

An Overview of Time Estimation in the Appraisal of Completeness for Software

Yukyong Kim*†

Abstract

The completeness appraisal of software is performed in various forms, such as assessing the completion level in the development process, calculating the defect rate, estimating the development cost, and calculating the redevelopment cost. Along with this, the problem that is often dealt with is estimation of the development time. Even in a dispute over completeness due to delays in software development, issues of calculating an appropriate development time required to develop a delivery software or a development time required for change requests are often included in the appraisal request.

In this paper, we introduce the procedure and method for estimating the appropriate project time of software development so that the appraiser can be applied to the appraisal work for determining the completeness. The method is based on the manual for calculating the appropriate project period of software development project.

keywords : Completeness evaluation, SW development time, guideline, development cost, function point

1. Introduction

The appraisal of computer programs can be largely classified into similarity appraisal and completeness appraisal. The completeness appraisal identifies functional problems occurring in a developed computer program, reliability issues in unit work processes, or operational problems of the program in use. It is to judge the degree of completion and defects[1][2]. In most cases, the completeness appraisal of

software is performed in various forms, such as assessing the completion level in the development process, calculating the defect rate, estimating the development cost, and calculating the redevelopment cost.

Along with this, the problem that is often dealt with is estimation of the development time. If Man-Month(MM) is used to calculate the cost of software development, it is based on the development time. In addition, even in a dispute over completeness due to delays in software development, issues of calculating an appropriate development time required to develop a delivery software or a development time required for a change request are often included in the appraisal request.

* Division of Basic Engineering, Sookmyung Women's University

† Corresponding Author : Yukyong Kim
(email: ykim.be@sookmyung.ac.kr)

Received: 2020.05.20. Revised: 2020.06.19.

Accepted: 2020.06.19.

In this paper, we introduce the procedure and method for estimating the appropriate project time of software development so that the appraiser can be applied to the appraisal work for determining the completeness.

In accordance with Article 23 of the Special Act on Promotion of Information and Communication and Activation of Convergence, which came into effect on February 14, 2014, published by the Ministry of Science and ICT, criteria for calculating appropriate project time for software development projects by state agencies are specified. Accordingly, a “Manual for Calculating the Appropriate Project Period of Software Development Project” has been prepared and used so that software developers can easily estimate the project time.

This paper is organized as follows. Chapter 2 briefly describes the completeness of software, and Chapter 3 introduces a method and a procedure for estimating the time required for software development. Chapter 4 presents the method for estimating the time period required for software appraisal. Chapter 5 describes the conclusion.

2. Completeness appraisal for software

Completeness appraisal of computer program is performed to determine to what extent certain systems and programs can be viewed as completed, or where there is a defect. The completeness appraisal should present quantitative and qualitative results through the presentation of specific numerical values.

The official definition of a defect refers to a case where there is a problem that does not meet the normal conditions expected by the law or the party. In the case of a defect, the legal action is canceled or invalidated or the occurrence of any right is blocked, or any effects such as the occurrence of obligations, are different according to the laws and regulations [3].

Defects in the public SW business means flaws in the SW project that do not meet the normal condition of the output after the contract between the SW operator and the state agency is completed [3]. A defect in a computer program refers to “a flaw or defect that does not meet the normal condition of the product after the contract between the customer and the developer is completed” and the defect type is classified as follows in the case of the existing appraisal.

- Incomplete
- Non-operating or partial operation
- Partial motion, non-implementation
- Un-finished, un-implementation, flaw
- Partially produced or not manufactured
- Unfulfilled tasks
- Error (malfunction)

Examples of cases where you can request a completeness assessment are as follows [1]. It is the case that the software ordering company A has signed a contract with the developing company B to promote the project, but the software supplied by the B company cannot be used normally. Or, SI(System Integration) developer B has entered into a contract with an ordering company and proceeded with the

project, but the contract was terminated in the middle of the contract period. In the former case, program development has been completed, while the latter is in the case where development was interrupted.

An example of appraisal of development time is to determine how much time it takes to develop a disputed program or to revise according to a change. It is not performed independently, but is mostly included as part of the appraisal request. In this paper, we introduce the method and procedure for calculating the appropriate time for software based on the manual for calculating the appropriate project period for software development projects developed by the National IT Industry Promotion Agency [4].

