A Comparative Study on the existing methods of Software Size Estimation

Manisha Vatsa [1], Rahul Rishi [2]
Department of Computer Science & Engineering,
University Institute of Engineering & Technology, Maharshi Dayanand University, Rohtak, India [1, 2]

Email: mvmanishavatsa@gmail.com [1] *, dir.uiet@mdurohtak.ac.in* [2]

Abstract- Software size estimation is an important phase in the software development life cycle. Software size is an important factor for the software development as it can predict the estimate cost and effort which are the key estimates that every client demands before starting the software development. There are various existing methods and techniques to estimate the software size before the start of software development phase. In this research work, a comparative review on the existing concepts of software size estimation is presented. The estimation of size of the software project ultimately helps in estimating the resources and time required which in turn is of utmost importance for the success of the project. Also, the major use of methods, concepts and parameters are highlighted.

Index Terms- Software Size Estimation; Software Development; Software cost; software effort & time; Software Development Life Cycle

1. INTRODUCTION

With the increasing demand of digitization, the need of software is also rises with advanced technology. These raises in software use also needs the software quality and size estimation before the start of software development process. Size estimation is the process used by an organization in order to forecast the cost for the development of a software project. All the estimation methods have to take as reference a software size metric. Software estimation is of prime importance in precisely controlling the various phases of the Software Development Life Cycle (SDLC). In SDLC, the initial process is requirement engineering stage to get all the estimates. According to Kotonya and Sommerville [1], requirements engineering stage consists of four stages of Requirements Elicitation, Requirements Analysis and Negotiation, Requirements Documentation and Requirements Validation. These steps are presented in figure 1.

As the software evolves during the life cycle, there should be some control on the product to be developed. Hence, to control the product, estimation should be performed prior to development. Estimation can be explored and determined based on the internal and external attributes. Internal attributes of a software project can be estimated in terms of size, complexity, and modularity. External attributes usually denote properties that can be measured by taking into account how this project is related to the environment. Examples of the external attributes are reliability, understandability, maintainability, etc. External attributes depend on their environment and they are very difficult to quantify. For that reason, size, which is an internal attribute, has been investigated in this work, to establish a model for estimation.

Estimation typically produces projections of the software size, effort, schedule, and quality required to complete the project. These estimates form the basis for initial project plans and subsequent re-planning. Generally, the same factors that are estimated need to be tracked in order to manage a project to a successful conclusion. Changes or mistakes in the assumptions driving the plans must be recognized as early as possible so that corrective action can be taken.

There are various methods related to line of code, function point analysis and use case modeling for the software size estimation. In this research work, existing concepts related to software size estimation are presented. This section has presented the basic software size estimation process. Next section presents the software size estimation models. Section 3 presents literature review on the existing software size estimation methods. Section 4 concludes the paper along with some future directions.

2. SOFTWARE SIZE ESTIMATION MODELS

Currently, many different functional methods and programming languages are in use. The natures of the method and language affect the availability of specific measures.

However, some measurable entities (e.g., procedures, functions) are common to all approaches. Getting a good size estimate is essential to getting good estimates of effort, schedule, and quality. The first step in size estimation is to select a size measure. Metrics are the powerful support tools in software development and maintenance. These are used to access software quality, to estimate complexity, cost, and effort, to control and improve processes. Typically, estimation begins by projecting the amount of software to be produced. Software size may be estimated in terms of Functional size (the amount of capability to be delivered), Physical size (the amount of software product to be delivered), and both [2]. These size estimation methods are discussed here:

2.1. *Physical size estimation*

Line of Code: The Line of code is the oldest, simplest and most widely used metrics for calculation of program size. It counts the "Number of Instructions‟ of a program excluding comments and blank lines in terms of SLOC (Source Lines of Code). SLOC is a key input for estimating project effort and is also used to calculate productivity and other measurements.

2.2. *Functional size estimation*

Function Point analysis (FPA): Function Point analysis is one of the best methods in traditional metrics for measuring the size of software. Function points represent logical size, as opposed to physical size (like SLOC or objects). Metrics are the powerful support tools in software development and maintenance. These are used to access software quality, to estimate complexity, cost, and effort, to control and improve processes.

Use case modeling (UCM) generally provides the complete information about all the functionality, addressed by any given software system. It also has certain distinct advantages in capturing the system requirements earlier, in the software life cycle. Further, it captures different granularities such as brief, fully specified and refinements in the requirements analysis phase in detail. UCM exhibits the various patterns of behavior that are adequate in capturing the system testing activities. Activities related to system testing can be estimated using UCM.

3. SOFTWARE SIZE ESTIMATION EXISTING TECHNIQUES

This section presents the literature survey of development size estimation approaches. Literature survey based on the development size estimation has been explored at the analysis and design phases. Further, table 1 presents the brief about the existing concepts.

