Software Project Estimation Report

Version 1.0
April 12, 2023

PassMan - Cryptographically stored password manager

Anushka Singh | 2021 IMG-014
Avijeet Jain | 2021 IMG-018
Yuvraj Kumar | 2021 IMG-066

Submitted in partial fulfillment

Of the requirements of
IMIT-3106 Software Engineering Lab

Table of Contents

1.0 SIZE ESTIMATION

It's important to understand that project size estimation is the most fundamental parameter. If this is estimated accurately, all other parameters like effort, duration, cost, etc., can be determined easily.

Currently, two techniques used to estimate project size are:

1. Lines of code or LOC
2. Function point

Both of the above serve as important project size estimation metrics. For our report, we will be using the Function Point metric.

1.1 Function Point Metric

The function point metric proposes that the size of the software project is directly dependent on the various functionalities it supports. The more the features supported, the more the size would be. This technique helps determine the size of the project directly from the problem specification, so it is helpful to project managers during project planning while determining size.

1.2 Step 1: UFP (Unadjusted Function Point) Computation

INPUTS:

1. Personal information of User (During Sign Up)
2. Personal information of User (During Login)
3. Saving a new password
4. Retrieving details of the stored password
5. Changing the stored password
6. Details of feedback provided in feedback forum form
7. Details of the query and solutions in the support forum form
8. Updated Details of User (While updating profile)
9. Change password input in Profile Screen

Total Number of Inputs = 9

OUTPUTS:

1. Confirmation message: User Successfully Signed Up
2. Confirmation message: User Successfully Logged In
3. Confirmation message: New Password created successfully
4. Confirmation message: Changed the stored password successfully
5. List of result that match the search query
6. Confirmation message: Successfully updated profile information
7. Confirmation message: Stored password deleted successfully
8. List of queries and related replies on the support forum
9. Error message: Invalid Sign Up
10. Error message: Invalid Login attempt
11. Error message: Failed to fetch creche organizations
12. Error message: Failed to update profile information

Total Number of Outputs = 12

INQUIRIES:

1. Request to sign into the system
2. Request to log into the system
3. Request to browse/search
4. Request to create a query on the forum
5. Request to add a reply on the forum
6. Request to update user profile information
7. Request to update creche profile information
8. Request to delete the account

Total Number of Inquiries = 8

FILES:

1. List of registered users
2. List of stored key
3. List of user queries
4. List of user solutions provided

Total Number of Files = 4

INTERFACES: 0

UFP Calculation:

UFP = (Number of inputs)*4 + (Number of outputs)*5 + (Number of inquiries)*4 + (Number of lines)*10 + (Number of interfaces)*10

UFP

= 9*4 + 12*5 + 8*4 + 4*10 + 0*10
= 168

1.3 Step 2: Refine Parameters

Below table is an FPA matrix that shows the weight adjustment factor Type:
Simple Average Complex

	Simple	Average	Complex
Inputs	3	4	6
Outputs	4	5	7
No. of inquiries	3	4	6
No. of files	7	10	15
No. of interfaces	5	7	10

UFP will be computed using the following equation:

$$\textbf{UFP} = \sum_{i=1 j=1}^{i=5 j=3} W_{ij} C_{ij}$$

Where Wi j is the Weight adjustment, and Ci j is the count of values.

	Simple	Average	Complex
Inputs	6	2	1
Outputs	6	4	2
No. of inquiries	3	3	2
No. of files	3	1	0
No. of interfaces	0	0	0

Inputs: 6 Simple + 2 Average + 1 Complex
Output: 6 Simple + 4 Average + 2 Complex
Inquiries: 3 Simple + 3 Average + 2 Complex
Files: 3 Simple + 1 Average
Interfaces: 0

Refined UFP

= ((6 * 3) + (2 * 4) + (1 * 6)) + ((6 * 4) + (4 * 5) + (2 * 7)) + ((3 * 3) + (3 * 4) + (2 * 6)) + ((7 * 3) + (10 * 1)) + (5 * 0) **= 154**

1.4 Step 3: Refine UFP based on the complexity adjustment factor

Function Point Relative Complexity Adjustment Factors Score
Requirement for reliable backup and recovery 4
Requirement for data communication 5
The extent of distributed processing 2
Performance requirements 3
Expected operational environment 4
The extent of online data entries 5
The extent of multi-screen or multi-operation online data input 4
The extent of online updating of master files 4
The extent of complex inputs, outputs, online queries and files 2
The extent of complex data processing 1
The extent of conversion and installation included in the design 4
The extent of multiple installations in an organization and variety of customer organizations 5
The extent of change and focus on ease-of-use 5
Degree of Influence (DI) 48

Complexity Adjustment Factor (CAF) = 0.65 + 0.01 * DI

= 0.65 + 0.01 * 48
= 1.13

Function Point = UFP * CAF = 154 * 1.13

= 174.02

2.0 EFFORT AND TIME ESTIMATION

2.1 COCOMO Model:

The COCOMO model is used to estimate time and effort for any software project. This software project is based on the organic category of development complexity. This is a reliable process for an approximate estimation of the project parameters. COCOMO model will estimate software parameters using the following expressions:

Effort = a1 × (KLOC)^ b1 PM
Tdev = c1 × (Effort)^ d1 months

where,

➔ KLOC is the estimated size of the software product expressed in Kilo Lines Of Code.

➔ a1, b1, c1, d1 are constants for each category of the software product. ➔ Tdev is the estimated time to develop the software, expressed in months.

