

*Sheila P. Dennis, Director of Sizing Services*

*David Garmus, Founder*

IBM first introduced the Function Point (FP) metric in 1978 <sup>[1]</sup>. Function Point counting has evolved into the most flexible standard of software sizing in the information technology (IT) domain today.

There are several characteristics that account for the flexibility which drives the popularity and usage of the function point methodology, but the foremost appeal is the ability to measure the size of any software deliverable in logical, user-oriented terms. All sizing is based upon identifying, evaluating and weighting functional entities (inputs, outputs, inquiries and data usage) in an application or project from a user perspective. The term user is defined to be any person and/or thing that interacts with the software. Clearly, the user could even be another system or even a machine. By applying this approach, function points can be universally applied regardless of platform, environment, language, or other technical considerations.

### **Using Function Points**

The IT industry, including the Department of Defense, has successfully used function point analysis to size a wide spectrum of applications and projects, including general business, complex financial and accounting, logistics and communications systems. It has been proven on a variety of development platforms and/or environments (e.g. mainframe, client-server, web, stand-alone PC, data warehouse); for a variety of development types (real-time, batch, interactive or control systems); for internal or external development efforts (e.g. on-shore, off-shore, contractor-based); and for vendor and/or COTS package integrations.

As projects are completed and software deliverables are produced, function point sizing, together with a collection of other meaningful measures, can be used in a variety of initiatives.

- **Balance Scorecard.** Using function points as the basis for size normalizes metrics across platforms and projects. Measures based upon function point sizing [e.g. delivery rate (hours/function point), defect rate (defects/1000 function points)] are currently being used as the cornerstone at select DOD installations supporting the enterprise IT Balance Scorecard.
- **Benchmarking.** Measuring DOD's IT performance against competing outsourcing environments, as well as other governmental organizations, has become increasingly important in identifying opportunities for improvement in time to delivery, cost reduction, and customer satisfaction. Cost per function point delivered and function points supported/developed per Full Time Equivalent are only two of many metrics used in current benchmark initiatives.
- **Outsourcing Service Level Agreements.** Commercial and governmental uses of delivery rates by platform and defect density are effective in providing a contractual basis for performance standards for outsourcing requirements when function points are used as the common size measure.

## Function Point Counting Process

The function point counting practices are governed by the International Function Point Users Group (IFPUG), a not-for-profit organization, consisting of IT measurement industry leaders and practitioners from over 30 countries. For more than 25 years, IFPUG has maintained the guidelines and rules for counting practices through committees whose members have extensive software development and measurement expertise. Each new edition, or version, of the rules (currently version 4.2) is verified by the IFPUG body through practical application and statistical methods to ensure consistency, usability and reliability. In this section, we will provide an overview of the counting process using simple counting examples. The counting process included in this article is not all-inclusive and must be complimented with the rules as defined in the IFPUG Counting Practices Manual <sup>[2]</sup>.

The function point method evaluates the software deliverable of a project and measures its size based on well-defined functional characteristics of a software system [3]. Therefore, one of the first steps in counting is to identify the functional processes of a project and categorize them into function point entities. After the identification, a complexity level (Low, Average, High) is evaluated for each of the entities, and then assigned a weight (3 - 15 points) based upon complexity. For simplicity, we will use examples based upon a Human Resources (HR) system.

**Identifying and Classifying.** The two major functional characteristics that are considered in function points are data types (files, tables, records) and activity-based transaction types (inputs, outputs, queries).

Data types are user defined and recognized logical data groups of data, usually physically stored as files or tables. We classify the data into two separate categories, internal and external.

- Data manufactured and stored within the system are internal logical files (ILFs). For HR, employee data would be an ILF.
- Data maintained within a different system but necessary to satisfy a particular process requirement are called external interface files (EIFs). For HR, IRS tax tables could be a potential EIF.

Transactional types are elementary processes that control, maintain or display data. We classify transactions according to whether they relate to data entering the system, or leaving the system.

- Data entering a system are called external inputs (EIs). For HR, examples of EIs could be (1) an incoming feed from another system or (2) adding an employee through screen entry.
- Data leaving the system are classified as external outputs (EOs) or external inquiries (EQs). For HR, an online display of employee data would be a typical EQ. Reports or feeds to other systems are also EOs or EQs.

**Evaluating Complexity.** After the logical entities for a project are classified into the function point entities (ILFs, EIFs, EIs, EOs, and EQs), a complexity level of low, average or high is assigned using IFPUG derived complexity matrices (Figure 1). The matrices are dependent upon the components of data types and transactions.

- Record Element Types (RETs) are mandatory or optional sub-groups of data.
- Data Element Types (DETs) are non-repeated fields or attributes.
- File Types Referenced (FTRs) are the internal or external data types (ILFs or EIFs) that are used and/or maintained by the transaction.

The number of different Record Element Types, and unique Data Element Types used and/or maintained, determine the complexity of data types. However, the complexity of transaction types are determined by the number of data types referenced (ILFs and EIFs) and unique Data Element Types. In general, the higher use of components results in a higher complexity.

