



## **Middleware and Function Points Do Mix!**

*Roger Heller, Vice President*

Most people agree that function points work perfectly well when measuring business systems. I often find myself, however, explaining why this robust software sizing metric works equally as well when measuring less traditional environments such as Middleware applications. There are often heated debates on whether International Function Point Users Group (IFPUG) function points can truly represent the "size" of non-business applications. In our client work, we have found that IFPUG function points do accurately represent the size of less traditional environments. Success depends on applying the rules consistently and analyzing the results against similar applications. With this in mind, there is no reason for organizations to risk potential failure by relying on proprietary and/or unsupported software sizing techniques.

Once over the acceptance hurdle, organizations can recognize many benefits that previously eluded them. While there are some function point rules and guidelines to be sensitive to, all in all the counting is very similar to traditional applications.

### **WHAT ARE MIDDLEWARE APPLICATIONS ANYWAY?**

In classes I teach I describe Middleware as being a software component that is in-between a user recognizable application and some other component of the technical architecture. The Middleware application could interact with another software application and/or a hardware component. The ultimate (human) end user typically does not recognize the Middleware application as an individual system even though they receive benefits from the functions it provides.

### **SO, WHO ARE THE "USERS" OF MIDDLEWARE APPLICATIONS?**

Function point counting of Middleware requires a broader definition of "User." When counting traditional systems, "User" typically refers to a person, namely the end business user. When using function points to count Middleware, we expand the definition of "User" to

include other software components and /or software engineers who are defining the requirements and using the Middleware. This expanded definition allows us to count from the Middleware perspective.

### **HOW DO YOU IDENTIFY MIDDLEWARE APPLICATION BOUNDARIES?**

Counting Middleware is ultimately a boundary issue. Middleware boundaries are based on separate and unique functions as seen by the user. Remember that we are not talking about a traditional human user but one as described above. With that in mind: data-management packages, security processing, memory managers, and data communications subsystems are Middleware applications and each could be considered to be a separate application boundary.

### **DO MIDDLEWARE APPLICATIONS HAVE FILES?**

Note: The following discussion is based on counting guidelines as defined in the IFPUG Counting Practices Manual (CPM v4.1).

Just as in counting traditional data functions, a file must contain data that is user identifiable, logically related and the data must be maintained (ILF) or referenced (ILF or EIF) through one or more elementary processes.

With this definition, a distinction needs to be made between files that contain temporary data and temporary files. For example, data may be received during an elementary process that is used for validation or to build a transaction that is sent to another application or user. In this case the data structure is permanent and it contains temporary data that is required only during the execution of the elementary process. It can be counted as a data function since the data is maintained through one elementary process even though it no longer resides within the boundary.

In some situations, a Middleware application may not store any data, since it could operate purely on control inputs.

## WHAT ABOUT TRANSACTION FUNCTIONS?

### External Inputs

A transaction that enters the boundary of the Middleware application to either control processing or results in storage of data required by the application is an External Input (EI). The External Input itself may only contain control information that does not have to update an ILF. Very often the External Input itself will be created by another application. For example, a telephone switch receives a message from a Customer Services application to turn-on a new feature for a customer. In this case the message to turn the feature on is an EI to the switch.

### External Inquiries

An External Inquiry causes data to exit the boundary of an application. The results of the External Inquiry cannot contain derived data or alter the behavior of the sending application. The process is the direct retrieval of previously stored data. The output will often be used by other applications to control specific processes required by the overall architecture. For example, the application retrieves a previously stored setting and sends it to another application that uses it as input. The function of retrieving and sending the setting would be considered an EQ.

### External Outputs

External Outputs are transactions that exit the boundary of an application and either contain derived data, cause the behavior of the system to be altered or update an ILF. The output is generated based on a specific set of system requirements. For example, given a set of parameters, an application has to derive a key that is passed to another application as input. The function of deriving and sending the key would be considered an EO.

## About the author

*Roger Heller, Vice President, Q/P Management Group Inc., wrote this article. This article originally appeared in Q/P's semi-annual newsletter, "The Q's & P's of Management"*

## ACCOUNTING AND REPORTING FOR MIDDLEWARE AREN'T THAT TRICKY!

This is another one of those sensitive items. Since the ultimate end-user functionality is already accounted for in the traditional applications portfolio, it would be double counting to add the Middleware function points to the same portfolio. Therefore, the results of the Middleware applications count should not be included in the business portfolio. It is more appropriate to establish a separate applications portfolio that only contains function point counts related to Middleware applications. This promotes specific tracking and reporting for these software products.

## WHY WOULD YOU WANT TO COUNT THIS STUFF?

There are many very good reasons to count Middleware applications. Organizations use the information for a wide variety of purposes including:

- Software size estimating
- Project scheduling
- Maintenance planning
- Cost estimation
- Software quality tracking and reporting

Using function points to size Middleware applications is a relatively new concept, but it is an important step forward. As a result of the rapid introduction of new technologies and computing architectures, software organizations are developing and utilizing Middleware at an increasing rate. To support this need, accounting for Middleware has taken on new importance. By utilizing Function Point Analysis, Middleware software engineers and developers can benefit from measurement just as their traditional counterparts have for several years. The use of this information will allow them to make better decisions on how best to manage and deliver their software development projects.