

Function Point Analysis

This section describes Function Point Analysis of Agri-Inn Project.

6.1. What' is Funct'ion Point'?

Function point (FP) measure the size of an application system based on the functional view of the system. Function Point relies on 5 things:

1. External inputs (EI)
2. External outputs (EO)
3. External inquiries (EQ)
4. External Interface Files (EIF)
5. Internal Logical Files (ILF)

The size is determined by counting the number of these factors in the system and adjusting that total for the functional complexity of the system.

$$FP = UFC * TCF$$

6.2. What' is Funct'ion Point' Analysis (FPA)?

Function Point Analysis (FPA) quantifies software functionality through a weighted measure. It suggests that a product showing greater functionality will have a larger size. Additionally, FPA provides a standardized method for estimating software size by aiding in project planning and resource allocation ultimately enhancing development efficiency and accuracy.

6.3. Typss or FPA

The types of FPA are discussed below:

- **External Input (EI):** EI takes data as input or control information that comes from outside the application's boundary. EG :input screen or business documents or external database or external tables or files i.e.the inputs to the application from outside but within business criteria boundary.
- **External Output (EO):** EO is an elementary process that generates and displays data or control information sent outside the application's boundary. Eg: reports, screens,error messages and so on.

- **External Inquiries (EO):** EQ is an elementary process made up of an input and output combination that results in data retrieval. Eg: Prompts, interrupts and so on. Eg: Prompts and interrupts, etc.
- **Internal Logical File (ILF):** A user identifiable group of logically related data or control information maintained within the boundary of the application. Eg: Databases and directories, etc.
- **External Interface File (EIF):** A group of user recognizable logically related data allusion to the software but maintained within the boundary of another software. Eg: Shared databases and shared routines and so on.

6.4. Advant'agss and Disadvant'agss or FPA

6.4.1. Advant'agss or FPA

The advantages of FPA include:

1. We can calculate Function Point before there are even any design or code documents.
2. They are handy for estimating project costs, effort and schedule early on in the project.
3. They assist in negotiating contracts.
4. Their applicability is not tied to any specific programming languages.

