Function Point Counting
In a Real-Time Environment

www.davidconsultinggroup.com

by
David Garmus

This article first appeared in the January 1996 edition of CrossTalk Magazine (*http://www.stsc.hill.af.mil/crosstalk/1996/01/function.asp*).

This article addresses the IFPUG extension of Function Points as a metric into the real-time systems environment. The Function Point metric is currently being applied successfully as a measurement tool by a large number of organizations in the real-time environment.

ARTICLE

What is a Real-time System

First, I will define what I consider to be included as real-time systems; this is significantly different from the definition ten or twenty years ago. This category is evolving rather than static. Examples of real-time systems might include:

radar systems	telephone switches
signal processors	satellite communications
missile guidance systems	telephone transmission systems
weapons systems	fax/modem communication systems
navigation systems	automated process control systems
safety systems	operating systems

The principal operating characteristic of a real-time system is that the software must execute almost instantly, at the boundary of the processing limits of the Central Processing Unit. It is an on-line, continuously available, critically timed system which generates event driven outputs almost simultaneously with their corresponding inputs. These inputs could occur at either fixed or variable intervals, selectively or continuously; and they could require interruption to process input of a higher priority, often utilizing a buffer area or queue to determine processing priorities. Data is frequently, but not always, stored in memory.

Consequently, real-time systems have some characteristics which are not fully supported by Function Points. These characteristics must be evaluated separately since they ultimately will impact the software delivery or maintenance rates of our software engineers. These characteristics include:

- Algorithms (mathematical and logical)
- Memory constraints
- Timing constraints
- Interruptions
- Execution speeds

1770 E. Lancaster Ave, Suite 15
Paoli, PA 19301
+1.610.644.2856 (F) +1.866.293.0120

- Continuous availability
- Process/event driven
- Communication constraints

The Process of Counting Function Points

The counting process described in this article is a high level summary and must be supplemented with the complete rules defined in the *IFPUG Counting Practices Manual* and supported by IFPUG case studies. Additionally, David Herron and I authored the book, *Measuring the Software Process: A Practical Guide To Functional Measurements* (Prentice Hall), which more completely describes the software measurement process. Our book also includes an example of a Function Point count of a Navy Fire Control System.

Function Point counting typically occurs when a development organization wishes to size and estimate development time/effort for a subsystem or a project. The development activity could either be accomplished internally or contracted out to an external developer. Other factors, which must always be considered in building any estimate of effort, include the selection of tasks to be performed, the language(s) to be utilized, the skill and knowledge levels of the development and user personnel, the process and technology to be applied and the environment in which development will take place. Other application characteristics outside of Function Points, which must also be evaluated, such as security, communications process, network transaction rate, functional domain, performance, algorithmic complexity, memory constraints, data relationships and system interfaces were discussed above..

This article addresses the sizing of a project, which might include the development of one or more new subsystems and/or the enhancement of one or more existing subsystems. New development projects represent the functionality of the subsystem delivered with the first install; as a result, a function point count serves a double purpose as both the project count and the initial count of the subsystem/application. Enhancement projects deliver changes in functionality to an existing subsystem, such as adding new transactional functions (new inputs, outputs or inquiries); changing existing transactional functions by adding/deleting fields or by changing the processing (different edits, validations, algorithms/calculations or files read/referenced/updated); deleting existing transactional functions or logical database files; adding new logical database files; or changing existing logical database files (different fields or sizes/structure of fields). These changes in functionality might result from a new or revised user requirement, statutory/regulatory changes or new users. Enhancements are a change to the processing within a subsystem as opposed to a maintenance fix. Maintenance includes the effort to maintain existing systems, excluding changes which are defined above as enhancements. Maintenance includes the effort to fix bugs/defects, respond to user questions about existing functionality (how it works), start/restart/install existing systems, change literals on screens/reports, prepare one-time ad-hoc reports, etc.

