Automated Function Point Counting – A Fact Based Analysis

By: Roger Heller

In recent years there has been much discussion around the ability to automate function point counting to produce results that would be similar to the results obtained by conducting a function point count using the International Function Point User's Group (IFPUG) counting process and methods. There have been a number of companies that have tried to achieve this goal but have fallen short for a variety of reasons. The most prominent reason is that the tools attempt to produce a logical size of a project or application by analyzing the software from a technical perspective.

With the introduction of new tools which produce automated function point (FP) counts, it is worth reviewing the accuracy and capabilities of past products that produced automated counts. Much of the literature available today that discusses the accuracy of the current batch of tools is marketing material produced by the vendors themselves or by their partner network, who are attempting to appear as independent voices.

This historical view is useful in order to gain an understanding of the challenges faced and the inaccuracies that can result, which in turn can lead to misuse of the measurement information produced. This article is based on work completed for organizations that were interested in an independent evaluation of the capabilities of several automated FP counting tools. It is based on past experiences and describes a structured method that can be used in current and future evaluations of these types of tools.

Background and Scope

Over the past 15 years a number of automated function point counting tools have become available with the intention of providing low cost function point counts. Typically, these tools analyze application code to identify data and transactional functions in order to attempt to derive an IFPUG comparable function point count. Each tool utilized methods that were based on proprietary algorithms that the companies stated had been designed specifically to evaluate code and data structures from a logical perspective.

The intent of the analysis described in this article was to determine how accurately these tools produced IFPUG comparable function point counts. The analysis is based on three tools that advertised their capabilities to produce accurate automated IFPUG function point counts. The names of the specific vendors have been excluded from this article due to confidentiality concerns. However, the background of each is an indication of their viability and their business focus. Each of the three vendors have been in business for more than 20 years. Each offers products and services to their clients that aid in the management of large-scale application software systems. All of the vendors offered a tool intended to produce automated function point counts as part of a larger portfolio management product suite. Over the years these vendor tools attracted the attention of numerous organizations due to the potential cost and labor savings by automating the manual function point counting approach. However, the results produced by these tools were viewed with great skepticism. This is primarily due to the fact that function point rules demand that all counting be performed from the user's logical perspective. The automated function point counting tools "count" from a physical perspective.

Each of the tools analyzed have several characteristics in common. All of the tools worked only on a subset of the available programming languages and had difficulty when an application containing multiple languages utilized one or more languages that were not supported. Additionally, each required a technician from the vendor to install the software and load each application into the tool. The technicians often had to meet with the application architects or database analysts to understand how the application was structured and how the data was organized. Since the results could vary significantly, after the initial executions each tool required additional fine-tuning at the application level to attempt to ensure consistency of the results. These major limitations prompted a number of organizations to conduct formal studies to determine the accuracy and repeatability of the results produced by these tools.

Evaluation Studies

Evaluations of these products have been conducted by several organizations over the past 15 years. All of the evaluation studies compared the automated function point counts produced by the tools to manual function point counts in order to determine the accuracy of the tool. The manual counts were performed by Certified Function Point Specialists (CFPS) who were independent of the tools' vendors.

Each study approached the analysis in a similar manner. The first step was to identify a representative sample of the applications that could be considered for the analysis given the tools' limitations related to the supported languages. At the time of these studies, the analysis focused on application counting only since establishing project function point counts is difficult or near impossible for the tools.

The inability to count projects is due to the fact that IFPUG function point analysis recognizes that both logical and some structural changes can result in project function points. However, when viewed from a technical perspective difficulties can arise:

- In certain circumstances, once a function was identified it was difficult for the automated tools to determine if the transaction provided add, change, delete and/or inquiry capabilities
- Since the tools were primarily focused on analyzing software and code structure, it was difficult to distinguish a changed transaction when a logic change had been made
- Additionally, it was challenging to identify new, changed and deleted functions if the software itself didn't change significantly

As a result, project counting was out of scope for these studies. The resulting scope of this analysis included twelve application counts.

