Function Points and Measurement
What's a Function Point?

by Carol A. Dekkers, Quality Plus Technologies, Inc.
email: dekkers@qualityplustech.com

Published in the QAI Journal
December 1998

In the past several months there have been a proliferation of software measurement conferences in the U.S. with a heavy emphasis on function points and functional size metrics. While the International Function Point Users Group (IFPUG) conferences naturally feature functional size measurement tracks, the Applications of Software Measurement (ASM) conference now devotes an entire track to function points, and the ASQC International Software Quality Conference featured many measurement presentations including ISO and CMM measurement topics. It appears the software industry is starting to realize the applications of and potential for functional size measurement in managing software development. Yet, I still meet people who think that function points are ***the*** measurement program in themselves. Function Points (and other functional size measures) alone do not make a measurement program!

As a measure of software size (similar to square feet in construction), function points alone are not enough to comprise a software measurement program. Function points measure the functional size of software -- no more -- no less. In the same way that square feet alone are insufficient for a builder to manage construction, function points alone are insufficient for a software developer to manage a project. Functional size is only relevant when used together with other fundamental measures to produce normalized software metrics. Managing the software process is possible with a coordinated suite of appropriate software metrics, some of which may be based on functional size.

This article provides an overview of functional size measurement and details how to calculate Function Points, which is one of the most popular methods to calculate functional size. It is intended to be a refresher for readers who are familiar with the concepts but not the details behind functional sizing of software, and an introduction for readers unaware of function points.

What is Functional Size?

Functional size is a measure of the size of software based on a standardized evaluation of the users' logical requirements. There are several measures currently in industry to measure functional size, the oldest of which is function points. Similar to the square feet of a house, function points are independent of the physical methods, tools or development language used to

build the software. The process to calculate function points is contained in the International Function Point User Group's (IFPUG) Counting Practices Manual (currently in release 4.0). Unlike Lines of Code, Function Points are not dependent on the physical implementation and languages used to develop software.

Other measures claiming to also measure software functional size include Mark II Function Points, Boeing 3D Function Points, and Feature Points. For the purposes of illustration, this article focuses on Function Points (also referred to as IFPUG Function Points), but does not preclude the reader's use of other functional size measures listed above. For further formation about Function Points or other functional size methods, please contact the author.

What are Function Points?

Just as the square foot size of a house does NOT equal the speed at which a house can be built or even its construction time, the Function Point size does NOT equal productivity or work effort. Function points measure the size of WHAT the software does, rather than HOW it is developed and implemented. This means that given a common set of logical user requirements, the function point (FP) size of the software will be the same whether it is developed using COBOL or DB2, or using rapid application development (RAD) or structured development methods.

How Old are the Function Point Concepts?

Function Point concepts were first introduced by Allan Albrecht of IBM in 1979 at a Guide/Share conference, refined into a formal methodology and published in the public domain in 1984. Subsequently, a community of avid users decided to further standardize the FP counting rules and formed the International Function Point Users Group (IFPUG) as a formally incorporated not-for-profit group in 1986. Since that time IFPUG has become a leader in the establishment of and publishing of FP related documents including the Counting Practices Manual (CPM currently in release 4.0[1]), the Guidelines to Software Measurement (currently in release 1.1, and which replaced FP as Assets) and several detailed FP Case Studies. Today, IFPUG remains largely a volunteer organization (the only employees are administrative staff), is an active participant in the ISO working group on Functional Size Measurement (ISO/IEC JTC1 SC7 WG12), administers certification for FP training materials, Certified Function Point Specialists (CFPS) and FP software, hosts its own WWW site (http://www.ifpug.org/ifpug), arranges software measurement education and holds at least two conferences annually. Currently, members from over 13 countries participate in IFPUG through service on committees or by attendance at training and conferences.

[1] IFPUG's Counting Practices Manual (CPM Release 4.0) can be obtained through the International Function Point Users Group in Westerville, OH at (614) 895-7130.

From small beginnings as an original concept by Allan Albrecht of IBM in 1979 to the establishment of an ISO workgroup on Functional Size Measurement in 1994, functional size measurement (which includes Function Points) has emerged into the mainstream. Today additional methods as mentioned previously are being used to measure functional size. Proponents of each method advocate their measure's use for a particular class of software and often for a particular use. For illustrative purposes, this article focuses on the method of counting Function Points following the current IFPUG published standard (CPM 4.0).

