Software Engineering

Name: Bruksha A. Patel
Sapid: 60004210126
Branch: Computer Engineering
Div: C2 Batch: 1

Experiment no. 06

Aim: Estimate effort and cost required using FP/COCOMO for the project. Create WBS and Gnatt chart for the same. Use PM Tool to depict a project plan.

Theory:

FP calculation:-

Functional units		Low	Avg	High	Total
EI	8 (L)	(3)	4	6	24
EO	6 (A)	4	(5)	7	30
EQ	5 (H)	3	4	(6)	30
ILF	2 (A)	7	(10)	15	20
ELF	3 (H)	5	7	(10)	30

External Information Domain Value	Count	Simple	Average	Complex	Total
External Inputs	4	3	4	6	4×4 = 16
External Inquiry	3	4	5	7	3×5 = 15
Internal Logical files	4	3	4	6	4×4 = 16
External outputs	2	7	10	15	2×10 = 20
External Interface files	0	5	7	10	0×7 = 0
			Total		67

Total count = 67

External Inputs (EI):
- Loan Application (from user)
- Verification of Employment (from Employer)
- Bank statements (from user)
- Credit History (from Bank)

External Inquiries (EQ):
- Balance Inquiry
- Transaction history
- Payment status check.

Internal Logic Files (ILF):
- User Information.
- Loan products
- Loan Applications
- User Accounts.

External outputs (EO):
- Loan Approval Notification (to user)
- Loan Disbursement (to user's Bank)

External Interface files (EIF)
- None

1] Does the system require reliable backup & recovery?

→ (4) Absolutely. Loan applications contain sensitive personal and financial data. System outages or data loss could have serious consequences. Reliable backup and recovery procedures are crucial.

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

→ (3) Potentially. The system might need to communicate with external sources like credit bureaus, banks, or government databases to verify information. However, the communication might not be highly specialized compared to complex e-commerce integrations.

3] Are there distributed processing functions?

→ (2) While not as critical as in a high-traffic e-commerce platform, the system might benefit from distributed processing during peak application periods. This could involve distributing workload across multiple servers to ensure smooth operation.

4] Is performance critical?

→ (3) High performance is desirable, but not necessarily as critical as in an e-commerce platform. Fast response times improve user experience, but loan applications generally don't require real-time processing for each step.

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

→ (4) Likely. The loan application system would likely integrate with existing bank or financial institution infrastructure, which

could be heavily utilized.

6] Does the system require online data entry?
→ (4) yes, users will interact with the system online to submit loan applications & potentially upload documents.

7] Does the online data entry require the input transaction to be built over multiple screens or operations?
→ (3) Probably. Loan applications typically involve multiple steps, gathering user information, financial details, & potentially document uploads.

8] Are the ILFs updated online?
→ (4) Ideally, internal logical files (eg. user information, loan products) should be updated online to reflect the latest data & avoid inconsistencies.

9] Are the inputs, outputs, files or inquiries complex?
→ (2) Loan application data might be complex, including personal information, income details, and asset information. However, compared to a full-fledged e-commerce system, the complexity might be lower.

10] Is the Internal processing complex?
→ (3) Internal processing could involve calculations, creditworthiness assessment, and rule-based decisions. The complexity depends on the loan types offered & the specific risk management strategies employed.

11] Is the code designed to be reusable?

(3) while code reusability is always desirable, it might not be the top priority if the loan application system is standalone. However, if it integrates with other systems within the organization, reusability becomes more important.

12] Are conversion & installation included in the design?

(3) Conversion from existing systems may be needed if the loan application system replaces an older process. Installation complexity depends on the chosen platform & infrastructure.

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

(2) Most likely, the loan application system will cater to a single organization (eg, a specific bank). Multi-organization support would require additional design considerations.

14] Is the application designed to facilitate change & ease of use by the user?

(4) absolutely. A user-friendly & intuitive interface is crucial for both the loan applicant & any internal users who manage the application process.

P. T. O.

$\Sigma(f_i) = 44$

The estimated no. of FP is derived.

FP estimated = count-total $\times$ $[0.65 + 0.01 \times \Sigma(F_i)]$

$= 67 \times [0.65 + 0.01 \times 44]$

$= 73.03$

Therefore, FP estimated is 73.03 pm

Conclusion :-

Thus, we implemented FP/ cocomo on our project go
This answer and made work break down structure
gnatt chart for our project

Preksha Ashok Patel

60004210126

C21

Software Engineering

Experiment No- 6

External Inputs (EI):

- Loan Application (from User)
- Verification of Employment (from Employer)
- Bank Statements (from User)
- Credit History (from Bank)

External Inquiries (EQ):

- Balance Inquiry
- Transaction history
- Payment status check

Internal Logical Files (ILF):

- User Information
- Loan Products
- Loan Applications
- User Accounts

External Outputs (EO):

- Loan Approval Notification (to User)
- Loan Disbursement (to User's Bank)

External Interface Files (EIF):

- None

Information Domain Value	Count	Simple	Average	Complex	Total
External Inputs	4	3	**4**	6	4*4 = 16
External Inquiry	3	4	**5**	7	3*5 = 15
Internal logical Files	4	3	**4**	6	4*4 = 16
External Outputs	2	7	**10**	15	2*10 = 20
External Interface Files	0	5	**7**	10	0*7 = 0
Total					**67**

Total count = 67

1. Does the system require reliable backup and recovery?

→ (4) Absolutely. Loan applications contain sensitive personal and financial data. System outages or data loss could have serious consequences. Reliable backup and recovery procedures are crucial.