In each case, the time to complete the manual function point count was tracked as well as the time it took to set up and run an automated FP count for the application being evaluated. In addition, any time required to fine-tune the tool in order to ensure a more accurate count was tracked when it related to a specific application. The initial product installation and setup time were not included as part of the time tracking for these analyses. However, this additional time and cost is significant and should be considered in the overall cost benefit analysis of acquiring and using a tool.

Once the function point counts were completed, a number of comparisons were established. These included:

1) The overall functional size of the application produced by the manual count compared to the automated count
2) The accuracy of the functions identified in the automated count compared to the manual function point count
3) The accuracy of the complexity assignment at the function level of the automated count compared to the manual count
4) The effort related to establishing the manual function point count compared to the effort required to complete and fine-tune the automated counts. Note, the effort for this analysis included time spent by the application subject matter expert (SME), the CFPS and the tool technicians.

Study Findings

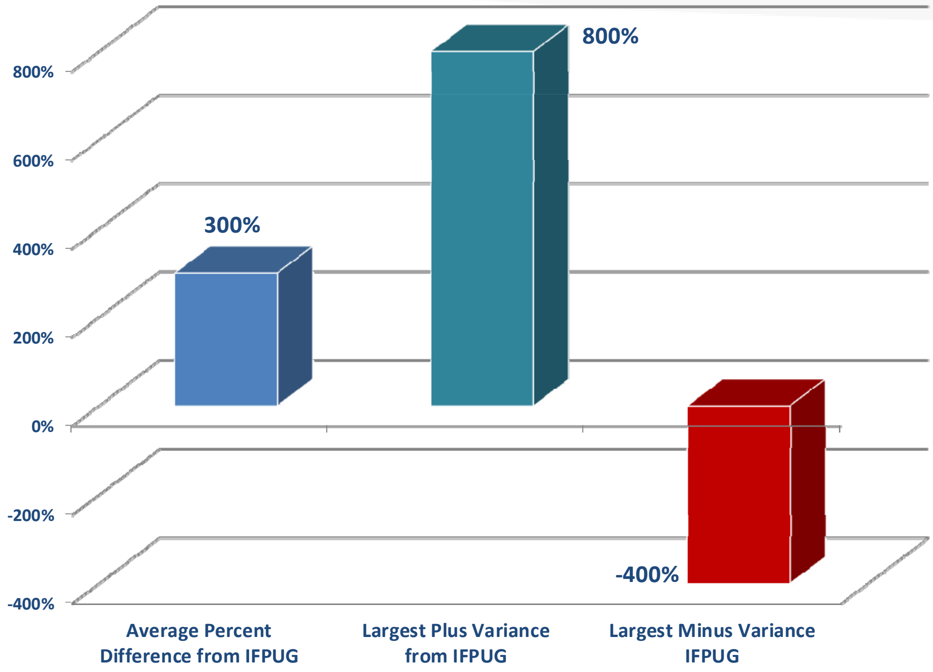
The individual studies included in this evaluation resulted in very similar findings. For the purposes of this article the results have been summarized below.

Summary

The automated function point counting tools evaluated were found to be grossly inaccurate in establishing function point counts. On average, the automated counts produced by the tools were 300% higher than the manual counts. In addition to the inaccuracies in counting, the expected time and potential cost savings through automated FP counting did not materialize. The time required to prepare, produce and review the automated counts was significant.

Chart 1 summarizes the major findings of the automated function point counts compared to the counts produced by an IFPUG Certified Function Point Specialist.

