Available Online at http://www.recentscientific.com

ISSN: 0976-3031

International Journal of Recent Scientific Research
Vol. 6, Issue, 6, pp.4796-4799, June, 2015

International Journal of Recent Scientific Research

RESEARCH ARTICLE

FUNCTION POINT ANALYSIS AN APPROACH TO PROJECT SCHEDULING

Pooja Narayan Patil

Lecturer, Computer Engineering Department, BGIT, Mumbai

ARTICLE INFO

Article History:

Received 2nd, May, 2015
Received in revised form 10th, May, 2015
Accepted 4th, June, 2015
Published online 28th, June, 2015

Key words:

Project Scheduling, Principles, Unadjusted Function Point, Function Point, Complexity Adjustment Factor.

ABSTRACT

An application is developed as an essential requirement for the need of any elementary process. Any application that needs to be developed is required to undergo the process of project scheduling to avoid delay in the delivery which costs huge for the company. An elementary process is not totally independent existing alone, but the elementary processes are together becoming interdependent. There are two basic types of elementary processes (data in motion and data at rest) in a software application. Data in motion has the characteristic of moving data inside to outside the application boundary or outside to inside the application boundary. An elementary process is similar to an acceptance test case. Function point analysis helps to calculate the project scheduling depending on the factors of the input, output, enquires, files and the external files it is dependent on. This paper tries to cover the concept of function point analysis in project scheduling.

INTRODUCTION

As changing customer requirements that are not reflected in schedule changes. To determine and estimate the amount of time and resources required for the development of the project. An honest underestimate of the amount of effort and/or the number of resources that will be required to do the job. Usually while doing the scheduling the project predictable and/or unpredictable risks that were not considered when the project commenced.Technical difficulties that could not have been foreseen in advance.

Factors That Delay Project Schedule

Although there are many reasons why software is delivered late, most can be traced to one or more of the following root causes:

1. An unrealistic deadline established by someone outside the software development group and forced on managers and practitioners within the group.
2. Changing customer requirements that are not reflected in schedule changes.
3. An honest underestimate of the amount of effort and/or the number of resources that will be required to do the job.
4. Predictable and/or unpredictable risks that were not considered when the project commenced.
5. Technical difficulties that could not have been foreseen in advance.
6. Human difficulties that could not have been foreseen in advance.
7. Miscommunication among project staff that results in delays.
8. A failure by project management to recognize that the project is falling behind schedule and a lack of action to correct the problem.

Principles Of Project Scheduling

1. ***Compartmentalization:*** The project must be compartmentalized into a number of manageable activities and tasks.
2. ***Interdependency:*** The interdependency of each compartmentalized activity or task must be determined.
3. ***Time Allocation:*** Each task to be scheduled must be allocated some number of work units (e.g., person-days of effort).
4. ***Effort Validation:*** the project manager must ensure that no more than the allocated number of people have been scheduled at any given time.
5. ***Defined Responsibilities:*** Every task that is scheduled should be assigned to a specific team member
6. ***Defined Outcomes:*** Every task that is scheduled should have a defined outcome.
7. ***Defined Milestones:*** Every task or group of tasks should be associated with a project milestone.

Corresponding author:* **Pooja Narayan Patil
Lecturer, Computer Engineering Department, BGIT, Mumbai

A milestone is accomplished when one or more work products has been reviewed for quality and has been approved.

Function Point Analysis

The FPA is a reliable method for measuring the size of computer software. It essentially measures functionality that the user requests and receives. Function point measures the functionality from the user point of view, that is, on the basis of what the user request and receives in return. Therefore, it deals with the functionality being delivered, and not with the lines of code, source modules, files etc. measuring size in this way has the advantage that size measure is independent of the technology used to deliver the function. It also measures the software development and maintenance cost and size independently of the technology used for implementation. The general approach that FP follows is

1. Count the number of inputs, outputs, inquiries, master files, and interfaces required, then calculate the Unadjusted Function Points (UFP)
2. Calculate the adjusted function point (AFP) by multiplying these counts by an adjustment factor; the UFP and the product complexity adjustment.
3. Calculate the Source Lines of Code (SLOC) with the help of the AFP and the Language Factor (LF)
4. Adjusted Function Point (AFP) = Unadjusted Function Point (UFP) * CAF (Complete Adjustment factor).

Advantages

1. It is not restricted to code
2. Language independent
3. The necessary data is available early in a project. We need only a detailed specification.
4. More accurate than estimated LOC.

