

QUALITY PLUS TECHNOLOGIES, INC.

Software and Technology Solutions

What's the Point of Function Point Analysis?

PART 1: Managing the Software Development Process

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

Published in <i>IT Metrics Strategies</i> <i>September 1998</i>

Introduction

Almost twenty years ago, the world was first exposed to Function Point Analysis when Alan Albrecht of IBM presented the concepts at a Guide/Share conference in 1979. Since then, the International Function Point Users Group (IFPUG) has refined the method into what is now referred to as "IFPUG 4.0" function point counting rules (the newest update of the method is targeted for release in January 1999). There have also been various metrics' groups throughout the world who have put forward their own variations of function points. Keeping in mind that function points (FP) are not the panacea of software metrics, it is also important to note that there are **many** areas where functional size is useful to software development and linking IT performance to business value. Although most commonly used as the input variable for product size in estimating models, function points can be used to normalize quality, productivity and financial aspects of software development. This article explores this range of usage, with the hope that companies will realize that the return on investment for function points can go well beyond a single process improvement or measurement initiative.

Function Points are Applicable to the Process and the Product

In a previous article, I mentioned that IFPUG function points, (and other variations of functional size measurement), are not the "Swiss army knife" of software metrics. What this means is that function points, like the square feet in house construction, have specific areas of applicability in software development. Just as other software metrics (MIPs, defects, DASD requirements, etc.), are applicable to particular aspects of the system development process, so too are function points. Function points represent the size of the software's "functional user requirements," independent of how the software is developed. This logical size of software is useful for tracking and

controlling the software development process, gauging the resultant software product, and managing the inventory of your software assets.

The range of uses for functional size measurement has been gathered from a number of sources including:

- Client experiences in using function points to solve real life system development challenges;
- Workflow Management measures featured in Howard Rubin's keynote article "IT Workflow Management Through Measurement" (ITMS, December 1997);¹
- The informative annex of the recently published International Organization for Standardization (ISO) standard: "ISO/IEC 14143-1:1998 -- Information technology -- Software measurement -- Functional size measurement -- Definition of concepts"² which outlines usage of functional size measurement for project management, and forecasting and performance measurement; and,
- Ongoing industry research and literature reviews on behalf of measurement clients.

This article is not meant to be exhaustive, but it is intended to illustrate how functional size measurement can be used for internal IT measurement and to link IT to business value. The three major areas of function point usage have been identified and are addressed in this and a subsequent article:

- Managing the Software Development **Process** (covered by this article: PART 1)
- Gauging the Software **Product** (covered by an article in the next 2 months: PART 2)
- Managing the **Inventory of Software Assets** (also covered by the PART 2 article).

Managing the Software Development Process

The range of uses presented in this section cover the software development process, but do not include uses of function points for Maintenance and Support. These are covered in PART 2: Gauging the Software Product and Managing the Inventory of Software Assets.

1. Project Estimation

Work effort and cost estimates are two of the most critical factors in project "go or no-go" funding decisions, and yet many project managers fail to do proper

¹ *IT Workflow Management Through Measurement*, by Howard Rubin, **IT Metrics Strategies**, Vol. III, No.12, December 1997, pages 1-5, Cutter Information Corp.

² ISO Standards may be obtained directly from the International Organization for Standardization (ISO) or your national standards body. In the U.S., ISO standards may be obtained through the American National Standards Institute (ANSI).

estimating. According to Robert Glass in *Building Quality Software*: “If there is one management danger zone to mark above all others it is software estimation.”

Both work effort and cost estimates become more accurate when we can maximize the number of known and minimize the number of unknown factors using proven estimating models. Using function points as the input value for software product size contributes to this increased level of accuracy. Many work effort and cost estimation models today include function points (FP) as the input measure for software size.

Additionally, as the project progresses and the user requirements solidify, it often becomes clear that the initial FP size may have missed some of the requirements. By updating the FP count and running the estimating model again, revised estimates of work effort and cost can be made. Over time, a history of product growth by phase can be built, and subsequently used on similar projects to estimate the anticipated size increase during the development process.