Typically, function point analysis follows specific steps once a new development or enhancement project or a installed subsystem has been selected for counting:

- Identification of the subsystem boundaries
- Identification of the data functions (internal logical files and external interface files)

1770 E. Lancaster Ave, Suite 15
Paoli, PA 19301
+1.610.644.2856 (F) +1.866.293.0120

- Identification of transactional functions (external inputs, external outputs and external inquiries)
- Calculation of the Unadjusted Function Point Count
- Determination of the Value Adjustment Factor using General System Characteristics
- Calculation of the Adjusted Function Point Count

Establish the Subsystem Boundary

A project may include one or more applications or subsystems. Measure applications or subsystems as they will ultimately be used. As an example, Microsoft Office includes Word for Windows, PowerPoint, Excel and Access, all separate applications or subsystems.

A military weapon systems may consists of separate subsystems such as radar tracking units, missile systems, fire control systems, missile guidance systems, etc. A telecommunications system should be broken down into switches, fax, modem, e-mail, sending/receiving units, satellite communication devices and so on.

Identify the Internal Logical Files

Identify logical data stores that are maintained or held in memory within the subsystem. External data, transactions, messages or controls (which will be described later as External Inputs) will populate, revise, update, change or add to the data stores. These logical data stores will ultimately be used within the subsystem to support External Outputs and/or External Inquiries (both of which will be identified later). A data group should not be dependent upon or attributive to another data group for its existence. Data groups are classified as Internal Logical Files (ILFs). An ILF is counted once per subsystem. Examples of ILFs could include radar tracking data, terrain maps, firing solutions, connection data, fax or voice mail set-up tables, routing instructions, usage tables or logs, alarm data, calls currently being processed, a mail box, user profiles, customer data, inventory data, parameter values, processing history, etc. The key to their identification is that the data actually exists or may exist when the software is in use and that it is dynamic, not hard-coded.

Each identified ILF must be assigned a functional complexity of Low (L), Average (A) or High (H) based upon the number of Data Element Types (DETs) and Record Element Types (RETs) associated with that ILF.

- Data element types (DETs) are unique user recognizable, non-repeating fields/attributes, including foreign key attributes, maintained on the ILF.
- Record element types (RETs) are user recognizable subgroups (optional or mandatory) of data elements contained within an ILF. Subgroups are typically represented in an entity relationship diagram as entity subtypes or attributive entities, commonly called parent-child relationships.

DETs and RETs are applied to the following matrix in order to determine the functional complexity of each ILF:

1770 E. Lancaster Ave, Suite 15
Paoli, PA 19301
+1.610.644.2856 (F) +1.866.293.0120

Internal Logical Files or External Interface File

Record Element Types	Data fields		
	1-19	28-68	61+
< 2	L	L	A
2-6	L	A	H
>6	A	H	H

What does all of this mean in terms of Function Points? ILFs receive the highest weights/values when compared to other functions. The IFPUG Unadjusted Function Point Table follows:

Function Point Counting Weights

Type	Low	Avg	High	Total
EI	__ x 3 +	__ x 4 +	__ x 6 =	___
EO	__ x 4 +	__ x 5 +	__ x 7 =	___
EQ	__ x 3 +	__ x 4 +	__ x 6 =	___
ILF	__ x 7 +	__ x10 +	__ x15 =	___
EIF	__ x 5 +	__ x 7 +	__ x10 =	___

Identify the External Interface Files

Data extracted or read from another subsystem's ILFs which do not directly update an ILF within the subsystem but are read/referenced only and used in processing are classified as External Interface Files (EIFs). An EIF is counted only once per subsystem. Common examples are Help, security, parameters, system tables/directories, edit/validation tables, operating system status, etc. The common denominator here is that they truly are ILFs in another subsystem and only read/referenced by the subsystem being counted.

1770 E. Lancaster Ave, Suite 15
Paoli, PA 19301
+1.610.644.2856 (F) +1.866.293.0120

Record Element Types and Fields/Attributes read/referenced are counted in the same way as for ILFs in order to determine the complexity of each EIF. The same matrix used for sizing ILFs is used to size EIFs.

