AIM: ESTIMATION USING FUNCTION POINT

THEORY:

Introduction To Function Point Analysis

Software systems, unless they are thoroughly understood, can be like an ice berg. They are becoming more and more difficult to understand. Improvement of coding tools allows software developers to produce large amounts of software to meet an ever expanding need from users. As systems grow a method to understand and communicate size needs to be used. Function Point Analysis is a structured technique of problem solving. It is a method to break systems into smaller components, so they can be better understood and analyzed.

Function points are a unit measure for software much like an hour is to measuring time, miles are to measuring distance or Celsius is to measuring temperature. Function Points are an ordinal measure much like other measures such as kilometers, Fahrenheit, hours, so on and so forth

Objectives of Function Point Analysis

Frequently the term end user or user is used without specifying what is meant. In this case, the user is a sophisticated user. Someone that would understand the system from a functional perspective --- more than likely someone that would provide requirements or does acceptance testing.

Since Function Points measures systems from a functional perspective they are independent of technology. Regardless of language, development method, or hardware platform used, the number of function points for a system will remain constant. The only variable is the amount of effort needed to deliver a given set of function points; therefore, Function Point Analysis can be used to determine whether a tool, an environment, a language is more productive compared with others within an organization or among organizations. This is a critical point and one of the greatest values of Function Point Analysis.

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. If the project has grown, there has been scope creep. 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.

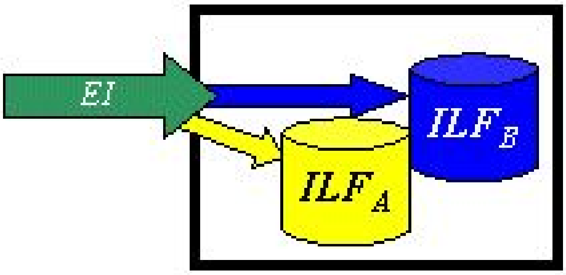
Characteristic of Quality Function Point Analysis

Function Point Analysis should be performed by trained and experienced personnel. If Function Point Analysis is conducted by untrained personnel, it is reasonable to assume the analysis will done incorrectly. The personnel counting function points should utilize the most current version of the Function Point Counting Practices Manual,

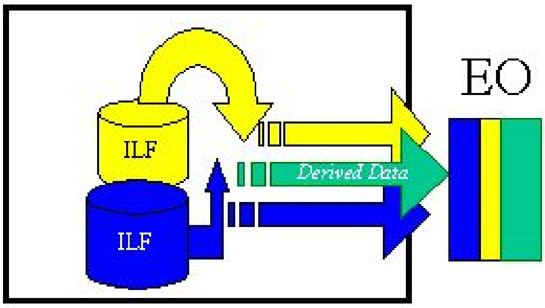
Current application documentation should be utilized to complete a function point count. For example, screen formats, report layouts, listing of interfaces with other systems and between systems, logical and/or preliminary physical data models will all assist in Function Points Analysis. The task of counting function points should be included as part of the overall project plan. That is, counting function points should be scheduled and planned. The first function point count should be developed to provide sizing used for estimating.

The Five Major Components

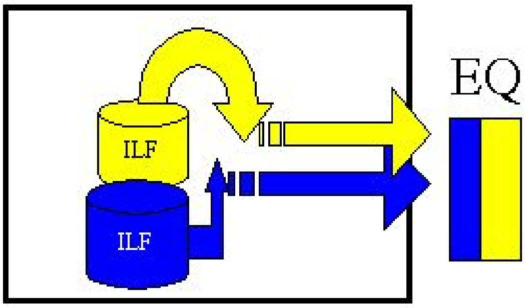
Since it is common for computer systems to interact with other computer systems, a boundary must be drawn around each system to be measured prior to classifying components. This boundary must be drawn according to the user's point of view. In short, the boundary indicates the border between the project or application being measured and the external applications or user domain. Once the border has been established, components can be classified, ranked and tallied.



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. The graphic represents a simple EI that updates 2 ILF's (FTR's).



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.

FUNCTION POINT COMPUTATION

	Weighting Factor				
Information Domain Value	**Count**	**DETs**	**FTRs/ RETs**	**Complexity Rating**	**Total**
External Inputs	1	2	1	Low(3)	3
External Outputs	1	6	2	Average(4)	4
External Inquiries	1	1	1	Low(3)	3
Internal Logical files	1	2	2	Low(3)	3
Count Total					13

Total number of Unadjusted Function Points=13

Value Adjustment Factors

Does the system require reliable backup and recovery?	0
Are specialized data communications required to transfer information to or from the application?	5
Are there distributed processing functions?	0
Is performance critical?	3
Will the system run in an existing, heavily utilized operational environment?	3
Does the system require on-line data entry?	1
Does the on-line data entry require the input transaction to be built over multiple screens or operations?	2
Are the internal logical files updated on-line?	2
Are the inputs, outputs, files, or inquiries complex?	3
Is the internal processing complex?	3
Is the code designed to be reusable?	2
Are conversion and installation included in the design?	3
Is the system designed for multiple installations in different organizations?	3
Is the application designed to facilitate change and for ease of use by the user?	4

Sum(F1) = 34

Number of Function points (FP)

= count total * [0.65 + 0.01 * sum(Fi)]

=13*[0.65+0.01*34]

=**12.87 (rounded up to 13)**

1 person-day = 1 FP

Therefore total person-days=13

CONCLUSION:

Hence, we have performed estimation using function point analysis.