Function Points as a Tool for the Appraisal of Software

by Curtis Graham, Guilherme Simões and Carlos Vazquez

Attributing a market value to a software system can be a complex process. There are various reasons to attribute monetary ($) value to a software application. Here are some of those reasons:

- Conduct IT Asset Management:
  - o Include software as part of the organization's assets
  - o Sell the application to another company
  - o Confirm the appraisal of a third party development firm
  - o Identify which components are considered as the most valuable
- Support Decision Making for IT Projects:
  - o Analyze if it's worth developing new software or if it's better to buy
  - o Help in cost and duration implication evaluation related to software decisions
  - o Help in decision making related to risk management

The problem arises when a company has the desire to appraise their software in $value terms but has no way to objectively calculate it. Function Point Analysis, amongst other tools, provides the foundation to logically calculate the first variable in appraising software - the cost related to developing it.

Introduction

Function Point Analysis is a method to measure a logical view of the software. It quantifies the functional size, as determined by the functional requirements. The method was developed at IBM in the 1970's and it is standardized, maintained and enhanced nowadays by IFPUG (The International Function Point Users Group).

Software Appraisal Method

When appraising a software application, it is important to express the monetary value as a range, using the cost to develop the application as the 'floor' (minimum value) and the results, the problems resolved and opportunities seized as the 'ceiling' (maximum value). Effort and cost are variables directly related to functional size. There are several estimation models that use functional size as an input to estimate effort or cost. For this article we will offer a simpler one: Cost = Functional Size x Delivery Rate x Person-Hour Value.

Value is something that can be perceived differently by any given entity. Therefore, we can say that this (business) value can be considered subjective. For example, a glass of water for someone who has been in a desert without drinking water for some time is more valuable in comparison to a person who has been swimming in a crystal clear river. An organization that is able to streamline an operational process by 50% with the use of said software, has a greater perceived value of this software than another organization that can only streamline the process by 5%. The added business value (ceiling) includes components such as *operational procedures* (flows that link different business functions), *quality levels* (number of defects), *performance and time to market levels*.

Keeping in mind that the perception of the added business value can be subjective, the floor of the appraisal (cost) based on its functional size is the only variable that can be objectively assessed to value the application. Other variables that can be calculated are the *quality levels* and *duration*. These two variables are considered under the discretion of the client as far as attributing a dollar amount to it.

Delivery Rates and Benchmarking for Software Development Projects

Once the application functional size is determined from its requirements [1][2], it is necessary to determine the delivery rate to develop an application like this one (of the same functional size) from the ground up. The best approach for this is to calculate the delivery rate using your own historical data from past projects. However, sometimes there is not enough data for this. An alternative however, yet not as good as the first, is to use a benchmark source to find a delivery rate. There are several benchmark sources for software projects data: Gartner Group, ISBSG and books from Capers Jones are some of them.

For example, we can extract a delivery rate from the International Software Benchmarking Standards Group (ISBSG) – dataset R11 [3] using the following fields as parameters for a given project:

a. Quality of Data: Records classified with insufficient quality were excluded.

