**Experiment No.08**

PART A

(PART A: TO BE REFFERED BY STUDENTS)

**Experiment 8**

**A.1 Aim: Cost Estimation of the proposed system, Schedule & Milestone using Function point estimation**

**A.2** **Prerequisite:**

Feasibility study of the system

**A.3** **Outcome:**

Cost Estimation, Time estimation

**A.4** **Theory:**

Project Estimation

* Project scope must be understood
* Elaboration (decomposition) is necessary
* Historical metrics are very helpful
* At least two different techniques should be used
* Uncertainty is inherent in the process

**Function Point(FP) estimation:**

The function point metric (FP), first proposed by Albrecht , can be used effectively as a means for measuring the functionality delivered by a system.

* Function points are derived using an empirical relationship based on countable (direct) measures of software's information domain and assessments of software complexity

Information domain values are defined in the following manner:

* number of external inputs (EIs)
* number of external outputs (EOs)
* number of external inquiries (EQs)
* number of internal logical files (ILFs)
* Number of external interface files (EIFs)

**Information Weighting Factor**

**Domain Value Count Simple Average Complex**

External Inputs \_\_\_\_\_ x 3 4 6 = \_\_\_\_\_

External Outputs \_\_\_\_\_ x 4 5 7 = \_\_\_\_\_

External Inquiries \_\_\_\_\_ x 3 4 6 = \_\_\_\_\_

Internal Logical Files \_\_\_\_\_ x 7 10 15= \_\_\_\_\_

External Interface Files \_\_\_\_\_ x 5 7 10 = \_\_\_\_\_  
**Count total** \_\_\_\_\_\_\_\_

**FP=count total\*[0.65+(0.01\*sigma(Fi)]**

External Inputs (EI) - is an elementary process in which data crosses the boundary from outside to inside. This data may come from a data input screen or another application. The data may be used to maintain one or more internal logical files. The data can be either control information or business information. If the data is control information it does not have to update an internal logical file.

External Outputs (EO) - an elementary process in which derived data passes across the boundary from inside to outside. Additionally, an EO may update an ILF. The data creates reports or output files sent to other applications. These reports and files are created from one or more internal logical files and external interface file.  The following graphic represents on EO with 2 FTR's there is derived information (green) that has been derived from the ILF's

External Inquiry (EQ) - an elementary process with both input and output components that result in data retrieval from one or more internal logical files and external interface files. The input process does not update any Internal Logical Files, and the output side does not contain derived data. The graphic below represents an EQ with two ILF's and no derived data.

Internal Logical Files (ILF’s) - a user identifiable group of logically related data that resides entirely within the applications boundary and is maintained through external inputs.

External Interface Files (EIF’s) - a user identifiable group of logically related data that is used for reference purposes only. The data resides entirely outside the application and is maintained by another application. The external interface file is an internal logical file for another application.

**Value adjustment factors:**

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 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 internal logical 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 for ease of use by the user?

**Solved example:**

The estimated number of FP is derived:

FPestimated = count-total 3 [0.65 + 0.01 3 S (Fi)]

FPestimated = 375

organizational average productivity = 6.5 FP/pm.

burdened labor rate = $8000 per month, the cost per FP is approximately $1230.

Based on the FP estimate and the historical productivity data, the total estimated project cost is $461,000 and the estimated effort is 58 person-months.

Task to be completed:

1. Complete the Function point estimation for your project

**PART B**

(PART B: TO BE COMPLETED BY STUDENTS)

**(Students must submit the soft copy as per following segments within two hours of the practical. The soft copy must be uploaded on the Blackboard or emailed to the concerned lab in charge faculties at the end of the practical in case the there is no Black board access available)**

|  |  |
| --- | --- |
| Roll No. B228 | Name: Pranav Kolhe |
| Program: CS | Division: A |
| Batch: A | Date of Experiment: |
| Date of Submission: | Grade: |