3. Guidelines for estimating software development time

Actually, the proper project period of the software development to be ordered does not calculate a reasonable project time considering the software development workload, business environment, and restrictions, etc. but calculates the project period in consideration of the customary accounting processing time.

In general, the appropriate time for a software development project is estimated by comprehensively reviewing the Function point (FP), basic business data, similar business data, and other peculiarities according to the 'General Standards for Software Business Management Supervision'.

In this paper, we will explain how to calculate the appropriate development period based on the function point and discuss ways to use it in the appraisal work in appraisal of completeness.

The procedure for estimating the appropriate development period based on the function point defined in [4] is shown in Figure 1 below.

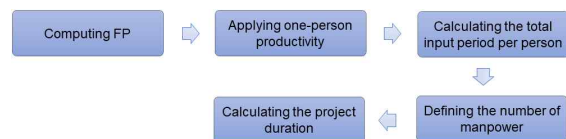


Fig. 1. The estimation procedure

3.1 Overview of function point

The function point method calculates the size of software development as a point for the identified function, and calculates the total function point by multiplying the unit cost and the correction factor per point. The function point is distributed annually through “SW Business Price Estimation Guidelines [5]” presented by the Korea Software Industry Association or “International Standards-Based Function Point Calculation Guide [6]” presented by the Korea Software Industry Association.

The function point is the amount of software function measured from the user’s point of view, and is a unit for measuring the scale of software functionalities provided to the user. Software functionalities are identified by referring to functional user requirements among development requirements as shown in Table 1.

Table 1. Requirements classification

Requirements	Function Point calculation
Functional requirements	Included in FP
Non-functional requirements	Excluded in FP (Identification target on adjustment)

The function point is a software scale measurement method that quantitatively calculates the functions requested by the user from the user's point of view and delivered to the user. It is an international standard for software size in ISO/IEC 14143. It is the most important factor in estimating the cost and resource requirements to develop, maintain, and operate software.

3.2 Procedure for estimating software development time

In [4], when calculating the appropriate project period for each project size, it is possible to secure the minimum project time, so as to reconsider the possibility of success in the ordered project.

Because the project time varies greatly depending on the project size, project characteristics, and ordering environment, it is necessary to take into account the business characteristics while using similar project data that was performed in the past.

The reference values for calculating productivity per person are also presented for step 2 of Figure 1. The unit of productivity

value per person is FP/MM.

After calculating the function point for each functional item, it is possible to estimate the time required for development by reflecting the productivity per person. Then, the total input period per person can be calculated as follows.

$$\text{Total input period per person (MM)} = \frac{\text{Project size}}{\text{Productivity per person}}$$

To compute the total business period using MM, it is needed the average number of manpower input per month which means the average number of people input per month based on the public and private software development projects. Using these values, the total business period required for software development can be calculated as follows.

$$\text{Total business period} = \frac{\text{Total input period per person}}{\text{Number of manpower}}$$

4. Time estimation in software appraisal

Time estimation is usually performed to define the number of hours a single developer needs to complete a particular task. As shown in Figure 2, software size is the most important factor influencing development effort and development time estimation. Therefore, the feasibility of estimating the development period has to depend on the validity of the estimated value of the measured software size. The most representative metrics for estimating software size are Lines of Code(LOC) and FP.

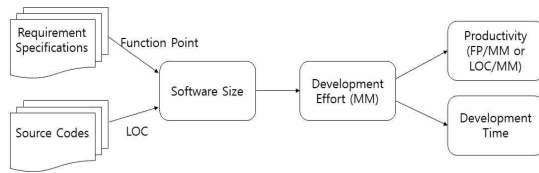


Fig. 2. Relationship between software size, development effort, and development time

As a study to estimate the development period based on the total effort to be put into development or the productivity, studies such as [7-10] have been conducted. However, since it is for a software development project, it is difficult to apply it directly to software appraisal.

In software completion evaluation, estimating the time period of completion can be applied to the same method as estimating the duration of software development. However, the guideline suggested in [4] cannot be applied to estimate the lead time for changes or additional requests. Therefore, the method for estimating the period must be defined differently according to the request for appraisal.

To estimate development time for requirements change during the implementation phase, we can use weights based on the type of change, as shown in Table 2.