- For the HR system, assume that “Employee Data” has two groups of data, employees and dependents, with a total of 35 unique, non-repeated, user recognizable fields. Then “Employee Data” would be rated Average.
- If the transaction to add an employee (EI) had at least 16 unique fields to enter on the screen in order to update “Employee Data”, and “Employee Data” was the only type of internal or external data to be used in this process, then the complexity would be based on 1 FTR with 16+ DETs, or Average.
- An HR report of employee tax data (EO) having 16 or more unique data fields, and using both the “Employee Data” and the external data from the IRS Tax tables, would be of High complexity.

| <b>(Data Types) ILF/EIF Complexity Matrix</b> |                  |                   |                 |
|---|------------------|-------------------|-----------------|
| <b>RETs</b>                                   | <b>1-19 DETs</b> | <b>20-50 DETs</b> | <b>51+ DETs</b> |
| 1   | Low              | Low               | Avg             |
| 2-5   | Low              | Avg               | High            |
| 6+  | Avg              | High              | High            |

  

| <b>(Transaction Type) EI Complexity Matrix</b> |                 |                  |                 |
|--|-----------------|------------------|-----------------|
| <b>FTRS</b>                                    | <b>1-4 DETS</b> | <b>5-15 DETs</b> | <b>16+ DETs</b> |
| 0-1  | Low             | Low              | Avg             |
| 2  | Low             | Avg              | High            |
| 3+   | Avg             | High             | High            |

  

| <b>(Transaction Type) EO/EQ Complexity Matrix</b> |                 |             |                 |
|---|-----------------|-------------|-----------------|
| <b>FTRS</b>                                       | <b>1-5 DETS</b> | <b>6-19</b> | <b>20+ DETs</b> |
| 0-1   | Low             | Low         | Avg             |
| 2   | Low             | Avg         | High            |
| 3+  | Avg             | High        | High            |

**Figure 1. Complexity Matrices** <sup>[2]</sup>

Assigning Weight. Once the complexity is identified, values for each entity are assigned using the IFPUG standard weights shown in Table 1.

| FP Type | Low | Avg | High |
|---------|-----|-----|------|
| EI      | 3   | 4   | 6    |
| EO      | 4   | 5   | 7    |
| EQ      | 3   | 4   | 6    |
| ILF     | 7   | 10  | 15   |
| EIF     | 5   | 7   | 10   |

**Table 1. Function Point Counting Weights** <sup>[2]</sup>

In the HR application, the previously identified entities would be weighted as follows:

- Employee Data (ILF) - Avg - 10 function points
- Add an Employee (EI) - Avg - 4 function points
- Employee Report (EO) - High - 7 function points

Calculating a Project Count. Assuming that the three functional processes listed above represented the requirements for a project, then the total unadjusted count for the project would be the sum of all the entities, or 21 function points.

At this milestone of the counting process, there is a Value Adjustment Factor (VAF) factor that is used to adjust the project count. The VAF, with a range of .65 to 1.35, is derived from evaluation of fourteen (14) General System Characteristics (GSCs) based upon the technical characteristics of the application. Complex processing, distributed processing, online data entry, security influence and transaction rates are a few of the aspects considered in the GSCs. In the HR project, if the GSC for HR was 1.1, then the project count would be adjusted by multiplying the unadjusted count (21 function points) by the GSC (1.1) for an adjusted total project count of 23 function points.

The final function point calculation yields a single number that represents the total amount of functionality being delivered. Once completed, the function point size of an application or a new development project can be communicated in a variety of ways. As a stand-alone value, the function point size of a system tells us how large the overall software deliverable will be. When the function point value is segmented into a more detailed display, it can communicate to end users the functional value of specific components of the system. Finally, more mature software measurement organizations can use function points to predict outcomes and monitor program progress.

### Summary

Function points is the most effective and flexible way of normalizing critical measures in the successful management and monitoring of both internal measurement initiatives and outsourcing arrangements. Use of function points as the standard size component of cost and quality measures can satisfy both the IT organization's need to monitor the outsourcing contract and the user's need to ensure the value of the deliverable. In addition, the use of function points provides the opportunity to make comparisons to industry performance levels.



### About the Author

Sheila P. Dennis is Director of Services, Vice President, Software Sizing and Software Measurement for the David Consulting Group (DCG). She is the Past President of the Rocky Mountain Function Point Users Group, has served on three International Function Point Users Group (IFPUG) committees, and has been a Certified Function Point Specialist for over 12 years. She came to DCG in 2004 after retiring from the Defense Finance and Accounting Service (DFAS). During her tenure at DFAS, she moved from software engineer to process engineer in 1993 when she was chosen as the organizational representative for the DOD Software Measurement Pilot, a collaborative effort of DOD and the Software Engineering Institute (SEI). She eventually became the manager of the process improvement (CMM), quality assurance and metrics programs for an organization of approximately 120 civilians. Her unit supported a portfolio of over 400,000 function points representing a variety of platforms and applications. She is currently an active consultant, conference speaker, and workshop teacher. She was a contributing author to *IT Measurement - Advice from the Experts* (Addison-Wesley, 2001) and a contributing editor of *Guidelines to Software Measurement* (IFPUG, 2003). She holds a Certificate of Management Studies from Golden Gate University and a B.A. in General Studies (Mathematics) from Columbia College.

#### Author Contact Information

Email: [s.dennis@davidconsultinggroup.com](mailto:s.dennis@davidconsultinggroup.com)

#### References

- [1] Jones, Capers, "The Expanding Roles of Function Point Metrics", *IT Measurement – Practical Advice from the Experts*, Addison-Wesley, 2002
- [2] *Function Point Counting Practices Manual - Release 4.2.1*, International Function Point Users Group, 2006
- [3] Herron, David and Garmus, David, *Function Point Analysis - Measurement Practices for Successful Software Projects*, Addison-Wesley, 2001

---

This article originally appeared in the June 2006 Edition (Vol. 9, Number 2) of the DOD Software Tech News.

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](mailto:info@davidconsultinggroup.com)