Raw Food Items E-Mart

Software Project Management Report

Group - 4 (IMG):

- Abhishek (2020-IMG-002)
- Aman Pachori (2020-IMG-007)
- Siddharth Butoliya (2020-IMG-061)

Submitted to Dr. Santosh Singh

Atal Bihari Vajpayee - Indian Institute of Information Technology & Management, Gwalior

Table of Contents

1. Size Estimation

Project Size estimation is the most crucial and fundamental parameter of a software project. Its accurate estimation simplifies the estimation of other parameters like cost, duration, effort, etc.

Currently, two of the most widely used for estimation project size are

1. Function Point
2. Lines of Code (LOC)

1.1. Function Point Metric

Functional Point Analysis (FPA) is used to estimate a software project. Its metric is directly dependent on the functionality or complexity of the software project. This technique enables managers to determine the size of the project from required software specifications and functionality. It helps track the software development process irrespective of the technologies in the project.

1.2. Unadjusted Function Point Estimation (UFP)

- Inputs:

    1. User’s details (username and password) to access their account. (Simple)
    2. ‘Browse Inventory’ search bar. (Simple)
    3. Sorting preference. (Average)
    4. Adding the desired quantity of an available product to the cart. (Simple)
    5. Promo codes for availing cash discounts. (Simple)
    6. Ratings and written reviews for the products. (Simple)
    7. Item details (including images) for listing an item in the store. (Average)
    8. Replying to the customer reviews. (Simple)
    9. Shipping details by the seller (Average).

Number of inputs: 9

- Outputs:

1. Confirmation message: “Login successful”. (Simple)
2. Confirmation message: “You are successfully logged out”. (Simple)
3. Confirmation message: “Review added successfully”. (Simple)
4. Confirmation message: “Item successfully added to the store”. (Simple)
5. Error message: “Invalid login credentials”. (Simple)
6. Error message: “Review cannot be blank”. (Simple)
7. Error message: “Reply cannot be blank”. (Simple)
8. Error message: “Shopping cart cannot be empty to checkout”. (Simple)
9. Logout Confirmation Prompt: “Please confirm the logout attempt”.(Simple)
10. Confirmation Prompt: “Please confirm to proceed with payment” (Simple)
11. Weekly Report generated by the system. (Complex)

Number of outputs: 11

- Inquiries:

1. View items. (Average)
2. Order status (tracking order). (Average)

Number of inquiries: 2

- Files:

1. List of registered users. (Average)
2. List of items. (Average)
3. Shipping and payment information. (Complex)

Number of files: 3

- Interface:

  1. Customer module. (Complex)
  2. Seller module. (Complex)
  3. Payment gateway API (Razorpay) for facilitating payments. (Average)

  Number of Interfaces: 3

UFP = $9 * 4 + 11 * 5 + 2 * 4 + 3 * 10 + 3 * 10 = 159$

1.3. Refined UFP

	Simple	Average	Complex
Input	6	3	0
Output	10	1	0
Inquiries	0	2	0
Files	0	2	1
Interface	0	1	2

Refined UFP: 6*3 + 3*4 + 10*4 + 1*5 + 2*4 + 2*10 + 1*15 + 1*7 + 2*10

Refined UFP = 145

1.4. DI, TCF, and FP

- Degree of Influence:

  Total Degree of Influence (DI) = 10*3 + 3*4 + 1*5 = 47

- TCF:

  TCF = (0.65 + 0.01*DI)
  = 0.65 + 0.01*47

  TCF = 1.12

- FP:

  FP = UFP*TCF
  = 159*1.12

  FP = 178.08

2. Effort and Time Estimation

2.1. COCOMO Model

The COCOMO model is used to estimate time and effort for any software project. As Boehm postulated that any software development project can be classified into one of the following three categories based on the development complexity—organic, semi-detached, and embedded.

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:

$$Effort = a1 * (KLOC)^{a2}\ PM$$
$$Tdev = b1 * (Effort)^{b2}\ months$$

Where,

- *Effort* is the total effort required to develop the software product, expressed in person-months (PMs).
- *Tdev* is the estimated time to develop the software, expressed in months.
- a1, a2, b1, b2 are constants for each category of the software product.
- KLOC is the estimated size of the software product expressed in Kilo Lines Of Code.

