Can we compare Cosmic Function Points to IFPUG Function Points and is there a correlation?

February 2012

Scope of this Report

Determine if we are able to compare the two different function point counting methodologies - IFPUG and COSMIC. Also, determine if there is a distinct correlation between the two methods.

- providing a brief history of the two methodologies – IFPUG and COSMIC
- providing brief description of each method – IFPUG and COSMIC
- comparison of counting methodologies
- conclusions

A brief history

Function Point Analysis also known as FPA was first introduced in 1979 by Alan Albrecht as an answer to a business problem concerning the size of software applications. In 1984, the International Function Point User Group (IFPUG) was formed in order to provide the guidelines and certification processes that are currently in place as a way to standardize the methodology of sizing a software application from a user’s perspective. The IFPUG methodology is currently the most widely used function point counting method.

In 1998, the Common Software Measurement International Consortium (COSMIC) introduced a new methodology known as Full Function Points (FFP). This methodology was developed as an answer to the real time and embedded software applications that were being developed at the time. The COSMIC method is also based on a view from the user’s perspective. The International Organization for Standardization (ISO) recognizes both FPA methodologies IFPUG and COSMIC as a standard method to measure the size of a software application.

A brief description of each method

The IFPUG methodology is based on five types of components that are used to measure a software application. The first two components ILF (Internal Logical File) and EIF (External Interface File) measure the data or information that is either maintained or referenced by the application during the business processes. The remaining three components EI (External Input), EO (External Output) and EQ (External Inquiry) measure the actual processes that are bringing the information in to the software application or
sending the information out from the software application. All of the components are evaluated and assigned a complexity that can vary in range from 3 to 15 function points and the total of all components is summed to produce the functional size for the software application. Hence, for any given functional component, the maximum number of IFPUG function points that can be assigned is 15.

The COSMIC methodology is based on four types of components that are used to measure a software application. All four types of components measure the movement of data that is related to the software application. The four types of components or data movements consist of Entry, Exit, Read and Write. The COSMIC methodology does not include the actual data or information that is moving when calculating the functional size of the software application. The total of all four components or data movements are summed to produce the functional size for the software application. This philosophy may lead to higher counts than are typical for the IFPUG methodology. The range of function points for each data movement can vary from 2 to infinity. Hence, for any given functional component, the maximum number of IFPUG function points that can be assigned is infinity.

**Comparison of counting methodologies**

When applying FPA to a software application it is important to be able to perform this process at several different stages of the software development lifecycle. Both the IFPUG and COSMIC methodologies can be used at different stages in the software lifecycle to accurately measure the size of a software application. The Functional User Requirements (FUR) can be used by both methodologies when performing the sizing of a software application at various stages throughout the development lifecycle.

The IFPUG methodology takes information from the FUR, data model, screens, reports, user’s guide as well as any information that is gained from interviews with the development team to count the distinct functional components that are provided to the user. Based on that information, a complexity value in the range 3-7 is assigned to each individual transaction that has been identified. In addition, the data or information used during the transactional processes is also assessed a complexity value in the range 5-15. The products of the transaction and data components and their complexities are summed to produce the functional size for the software application.

The COSMIC methodology also takes the information from the FUR in order to assess the functionality that is provided to the user. From the COSMIC view point, this could occur at several different layers or components of a software application, for example each layer or component is solely responsible for a specific function within the software application. The processes or movements of data are identified and assessed a complexity. Each process or movement of data takes the total number of entries, reads, writes or exits in order to assess the complexity for each process. The total of each process is summed together to produce the total functional size for the software application.

A key difference in the two counting methodologies is that as individual functional components in software application become more complex, the size calculated by IFPUG is constrained (to 15) whereas in COSMIC it is open-ended. Proponents of COSMIC sometimes claim that this makes COSMIC more suited to software applications with higher algorithmic complexity (although this claim is not officially supported by the COSMIC organization). IFPUG supporters counter-claim that this effect is not significant.
Now let’s take a look at some measurement results from a study performed by Gu Xunmei, Song Guoxin, and Zheng Hong. There were seven application counted with both the IFPUG and COSMIC methodology.

Table 1 Comparison of different sizing methodologies applied to applications (Source: Gu Xunmei, Song Guoxin, and Zheng Hong)

<table>
<thead>
<tr>
<th>Product</th>
<th>Type</th>
<th>Size (IFPUG FPA)</th>
<th>Size (COSMIC-FPP)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Real-time</td>
<td>210</td>
<td>794</td>
</tr>
<tr>
<td>B</td>
<td>Real-time</td>
<td>115</td>
<td>183</td>
</tr>
<tr>
<td>C</td>
<td>Real-time</td>
<td>---</td>
<td>2604</td>
</tr>
<tr>
<td>D</td>
<td>Real-time</td>
<td>43</td>
<td>318</td>
</tr>
<tr>
<td>E</td>
<td>Mostly MIS</td>
<td>764</td>
<td>791</td>
</tr>
<tr>
<td>F</td>
<td>MIS (batch)</td>
<td>272</td>
<td>676</td>
</tr>
<tr>
<td>G</td>
<td>MIS</td>
<td>878</td>
<td>896</td>
</tr>
</tbody>
</table>

Based on the information from Table 1, we can see that the results are similar when the methodologies were applied to the MIS software applications. However, we can also see that there was no similarity when the methodologies were applied to the real-time software applications. While the information used is just a small sample of applications it clearly shows that there is no distinct correlation between the IFPUG and COSMIC methodologies.

Each of the methods is a different standardized process that is producing quantitative results for the companies that have implemented them into their software development lifecycle process. Companies rarely use both methodologies in their software process and usually have a preference as why to use one method over the other. In addition, both methodologies are competing against one another for their place in the software industry as the primary method for sizing software applications.

Conclusions

In conclusion, we believe that while the results from the two methodologies can seem to be comparable for certain types of applications, in reality this is simply coincidence. There is no correlation between the two methodologies.

Sources: