Q/P MANAGEMENT GROUP, INC.

A Publication for Information Technology Professionals

An Introduction to Function Point Analysis

Roger Heller, Vice President

The purpose of this article is to provide an introduction to Function Point Analysis and its application in non-traditional computing situations. Software engineers have been searching for a metric that is applicable for a broad range of software environments. The metric should be technology independent and support the need for estimating, project management, measuring quality and gathering requirements. Function Point Analysis is rapidly becoming the measure of choice for these tasks.

Function Point Analysis has been proven as a reliable method for measuring the size of computer software. In addition to measuring output, Function Point Analysis is extremely useful in estimating projects, managing change of scope, measuring productivity, and communicating functional requirements.

There have been many misconceptions regarding the appropriateness of Function Point Analysis in evaluating emerging environments such as real time embedded code and Object Oriented programming. Since function points express the resulting work-product in terms of functionality as seen from the user's perspective, the tools and technologies used to deliver it are independent.

The following provides an introduction to Function Point Analysis and is followed by further discussion of potential benefits.

Introduction to Function Point Analysis

One of the initial design criteria for function points was to provide a mechanism that both software developers and users could utilize to define functional requirements. It was determined that the best way to gain an understanding of the users' needs was to approach their problem from the perspective of how they view the results an automated system produces. Therefore, one of the primary goals of Function Point Analysis is to evaluate a system's capabilities from a user's point of view. To achieve this goal, the analysis is based upon the various ways users interact with computerized systems. From a user's perspective a system assists them in doing their job by providing five (5) basic functions. Two of these address the data requirements of an end user and are referred to as Data Functions; the remaining three address the user's need to access data and are referred to as Transactional Functions.

The Five Components of Function Points

Data Functions
- Internal Logical Files
- External Interface Files

Transactional Functions
- External Inputs
- External Outputs
- External Inquiries

Internal Logical Files - The first data function allows users to utilize data they are responsible for maintaining. For example, a pilot may enter navigational data through a display in the cockpit prior to departure. The data is stored in a file for use and can be modified during the mission. Therefore the pilot is responsible for maintaining the file that contains the navigational information. Logical groupings of data in a system, maintained by an end user, are referred to as Internal Logical Files (ILF).

External Interface Files - The second Data Function a system provides an end user is also related to logical groupings of data. In this case the user is not responsible for maintaining the data. The data resides in another system and is maintained by another user or system. The user of the system being counted requires this data for reference purposes only. For example, it may be necessary for a pilot to reference position data from a satellite or ground-based facility during flight. The pilot does not have the responsibility for updating data at these sites but must reference it during the flight. Groupings of data from another system that are used only for reference purposes are defined as External Interface Files (EIF).

The remaining functions address the user's capability to access the data contained in ILFs and EIFs. This capability includes maintaining, inquiring and outputting of data. These are referred to as Transactional Functions.

External Input - The first Transactional Function allows a user to maintain Internal Logical Files (ILFs) through the ability to add, change and delete the data. For example, a pilot can add, change and delete navigational information prior to and during the mission. In this case the pilot is utilizing a transaction referred to as an External Input (EI). An External Input gives the user the capability to maintain the data in ILF's through adding, changing and deleting its contents.

External Output - The next Transactional Function gives the user the ability to produce outputs. For example a pilot has the ability to separately display ground speed, true air speed and calibrated air speed. The results displayed are derived using data that is maintained and data that is referenced. In function point terminology the resulting display is called an External Output (EO).

External Inquiries - The final capability provided to users through a computerized system addresses the requirement to select and display specific data from files. To accomplish this a user inputs selection information that is used to retrieve data that meets the specific criteria. In this situation there is no manipulation of the data. It is a direct retrieval of information contained on the files. For example if a pilot displays terrain clearance data that was previously set, the resulting output is the direct retrieval of stored information. These transactions are referred to as External Inquiries (EQ).

In addition to the five functional components described above there are two adjustment factors that need to be considered in Function Point Analysis.

Functional Complexity - The first adjustment factor considers the Functional Complexity for each unique function. Functional Complexity is determined based on the combination of data groupings and data elements of a particular function. The number of data elements and unique groupings are counted and compared to a complexity matrix that will rate the function as low, average or high complexity. Each of the five functional components (ILF, EIF, EI, EO and EQ) has its own unique complexity matrix.

The following is the complexity matrix for External Outputs.

	1-5 DETs	6 - 19 DETs	20+ DETs
0 or 1 FTRs	L	L	A
2 or 3 FTRs	L	A	H
4+ FTRs	A	H	H

Complexity	UFP
L (Low)	4
A (Average)	5
H (High)	7

Using the examples given above and their appropriate complexity matrices, the function point count for these functions would be:

Function Name	Function Type	RET	DET	FTR	UFP
Navigational data	ILF	3	36	n/a	10
Positional data	EIF	1	3	n/a	5
Navigational data - add	EI	n/a	36	1	4
Navigational data - change	EI	n/a	36	1	4
Navigational data - delete	EI	n/a	3	1	3
Ground speed display	EO	n/a	20	3	7
Air speed display	EO	n/a	20	3	7

Function Name	Function Type	RET	DET	FTR	UFP
Calibrated air speed display	EO	n/a	20	3	7
Terrain clearance display	EQ	n/a	1	1	3
Total unadjusted count					50 **UFPs**

All of the functional components are analyzed in this way and added together to derive an Unadjusted Function Point count.