2. Requirements “Completeness”

A lingering question for project managers at the outset of a project is “Are the documented user requirements complete?” Function points can be used to validate the completeness of user requirements through a comparison to similar project FP *profiles*. A project FP profile is simply the completed project FP count broken down by function type.³ Here’s how it works: Once the initial FP count for a new project is done, the relative percentage breakdown of each function type is compared to the FP profile of similar projects. If a particular function type (for example, external queries) is found to be outside the standard range of contribution for similar projects, the project manager should go back to the users and confirm whether the gathered requirements are different from the norm or are incomplete. (For example, if your historical average FP profile for on-line banking systems consists of at least 40% of the FP as External Inputs, then if a new project’s FP estimate consists of only 20% of its FP as External Inputs, there may be some input functions missing from the documented requirements.) There have been cases of fixed price bids being corrected, when users were confronted with an anomalous FP profile and identified missing requirements that they thought would be included later at no charge.

3. Change and Scope Management

The exponential impact of rework on project schedules and cost is well documented. Although not the only reason for rework, change introduced late in the development life cycle can set off rework spirals that can spin projects out of control. It is a fact of life that requirement changes will occur during software development, however, it is critical to assess and quantify their impact on the project *before* the project schedule is interrupted.

³ Function type refers to the specific IFPUG 4.0 components: External Input, External Output, External Query, Internal Logical File and External Interface File, as contained in the copyrighted IFPUG Counting Practices Manual 4.0, maintained by the International Function Point Users Group. For further information about this manual or other IFPUG publications or membership, please call the IFPUG administrative office at (614) 895-7130 or send an email from the IFPUG web site at www.ifpug.org

Unmanaged change during a project increases the overall cost in terms of effort, resources, duration, rework, and morale on a project. Still, many project managers are forced to do whatever it takes to “get the changes made.” Using function points to quantify the size of change, it is possible to create estimates for the work effort, cost, resources and revised project schedule(s). Armed with these estimates, and the knowledge about the *Mythical Man Month*⁴ (nine women and one month cannot make a baby), project managers can better assess the impact of change on their project. Customers and developers can then make better decisions about the project scope and project plan, and whether to make or defer particular changes.

Keeping track of project change and the actual project impact (actual versus estimated effort, costs, etc.) can be invaluable in process improvement efforts focused on change management.

4. *Project Tracking and Completion*

The project size in function points and the detailed breakdown of functions can be used as an inventory against which to gauge the progress and completion status of a project. As functions are completed or pass a particular milestone, the percentage of their function points divided by the total project function points provides a figure of the project’s progress. This can be used equally well with new development or enhancement projects and can be adapted to include scope changes..

5. *Functionality Fit of Software Packages*

Using function point methods to predict the relative “fit” between the logical user requirements and a Commercial Off the Shelf (COTS) package is one of the emerging uses for functional size. The functional fit can be calculated in several ways depending on the needs of the organization. Some companies calculate the functional fit as the percentage result of dividing the project function points delivered by the package (without modification) by the project function points (based on the user requirements). If the percentage is less than 40%-50%, most companies judge that the target package is a poor fit.

Other companies judge the package fit based on the project function points divided by the total function points delivered by the package. This can be done projected over several phases of the project as the user requirements increase. A low percentage package fit gives warning to the developers that they will be testing and installing a high percentage of package functionality that the users will not use. (This must be tested to ensure that the package works.) Additionally, it indicates that there will be function points in the delivered product inventory will take up hardware space and support resources even though the functionality is not required.

6. *Cost Analysis of Project Alternatives*

⁴ Fred Brooks’ book: *The Mythical Man Month* was first published in 1972, re-released in 1997 and is still one of the authoritative project management books referenced in IT today.

When analyzing the relative cost of various project solutions, it helps to have a per unit cost figure on which to compare. Function points provide the normalized basis for calculating such a value. To do so, take the total projected cost for each alternative (custom development costs, hardware, software, package acquisition, etc.) and divide it by the project function points. This provides a cost per function point that can be easily compared across the differing alternatives.

Multiple phase or release unit costs can also be projected by estimating the incremental projected costs for those phases, estimating the project function points for each phase, and then dividing the two. This may illustrate the fact that a package may result in a higher first phase unit cost (cost per project FP), and yet be the most cost effective alternative when subsequent phases are included. (Note that the costs of modifying or customizing package functionality must be included in the cost of the applicable alternative.)

7. *Productivity and Workflow Trending*

In the December 1997 issue of ITMS, Howard Rubin identified a series of workflow and productivity measures of interest to senior management. Suggested metrics included:⁵

- Project delivery (in FP/work effort),
- Development trends (Total new FP/total project effort)
- Delivery rate (Total FP/total duration)
- Enterprise productivity (Total new FP and maintenance FP/ total effort)

Because of their independence, Function Points are useful to normalize delivery rates and productivity ratios across disparate technologies, hardware platforms, languages, etc. Trend analysis of these figures can illustrate process improvement gains, prove the value of new tools and methods, and identify opportunities for corrective action.