b. Type of Count Used: IFPUG, NESMA

c. Functional Size: Not Null

d. Level of Effort: Not Null

e. Project Year: Post 2002

f. Type of Development: New Development

g. Primary Programing Language: JAVA

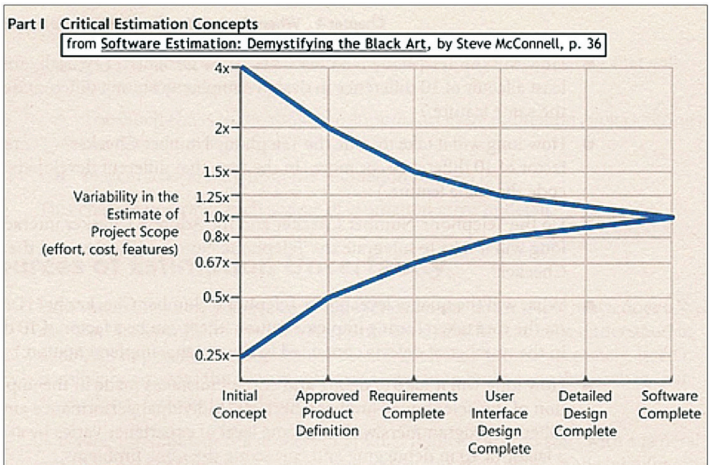
Using the percentage frequency of the benchmark information, the *Delivery Rate* was calculated to be at 14 Person-Hours per Function Point (PH/FP) with an 80% confidence factor that this number will not be underestimated.

The Delivery Rate alongside the functional size would theoretically allow for the calculation of the required effort (Required Effort = Function Points *Delivery Rate) to deliver

(continued on next page)

(continued from page 7)

the software. However, since this delivery rate was estimated, uncertainty had to be considered in the calculation. This was considered by both the Cone of Uncertainty by COCOMO II and the Project Evaluation and Review Technique (PERT). COCOMO II is an estimation model that allows for the estimation of cost, effort and schedule [4]. Moreover, COMOMO II is supplemented by the Cone of Uncertainty which describes that throughout the project life cycle, the amount of uncertainty decreases as time passes. One example of this can be the uncertainty that new requirements might suddenly 'appear', this is also known as scope creep.



Using the Cone of Uncertainty, a respective variable range needs to be selected in accordance to the stage of the life cycle that the project is in. If this information is not readily available, it can be determined by the level/detail of documentation provided by the client. The figure above represents the Cone of Uncertainty [5].

After the present ranges have been selected in the Cone of Uncertainty, the Project Evaluation and Review Technique (PERT) uses the selected ranges as inputs. The PERT formula is a method that was developed by Booz, Allen and Hamilton in the 1950s that allowed for more precise duration forecasting. It accomplishes this by calculating the average of three factors: the nominal estimate, pessimistic estimate and optimistic estimate [6].

Weighted Average for Project Estimates

Weighted Average = [Optimistic Estimation + 4(Nominal Estimation) + Pessimistic Estimation]/6

The optimistic and pessimistic estimation variables in the PERT formula above were replaced by the corresponding value of the range of the Cone of Uncertainty times the nominal estimation.

Once the PERT weighted average was calculated, it was possible to calculate the cost of the application using as a reference the Person-Hour value as defined by the client.

Cost of Application = Weighted Required Effort * Person-Hour Value

However, this result represents the floor value for the appraisal of the application as it was mentioned at the beginning of this article. The other variables that indirectly contribute to value were also calculated using the Functional Size as inputs. According to Estimating Software Costs, duration can be calculated with the following formula:

$$Duration\ (months) = Scope^{FP\ Constant}$$

Scope

In this case, Scope refers to the Functional Size count of the application. The FP constant, also provided by *Estimating Software Costs* [7], was selected from a list with different constants per type of project (commercial package was chosen). With these two variables, duration to develop a similar application was calculated. With this kind of information, any given client has the ability to recognize how long it would take to develop an application like this one and perform an opportunity cost analysis to see if it's worth using its resources to develop it.

Quality Level

Quality Level is another indirect component that can be used in the opportunity cost analysis. This variable can be defined in terms of Defects per Function Point (Defects/FP). Since *Quality Level* is affected by the type of development methodology being used to develop the application, the book *Applied Software Measurement – Global Analysis of Productivity and Quality* [8] was used as a reference. The Potential Defects and Defects Delivered constants were used for software with the same maturity level of the application being measured, which in this case was CMM Level 5.

Potential Defects = 5.5 * Functional Size

Defects Delivered = 0.22 * Functional Size

With this information, any particular business client interested in developing this kind of software application will have to consider the cost related to fixing said defects even after the application has been developed.