2.2. Estimation of Development Effort

For our project we have estimated the size (in KLOC) by analyzing the already existing similar projects. Estimated size =80 KLOC Our application falls under the category of application software. So, the formulae for organic software will be used to calculate the effort and time of development.

$$
\begin{aligned}
\mathit{Effort} &= a1 * (KLOC)^{a2}\ PM \\
&= 2.4 * (KLOC)^{1.05}\ PM \\
&= 2.4 * (80)^{1.05}\ PM \\
&= 239.031\ PM
\end{aligned}
$$

2.3. Estimation of Development Time

$$
\begin{aligned}
\mathit{Tdev} &= b1 * (Effort)^{b2}\ months \\
&= 2.5 * (239.031)^{0.38}\ months \\
&= 20.033\ months
\end{aligned}
$$

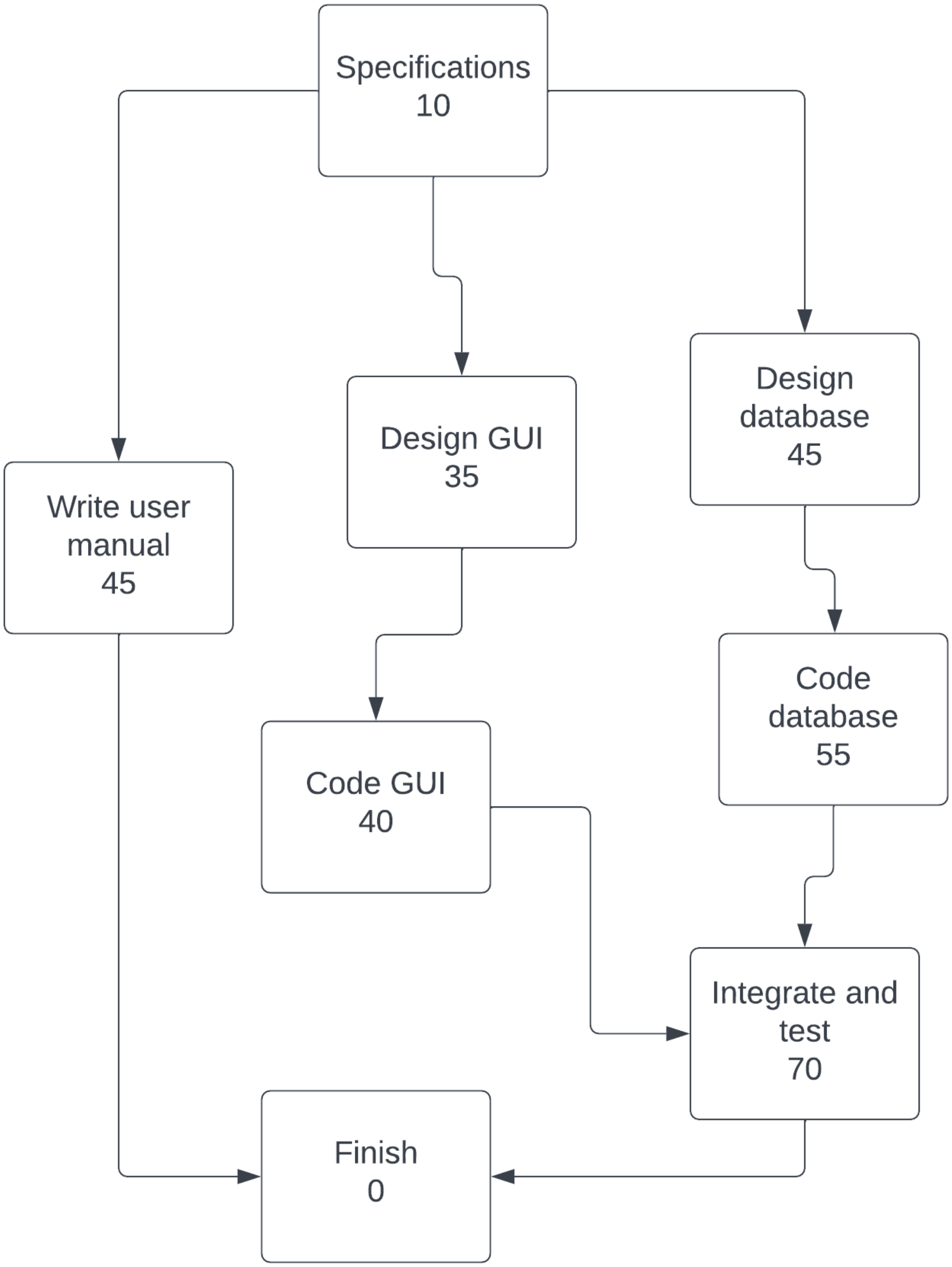
3. Project Schedule Breakdown

3.1. Activity Network Representation

An activity network shows the different activities making up a project, their estimated durations, and their interdependencies. Each activity is represented by a rectangular node and the duration of the activity is shown alongside each task in the node. The inter-task dependencies are shown using directional arrows. The activity representation is given in the table below.

Task Number	Task	Duration	Dependent on Task
T1	Specification	10	-
T2	Design database	45	T1
T3	Design GUI	35	T1

T4	Code database	55	T2
T5	Code GUI part	40	T3
T6	Integrate and test	70	T4 & T5
T7	Write user manual	45	T1



3.2. PERT Chart

PERT (project evaluation and review technique) charts are a more advanced version of the activity chart. The probability timelines for attaining various project milestones, including the final milestone, can be determined using PERT charts.