Chart 1 - Automated FP Counts Compared to Manual IFPUG FP Counts



Detailed Findings

- In general, the automated function point counts were significantly higher than the manual counts
  - Eleven out of the twelve counts analyzed had a variance of 40% or greater.
  - In one instance, the automated count was almost 800% higher than the manual count
  - In another case, the automated count was almost 400% lower than the manual count

- Only one count resulted in a 10% variance. After a close review of this count it was found that the details of the count varied greatly between the automated and manual counts. The automated tool inaccurately identified over 150 logical files. This was offset by the automated tool's unsuccessful identification of over 300 inputs, outputs and inquiries.

- The largest discrepancy occurred in one automated count which produced a size that was overstated by approximately 800 function points. The CFPS manual count resulted in less than 20 Internal Logical Files (ILFs) and External Interface Files (EIFs). The automated function point counting tool counted over 500 ILFs and EIFs.

- One of the tools did not identify any External Inquiries in one application even though over 15 were present.

- The extraordinarily high count produced by one of the tools was due in part to its process of counting as many functional components as possible and relying on a post-analysis manual review to eliminate any inappropriately counted functions
  - The technicians for this particular tool also recommended a review be conducted and adjustments to the automated count results be made by staff who are familiar with the application and who also have function point counting expertise

- In general, the manual process to review and adjust the automated FP counts was labor intensive

- The time required to utilize the code analysis tools was significant. A great deal of time was required to locate and load application modules. Significant time was also required to operate the tool, produce reports, review results and adjust counts. In some instances it was necessary to reload the code after the initial load.

- On average, it took two thirds of the manual counting time to utilize the automated counting tool. This does not include time required to review and adjust the automated FP counts.

- In one instance it took 50 hours to perform a manual FP count and 55 hours to produce an inaccurate FP count using an automated counting tool. If the time required to install the tool was included in the analysis, the effort to produce a manual FP count would be significantly shorter than the time required to produce an automated FP count.

Conclusions

Historically, the automated function point counting tools may have a number of beneficial uses such as determining code quality and complexity. These tools could not be relied on to provide automated function point counts that are within an acceptable variance of actual IFPUG certified counts. Even if these automated function point counts could be reviewed and adjusted to make them accurate, the time to utilize the tool and correct the counts was equivalent to, or greater than, the time needed to perform a more accurate manual function point count.

IFPUG has a process to evaluate and certify these types of tools but no tool in the 20+ year history of the organization has ever been certified. Some vendors have tried and failed. Others have endeavored to have the IFPUG certification requirements modified in an attempt to gain certification. It is logical to believe that large organizations that develop and sell these tools would have no problem spending the time and money to get their tools certified by IFPUG, if certification was possible.

There is a current attempt, sponsored by a non-IFPUG standards organization, to define a standard for automated IFPUG function point counting tools. However this is being done independently of the International Function Point User's Group. This current effort could result in a standard that professes to count IFPUG function points but in fact it may not.

The tools available today continue to promote that they provide the same benefits of these earlier entries into the market place. Today's tool may be able to achieve their stated objectives but the jury is still out. The available literature is mostly provided by the companies developing the automated tools and leans heavily towards marketing. Some of the available evaluations are being produced by partner companies which benefit from the sale of the tools.

Software organizations that are interested in evaluating the appropriateness of these automated tools for function point counting should take it upon themselves, or seek outside assistance, to conduct a due diligence assessment using independent resources and applying an evaluation approach similar to the one described above.

About the author

Roger Heller, Senior Vice President Q/P Management Group, is a recognized consultant, instructor and speaker with over 30 years of Information Systems and consulting experience. Roger has been a Certified Function Point Specialist since 1994 and is a past chairperson of the IFPUG New Environments counting standards committee. He is currently the chairperson of the IFPUG Membership committee. He specializes in helping organizations improve software quality and productivity through measurement. He has conducted numerous benchmark studies comparing internal and outsourcer performance to industry benchmarks. He has assisted numerous organizations in establishing software measurement programs and quality improvement initiatives. His other areas of expertise include: function point analysis; applying function point analysis techniques to emerging environments; software estimating; and application and technology planning.