In table 2, the term 'major grow modification' means that the change request requires an estimated three quarter additional code implementation to the original code. The term 'negligible modification' is that the change request does not require significant change to the original code.

The biggest problem with estimating the

development duration of requirement changes is that the size of the requirement changes cannot be estimated simply using LOC or FP. In software development, changing requirements includes not only adding new features, but also modifying and deleting them. It is difficult to estimate the scale for this change by applying LOC or FP as it is. In the case of modifications, it may have to change most of the code, or it can simply fix very few. Since all of these cases cannot be calculated uniformly, it is necessary to estimate the change size by weighting.

Table 2. Weights for the type of change

Change type	Weight	
Add	1.0	
Modify	major grow/shrink	0.75
	grow/shrink	0.5
	minor grow/shrink	0.25
	negligible	0.1
Delete	1.0	

For each change request, the change type is classified, and the weight for each change item is multiplied to calculate the FP.

5. Conclusions

The method of calculating the development period of the computer program introduced in this paper is a guideline that can be used in the completeness appraisal task and suggests the period estimation process step by step.

It is expected to be applied to both the appraiser's work performing the appraisal of completeness and as a guideline for related business personnel to be used in both the development cost evaluation and the appraisal of completeness.

References

- [1] K. T. Kwon, "Overview of the Appraisal of Completeness for Software", Journal of Korea Software Assessment and Valuation Society, vol.11, no.2, pp.1-8, Dec. 2015. [http://www.i3.or.kr/html/paper/2015-2/\(1\)2015-2.pdf](http://www.i3.or.kr/html/paper/2015-2/(1)2015-2.pdf)
- [2] K. T. Kwon, "Problem of SW Cost Estimation Guideline for the Appraisal of Software Development Cost", Journal of Korea Software Assessment and Valuation Society, vol.12, no.1, p.11-19, June 2016. [http://www.i3.or.kr/html/paper/2016-1/\(2\)2016-1.pdf](http://www.i3.or.kr/html/paper/2016-1/(2)2016-1.pdf)
- [3] Korea Copyright Commission, "Collection of cases of disputes such as SW copyright, vol. 1", Feb. 2011. <http://www.copyright.or.kr/information-materials/publication/research-report>
- [4] National IT Industry Promotion Agency, "Manual for Calculating the Appropriate Project Period of Software Development Project", July 2016. https://www.swit.or.kr/IS/web/isCbmRefView.jsp?ref_sq=662
- [5] Korea Software Industry Association, "SW Business Price Estimation Guidelines (2019 Revision)", June 2019. http://www.kdata.or.kr/info_05_download
- [6] Korea Software Industry Association, "International Standards-Based Function Point Calculation Guidelines", Nov. 2014. http://kdata.or.kr/info_05_download
- [7] S. Oligny, P. Bourque and A. Abran, "An Empirical Assessment of Project Duration Models in Software Engineering", Proceedings of the 8th European Software Control and Metrics Conference (ESCOM'97), 24-29; Berlin, Germany, May, 1997.
- [8] S. Oligny, P. Bourque, A. Abran and B. Fournier, "Exploring the Relation Between Effort and Duration in Software Engineering Projects", Proceedings of 16th World Computer Congress 2000 (WCC 2000), 175-178; Beijing, China. August, 2000. <https://pdfs.semanticscholar.org/416b/66f0c3969f12d922b6b3758795f17a71b928.pdf>
- [9] B. A. Kitchenham, "Empirical Studies of Assumptions That Underlie Software Cost estimation Models", Information and Software Technology, vol.34, no.4, pp. 211-218, April 1992. DOI: [https://doi.org/10.1016/0950-5849\(92\)90077-3](https://doi.org/10.1016/0950-5849(92)90077-3)
- [10] O. K. Harfoushi, "Software project scheduling techniques: A comparison study", International Review on Computers and Software, vol.8, no.3, pp.876-880, March 2013.

Authors



Yukyong Kim

2001.8 Ph.D. in Computer Science from Sookmyung Women's University.
 2005.9 - 2006.8 Post-doc. in University of California at Davis.
 2006.9 - 2013.9 Research professor in Dept. of Computer Engineering at Hanyang University.
 2018.2 - present. Faculty member in Division of Basic Engineering at Sookmyung Women's University.
 <Research interests> Web services, QoS Evaluation, Trust Assessment on SOA based IoT, Software Quality Metrics