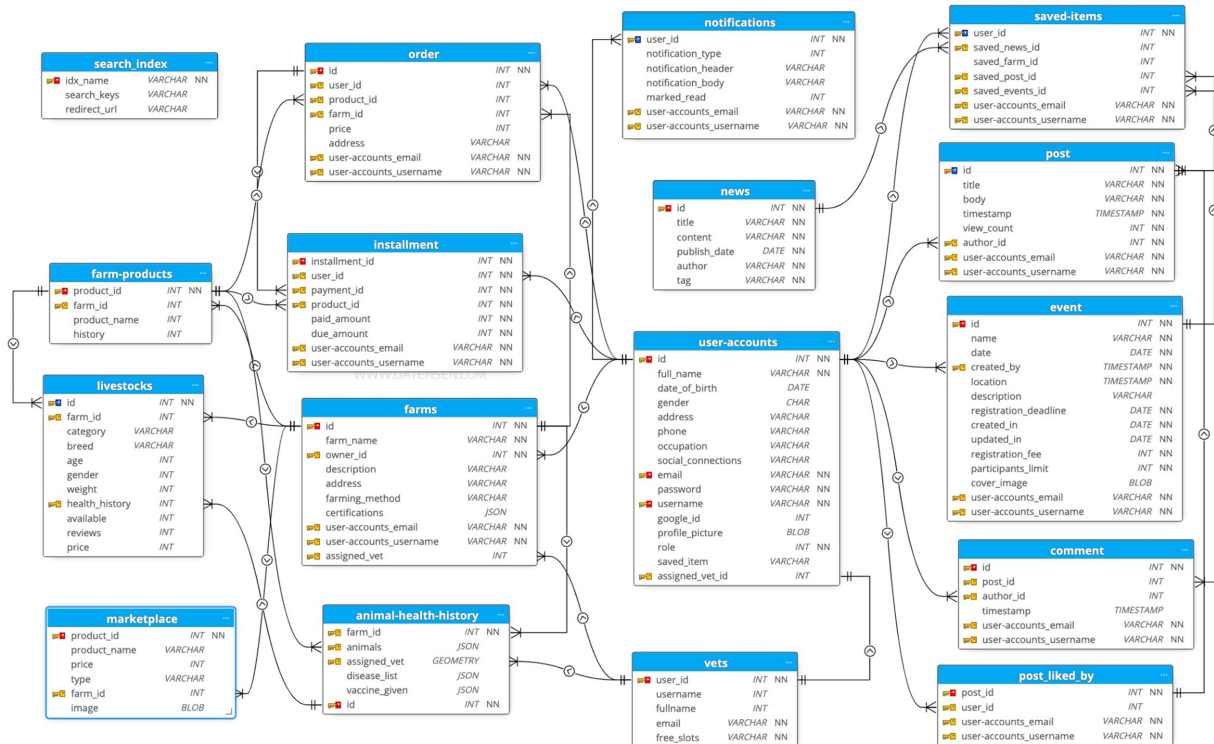
6.4.2. Disadvant'agss or FPA

The disadvantages of FPA include:

1. Function Point can vary based on how users choose to assign weights by introducing subjectivity. Some of them are :
 - Issues arise with subjectivity particularly concerning technology factors.
 - Instances of double counting can occur.
 - Subjective weighting presents challenges.
2. Calculating Function Points necessitates a comprehensive software system specification by making it challenging to utilize them in the early stages of software development.
3. Challenges arise with changing requirements.
4. Function Points are not ideal for complex software project types like real time and embedded applications.

6.5. ER-Diagram of Agri-Inn

This section shows an abstract view of the ER-diagram of Agri-Inn. Note that the diagram does not represent the complete system overview.



6.6. Function Point Calculation of Agri-Inn

This section calculates the Function points for the Agri-Inn project.

The formula for calculating function point is $FP = UFC \times TCF$; where FP is the Function Point, UFC is the Unadjusted Function Point Count and TCF is the Technical Complexity Factor. UFC is also known as UFP, Unadjusted Function Point and TCF is also known as VAF, Value Adjustment Factor or CAF, Complexity Adjustment Factor. To calculate Function Point for Agri-Inn, we need to calculate UFC and TCF .

6.6.1. Unadjusted FP count (UFC)

The unadjusted function point (UFP) count in COCOMO is the initial calculation of the size of a software project based on its functionality. It's determined by counting the five types of functional components (See Section 6.3) in the system and assigning a weight to each.

The table shows all the five function counts for Agri-Inn, considering all the features:

Function Name	EI (3)	EO (4)	Eq (4)	ILF (15)	EIF (10)
1. Disease History	1	1	1	1	1
2. Appointments	1	1	1	1	1
3 Vaccination	1	1	1	1	1
4. Livestock Management	3	3	1	1	1
5. Financial Analysis	5	5	3	3	1
6. Product Listing System	3	3	2	3	1
7. Inventory	1	2	1	2	1
8. Reviews	3	3	0	1	1
9. Cart	2	2	0	0	0
10. Secure Payment System	3	1	0	1	1
11.Order Management	6	2	1	1	1
13.Installment	1	2	0	1	1
14. News	0	1	1	1	1
15.Announcements	0	1	1	1	1
16.Search and query	0	1	9	1	1
17.Notifications	0	1	0	1	1
18.Saved Items	0	1	1	1	1
19.Chat	1	1	0	1	1
20.Post Interactions	4	4	0	1	1
21.Post Management	3	3	1	1	1
22.Event Management	3	2	3	1	1
23.Events Participation	2	3	1	1	1
TOTAL FP COUNT	43	44	28	26	21

As it is shown in the table, the functional component count of external input (EI) is 43; external output (EO) is 44; external inquiries (EQ) is 28; internal logical file (ILF) is 26 and external interface file (EIF) is 21.

The equation of calculating unadjusted function point count (UFC) is:

$$UFP = w_{EI} \times N_{EI} + w_{EO} \times N_{EO} + w_{EQ} \times N_{EQ} + w_{ILF} \times N_{ILF} + w_{EIF} \times N_{EIF}$$

Here, all the component counts (N) are determined. To determine the weights, the following table is followed.