Drawbacks

1. Subjective counting
2. Hard to automate and difficult to compute
3. Ignores quality of output
4. Oriented to traditional data processing applications

Abbreviations Used

AFP	Adjusted Function Point
UFP	Unadjusted Function Point
GSC	General System Characteristics
FTR	File Types Referenced
FP	Function Point
ILF	Internal Logical File.
EIF	External Interface file
EI	External Inputs
EO	External Outputs
EQ	External Enquiries
RET	Record Element Type
DET	Data Element Type
FTR	File Type Reference
GSC	General System Characteristic
VAF	Value Adjustment Factor
LOC	Line of code
EAF	Effort Adjustment Factor
SLOC	Source Lines of Code
CPLX	Development/Technical Complexity Factor
TOOL	Development/Technical Tool Complexity Factor
TDEV	Development Time

Calculation Of The Unadjusted Function Points (Ufp)

1. The FPA measures functionality that the user requires.
2. The specific user functionality is a measurement of the functionality delivered by the application as for user request.
3. The 5 function types identified are

- ***External Input*** which receives information from outside the application boundary,
- ***External Output*** which presents information of the information system,
- ***External Inquiries*** which is special kind of an external output. An external inquiry presents information of the information system based on a uniquely identifying search criterion, without applying additional processing (such as calculations).
- ***Internal Logical Files*** contains permanent data that is relevant to the user. The information system references and maintains the data.
- ***External Interface Files*** also contains permanent data that is relevant to the user. The information system references the data, but the data is maintained by another information system.

Table 1 Function type for the various Inputs

Function type	Simple	Average	Complex
External Input	3	4	6
External Output	4	5	7
External Inquiry	3	4	6
Internal Logical File	7	10	15
External Interface File	5	7	10

4. For each function identified above the function is further classified as simple, average or complex and a weight is given to each.
5. The sum of the weights quantifies the size of information processing and is referred to as the Unadjusted Function points.

Table 2 Adjusted Function Point

Number	Complexity Weighting Factor	Value
1	Backup and recovery	1
2	Data communications	2
3	Distributed processing	2
4	Performance critical	5
5	Existing operating environment	3
6	On-line data entry	3
7	Input transaction over multiple screens	1
8	Master files updated online	3
9	Information domain values complex	5
10	Internal processing complex	4
11	Code designed for reuse	5
12	Conversion/installation in design	4
13	Multiple installations	4
14	Application designed for change	4
	Total complexity adjustment value	46

Calculate Adjusted Function Point

1. To calculate the Complexity adjustment value, several factors have to be considered, such as Backup and recovery, code design for reuse, etc.
2. All the factors and their estimated values in this project are shown in the following table.
3. The adjusted function point denoted by FP is given by the formula:
4. FP = total UFP * (0.65 + (0.01 * Total complexity adjustment value)) or
5. FP = total UFP * (Complexity adjustment factor)
6. Total complexity adjustment value is counted based on responses to questions called complexity weighting factors in the table below.
7. Each complexity weighting factor is assigned a value (complexity adjustment value) that ranges between 0 (not important) to 5 (absolutely essential).

Benefits And Uses

A function point count has many uses

1. Function Points can be used to communicate more effectively with business user groups.
2. Function Points can be used to reduce overtime.
3. Function points can be used to establish an inventory of all transactions and files of a current project or application.
4. Function Points can be used to size software applications. Sizing is an important component in determining productivity (outputs/inputs), predicting effort, and understanding unit cost.
5. Unlike some other software metrics, different people can count function points at different times, to obtain the same measure within a reasonable margin of error. That is, the same conclusion will be drawn from the results.
6. FPA can help organizations understand the unit cost of a software application or project.

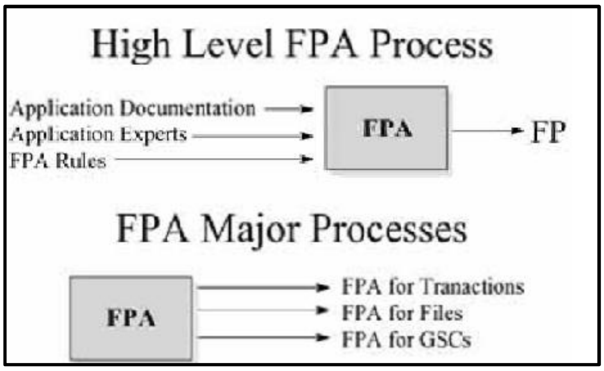

Figure 1 FPA Process

Types Of Function Point Counts

Function point counts can be associated with either projects or applications. There are three types of function point counts.

1. Development project function point count
2. Enhancement project function point count
3. Application function point count.

Fpa Steps For Transactional Function Types

Each transaction must be an elementary process. An elementary process is the smallest unit of activity that is meaningful to the end user in the business.T1. Application documentation and transaction rules are used to identify transactions. T2. The application documentation and transaction rules are used to determine type of transaction (external input, external output, or external inquiry). T3. With the help of application documentation (data model and transaction model) and transaction rules the number data elements and file type referenced are determined. T4. Each identified transaction is assigned a value of low, average or high based upon type, data elements, and files referenced. T5. A distinct numerical value is assigned based upon type and value (low, average, or high). T6. All transactions are summed to create a transaction unadjusted function point count.

