**SPM Lab Activity-04**

**Project Estimation using Function Point Analysis**

**1. Understanding Software Estimation Techniques**

**1.1 Introduction**

Once a project is found to be feasible, software project managers undertake project planning. Project planning is undertaken and completed even before any development activity starts. Estimating the following attributes of the project is an essential activity of project planning.

***1. Project size****: What will be problem complexity in terms of the effort and time required to develop the product?*

***2. Cost****: How much is it going to cost to develop the project?*

***3. Duration****: How long is it going to take to complete development?*

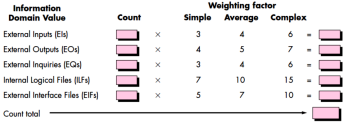
***4. Effort****: How much effort would be required?*

**1.2 Function point (FP)**

Function point metric was proposed by Albrecht [1983]. This metric overcomes many of the shortcomings of the LOC metric. Since its inception in late 1970s, function point metric has been slowly gaining popularity. One of the important advantages of using the function point metric is that it can be used to easily estimate the size of a software product directly from the problem specification.

This is in contrast to the LOC metric, where the size can be accurately determined only after the product has fully been developed. The conceptual idea behind the function point metric is that the size of a software product is directly dependent on the number of different functions or features it supports. A software product supporting many features would certainly be of larger size than a product with less number of features. Each function when invoked reads some input data and transforms it to the corresponding output data.

The size of a product in function points (FP) can be expressed as the weighted sum of these five problem characteristics. The weights associated with the five characteristics were proposed empirically and validated by the observations over many projects. Function point is computed in two steps. The first step is to compute the **unadjusted function point** (UFP).



**Number of inputs:** Each data item input by the user is counted. Data inputs should be distinguished from user inquiries. Inquiries are user commands such as print-account-balance. Inquiries are counted separately. It must be noted that individual data items input by the user are not considered in the calculation of the number of inputs, but a group of related inputs are considered as a single input.

**Number of outputs:** The outputs considered refer to reports printed, screen outputs, error messages produced, etc. While outputting the number of outputs the individual data items within a report are not considered, but a set of relate data items is counted as one input.

**Number of inquiries:** Number of inquiries is the number of distinct interactive queries which can be made by the users. These inquiries are the user commands which require specific action by the system.

**Number of files:** Each logical file is counted. A logical file means groups of logically related data. Thus, logical files can be data structures or physical files.

**Number of interfaces:** Here the interfaces considered are the interfaces used to exchange information with other systems. Examples of such interfaces are data files on tapes, disks, communication links with other systems etc.

Once the unadjusted function point (UFP) is computed, the **technical complexity factor** (TCF) is computed next. TCF refines the UFP measure by considering fourteen other factors such as high transaction rates, throughput, and response time requirements, etc. Each of these 14 factors is assigned from 0(not present or no influence) to 5 (strong influence). The resulting numbers are summed, yielding the total degree of influence (DI).

The Di (i = 1 to 14) are value adjustment factors (VAF) based on responses to the following questions [Lon02]:

1. Does the system require reliable backup and recovery?

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

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 online data entry?

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

8. Are the ILFs updated online?

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?

Now, TCF is computed as(0.65+0.01\*DI).

As DI can vary from 0 to 70, TCF can vary from 0.65 to 1.35.

Finally, FP=UFP\*TCF.

**Example**

Compute function point value for a project with the following domain characteristics: No. of I/P = 30 (Simple), No. of O/P = 50 (Average) , No. of user Inquiries = 24( Average), No. of files = 12 (Simple),No. of external interfaces = 3(Complex) and Σ Di = 45.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Sr.  No. | Project  Characteristics | Count | Type | | | Weighing  factor | | | Value |
|  | Simple Average Complex |  | S | A | C |
| 1 | Inputs | 30 | 30 | 0 | 0 | 3 | 4 | 6 | 90 |
| 2 | Outputs | 50 | 0 | 50 | 0 | 4 | 5 | 7 | 250 |
| 3 | User Inquiries | 24 | 0 | 24 | 0 | 3 | 4 | 6 | 96 |
| 4 | Files | 12 | 12 | 0 | 0 | 7 | 10 | 15 | 84 |
| 5 | External Interfaces | 3 | 0 | 0 | 3 | 5 | 7 | 10 | 30 |
|  |  |  |  | | | UFP | | | 550 |