**B.1 Tasks given in PART A to be completed here**

*(****Students must write the answers of the task(s) given in the PART A)***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **COMPONENTS** | **WEIGHING FACTOR** | | | **TOTAL COUNT** |
| **Low** | **Average** | **High** |
| **EI** | **3\*6=18** | **0** | **0** | **18** |
| **EO** | **4\*2=8** | **0** | **0** | **8** |
| **EQ** | **0** | **0** | **0** | **0** |
| **ILF** | **0** | **7\*1=7** | **0** | **7** |
| **EIF** | **0** | **10\*1=10** | **0** | **10** |
|  |  |  | **Total FPs** | **43** |

**Thus, Functional Points are:**

**FP = Total I/P + Total O/P + Total ILF + Total EIF + Total Enquiries**

**18 + 8 + 0 + 7 + 10**

**= 43**

**Total Unadjusted Functional Points = 282**

**CAF = 0.65 + (0.01 \*Σfi)**

**= 1.02**

**Functional Points(FP) = FP \* CAF**

**= 287.64**

**Assuming that the cost is Rs.5000/pm, i.e., 5000 per month.**

**Cost = cost per person month \* effort**

**= Rs. (5000\*3) = Rs.15,000.**

**B.2 Observations and Learning:**

I have successfully learnt to identify project functional points and the cost.

**B.3 Conclusion:**

Experiment was completed successfully.

**B.4 Question of curiosity:**

1. What is Function Point Analysis? What is A Function Point?

Ans1. FPA is used to make estimate of the software project, including its testing in terms of functionality or function size of the software product. However, functional point analysis may be used for the test estimation of the product. The functional size of the product is measured in terms of the function point, which is a standard of measurement to measure the software application. A function point (FP) is a component of software development which helps to approximate the cost of development early in the process. It is a process which defines the required functions and their complexity in a piece of software in order to estimate the software's size and scope upon completion.

2. Who created Function Points Analysis? Why it was created?

Ans2. Allan J. Albrecht initially developed function Point Analysis in 1979 at IBM and it has been further modified by the International Function Point Users Group (IFPUG). The basic and primary purpose of the functional point analysis is to measure and provide the software application functional size to the client, customer, and the stakeholder on their request. Further, it is used to measure the software project development along with its maintenance, consistently throughout the project irrespective of the tools and the technologies.

3. Is the Function Point Analysis technique owned by some company?

Ans3. IFPUG owns Function Point Analysis (FPA) as defined in ISO standard 20296:2009 which specifies the definitions, rules and steps for applying the IFPUG's functional size measurement (FSM) method.

4. What are Function Point Analysis benefits?

Ans4. Benefits of FPA:

FPA is a tool to determine the size of a purchased application package by counting all the functions included in the package.

It is a tool to help users discover the benefit of an application package to their organization by counting functions that specifically match their requirements.

It is a tool to measure the units of a software product to support quality and productivity analysis.

It s a vehicle to estimate the cost and resources required for software development and maintenance.

It is a normalization factor for software comparison.

5. Is it necessary to be a software developer to do Function Point Analysis?

Ans5. No, I don’t think so that it is necessary to be a software developer to do Function Point Analysis.

6. Who uses Function Point Analysis in the world?

Ans6. Several standards exist in the industry, but the International Function Point Users Group (IFPUG) is the most widely used. IFPUG produces the Function Point Counting Practices Manual, used by Certified Function Point Specialists (CFPS) to conduct function point counts. IFPUG is one of the ISO standards for software sizing (ISO/IEC 28926:2009).

7. What tools are suitable for support and/or to automate the use of FPA?

Ans7. The first point to note in this issue is that there are no tools available that automatically count function points reliably. However there are tools available that can support and partially automate the process of function point counts and also to store and manage the results of the counts.The simplest tool to be used to record a function point count is a spread sheet. In the “resources” section of our website, there is a free and formatted spreadsheet for function point counts available for download. Despite being the first and simplest tool to be used by many professionals, its use begins to be impractical as the number of counts increases. The control of the count repository is usually manual, and with the increasing amount of data, the task becomes costly.When the organization realizes that the spreadsheet no longer meets it needs, a natural course of action is to search tools with more capabilities on the market. The IFPUG has a certification process for the tools to support the function point counts. The list of tools currently certified can be viewed here: http://www.ifpug.org/?page\_id=316. According to this process, the tools can be classified into three categories:

Type 1: The user does the function points count manually and the software provides functionalities for data collection and calculations.

