



Function Points Overview

Neville Turbit

Software Development Metrics

Prior to 1980, there was no method to estimate the size of IT developments other than to guess based on comparable developments. An IBM employee came up with a concept called Function Points. Since 1980, Function Points have been refined and developed to the point where they can be accurate within a few percent.

Counting Function Points

To build any system, you need to spend time building files to hold information, and interfaces to other systems (Files and Interfaces). You need to spend time building input screens (Inputs), enquiry screens (Enquiries) and reports (Outputs). If you count the:

- Files
- Interfaces
- Inputs
- Outputs and
- Enquiries

you have the start of a measure of the work to be undertaken.

International Function Point Users Group

To build a file as opposed to building an output will take a different amount of time. Through an organisation known as IFPUG (International Function Point Users Group) averages have been established for these five types of construction. An enquiry for example may take twice as long to build as an input. If these weightings are applied to the inputs, outputs, enquiries, files and interfaces, an Unadjusted Function Point Count can be calculated.

Adjust for Technical Complexity

This of course does not take into account the complexity of the application being built. Is it a simple listing of names and addresses, or is it calculating the orbit of a satellite around Mars? A further adjustment to the unadjusted function point count needs to take place for the Technical Complexity of the application.

Adjust for Environmental Complexity

The final adjustment relates to Environmental Complexity. Is the application to run on a single PC, or over a wide area network? Does it need to cope with 1 user or 1,000 users? Is it designed for on line update or is it batch processing.

Adjusted Function Point Count

The Adjusted Function Point Count is the final figure. It is derived by multiplying the Unadjusted Function Point Count by the Technical Complexity and the

Environmental Complexity. Depending on what sort of language is being used, different productivity figures exist. For example, a Cobol programmer may be able to deliver 10 function points per man month. A VB programmer 40 function points.

Calculating Function Points

To calculate how long it will take:

Adjusted Function Point Count / Productivity for that language = Development Time

e.g. for a VB development which can deliver 40 Function Points per man month

4,000 Function Points / 40 Function Points per man month = 100 man months of development.

If the cost of a man month of development is known a cost can be calculated. e.g. Assume a developer with oncosts is worth \$84k per annum (i.e \$7k per month) In the example if the development is 100 man months it will cost 100 x \$7k or \$700k.

100 man months may take 10 developers 10 months or 5 developers 20 months.

Project Scope

Another value of function points is managing project scope. As all the screens, reports, files and interfaces have been counted early in the development lifecycle, we have an inventory of what is in scope. If later in the development, 10 more reports are requested, it may equate to another 60 Adjusted Function Points. In real terms it means another 6 man weeks work costing \$10.5k. It can either be justified and additional resources and/or time added to the project, or rejected for the present.

By benchmarking completed or existing developments, a model can be established. It will show the relationship between the language, the skills of the developers, the completeness of the requirements, the development tools and all the other variables that normally exist in a project. The more information gathered, the more accurate the estimate of productivity will be for new projects.

Function Points Summary

Function Points are not absolutely precise. There is no absolutely precise measure for sizing developments. Function Points are open to some interpretation and often not comparable between organisations. If they are used consistently within an organisation they provide the only way to measure the size of a development effort and should be accurate within 5%.

For further information see the International Function Point Users Group (IFPUG) web site <http://www.ifpug.org>

Neville Turbit has had over 15 years experience as an IT consultant and almost an equal time working in Business. He is the principal of Project Perfect. Project Perfect is a project management software consulting and training organisation based in Sydney Australia.

Project Perfect sell “Project Administrator” software, which is a tool to assist organisations better manage project risks, issues, budgets, scope, documentation planning and scheduling. For more information on Project Administrator or Project Management visit www.projectperfect.com.au