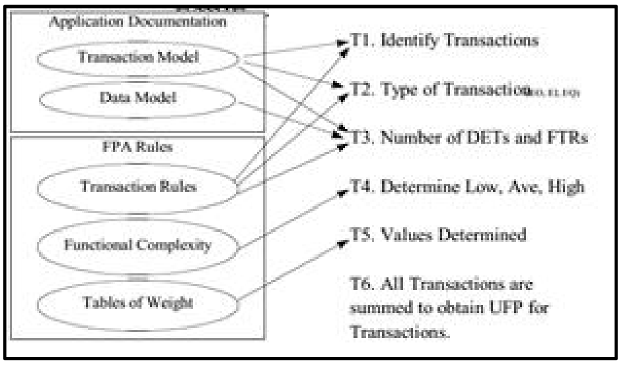
Application Documentation
Transaction Model
Data Model
FPA Rules
Transaction Rules
Functional Complexity
Tables of Weight
T1. Identify Transactions
T2. Type of Transaction(EO, EI, EQ)
T3. Number of DETs and FTRs
T4. Determine Low, Ave, High
T5. Values Determined
T6. All Transactions are summed to obtain UFP for Transactions.

Figure 2 Transactional Function Point

Fpa Steps For Files

F1. Application documentation and file rules are used to identify files. F2. The application documentation (transaction model and data model) is used to determine type of file (either external interface file or internal logical file). F3. With the help of application documentation (data model) and file rules the number data elements and record element types are determined. F4. Each identified file is assigned a value of low, average or high based upon type, data elements and record types. F5. A distinct numerical value is assigned based upon type and value (low, average, or high). F6. All files are summed to create a file unadjusted function point count.

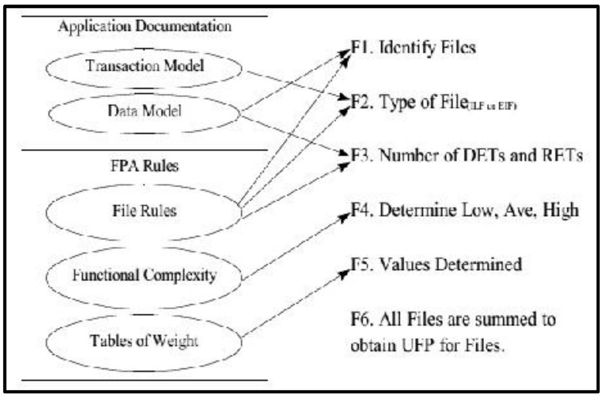
Application Documentation
Transaction Model
Data Model
FPA Rules
File Rules
Functional Complexity
Tables of Weight
F1. Identify Files
F2. Type of File(ILF or EIF)
F3. Number of DETs and RETs
F4. Determine Low, Ave, High
F5. Values Determined
F6. All Files are summed to obtain UFP for Files.

Figure 3 FPA Steps for Files

Case Study: Study of scope of Library Automation project has produced following results: need 7 inputs, 10 outputs, 6 inquiries, 17 files and 4 external interfaces. Input and external interface function point attributes are of average complexity and all other function points attributes are of low complexity. Determine adjusted function points assuming complexity adjustment factor is 32.

Solution: As given 7 Inputs, 10 outputs, 6 inquiries, 17 files and 4 external interfaces

Function type	Simple	Average	Complex
Internal Logical File	7	10	15
External Interface File	5	7	10
External Input	3	4	6
External Output	4	5	7
External Inquiry	3	4	6

Count the number of inputs, outputs, inquiries, master files, and interfaces required, then calculate the Unadjusted Function Points (UFP)

Calculate the Adjusted Function Point (AFP) by multiplying these counts by an adjustment factor; the UFP and the product complexity adjustment

UFP = 7*4 + 10 * 4+6 * 3 + 17 * 7 + 4 * 7
= 28+40+18+119+28
= 233
AFP = UFP * CAF (Complete Adjustment factor)
CAF given 32
Hence
FP = 233 * (0.65+0.01*32)
= 233 * (0.65+0.32)
=233 * 0.97
= 226.01
= 227

References

1. www.SoftwareMetrics.Com
2. www.anuradhabhatia.com
3. www.functionpoints.org
4. www.functionpointmodeler.com

How to cite this article:
Pooja Narayan Patil *et al*, Rao Sethumadhavan., Function Point Analysis an Approach to Project Scheduling. *International Journal of Recent Scientific Research Vol. 6, Issue, 6, pp.4796-4799, June, 2015*

****