Value Adjustment Factor - The Unadjusted Function Point count is multiplied by the second adjustment factor called the Value Adjustment Factor. This factor considers the system's technical and operational characteristics and is calculated by answering 14 questions. The factors are:

1. Data Communications
The data and control information used in the application are sent or received over communication facilities.

2. Distributed Data Processing
Distributed data or processing functions are a characteristic of the application within the application boundary.

3. Performance
Application performance objectives, stated or approved by the user, in either response or throughput, influence (or will influence) the design, development, installation and support of the application.

4. Heavily Used Configuration
A heavily used operational configuration, requiring special design considerations, is a characteristic of the application.

5. Transaction Rate
The transaction rate is high and influences the design, development, installation and support.

6. On-line Data Entry
On-line data entry and control information functions are provided in the application.

7. End -User Efficiency
The on-line functions provided emphasize a design for end-user efficiency.

8. On-line Update
The application provides on-line update for the internal logical files.

9. Complex Processing
Complex processing is a characteristic of the application.

10. Reusability
The application and the code in the application have been specifically designed, developed and supported to be usable in other applications.

11. Installation Ease
Conversion and installation ease are characteristics of the application. A conversion and installation plan and/or conversion tools were provided and tested during the system test phase.

12. Operational Ease
Operational ease is a characteristic of the application. Effective start-up, backup and recovery procedures were provided and tested during the system test phase.

13. Multiple Sites
The application has been specifically designed, developed and supported to be installed at multiple sites for multiple organizations.

14. Facilitate Change
The application has been specifically designed, developed and supported to facilitate change.

Each of these factors is scored based on their influence on the system being counted. The resulting score will increase or decrease the Unadjusted Function Point count by 35%. This calculation provides us with the Adjusted Function Point count.

An Approach to Counting Function Points

There are several approaches used to count function points. Q/P Management Group, Inc. has found that a structured workshop conducted with people who are knowledgeable of the functionality provided through the application is an efficient, accurate way of collecting the necessary data. The workshop approach allows the counter to develop a representation of the application from a functional perspective and educate the participants about function points.

Function point counting can be accomplished with minimal documentation. However, the accuracy and efficiency of the counting improves with appropriate documentation. Examples of appropriate documentation are:

- Design specifications
- Display designs
- Data requirements (Internal and External)
- Description of user interfaces

Function point counts are calculated during the workshop and documented with both a diagram that depicts the application and worksheets that contain the details of each function discussed.

Benefits of Function Point Analysis

Organizations that adopt Function Point Analysis as a software metric realize many benefits including: improved project estimating; understanding project and maintenance productivity; managing changing project requirements; and gathering user requirements. Each of these is discussed below.

Estimating software projects is as much an art as a science. While there are several environmental factors that need to be considered in estimating projects, two key data points are essential. The first is the size of the deliverable. The second addresses how much of the deliverable can be produced within a defined period of time. Size can be derived from Function Points, as described above. The second requirement for estimating is determining how long it takes to produce a function point. This delivery rate can be calculated based on past project performance or by using industry benchmarks. The delivery rate is expressed in function points per hour (FP/Hr) and can be applied to similar proposed projects to estimate effort (i.e. Project Hours = estimated project function points FP/Hr).

Productivity measurement is a natural output of Function Points Analysis. Since function points are technology independent they can be used as a vehicle to compare productivity across dissimilar tools and platforms. More importantly, they can be used to establish a productivity rate (i.e. FP/Hr) for a specific tool set and platform. Once productivity rates are established they can be used for project estimating as described above and tracked over time to determine the impact continuous process improvement initiatives have on productivity.

In addition to delivery productivity, function points can be used to evaluate the support requirements for maintaining systems. In this analysis, productivity is determined by calculating the number of function points one individual can support for a given system in a year (i.e. FP/FTE year). When compared with other systems, these rates help to identify which systems require the most support. The resulting analysis helps an organization develop a maintenance and replacement strategy for those systems that have high maintenance requirements.

Managing Change of Scope for an in-process project is another key benefit of Function Point Analysis. Once a project has been approved and the function point count has been established, it becomes a relatively easy task to identify, track and communicate new and changing requirements. As requests come in from users for new displays or capabilities, function point counts are developed and applied against the rate. This result is then used to determine the impact on budget and effort. The user and the project team can then determine the importance of the request against its impact on budget and schedule. At the conclusion of the project the final function point count can be evaluated against the initial estimate to determine the effectiveness of requirements gathering techniques. This analysis helps to identify opportunities to improve the requirements definition process.

Communicating Functional Requirements was the original objective behind the development of function points. Since it avoids technical terminology and focuses on user requirements it is an excellent vehicle to communicate with users. The techniques can be used to direct customer interviews and document the results of Joint Application Design (JAD) sessions. The resulting documentation provides a framework that describes user and technical requirements.

In conclusion, Function Point Analysis has proven to be an accurate technique for sizing, documenting and communicating a system's capabilities. It has been successfully used to evaluate the functionality of real-time and embedded code systems, such as robot based warehouses and avionics, as well as traditional data processing. As computing environments become increasingly complex, it is proving to be a valuable tool that accurately reflects the systems we deliver and maintain.

About the author

Roger Heller, Vice President Q/P Management Group, Inc., wrote this article. Mr. Heller is a recognized consultant, instructor and speaker with over thirty years of Information Systems and consulting experience. *The article was originally published in the STC Crosstalk publication.*