

DEVELOPMENT OF A MEASUREMENT TOOL FOR PHP LANGUAGE WITH MYSQL DATABASE

Chaknarin Kongcharoen* and Pichayotai Mahatthanapiwat

School of Computer Engineering Suranaree University of Technology
Nakhon Ratchasima 30000, Thailand

ABSTRACT

The estimate of effort and schedule for Web development is a difficult task because there are many Web components. Function points are used to quantify the amount of functionality of a software product based on its requirement. However, they are widely used for general programming languages. This paper shows how to apply them with PHP languages with MYSQL Database in order to quantify the functionality for Web development. After calculating the size of the program in SLOC, we can use COCOMOII to estimate the effort and schedule length.

KEYWORDS : Function Point Analysis, Source lines of code, COCOMO II , Web Component

1. INTRODUCTION

Function Point Analysis (FPA)[1][3] is an attempt to overcome difficulties associated with Source lines of code (SLOC) as a measure of software size, and to assist in developing a mechanism to predict effort associated with software development. Practically, we will translate FPs into Source lines of code and use another model (e.g., COCOMO II) that uses SLOC as the independent variable to derive estimates of schedule.

This paper describes such a process which was modified for a web development from PHP Language with MYSQL database environment. FPA was used to produce a level of effort estimate in terms of SLOC, which can be add a web component (e.g., picture , multimedia file) to FPs components. The Constructive Cost Model II (COCOMO II) was used with FPA results to derive an estimate of resource requirements, which can be refined to account for complexities within the development environment.

2. FUNCTION POINT WITH WEB COMPONENT ANALYSIS

Function Points with Web Component Analysis are an extension of function points. Web components in the Web applications are central to the size of such applications. Table 1 shows each of the five FPs and one Web component category with the associated weight factors representing level of complexity.

2.1 Function Point with Web Component Categories

The base of our estimate are still function points. To extended them, we have shown that they still can be used to accurately estimate the size of web applications[4]. To do this, a Multi-Media Files need to be added to take the new factors that drive the size of web applications into account. To count Function Points with Web Component, we evaluate the following six components of a web system based upon user requirements and page layouts:

Corresponding author. Tel: 0-1975-7856

E-mail: csnerk@ku.ac.

Components	Functions Levels			Notes
	Low	Average	High	
Internal Logical Files (ILF)	7	10	15	Unadjusted function points computed using these standard weightings using IFPUG counting conventions
External Interface Files (EIF)	5	7	10	
External Inputs (EI)	3	4	6	
External Outputs (EO)	4	5	7	
External Inquires (EQ)	3	4	6	
Multi-Media Files (MMF)	1	2	4	Multi-media files (graphics , animation , audio , video , etc.) -JPEG.GIF (Low) -A2b music, Microsoft picture (Average) -PCX Image, XIF: Image, AIFF Audio, Liquid Audio, Streaming Audio/Video (High)

Table 1. Function Point with Web Component Factors

2.2 Function Point with Web Component Calculations

Sample FPs calculations are shown in Table 2. To translate FPs into SLOC, a SLOC estimate is provided for each FPs category to take into account category-specific requirements common in PHP Language with MYSQL database development. It should be noted that the SLOC values presented in the table were derived from the case study of PHP Language with MYSQL database development projects into Source Lines of Code per Function point (SLOC/FP). The values of SLOC/FP should not be used as absolute guidelines until the values can be calibrated to a much larger sample of case study. It should also be noted that the definition of SLOC used is expected by COCOMO II. If a different parametric model had been used, the definition of SLOC would have to be modified accordingly.

The Function Point with Web Component Calculations are show in Table 2.

2.3 COCOMO II

COCOMO II provides an estimate of the effort and schedule required to complete a software development project. The primary cost driver is Delivered Source Instructions (DSI). The SLOC estimate derived in the FPA can be used as the DSI estimate in COCOMO II calculations. Using this approach, we can convert the level of effort estimate in FPA (i.e., SLOC) into personnel and schedule requirements as shown in Table 3.