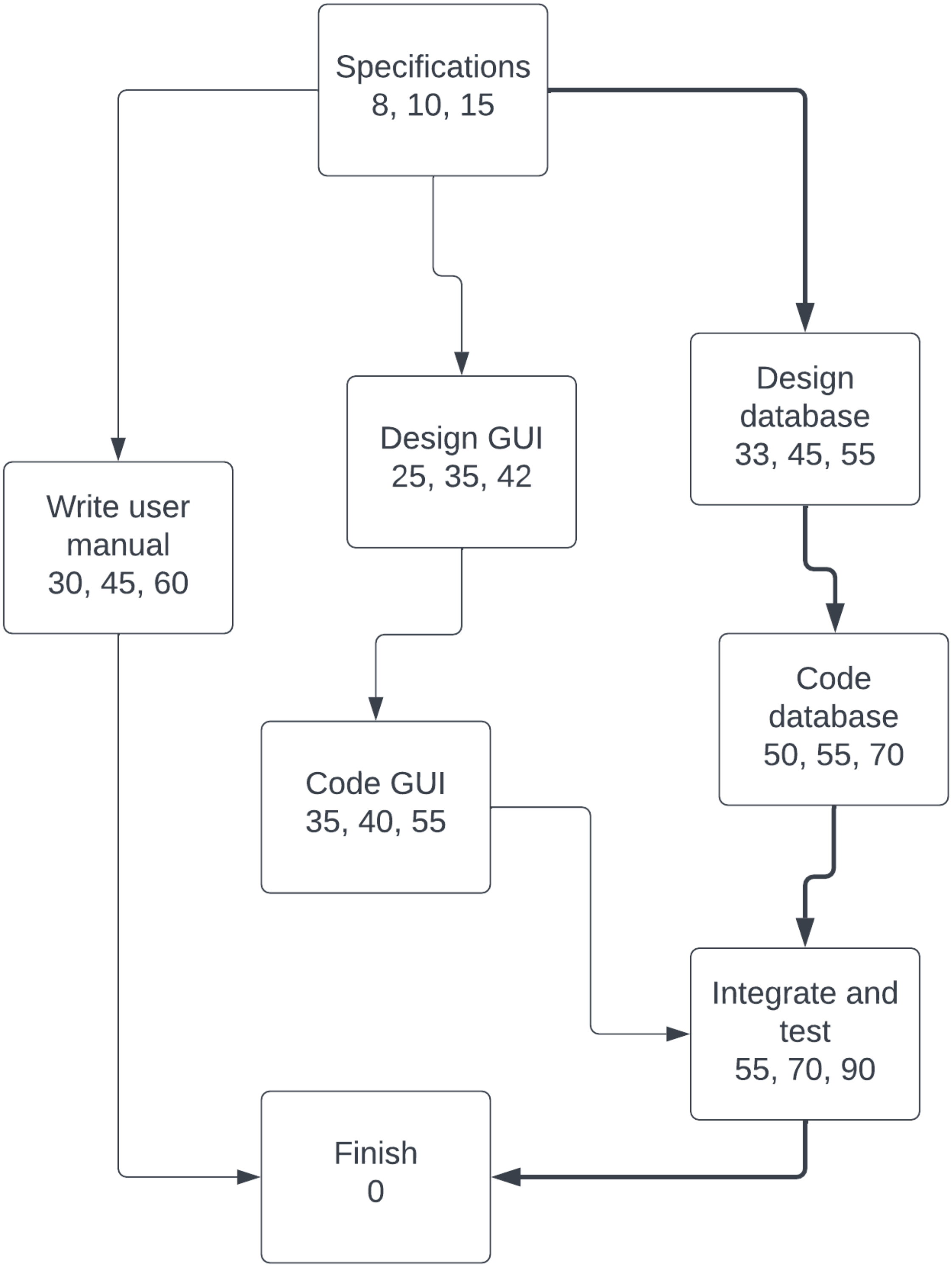
PERT charts are made up of a network of boxes and arrows, similar to activity networks. The arrows reflect task dependencies, whereas the boxes represent activities. The statistical fluctuations in the project estimates are represented by a PERT chart, which assumes a normal distribution.

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 project parameters are given in the following table.

Task	ES	EF	LS	LF	ST
Specification	0	10	0	10	0
Design database	10	55	10	55	0
Design GUI	10	45	35	70	25
Code database	55	110	55	110	0
Code GUI part	45	85	70	110	25
Integrate and test	110	180	110	180	0
Write user manual	10	55	135	180	125


Specifications
8, 10, 15
Design GUI
25, 35, 42
Design database
33, 45, 55
Write user manual
30, 45, 60
Code database
50, 55, 70
Code GUI
35, 40, 55
Integrate and test
55, 70, 90
Finish
0