Identify the External Inputs

Data received from outside the subsystem boundary that either maintain (add, change, maintain, populate or delete data in) an ILF or provide control functions are identified as External Inputs (EIs). Examples of data EIs include radar tracking data, alarms, analog data, transactional data, incoming voice or fax messages, inventory issues or receipts, parameter selections, etc. Examples of control EIs include commands/messages from other subsystems, users, sensors, equipment, satellites, hardware, etc. which might contain warnings, processing instructions, firing commands, setup or shut-down commands, volume controls, temperature controls, etc. Data with unique processing requirements are counted as separate EIs. These inputs could arrive via data packets/files, voice, tones, wireless, teleprocessing signals, calls, control panels, screens, TTY, bar code, button controls, etc.

Each identified EI must be assigned a functional complexity a functional complexity of Low (L), Average (A) or High (H) based upon the number of Data Element Types (DETs) and File Types Referenced (FTRs) associated with the EI.

- Data element types (DETs) are usually unique user recognizable, non-repeating fields/attributes, including foreign key attributes, that enter the boundary of the subsystem/application.
- File types referenced (FTRs), or more simply files referenced, totals the number of internal logical files (ILFs) maintained, read or referenced and the external interface files read or referenced by the EI transaction.

DETs and FTRs are applied to the following matrix in order to determine the functional complexity of each EI:

External Input

	Data fields		
File Types Referenced	1-4	6-16	16+
< 2	L	L	A
2	L	A	H
>2	A	H	H

Identify the External Outputs

1770 E. Lancaster Ave, Suite 15
Paoli, PA 19301
+1.610.644.2856 (F) +1.866.293.0120

Data generated within the subsystem which exit the boundary are counted as External Outputs (EOs). Outgoing data, messages, commands, alarms, processing instructions, routed calls, faxes, printed information and screen displays of information could all be EOs. Weapons firing solutions, status reports, commands to another system or outgoing e-mail could be counted as EOs. One exception to this rule is that all operator requested retrievals of information from any combination of ILFs/EIFs, that do not contain any derived or calculated information, are counted as External Inquiries (EQs) and not as EOs.

Each identified EO must be assigned a functional complexity of Low (L), Average (A) or High (H) based upon the number of Data Element Types (DETs) and File Types Referenced (FTRs) associated with the EO.

- Data element types (DETs) are usually unique user recognizable, non-repeating fields/attributes, including foreign key attributes, that exit the boundary of the subsystem/application.
- File types referenced (FTRs), or more simply files referenced, totals the number of internal logical files (ILFs) read or referenced and the external interface files read or referenced by the EO transaction.

DETs and FTRs are applied to the following matrix in order to determine the functional complexity of each EO:

External Output

	Data fields		
File Types Referenced	1-5	6-19	20+
< 2	L	L	A
2-3	L	A	H
>3	A	H	H

Identify the External Inquiries

Identify unique operator trigger (question)/response (answer) combinations that retrieve stored data without any mathematical calculation. Inquiries (EQs) are usually common retrieve, view, extract, display, browse or print functions. Typical examples might include a view of the current location of a target, a digitized map image, an event report or log, a view of incoming mail, the retrieval of voice or fax messages, a display of a user profile or current operational settings, diagnostic data, security data, Help, parameter settings, status of the system or retrieval of phone listings. Retrievals of data upon operator request from any combination of ILFs/EIFs, that also

1770 E. Lancaster Ave, Suite 15
Paoli, PA 19301
+1.610.644.2856 (F) +1.866.293.0120

contain any derived or calculated information, are counted as External Outputs (EOs) and not as EQs.

FTRs and DETs are counted separately for both the input and output sides of each EQ. The higher of the two complexities from either the previously provided EI matrix or EO matrix determines the value of the EQ.

Unadjusted Function Points Are Computed

Low, Average and High ILFs, EIFs, EIs, EOs and EQs are totaled using the appropriate IFPUG matrices. Value for each category are calculated using standard weights, using the previously presented table, in order to determine that Total Unadjusted Function Points for the subsystem.

General System Characteristics (GSCs) Are Determined, And The Value Adjustment Factor (VAF) Is Calculated

The degree of Influence (0-5) for each of 14 General System Characteristics (GSCs) are determined. These questions are answered using Degrees of Influence (DI) on a scale of zero to five.