Item	Simple	Average	Complex
EI	3	4	6
EO	4	5	7
Eq	3	4	6
ILF	7	10	15
EIF	5	7	10

For Agri-Inn the weight of external input is simple (3); external output is simple (4); external inquiry is average (4); weight of internal logical file (ILF) is complex (15). The weight of the external interface file (EIF) needs to be calculated. From the section 6.5, the ER-Diagram shows that:

- Number of RET: 17
- Number of DET: 145

With this, the weight of external interface files can be calculated by the following table:

RETs	DETs		
	1 - 19	20 -50	51
1	Low	Low	Average
2 to 5	Low	Average	High
> 5	Average	High	High

As the RET is greater than 5 and DET is greater than 51; the weight of the external interface file is High (10).

Since all the values are determined, the unadjusted function point is:

$$UFC = (3 \times 43) + (4 \times 44) + (4 \times 28) + (15 \times 26) + (10 \times 21) = 1017$$

So, Unadjusted Function Point Count is 1017.

6.6.2 Technical Complexity Factor (TCF)

In COCOMO, the technical complexity factor (TCF) is a concept present in the Intermediate COCOMO model, not the Basic COCOMO which uses unadjusted function points (UFP). The TCF refines the UFP count by considering additional factors that influence the inherent complexity of the software being developed. These factors go beyond the basic functionality and delve into technical characteristics that can make the project more challenging to build.

There are a total of 14 factors, each ranging from zero to five. The values represent the level of priority or importance in a system. The level of importance are given below:

- 0 * No influence
- 1 * Incidental
- 2 * Moderate
- 3 * Average
- 4 * Significant
- 5 * Essential

Each factor is assigned one of the points mentioned above. Below is the table that shows all the complexity factors assigned for Agri-Inn:

Technical Factors	Factor points
F1 Reliable backup and recovery	4
F2 Data communications	5
F3 Distributed functions	3
F4 Performance	2
F5 Heavily used configuration	3
F6 Online data entry	5
F7 Operational ease	4
F8 Online update	5
F9 Complex interface	2
F10 Complex processing	5
F11 Reusability	4
F12 Installation ease	4
F13 Multiple sites	5
F14 Facilitate	4

In the table, ΣF shows the total summation of all factor points. The equation for calculating the technical complexity factor (TCF):

$$TCF = 0.65 + (0.01 \times \Sigma F)$$

As we calculated ΣF to be 55, we can calculate the TCF by the equation. So, TCF is $0.65 + (0.01 \times 55) = 1.20$.

From this section, the Technical Complexity Factor is 1.20

6.6.3. Counting Function Point (FP)

As mentioned above, to calculate function point (FP), unadjusted function point count and technical complexity factor is multiplied. From section 6.6.1 and 6.6.2:

- Unadjusted Function Point * 1017
- Technical Complexity Factor * 1.20

So, function point is $1017 \times 1.20 = 1220.4$.

From this section, the Function Point is 1220.4

6.7. Explanation or Terminology

This section explains the terms **External Input (EI)**, **External Output (EO)**, **External Inquiries (EO)**, **Internal Logical Files (ILF)**, **External Interface Files (EIF)**, **Total DET** and **Total RET** in context of Agri-Inn mentioned in 6.6.1 :

1. **Disease History :**
 - **EI** = data entry for animals' disease
 - **EO** = viewing the animal's disease history
 - **EO** = queries for specific disease information
2. **Appointments :**
 - **EI** = Data entry related to scheduling appointments
 - **EO** = Displaying appointment information on the screen for users
 - **EO** = Looking for information about available appointment slots or specific veterinarian availability
3. **Vaccination :**
 - **EI** = Data entry related to vaccination
 - **EO** = vaccination messages
 - **EO** = vaccination information about a specific animal

4. **Livestocfi Management :**

- **EI** =adding data entry for adding new livestock to the system, including details such as species, breed, age, gender etc ; deleting them ; update their information
- **EO** =viewing the these information ; generating reports ; about them; sending farm owners alert incase of any emergency of an animal
- **EO** = searching any particular information about an animal.

5. **Financial Analysis :**

- **EI** =Importing financial data from accounting software or spreadsheets like income statements,etc;Input data related to expenses like as operating costs etc;Entering data about revenue sources , update , delete any item
- **EO** =Generating financial reports such as income statements etc,Producing graphical visualizations ,alerts,producing summaries, interactive simulations
- **EO** = searching previous financial records,budget analysis queries,sorting financial information.