Total Complexity Factor (TCF)= 0.65+0.01\***45= 1.1**

**Total Function Points= TCF \* UFP = 1.1\* 550 =605**

**Lab Activity:**

**(a) List out the following five project characteristics for the mini-project undertaken by you.**

**List of Inputs:**

|  |  |  |  |
| --- | --- | --- | --- |
| Sr. no | Input Name | Nature of Input | Justification (if required) |
| 1 | User Profiles | Simple | Information about job seekers |
| 2 | Education Level | Simple | Highest level of education attains by user |
| 3 | Skills | Simple | Previous work experience and job history |
| 4 | Location | Average | Preferred work location |

**List of Outputs:**

|  |  |  |  |
| --- | --- | --- | --- |
| Sr. no | Output Name | Nature of Input | Justification (if required) |
| 1 | Application Status | Average | Status of job application submitted by user |
| 2 | Employer Contact Info | Simple | Information to connect with potential employers |
| 3 | Job Recommendation | Complex | Suggested job based on user preferences |
| 4 | Interview Invitation | Average | Invitations from employers for job interviews |

**List of Inquires:**

|  |  |  |  |
| --- | --- | --- | --- |
| Sr. no | Inquiry Name | Nature of Input | Justification (if required) |
| 1 | Job Search | Simple | Search for specific job positions |
| 2 | Employer Details | Simple | Request details about potential employers |
| 3 | Employer Requirements | Average | Requirements that employer need for the work |
| 4 | Interview Schedule | Average | To schedule the interview for best employee to select |

**List of Internal Logical Files/Tables:**

|  |  |  |  |
| --- | --- | --- | --- |
| Sr. no | Files/Table Name | Nature of Input | Justification (if required) |
| 1 | User Profiles | Simple | Information storage for user profile |
| 2 | Application Forms | Average | Forms submitted by users when applying for the job |
| 3 | Employer Profile | Simple | Information repository for employer profiles |
| 4 | User Skills | Simple | Database storing specific skills of job seekers |

**List of Interfaces:**

|  |  |  |  |
| --- | --- | --- | --- |
| Sr. no | Interface Name | Nature of Input | Justification (if required) |
| 1 | User Registration | Simple | Interface for users to create a user account and profiles |
| 2 | Job Posting | Complex | Interface for employers to post job vacancies |
| 3 | User Login | Simple | Interface to login a employer and job seekers |
| 4 | Profile Update | Average | Interface for an update an profile of job seekers and employer |

**(b)Calculate unadjusted function points based on above information.**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Sr.  No. | Project  Characteristics | Count | Type | | | Weighing  factor | | | Value |
| Simple | Average Complex | Complex | S | A | C |
| 1 | Inputs | 4 | 3 | 1 | 0 | 3 | 4 | 6 | 13 |
| 2 | Outputs | 4 | 1 | 2 | 1 | 4 | 5 | 7 | 21 |
| 3 | User Inquiries | 4 | 2 | 2 | 0 | 3 | 4 | 6 | 14 |
| 4 | Files | 4 | 3 | 1 | 0 | 7 | 10 | 15 | 31 |
| 5 | External Interfaces | 4 | 2 | 1 | 1 | 5 | 7 | 10 | 27 |
|  |  |  |  | | | UFP | | | 106 |

**(c) Assume suitable values (0 to 5) for each of following 14 complexity factors and find DI. Calculate TCF using formula 0.65+0.01\*DI and Total Function Points (FP).**

|  |  |  |
| --- | --- | --- |
| **Sr.**  **No.** | **Complexity factor** | **Rate**  **(0-5)** |
| 1 | Does the system require reliable backup and recovery? | 4 |
| 2 | Are specialized data communications required to transfer information to or from the application? | 3 |
| 3 | Are there distributed processing functions? | 2 |
| 4 | Is performance critical? | 2 |
| 5 | Will the system run in an existing, heavily utilized operational environment? | 2 |
| 6 | Does the system require online data entry? | 5 |
| 7 | Does the online data entry require the input transaction to be built over multiple screens or operations? | 3 |
| 8 | Are the ILFs updated online? | 4 |
| 9 | Are the inputs, outputs, files, or inquiries complex? | 1 |
| 10 | Is the internal processing complex? | 1 |
| 11 | Is the code designed to be reusable? | 3 |
| 12 | Are conversion and installation included in the design? | 2 |
| 13 | Is the system designed for multiple installations in different organizations? | 2 |
| 14 | Is the application designed to facilitate change and ease of use by the user? | 4 |
|  | Total DI | 38 |

Total Complexity Factor (TCF)= 0.65+0.01\***38= 1.03**

**Total Function Points= TCF \* UFP = 1.03\* 106 =109.18**

**(d) Assume suitable value for one function point and find size of project size ( in terms KLOC).**

**Project Size = Total Function Points \* 1000 = 109.18 \* 1000 = 109180 lines of code (KLOC)**