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

→ (3) Potentially. The system might need to communicate with external sources like credit bureaus, banks, or government databases to verify information. However, the communication might not be highly specialized compared to complex e-commerce integrations.

3. Are there distributed processing functions?

→ (2) While not as critical as in a high-traffic e-commerce platform, the system might benefit from distributed processing during peak application periods. This could involve distributing workload across multiple servers to ensure smooth operation.

4. Is performance critical?

→ (3) High performance is desirable, but not necessarily as critical as in an e-commerce platform. Fast response times improve user experience, but loan applications generally don't require real-time processing for each step.

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

→ (4) Likely. The loan application system would likely integrate with existing bank or financial institution infrastructure, which could be heavily utilized.

6. Does the system require online data entry?

→ (4) Yes, users will interact with the system online to submit loan applications and potentially upload documents.

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

→ (3) Probably. Loan applications typically involve multiple steps, gathering user information, financial details, and potentially document uploads.

8. Are the ILFs updated online?

→ (4) Ideally, internal logical files (e.g., user information, loan products) should be updated online to reflect the latest data and avoid inconsistencies.

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

→ (2) Loan application data might be complex, including personal information, income details, and asset information. However, compared to a full-fledged e-commerce system, the complexity might be lower.

10. Is the internal processing complex?

→ (3) Internal processing could involve calculations, creditworthiness assessments, and rule-based decisions. The complexity depends on the loan types offered and the specific risk management strategies employed.

11. Is the code designed to be reusable?

→ (3) While code reusability is always desirable, it might not be the top priority if the loan application system is standalone. However, if it integrates with other systems within the organization, reusability becomes more important.

12. Are conversion and installation included in the design?

→ (3) Conversion from existing systems may be needed if the loan application system replaces an older process. Installation complexity depends on the chosen platform and infrastructure.

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

→ (2) Most likely, the loan application system will cater to a single organization (e.g., a specific bank). Multi-organization support would require additional design considerations.

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

→ (4) Absolutely. A user-friendly and intuitive interface is crucial for both the loan applicant and any internal users who manage the application process.

Σ (Fi) = 44

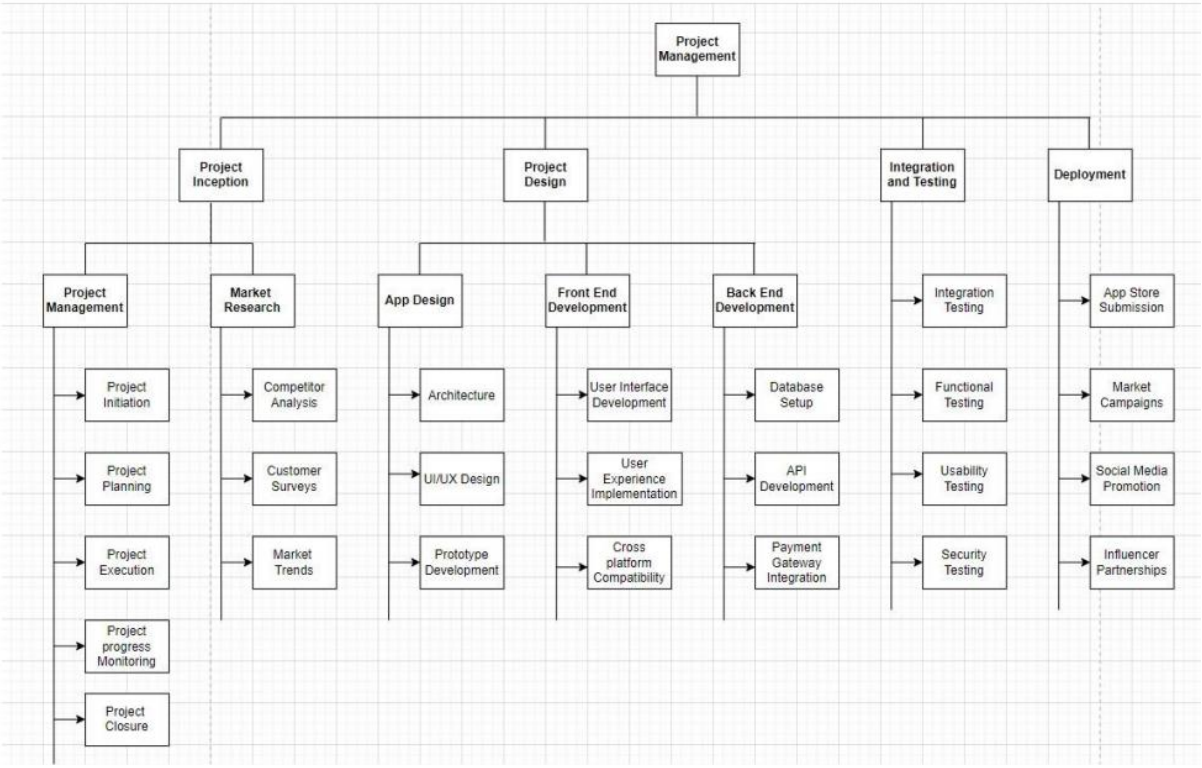
The estimated number of FP is derived:

FP estimated = count-total x [0.65 + 0.01 x Σ (Fi)] =

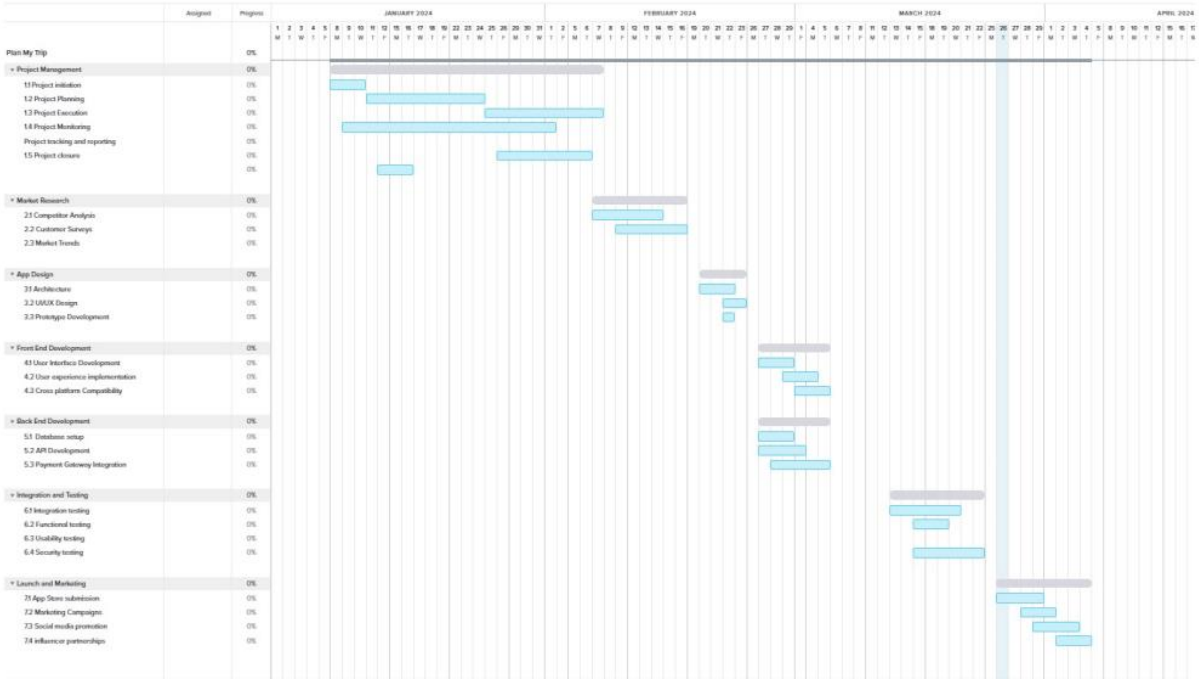
= 67* [0.65 + 0.01 x 44] = 73.03

Therefore, FP estimated is 73.03 pm

Work Breakdown Structure



Gantt Chart


Assignee
Progress
JANUARY 2024
FEBRUARY 2024
MARCH 2024
APRIL 2024
Plan My Trip
0%
Project Management
1.1 Project initiation
1.2 Project Planning
1.3 Project Execution
1.4 Project Monitoring
Project tracking and reporting
1.5 Project closure
Market Research
2.1 Competitor Analysis
2.2 Customer Surveys
2.3 Market Trends
App Design
3.1 Architecture
3.2 UI/UX Design
3.3 Prototype Development
Front End Development
4.1 User Interface Development
4.2 User experience implementation
4.3 Cross platform Compatibility
Back End Development
5.1 Database setup
5.2 API Development
5.3 Payment Gateway Integration
Integration and Testing
6.1 Integration testing
6.2 Functional testing
6.3 Usability testing
6.4 Security testing
Launch and Marketing
7.1 App Store submission
7.2 Marketing Campaigns
7.3 Social media promotion
7.4 Influencer partnerships