# VERIFYING FUNCTION POINT VALUES

E.E. Rudolph, HS Bremerhaven, erudolph@hs-bremerhaven.de
G.E. Wittig, Bond University, gerhard_wittig@bond.edu.au
G.R. Finnie, Bond University, gavin_finnie@bond.edu.au
P.M. Morris, Total Metrics, Pam.Morris@Totalmetrics.com

**September 1998**

## INTRODUCTION

It is common practice to size computer software by Function Points (FP) when estimating development costs, assessing development productivity, or specifying software contracts. Function Points are also used to normalize results when analyzing or comparing software.

The Function Point technique was introduced by Albrecht [1979] and since has been refined (see IFPUG [1994]), or modified (see Symons [1988]). The concept of quantifying software by its functional size has been formalized by an international standard (see ISO/IEC [1997]).

Function Points, however, are not without controversy. From a theoretical point of view Function Points violate measurement frameworks (see Kitchenham et al. [1995]). Some of those theoretical objections may be too harsh, as suggested by Rudolph [1997]. There is, however, the practical concern that the absolute functional size values, produced by the popular Function Point techniques, have no (or dubious) justification.

Albrecht [1979] and IFPUG [1994] use a weight table to assign size values to individual function types. No justification is given for the specific selection of those weights, or the selection of functional size indicators. Symons [1988] derives the weights for his function types from the effort required to develop the function type. Such correlation between functional size and effort is beneficial for the estimating of development time. However, it limits the validity of the metric when normalizing software, or establishing development productivity.

A formal methodology has been proposed by Wittig et al. [1996] to determine empirically the weight coefficients for the IFPUG [1994] Function Point method. The IFPUG Function Point method defines Function Point analysis as “a standard method for measuring software development from the customer‘s point of view“ (see IFPUG [1994]). Wittig et al. use the Analytic Hierarchy Process (AHP) developed by Saaty [1980] to establish the weight factors. AHP is a proven technique of multiple criteria decision making. It has been successfully applied in industry, government and research establishments (see Saaty [1990]). Comparing five methods for determining weights in additive utility models Schoemaker and Waid [1982] found AHP to be the best technique.

## RESEARCH METHODOLOGY

The Analytic Hierarchy Process (AHP) technique compares and relates pairs of individual components (in this research the components are function types). The user of

an information system will be asked to assess, which of two function types is the larger one, and how much larger it appears to be. With sufficient samples of all combinations of the various function types, and complexity levels, this correlation process will produce a value set for FPA function types as perceived by the users of such systems.

Design of the data collection questionnaire

With 15 different components of function types (5 types with 3 complexity levels each) a total of 105 pairs of components would have to be assessed. Assuming that each pair would require 2-3 minutes to evaluate, the overall assessment would take at least half a day. Such highly repetitive and lengthy procedure would exceed the willingness of most organisations to co-operate with the data collection.

Instead, the questionnaire was restricted to a total of 25 component pairs. Within each of the 5 function types all three combinations between the three complexity levels were included. Additionally all combinations of the five function types at average level were covered.

The 15 distinct functions (one for each function type and complexity level) were randomly selected from an application system recently counted and familiar to the assessor. For most observations this required an individual selection of the sample functions. It ensured, however, that the assessor was familiar with the function types. It further provided a variety of different instances of individual components. In some cases different individuals assessed the same functions.

The 15 function components could be selected prior to the assessment. The assessor will typically complete the questionnaire in 40 – 60 minutes.

Assessing Functionality

The most difficult part of the assessment is to find a common basis on which to decide on the amount of “functional size”. IFPUG [1994] has no definition for functional size or functionality. ISO/IEC [1997] defines functional size as “a size of the software derived by quantifying the functional user requirements”. ISO/IEC [1997] defines functional user requirements as “part of the user requirements representing the user practices and procedures that the software must perform to fulfil the users’ needs excluding quality and any technical requirements”.

