

Laboratory

3

FP (Function Point) ORIENTED ESTIMATION MODEL

Calculating effort for Attendance Management System using FP oriented model.

Objective

Calculating effort for Attendance Management System using Function Point oriented estimation model. It is a method to break systems into smaller components, so they can be better understood and analyzed. It is used to express the amount of business functionality, an information system (as a product) provides to a user. Fps measure software size. They are widely accepted as an industry standard for functional sizing. Function points are used to compute a functional size measurement (FSM) of software. The cost (in dollars or hours) of a single unit is calculated from past projects. Function Point Analysis can provide a mechanism to track and monitor scope creep. Function Point Counts at the end of requirements, analysis, design, code, testing and implementation can be compared. The function point count at the end of requirements and/or designs can be compared to function points actually delivered. The amount of growth is an indication of how well requirements were gathered by and/or communicated to the project team. If the amount of growth of projects declines over time it is a natural assumption that communication with the user has improved.

Overview

Function-oriented software metrics use a measure of the functionality delivered by the application as a normalization value. Since 'functionality cannot be measured directly, it must be derived indirectly using other direct measures. Function-oriented metrics were first proposed by Albrecht, who suggested a measure called the function point. Function points are derived using an empirical relationship based on countable (direct) measures of software's information domain and assessments of software complexity. Function points are computed by completing the table as

shown below. Five information domain characteristics are determined and counts are provided in the appropriate table location. Information domain values are defined in the following manner:

| Measurement parameter | Count | Weighting factor | | | | |
|-------------------------------|----------------------|------------------|---------|---------|---|----------------------|
| | | Simple | Average | Complex | | |
| Number of user inputs | <input type="text"/> | x 3 | 4 | 6 | - | <input type="text"/> |
| Number of user outputs | <input type="text"/> | x 4 | 5 | 7 | - | <input type="text"/> |
| Number of user inquiries | <input type="text"/> | x 3 | 4 | 6 | - | <input type="text"/> |
| Number of files | <input type="text"/> | x 7 | 10 | 15 | - | <input type="text"/> |
| Number of external interfaces | <input type="text"/> | x 5 | 7 | 10 | - | <input type="text"/> |
| Count total | → | | | | | <input type="text"/> |

Number of user inputs: Each user input that provides distinct application oriented data to the software is counted. Inputs should be distinguished from inquiries, which are counted separately.

Number of user outputs: Each user output that provides application oriented information to the user is counted. In this context output refers to reports, screens, error messages, etc. Individual data items within a report are not counted separately.

Number of user inquiries: An inquiry is defined as an on-line input that results in the generation of some immediate software response in the form of an on-line output. Each distinct inquiry is counted.

Number of files: Each logical master file (i.e., a logical grouping of data that may be one part of a large database or a separate file) is counted.

Number of external interfaces: All machine readable interfaces (e.g., data files on storage media) that are used to transmit information to another system are counted. Once these data have been collected, 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.

Procedure

FPA provides different estimation mechanism within it for development and maintenance projects. (having different multiplication factors). This approach computes the total function points (FP) value for the project, by totaling the number of external user inputs, inquiries, outputs, and master files, and then applying the following weights: inputs, outputs, inquiries, and master files.

FP POINTS COMPUTATION

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

$$FP = \text{count total} [0.65 + 0.01 \sum (F_i)]$$

where count total is the sum of all FP entries .

The F_i ($i = 1$ to 14) are "complexity adjustment values" based on responses to the following questions :

1. Does the system require reliable backup and recovery?
2. Are data communications required?
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 on-line data entry?
7. Does the on-line data entry require the input transaction to be built over multiple screens or operations?
8. Are the master files updated on-line?
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?

Each of these questions is answered using a scale that ranges from 0 (not important or applicable) to 5 (absolutely essential). The constant values in Equation and the weighting factors that are

applied to information domain counts are determined empirically. Once function points have been calculated, they are used in a manner analogous to LOC as a way to normalize measures for software productivity, quality, and other attributes:

Errors per FP.

Productivity = FP/ Person-Month

Quality = No of faults / FP

Cost= \$/FP

Documentation = Pages count / FP.

Effort = FP/ Person-Month

Count Total can be obtained using the following table

| Domain characteristics | Count | | Weighting factor | | | Count |
|---------------------------|-------|---|------------------|---------|---------|-------|
| | | | Simple | Average | Complex | |
| No of user input | | * | 3 | 4 | 6 | |
| No of user output | | * | 4 | 5 | 7 | |
| No of user queries | | * | 3 | 4 | 6 | |
| No of files | | * | 7 | 10 | 15 | |
| No of external interfaces | | * | 5 | 7 | 10 | |
| Count total: | | | | | | |

Example:

Assume that....

Number of user input : 5

Number of user output : 5

Number of user enquires : 6

Number of files :5

Number of external interfaces : 5

Apply these assumptions on a **simple project** and calculate the Count Total

| Domain characteristics | Count | * | Weighting factor | | | Count |
|---------------------------|-------|---|------------------|---------|---------|-------|
| | | | Simple | Average | Complex | |
| No of user input | 5 | * | 3 | 4 | 6 | 15 |
| No of user output | 5 | * | 4 | 5 | 7 | 2 |
| No of user queries | 6 | * | 3 | 4 | 6 | 18 |
| No of files | 5 | * | 7 | 10 | 15 | 35 |
| No of external interfaces | 5 | * | 5 | 7 | 10 | 25 |
| Count total: | | | | | | 113 |

Therefore Count Total =113

Now calculate the Functional Points using $FP = \text{count total} * 0.65 + 0.01 \sum (Fi)$

$$FP = \text{count total} * 0.65 + 0.01 \sum (Fi)$$

$$= 113 * (0.65 + 0.01 * 25) \text{ where } \sum (Fi)$$

$$=25$$

i.e the questions answered using a scale that ranges from 0 (not important or applicable) to 5 (absolutely essential) in total 14 questions

$$=113 * (0.65 + 0.25)$$

$$=113 * 0.9$$

$$= 101.7$$

$$FP = 101.7$$

$$Effort = FP / \text{person-month}$$

ADVANTAGES:

1. This method is independent of programming languages.
2. It is based on the data which can be obtained in early stage of project
3. Function Points are easily understood by the non technical user. This helps communicate sizing information to a user or customer.

DISADVANTAGES:

1. This method is more suitable for Business systems and can be developed for that domain
2. Many aspects of this method are not validated
3. The functional point has no significant ant meaning, it's just a numerical value.

Viva-Voice Questions

1. What is function point analysis?
2. How do you estimate effort for a software project by using function point analysis?
3. What are the five domain characteristics of FPA?
4. How to calculate the function point effort?
5. What are the pros and cons of a FPA?

Lab Report

After successful completion of this lab, the student will have the ability to calculate the effort of a software project by using Function Point Analysis.

Author

Y.Siva Ramaiah
Asst.Professor
CSE Department
Teaching Exp: 2 Yrs
Subjects Taught:PPL,HCI,SE,CO



MVR COLLEGE OF ENGINEERING AND TECHNOLOGY

E-m@il id:sivaramaiah999@gmail.com