0 Not present, or no influence

1 Incidental influence

2 Moderate influence

3 Average influence

4 Significant influence

5 Strong influence throughout

Characteristics such as redundancy, complexity, communication, distributed processing, security, etc. are recognized here. A special set of General System Characteristics are currently being developed by the New Environments Committee of IFPUG to correspond more directly to the real- time environment. The categories for the current GSCs follow:

1. Data Communications
2. Distributed Data Processing
3. Performance
4. Heavily Used Configuration
5. Transaction Rate
6. On-Line Data Entry
7. End-User Efficiency
8. On-Line Update
9. Complex Processing
10. Reusability
11. Installation Ease
12. Operational Ease
13. Multiple Sites
14. Facilitate Change

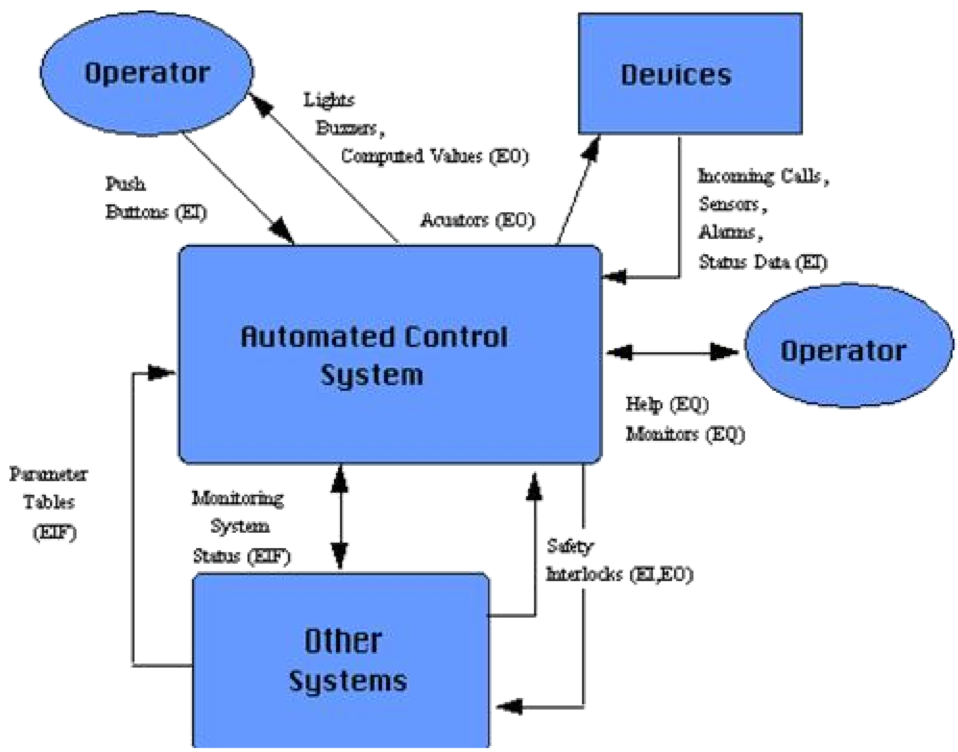
1770 E. Lancaster Ave, Suite 15
Paoli, PA 19301
+1.610.644.2856 (F) +1.866.293.0120

The score for all 14 GSCs is totaled and multiplied by .01. That value is added to .65. The result is the VAF which will be within a range of .65 to 1.35.

The Adjusted Function Point Count Is Calculated

The VAF (value between .65 and 1.35) is multiplied by the Unadjusted Function Point Count to determine the Final Function Point Count.

A Simple Example



Function Points is a Standard

IFPUG has worked closely with ISO, IEEE and NIST to establish the Function Point metric as an international standard. Although its initial use was principally extended to management information systems, Function Points have been extended to the measurement of real-time operating systems, process control systems, weapons systems, telecommunication systems, knowledge based systems and scientific systems.

For further information on this topic or to talk to a DCG expert, contact us at 610.644.2856 or send inquiry e-mail to info@davidconsultinggroup.com

1770 E. Lancaster Ave, Suite 15
Paoli, PA 19301
+1.610.644.2856 (F) +1.866.293.0120