Type 2: The software provides the functionalities for data collection and calculations, and the user and the system do the interactive function points count, using questions submitted by the system and actions being taken automatically depending on the answers provided.

Type 3: The software automatically produces a function point count using various sources of information such as the database application, the application itself and artifacts of the development tools. The user can enter the data interactively, but his involvement is minimal during the count. It is important to note that there are no such tools certified.

Although there are several options of tools on the market to support the use of function points, many organizations choose to develop an inhouse tool integrated with its systems of internal control. Some reasons for this may be:

The cost to develop an internal solution is less than the cost of acquisition and maintenance of packages available on the marke.

Lack of local support for the solution, due to the fact that most tools on the market are foreign

Needs to integrate with internal systems

8. Why automatic tools cannot correctly count function points?

Ans8.

There are several software products that from a program model or its source code, cal-culate its size in function points. However, comparisons between the results produced by different tools for the same system, frequently have an unacceptable variation. These numbers, also often differ greatly from a manual count.The answer to this variation is in how these tools calculate the number of function points. Some are based on files, screens, reports and other elements to derive a num-ber. Although there is often a direct relationship between these objects and data functi-ons and transactions functions of Function Point Analysis (FPA), it must be remembe-red that the technique measures only the logical functions of the system. And these tools have difficulties in differentiating logic functions from physical functions. For example, not every file or table from a program file corresponds to an internal logical file or external interface file. Or even an elementary process can be implemented through multiple screens. To do the measurement in a correct way, the software should have enough intelligence to make this judgment. That is, this software would have to have the skill to read the program and interpret the user´s requirements. However, there is no software with this artificial intelligence.Other tools are based on the backfiring technique, which is to derive the number of function points from the program number of lines of code, based on a previous relati-onship established between LOC and FP. However, this is a technique that has been widely criticized, and whose application is restricted.There are software products to support the process of counting function points that automate a part of the process, but the decision and analysis of that should be consi-dered, remains as the responsibility of the human user who enters the data, and not of the software.

9. What kind of software can be measured by Function Points?

Ans9.FPA is a technique to measure the functionalities given by a software to the users; and this measurement is always made on an external perspective, the users’ perspective. However, it is important to say that the concept of user for FPA is not only the one of the enduser of the software. The user for the FPA is any person or thing that interacts with the software at any time. In other words, the user for FPA can be both the person acting as enduser to the software and another software that uses the services of the software in analysis.Considering that every and any software exists to offer one or more services (functions) to someone (person or thing); it is concluded that every and any software can be measured by Function Points.A common mistake for beginners with FPA is to only consider the endusers´point of view. In this case some types of software will be partially (or completely) “invisible” to this user. Then they mistakenly conclude that FPA does not work for that kind of software. The most common is for the person to learn the principles of the FPA applied to systems with screens and reports. However, when this person faces some software domain which do not have screens, like batch processing, middlewares, basic softwares, it is natural to have some difficulties on measuring it.Let’s imagine that the goal was to measure a printer’s driver. Well, there is no enduser (person) for this kind of software. In this perspective, the printer’s driver is invisible to the enduser. However it exists to offer services to someone; in this case, the operating system. Thus, analyzing the printer’s driver in the perspective of the operating system, it is possible to see functions, for example: to start the the printer, inform the general situation of the device, eject a sheet of paper, print, alert the level of the ink, etc...

10. Is it possible to use FPA in a project using agile methodology?

Ans10.Yes, but not all of the software maintenances are likely to be measured with Function Point Analysis (FPA). Only the maintenances that change the software functional requirements can be measured by the Function Point Analysis (FPA), in this case IFPUG uses the term “improvement” instead of “maintenance”, exactly to make the point that the improvement is not any kind of maintenance. In IFPUG’s concept, the improvement measures all the functions that will be added, changed or excluded from the application, as well as eventual functions of data conversion.Maintenance for correction of defects or to keep only nonfunctional requirements are not measured by Function Point Analysis (FPA).

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*