During the first field-tests it became obvious that assistance was required to provide assessment criteria for “functional size”. The IFPUG classification criteria would only apply to individual function types and did not cover comparisons between different function types. There were even within a function type additional criteria, which could influence a decision. Furthermore there is the danger that either the development effort or the benefit to the business (or the assessor) may influence the judgement of functional size of a component. And last but not least most assessors found it very difficult to compare the transaction function types (External Inputs (EI), External Outputs (EO) and

External Inquires (EQ)) with the data function types (Internal Logical Files (ILF) and External Interface Files (EIF)).

The following assistance was provided, encouraging the assessor to view the function from a manual point of view. The assessor then would concentrate on characteristics reflecting what is involved in order to execute and complete a function manually without the assistance of a computer. These criteria could be considered when comparing the amount of functionality delivered by different functional components. In addition to these the assessor may have other criteria which guided his or her judgement. So far, however, not many additional criteria were brought forward.

For the transaction oriented function types (External Inputs (EI), External Outputs (EO) and External Inquires (EQ)) consider:

1. The number of tasks necessary to complete the function, ie
   - the number of decisions needed to be made
   - the number of questions that need to be asked
2. The amount of information that needs to be handled by the function, ie.
   – received, - analysed, - produced, - written down (data elements, files)
3. The skill level of the person required to complete the function, ie their:
   - time required to understand the problem (assessed in hours)
   - capability to perform the necessary calculations and algorithms (education level)
   - relevant experience with similar problems (years of business experience)
   - knowledge of the correct rules to be applied to the problem (number of rules or facts)

For the data oriented function types (Internal Logical Files (ILF) and External Interface Files (EIF)) it may be difficult to visualise functionality in context with file types. Functionality of file types could be viewed as a measure to include one or more of the following:

- the amount of information that is contained in one file record (data elements)
- the capacity of the record to provide a range of information (# of different usages)
- the capacity of the file to be able to cross reference data in another file (# of links)
- steps, data volume or skill level necessary to access or collect such file

Example

Fig. 1 shows the example of a comparison of an average EI (function A) and a complex EI (function B). In this case function B was assessed to be twice the size of function A. The decision was primarily based on the fact that function B contained more tasks than function B.

| ID | Name | Larger |
|---|---|---|
| **A** | *Create new customer* | |
| **B** | *Change inventory* | ✔ |

How much more functionality? **2 X**

**Reason?:** more tasks *x*; more information __; higher skill required __ ; other __

Comments: *function B implies many consequential changes and verifications*

**Fig. 1: Example of a comparison result**

## FIRST RESULTS

Weight Coefficients

Initial results based on 23 projects were published by Wittig et al [1997]. The results are shown in the first data column of Table 1. Based on the experience of the first data collection more guidance for the assessment of functionality was provided in a second round of data collection. A further 22 projects were collected from organisations in Australia at the end of 1997. The combined results, based on 45 projects are shown in the second data column of Table 1. The third data column of Table 1 shows the weight factors introduced by Albrecht and used by IFPUG [1994].

**Table 1**: Scaled Function Point weights

| | | **23 projects** | **45 projects** | **Albrecht** |
|---|---|---|---|---|
| **ILF** | low | 3 | 3 | 7 |
| | avg. | 5 | 5 | 10 |
| | high | 12 | 12 | 15 |
| **EIF** | low | 3 | 2 | 5 |
| | avg. | 5 | 5 | 7 |
| | high | 9 | 9 | 10 |
| **EI** | low | 4 | 4 | 3 |
| | avg. | 7 | 7 | 4 |
| | high | 12 | 13 | 6 |
| **EO** | low | 3 | 3 | 4 |
| | avg. | 5 | 5 | 5 |
| | high | 10 | 10 | 7 |
| **EQ** | low | 3 | 3 | 3 |
| | avg. | 5 | 5 | 4 |
| | high | 9 | 9 | 6 |

The additional observations are very close to the initial results of the AHP study. Overall the results appear to confirm Albrecht‘s weight factors. The transaction function types, however, seem to score higher at the expense of the data oriented function which appear to be overrated.