6. **Product listing :**

- **EI** = Adding ; updating ;deleting –product details
- **EO** = Product display; product recommendations; generating product reports
- **EO** = product search, inventory availability check.

7. **Reviews :**

- **EI**=User adding reviews, admin approving review , admin Rating Scale Customization
- **EO** = Displaying reviews;displaying rating;Providing review summaries
- **EO** : nothing

8. **Cart :**

- **EI** = Adding items to the cart ; Updating cart contents
- **EO** = Displaying the cart summary;Order confirmation message
- **EO** : nothing

9. **Secure Payment System :**

- **EI** = adding User payment information, adding Transaction details ; adding User authentication data
- **EO** = Payment Confirmation message
- **EO** = nothing

10. **Order Management :**

- **EI** = Addition of Customer Order Entry ;Order Modification,Order Cancellation;Price Adjustments,Inventory Updates,Adding Order Discount
- **EO** = Order Confirmation message
- **EO** : Order Status Inquiry

11. **Inventory :**

- **EI** = Inputting information about new products
- **EO** =Inventory Reports;Low Stock Alerts

- **EO** = Product Availability Inquiry.
12. **Installment :**
 - **EI** : Customer Payment Information
 - **EO** =Payment Confirmation Notifications;Payment Reminders
 - **EO** : Nothing.
 13. **News :**
 - **EI** : nothing
 - **EO** :News Display
 - **EO** : news search
 14. **Announcements :**
 - **EI** : nothing
 - **EO** : notifications
 - **EO** : query related to particular notification
 15. **Search and Query :**
 - **EI** =nothing
 - **EO** =Search Results Display
 - **EO** = Query for any particular thing
 16. **Notifications :**
 - **EI** =nothing /0
 - **EO** =Push Notifications
 - **EO** = nothing
 17. **Saved Items :**
 - **EI** = 0
 - **EO** = view Saved Items List
 - **EO** = Search Saved Items
 18. **Chat :**
 - **EI** = input text
 - **EO** = view message
 - **EO** = 0
 19. **Post Interactions :**
 - **EI** =User Comments,Upvotes,Shares, Post Tags
 - **EO** =view comments,upvotes ,shares , tags
 - **EO** = 0
 20. **Post Management:**
 - **EI** =Post Creation Inputs,Content Submission Inputs,Metadata Entry Inputs
 - **EO** = Post view,Engagement Metrics,post submission confirmation message
 - **EO** = specific post search
 21. **Event Management :**
 - **EI** = Addition of Attendee Registration,event creation, event deletion
 - **EO** =view events,event report
 - **EO** = search events
 22. **Events Participation :**

- **EI** = Participant Registration data , Event Details Update
- **EO** =Event Confirmation Emails ,Participant Lists
- **EO** = Event Availability Inquiry.

ILF : the files API consisting of the information of the above mentioned features or functions that stores their related information.

EIF : All the documents as the databases of the MONGODB used.

DET : All the unique attributes or fields from each documents.For eg:

For the livestock Management feature mentioned above in 4 no. in 6.6.1- the Det are : product_id,farm_id,farm_name,history.

RET : the record consisting of a subgroup of data elements.For example in 4 no. Feature or function in 6.6.1 :

a record representing the information of a specific livestock.

6.8. Justification or Complexity Weights or Functional Components

For Agri-Inn, while calculating the Functional Components, the complexity weights is assigned as the below:

- External Input (EI) * Simple (3)
- External Output (EO) * Simple (3)
- External Inquiries (Eq) * Average (4)
- Internal Logical Files (ILF) * Complex (15)
- External Interface Files (EIF) * Complex (10)

Now, this section describes the reasoning behind assigning the complexities to the 5 parameters

Justification or External Input (EI)

In the Agri-Inn project, all the external input contains filling up simple forms with simple data representation. These inputs require less processing power than manipulating and storing that data in the system. It always involves basic validation or formatting checks before entering the system and the real process happens after the system receives the complete data. Also, the input data types are basic and simple, for example, names, id, address or date. This reduces the complexity of handling them compared to unique or custom data structures within the system. All the external input has simple or zero underlying logic as it does not