COCOMO II is composed of three submodels i.e., Application Composite Model, Early Design Model and Post Architecture Model. In this research, we will use Post Architecture Model and Early Design Model

These two models are used in the development of Application Generator, System Integration, or Infrastructure developments. The Post-Architecture is a detailed model that is used once the project is ready to develop and sustain a fielded system. The system should have a life-cycle architecture package, which provides detailed information on cost driver inputs, and enables more accurate cost estimates. The Early Design model is a high-level model that is used to explore of architectural alternatives or incremental development strategies. This level of detail is consistent with the general level of information available and the general level of estimation accuracy needed.[2]

Both the Post-Architecture and Early Design models use the same functional form to estimate the amount of effort and schedule it will take to develop a software project. These nominal-schedule (NS) formulas exclude the cost driver for Required Development Schedule, SCED. The amount of effort in person-months, PM_{NS}, is estimated by the formula:

$$PM_{NS} = A \times Size^E \times \prod_{i=1}^n EM_i \tag{Eq.1}$$

$$\text{where } E = B + 0.01 \times \sum_{j=1}^5 SF_j$$

The amount of schedule, TDEV_{NS}, it will take to develop the product is estimated by the formula:

$$TDEV_{NS} = C \times (PM_{NS})^F \quad \text{Eq.2}$$

where $F = D + 0.2 \times 0.01 \times \sum_{j=1}^5 SF_j$

$$F = D + 0.2 \times [E - B]$$

The value of n, the number of effort multipliers, EM_i , is 17 for the Post-Architecture model effort multipliers, EM_i , and 7 for the Early Design model. SF_j stands for the exponential scale factors. The values of A, B, C, and D are A = 2.94, B = 0.91, C = 3.67 and D = 0.28

Function	Type ^a	Value
Index Page Books Cart	EQ	6
	EI	17
	MMF	2
News	EQ	3
	MMF	1
News Admin	EQ	9
	EI	16
Books Detail	EQ	3
	MMF	1
Books Order	EI	13
Books Cart Admin	EQ	9
	EI	23
Book Database	ILF	21
Order Book Database	ILF	14
Mailing Database	ILF	14
News Database	ILF	14
Total unadjusted function point count		166
Source lines of code (SLOC) ^b		166*23
		3818

^a ILF = Internal Logical Files ; EIF = External Interface Files
 EI = External Inputs ; EO = External Outputs
 EQ = External Inquires ; MMF = Multi-Media Files
^b SLOC / UFP ≈ 24

Table 2. Calculation the Function Point Count

Simple Calculations
Size = KSLOC × $\left(1 + \frac{REVL}{100}\right)$: Percentage of Requirement Evolution and Volatility (REVL)
Size = 3.818 × $\left(1 + \frac{10}{100}\right)$; REVL = 5 = 4.008 KSLOC
$PM_{NS} = A \times Size^E \times \prod_{i=1}^n EM_i$ where $E = B + 0.01 \times \sum_{j=1}^5 SF_j$, A = 2.94, B = 0.91
$PM_{NS} = 2.94 \times 4.2^E \times \prod_{i=1}^n EM_i$ = 13.5 person-months
$TDEV_{NS} = C \times (PM_{NS})^F$ where $F = D + 0.2 \times [E - B]$, B = 0.91, C = 3.67, D = 0.28
$TDEV_{NS} = 3.67 \times (27.1)^F$ = 8.4 months

Table 3. Calculation the COCOMO II

3. CONCLUSIONS

Managers of software development projects need cost estimation tools to help plan and control costs. FPA and COCOMO were developed with general language environments in mind. PHP languages with MYSQL Database present significant opportunities for achieving development productivity improvements. Tools such as FPA can be extended with web component for using in these new environments and then used in conjunction with COCOMO II

The resulting tool is not only useful for projecting costs before a project begins, but also for assessing the sensitivities of effort and schedule due to changes in software development parameters (e.g., skill-level of programmers). Your estimates will improve as you become more familiar with the application and tools.

REFERENCES

- [1] Albrecht, Alan J. Gaffney and John E., Jr. **1983** Software Function Points, Lines of Code, and Development Effort Prediction: A Software Science Validation. *IEEE Transactions on Software Engineering*, 9(6), pp.639-648.
- [2] Boehm, Barry W., Abts, Chris, Brown, A. Winsor, Chulani, Sunita, Clark, Bradford K., Horowitz, Ellis, Madachy, Ray, Reifer, Donald and Steece, Bert. **2000**. *Software Cost Estimation with cocomoII*. New Jersey, Prentice-Hall.
- [3] Garmus, David, and David Herron. **2001**. *Function Point Analysis Measurement Practices for Successful Software Projects*. New York, Addison-Wesley.
- [4] Reifer, Donald J. **2002** Estimating Web Development Costs: There Are Differences. *Crosstalk*, pp.13-17