More observations, however, are required, particularly for the data oriented function types which often could not be found at all complexity levels in the projects surveyed. This by itself is interesting since most of the projects surveyed were in excess of 1000 Function Points in total.

The data also showed a considerable amount of controversy in the assessment of functional size. Overall and on geometrical mean the original Function Point types appear to be not too far appart from the AHP results. On an individual basis, however, on average the AHP results differed by 150% from the established Function Point results. More data is needed from different geographical regions to confirm or correct the results obtained so far.

Lessons learned

For a technique which claims to represent the users view it is surprising to experience great difficulties in finding users who can relate to the function types identified by Function Points. Function Point counting specialists would claim that users did not appreciate the functions counted or the scope of those functions. At the end at least 50% of the AHP assessments were made by technical or business analyst staff and not by users from the business areas.

Part of this problem resulted from the fact that Function Point counts did often reflect the developers' views. Individual users on the other side were also often not aware of the full business requirements and judged on their limited views. Some functions therefore were underrated.

## CONCLUSIONS

The AHP approach and the weight coefficient results identified earlier by Wittig et al [1997] could be confirmed in this refinement of the initial AHP study.

Considering the limited amount of data oriented function types included in this study, and some difference of opinion in the assessment of the functional size of individual functional components more data should be collected from a wider range of users.

The practical and theoretical implications of a successful verification of the Function Point values would be substantial. From a business point of view Function Point results would become more credible. Scientifically the Function Point technique would no longer violate major parts of the software metric framework introduced by Kitchenham et al. [1995].

## ACKNOWLEDGMENTS

The authors would like to thank Bankers Trust Australia, Deutsche Telekom, ISL, LOGAS, Optus, Robert Bosch (Australia), Shell Australia, Telstra and the Victorian Dept. of Justice for submitting data to this study. This project has been supported financially by the Australian Research Council Small Grants Scheme.

## REFERENCES


**Albrecht**, A.J. 1979. "Measuring application development productivity", Proc. Joint SHARE/ GUIDE/IBM Symposium on Application Development, Monterey, Calif., Oct. 14-17, pp. 83-92.

**Kitchenham**, B.A., **Pfleeger**, S.L. and **Fenton**, N.E. and 1995. "Towards a Framework for Software Measurement Validation", IEEE Trans. S.E., Vol. 21, No. 12, pp. 929-943.

**IFPUG** 1994. Function Point Counting Practices Manual, Release 4.0, International Function Point User Group, 5008-28 Pine Creek Drive, Westerville, OH 43081-4899, USA.

**ISO/IEC** 1997. Information technology - Software measurement - Part 1: Definition of functional size measurement, Draft International Standard ISO/IEC DIS 14143-1.

**Rudolph**, E.E. 1997, "Software Metrics: Theory and practice", Proc. 8th ESCOM Conference, Berlin, May 26-28, pp.12-16.

**Saaty**, T.L. 1980, The Analytic Hierarchy Process, McGraw-Hill, New York.

**Saaty**, T.L. 1990, "How to make a decision: The analytic hierarchy process", European Journal of Operations Research, no. 48, pp. 9 - 26.

**Schoemaker**, P.J. and **Waid**, C.C. 1982, "An Experimental Comparison of Different Approaches to determining Weights in Additive Utility Models", Management Science, no. 28, pp. 182 - 196.

**Symons**, C.R. 1988. "Function Point Analysis: Difficulties and Improvements", IEEE Trans. S.E., Vol. 14, No. 1, pp. 2-11.

**Wittig**, G.E., **Morris**, P.M., **Finnie**, G.R. and **Rudolph**, E.E. 1996, "Project Research Design to Validate FPA Coefficients using AHP", ACOSM 96 Third Australian Conference of Software Metrics, Melbourne, 19-21 Nov.

**Wittig**, G.E., **Morris**, P.M., **Finnie**, G.R. and **Rudolph**, E.E. 1997, "Formal Methodology to establish Function Point Coefficients", Proc. IFPUG Fall Conf., Scottsdale, Sept. 17-19.