass/Fail	Pass
	Pass

Chapter 07

Conclusion and Future Enhancement

7.1 INTRODUCTION

This document describes the project conclusions and future enhancements.

7.1.1 Summary

This application allows the project manager to estimate the project's cost, duration, size and effort. This estimation is based on the previous data of the company. For this purpose sufficient numbers of completed project's data should be present which will use in the estimation. This application provide the currency conversion module which helps the project manager to convert the amount into different currencies on current rates. For better metric record this application provides the facility to count source lines of code in many languages such as java, pascal, javascript, php, python, C, C++, html, C#, assembly, visual basic and xml.

Basic Cocomo model and empirical model estimate the project on the basis of predefined variables while function point and use case point metric use previous completed project's data. It compares the function points and use case points with the effort, size and duration and take average of all the values to calculate the product factor of duration, size and effort.

7.1.2 Conclusions

We are now able to estimate the project's cost, size, duration and effort required for the completion of the project. Its better results depends upon the availability of previous data and analysis of the project manager. As it is based on estimation so results will not be 100% accurate. 100% accurate metric is only possible after the completion of project. We are now able to check the performance of team by analyzing the metrics. This application help the organization to do projects in an efficient way and increase the revenue.

An early project estimate helps managers, developers, and testers plan for the resources a project requires. These metrics can produce an early estimation of the project and closer to the actual effort than experts.

7.1.3 Future Enhancements

In future application can be enhanced by:

- In future more estimation models will be added for better estimations.
- It will predict the team or person for the project development corresponding to person's experience and project requirements.
- It will notify the each developer about his performance.
- It will support more languages in future.
- In will fetch the requirements from the project's description and made estimation in fully or semi-automated way.
- It will fetch the use cases from the use case diagram for use case point estimation.

8. References

[1] "Acceptance Testing," [Online]. Available: http://www.tutorialpoint.com/software testing dictionary/acceptance testing.htm. [Accessed 15 1 2017].

[2] "System architecture," wikipedia, [Online]. Available: http://en.wikipedia.org/wiki/Systems architecture. [Accessed 15 1 2017].

[3] R. K. Clemmons, "Project Estimation With Use Case Points," [Online]. Available: http://www.bfpug.com.br/Artigos/UCP/Clemmons-Project_Estimation_with_UCP.pdf. [Accessed 27 1 2017].

[4] "SLOC Master," MIT, [Online]. Available: http://github.com/flosee/sloc. [Accessed 31 3 2017].

[5] "Google jsapi," Google, [Online]. Available: http://www.google.com/jsapi. [Accessed 5 5 2017].

[6] "SB Admin," MIT, [Online]. Available: http://startbootstrap.com/template-overviews/sb-admin/. [Accessed 21 2 2017].

[7] "Data tables," Mit Spry Media, [Online]. Available: https://datatables.net/examples/styling/bootstrap.html. [Accessed 3 3 2017].

[8] R. S. Pressman, Software engineering : a practitioner’s approach, McGraw-Hill, 2007.

[9] "Help Page," http://groups.engin.umd.umich.edu/CIS/course.des/cis525/js/f00/gamel/help.html#anchor2. [Online]. [Accessed 27 1 2017].