Function Point Analysis, as seen in the scenario above, can be considered more than just a sizing technique when used with other tools. Using benchmarking and historic data, FPA can be used to logically estimate the cost, duration and quality levels of any software to eventually attribute both a direct (cost) and indirect (business value) dollar amount to it.

REFERENCES

[1] Counting Practices Manual, International Function Point Users' Group (IFPUG). 4.3.1.

[2] Function Point Analysis For Software Enhancement Guidelines, NESMA, 2.2.1.

[3] International Software Benchmarking Standards Group (ISBSG). 2009. New Development & Enhancements Repository R11. http://www.isbsg.org.

[4] Software Cost Estimation with COCOMO II, Boehm, B.W. and Abts, C. and Brown, A.W. and Chulani, S. and Clark, B.K., 9780137025763, 2009, Prentice Hall.

[5] Software Estimation: Demystifying the Black Art (Developer Best Practices), Steve McConnell, 9780735605350, 2006, Microsoft Press; 1 Edition.

[6] A Guide to the Project Management Body of Knowledge: PMBOK(R) Guide, Project Management Institute, 9781935589679, 2013, Project Management Institute.

[7] Estimating Software Costs, Caper Jones, 9780070659490, 2007. McGraw-Hill Education.

[8] Applied Software Measurement: Global Analysis of Productivity and Quality, Caper Jones, 9780071502443, 2008, McGraw-Hill Education.

Partners' World!

David Consulting Group

Pennsylvania, USA

David Consulting Group (DCG) is a global provider of software analytics, software quality management and Agile development solutions. As the industry's leading independent provider of software sizing, we offer both Function Point Analysis and SNAP sizing services.

Since opening its doors in 1994, we have earned and maintained a reputation in the industry for our ability to help companies achieve their IT-related goals – and for our knowledge of how to do so quickly and effectively. Our consulting expertise enables us to position IT organizations for high quality, on-time and on-budget delivery of software through all phases of the development lifecycle. Our goal is always to improve our clients' bottom line via quantifiable changes in their software development.

Our clients span industries, regions, and size, including the Fortune 100 and Global 1000. Large enterprises, growing mid-cap companies and emerging technology leaders all rely on our expertise in helping them optimize their software production.

DCG maintains a U.S. corporate office in Malvern, P.A., and a European corporate office, DCG-SMS, in the U.K.

For more information, visit www.davidconsultinggroup.com or call 610-644-2856.

CHARISMATEK Software Metrics

Melbourne, Australia

Created in 1991, and now well established as a significant contributor across the world and as the hub of an international network of Metrics Partners, CHARISMATEK Software Metrics provides the highest level of software metrics and measurement based consulting & training services and products & tools to our clients.

At CHARISMATEK, we focus on using software sizing, metrics and quantitative analysis as a pragmatic and objective basis for addressing specific business issues. Real business and IT experience, in conjunction with unparalleled technical expertise, means that CHARISMATEK Software Metrics can provide you with practical and realistic assistance in a range of software and IT areas.

- Project Estimation - to assess the validity of budgets and schedules within a proposed business case or to determine a project's risk profile
- Scope Management - to clarify and negotiate software deliverables at project initiation and to track and control change throughout delivery
- Value for Money Assessments - to ensure you continue to receive value from your software delivery and support suppliers
- Contract Management - to devise project budgets and undertake software portfolio assessments
- Function Point Analysis - to determine the size of your software projects and applications.

CHARISMATEK publishes a high quality toolset for Function Point Analysis - the Function Point WORKBENCH. The foundation for any detailed count recorded in the WORKBENCH is a functional model which represents the software system. This model expresses the software graphically and allows the interrelationships between the software functions to be recorded and illustrated. By associating the details of a Function Point Analysis with a model of the software system, the WORKBENCH provides vivid and graphic support for the counting process. In addition, size can be approximated from physical artefacts using the Approximator module. Download a product brochure from http://www.charismatek.com/_public4/html/fpw_brochure.htm.