8. *Contract management*

Function points can assist in managing the cost and schedule of software development by contractors or software suppliers. Vendors can, and do, bid on the basis of the functional user requirements, function point size, productivity rates, cost per FP (in development and for scope changes). Both the vendor and the customer require a good understanding of function points to ensure that contract management on this basis can be fair and successful.

Full outsourcing management with function point based measurement components are also frequent uses of function points. Outsourcing management is included in the PART 2 article under the topic of Managing the Inventory of Software Assets.

9. *Project Planning to Optimize Quality and Productivity*

The knowledge of which project attributes (and combinations of attributes) contribute to higher levels of quality and productivity helps managers to plan projects

⁵ *IT Workflow Management Through Measurement*, by Howard Rubin.

for success. Questions such as the following can be answered confidently when enough historical information about workflow metrics⁶ and project attributes are collected:

- Does our use of Joint Application Design (JAD) sessions increase requirements' accuracy? Do they reduce project rework?
- Does training contribute to higher levels of productivity on projects using new tools?
- Does frequent project turnover result in lower productivity and late delivery?
- What is the most productive team size for a project of approximately 1000 FP?
- Are user walkthroughs a good investment in terms of making sure the requirements are right?

10. *Communication with Customers and Users*

Because function points reflect the size of a project or application based on the functional *user* requirements, they are more easily understood by customers than subroutines, objects, or modules. The functions in function points are based on the concept of an "elementary process"⁷ that reflects a unit business process from the customer or user perspective. Communication of results, progress, scope, impact of change, etc., can be facilitated by using functional size as the common denominator. Customers and users more easily buy-in to the development process when business friendly terms such as function points are used in place of technical jargon.

(Note that customers and users will need training in at least the concepts of Function Point Analysis, prior to using them as a communication vehicle.)

Gauging the Software Product

The uses of function points related to the software product will be covered in PART 2: Gauging the Software Product and Managing the Inventory of Software Assets. The following list outlines the uses that are included:

- Product quality evaluation and comparison (product quality, repair cost ratio, defect ratio)
- Product stability
- Workflow and productivity (maintenance)
- Gauging product results due to process improvement

Managing the Inventory of Software Assets

The uses of function points related to the software portfolio are covered in PART 2: Gauging the Software Product and Managing the Inventory of Software Assets. The following list outlines some of the uses that are included:

⁶ : See the previous section: 4. Productivity and Workflow Metrics

⁷ For further information on the concept of "elementary process" refer to the International Function Point Users Group (IFPUG) Counting Practices Manual 4.0, available at a cost to members and non-members by calling IFPUG in the U.S. at (614) 895-7130. (or via the IFPUG website at www.ifpug.org)

- Portfolio trend analysis (Growth and change)
- Workflow and productivity (On an enterprise level overall and broken down by maintenance, development /enhancement)
- Maintenance and support management (Support load, new project load, maintenance backlog)
- Financial comparisons and trending (Unit costs of maintenance and development, total cost of replacement for re-engineering, total asset value)
- Process improvement (identification of opportunities, result analysis)
- Corporate IT strategy development (demographics, build versus buy)
- Outsourcing management

Summary

The range of uses for function points in managing the software development process is diverse and can assist in quantifying and normalizing comparative metrics. Additionally, there are many applications of function points to gauge the software product and contribute to the management of your overall software portfolio.

Function points are still not *the* panacea of software metrics, however, their application and usage in system development today goes far beyond their inclusion in a single process improvement or measurement initiative.

=====

Carol A. Dekkers is the President of Quality Plus Technologies, Inc. a management consulting firm specializing in quality initiatives, software metrics, and process improvement. She is a frequent presenter and trainer at both U.S. and international quality and measurement conferences. Carol holds credentials as a Certified Management Consultant (CMC), a Certified Function Point Specialist (CFPS), a professional engineer (Canada) and an Information Systems Professional (ISP). She is the President of the International Function Point Users Group (IFPUG) Board of Directors and formerly held the positions of Vice President and Director of Counting Standards. She is also the project editor for Part 5 of the ISO/IEC functional size measurement project (“Definition of Functional Domains for use with Functional Size Measurement”) which is currently at the PDTR stage of international balloting.