What is the Function Point Counting Process?

The basic steps[2] involved in function point counting include:

- Determine type of count (can be a new development project, an application/ base count or an enhancement project count).
- Identify the application boundary (i.e., what functions must the software perform?)
- Count the data function types (split into: i) Internal Logical Files or ILFs, which are logical data groups maintained within the application boundary, and ii) External Interface Files or EIFs, which are used only for reference by the application).
  Each ILF is worth 7,10, or 15 FP, while each EIF is worth 5,7 or 10 FP.
- Count the transactional function types (split into: a) External Inputs or EIs, which are data entry processes, b) External Outputs or EOs, for example, reports, and c) External Queries or EQs, for example, browse employee details).
  Each EI or EQ is worth 3, 4 or 6 function points, while each EO is worth 4,5 or 7 function points.
- Several simple matrices based on data element types (user recognizable, non-recursive data fields), together with record element types (subsets of user recognizable data) or file types referenced (number of logical data groupings required to complete a process) are used to determine the whether each function is Low, Average or High in complexity. The following IFPUG table summarizes the number of function points assigned to each function type:

[2] The full details of FP counting are included in the IFPUG Counting Practices Manual (which including examples of FP counting scenarios is over 600 pages). Additionally, there are full Case Studies of FP counts done at differing phases of application development that can also be ordered through the IFPUG office at (U.S. phone number): (614) 895-7130.

Function Type	Low	Average	High
EI	x 3	x 4	x 6
EO	x 4	x 5	x 7
EQ	x 3	x 4	x 6
ILF	x 7	x 10	x 15
EIF	x 5	x 7	x 10

- Determine the value adjustment factor (VAF) based on an equation (VAF = 0.65 + (Sum of General System Characteristics x .01) and the evaluation on a scale of 1 to 5 of the following fourteen General Systems Characteristics. Specific evaluation guidelines are provided in the IFPUG CPM:
    1. Data Communication
    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
- Calculate the final adjusted FP count. (Final FP count = unadjusted count x VAF)

How can One Use Function Points?

Given the FP sizes of software applications or projects together with other measures, metrics normalized by FP size can be calculated and used for comparative analysis. For example, it is possible to compare the Development Productivity rates for different methodologies given work effort and product size measures. To calculate productivity rates take the size of each development product divided by the work effort expended on each. Analysis of the differences between the two rates could provide process improvement opportunities.

In addition, other productivity and delivery metrics can be calculated given function points together with other measures. Quality metrics (for example, defect density) and support ratios (size of application supported per person in maintenance) are also possible using function points and other collected and correlateable measures. Note, that function points provide us with a functional size measure from the user perspective and are NOT a silver bullet solution.

Just as other software measures are specific in their application and usage, function points can be correlated with other measures to produce specific software metrics. Function points do not correlate to everything (e.g., it does not make sense to correlate customer satisfaction to FP size), nor can they control behavior. Software metrics are passive tools used to quantify and report the results of change, and function points are no exception.

How do Software Metrics and Function Points fit into a Measurement Program?

Function Points do provide us with a standard, normalized software measure of the ***functional size*** of an application's logical user requirements and together with other measures can illustrate various points of the software development process that warrant improvement. In a future article, we will examine how and when functional size fits into a software measurement program and the primary considerations when initiating software measurement based on function points.

Carol A. Dekkers is the President of Quality Plus Technologies, Inc. a management consulting firm specializing in training and consulting in function points, software metrics, requirements and estimation process improvement. She is the President of the International Function Point Users Group (IFPUG) Board of Directors and is a project editor within the ISO Functional Size Measurement workgroup (ISO/IEC/JTC1/SC7 WG12). She is a frequent presenter and trainer at both U.S and international quality and measurement conferences and is credentialled as a Certified Management Consultant (CMC), a Certified Function Point Specialist (CFPS), a professional engineer (Canada) and an Information Systems Professional (ISP). Carol may be contacted via email at dekkers@qualityplustech.com.

==========end of article ==========