➔ Effort is the total effort required to develop the software product, expressed in person-months (PMs).

LOC Estimation:

1. Registration: 250
2. Login: 150
3. Home Page: 500
4. Securing the passwords using cryptography: 300
5. Profile Pages: 350
6. Discussion Forum: 470
7. Miscellaneous: 430

2.2 Estimation of development effort:

= 2.4 * (KLOC)^1.05
= 2.4 * (2.45)^1.05
= 6.14944 PM

2.3 Estimation of development time:

= 2.5 * (EFFORT)^0.38
= 2.5 * (6.14944)^0.38
= 4.9853 months

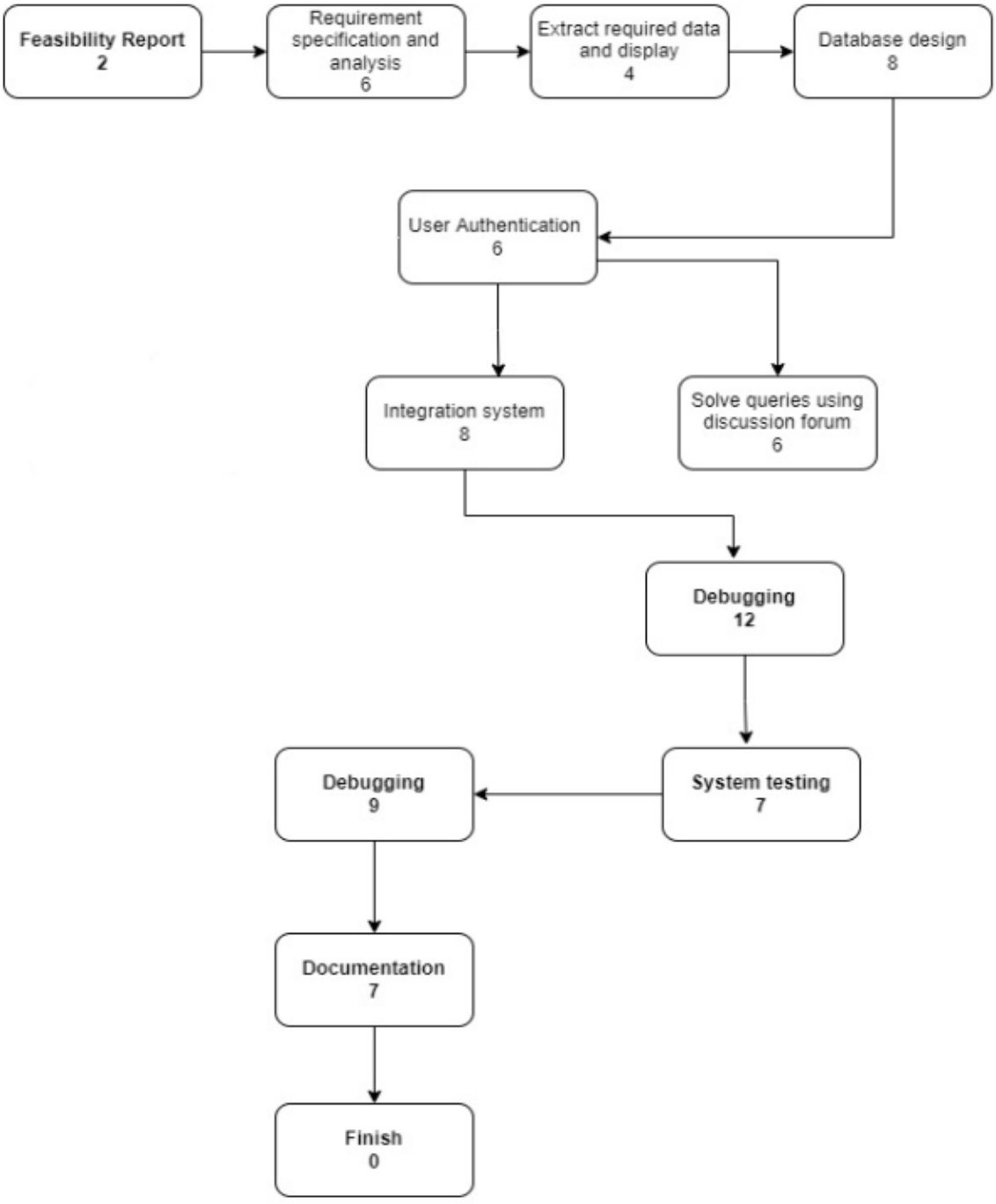
3.0 PROJECT SCHEDULE BREAKDOWN

3.1 Activity Network

An activity network shows different activities making up a project, their estimated durations, and their interdependencies. Two equivalent representations for activity networks are possible are in use:

➔ Activity On Node
➔ Activity On Edge

THE ACTIVITY NETWORK



3.2 PERT Chart

Project Evaluation and review technique (PERT) charts are a more sophisticated form of activity chart. PERT charts like activity networks consist of a network of boxes and arrows. The boxes represent activities, and the arrows represent task dependencies. A PERT chart represents the statistical variations in the project estimates, assuming these to be a normal distribution. PERT allows for some randomness in task completion times and therefore provides the capability to determine the probability of achieving project milestones based on the probability of completing each task along the path to that milestone. Each task is annotated with three estimates:

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

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

Worst case (W): The worst possible case task completion time.

The PERT Chart