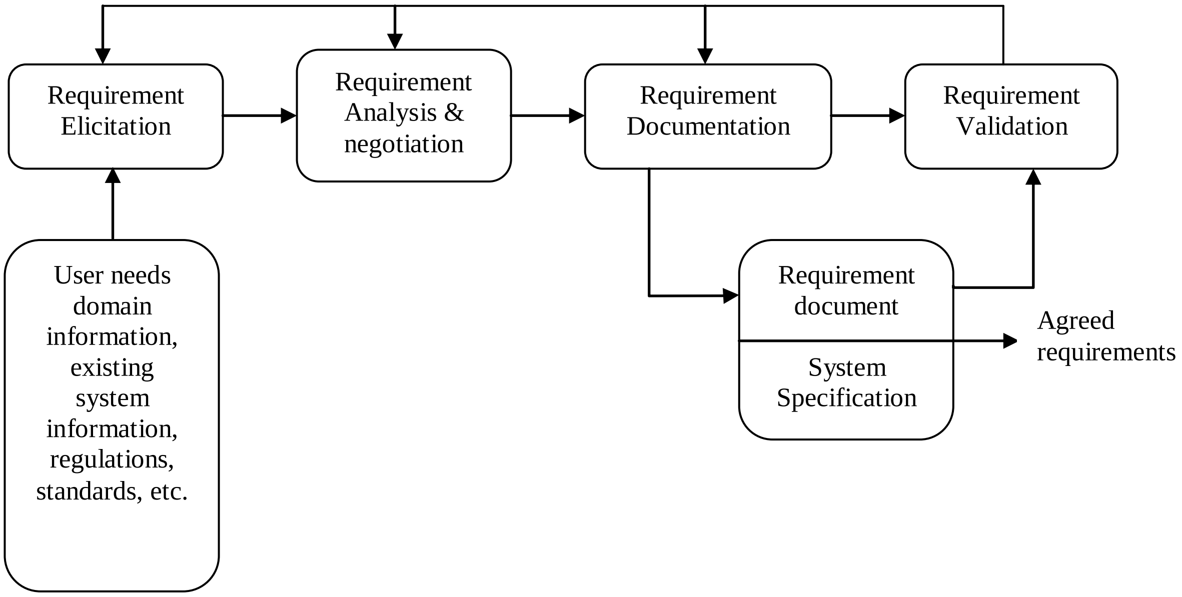
Requirement Elicitation
Requirement Analysis & negotiation
Requirement Documentation
Requirement Validation
User needs domain information, existing system information, regulations, standards, etc.
Requirement document
System Specification
Agreed requirements

Figure 1. Requirement Phases

Authors (Sholiqa et al., 2017) [3] have presented the comparative analysis based on the concepts of Function Points (FP) and Use Case Points - Activity Based Costing models (UCPabc). Authors compared the concepts based on deviation, complexity factors, process, parameters, Function Points (FP). Authors have observed better deviation for the UCPabc in comparison with FP. Another thing reported by authors was that FP have only 14 complexity factor whereas UCPabc have 7 team environment factors are 14 technical factors.

Authors (Zhou and Huang, 2016) [4] summarizes existing scale estimation methods in software engineering field, gives a proper understanding to the research goals and the basic requirements, and proposes a new business process testing size estimation method. The Science Center of the international corporation mission SVOM adopts an iterative development approach. The development process is divided into developing three systems: the basic system, the core system and the complete system. Current stage is the basic system, where the development of an optical payload data processing system focuses on correctness of basic data path. Therefore, the strategy of the User Acceptant Test is to cover certain critical and typical business scenarios. In order to reasonably estimate software functional testing cost before test implementation and fully use the resource efficiently during testing process, this paper studies the methods of software functional testing size of this system for the User Acceptant Test.

Authors (Sabahat et al. 2016) [5] have used the step forward regression based parametric model for the prediction of software size estimation of board based desktop games. To access the model, authors have used prediction accuracy matrices like MMRE, PRED (x), MdMRE etc. For the validation of project, K-fold cross validation framework has been considered. For the prediction of software size during this model, authors have provided the inputs like number of rules, miscellaneous game options, number of players and animation etc and obtained the output results as functional points which is further converted into source lines of code. For this analysis, dataset of 65 open source board-based desktop games is used.

Authors (Nassif et al., 2016) [6] present some techniques using fuzzy logic and neural networks to improve the accuracy of the use case points method. This paper introduces a new approach to overcome the absence of the graduation limitation of the UCP when classifying the complexity of the use cases. First, rather than classifying a use case as simple, average, or complex, the use case will be classified as u, such as x ∈ [1,10] where x represents the number of transactions. The proposed approach was implemented in two independent stages. First, a fuzzy logic approach is applied to determine the complexity factor of ux. The second stage of the proposed approach is implemented through a neural network model. The output of the neural network will be the size of the software. Results showed that an improvement up to 22% can be obtained using the proposed approach.

In order to streamline size measurement and effort estimation, a study (Ayyildiz, 2016) [7] has been conducted that exploits the correlations between the problem domain measures such as the number of distinct nouns and distinct verbs in the requirements artifacts and the solution domain measures such as the number of software classes and methods in the corresponding object oriented software. In this case study, 12 commercial software projects are analyzed and multiple regression analysis is carried out to develop an estimation model for the solution domain metrics in terms of problem domain metrics. The results suggest that, for the projects examined, it is possible to use problem domain measures to make plausible predictions for the solution domain metrics.

An approach has been proposed (Kiewkanya, 2016) [8] for constructing C++ software size estimation model using a statistical technique called Multiple linear regression analysis. The proposed model is constructed from structural complexity metrics that can be measured from class diagram. The paper also presents an automated tool for measuring these metrics, and in effect, estimating the C++ software size.

Authors (Moeyersoms, de Fortuny et al. 2015) [9] are of the opinion that the predictive models not only need to be accurate but also comprehensible, demanding that the user can understand the motivation behind the model’s pre-diction. Unfortunately, to obtain predictive performance, comprehensibility is often sacrificed and vice versa. To overcome this problem, they extract trees from well performing Random Forests (RFs) and Support Vector Machines for regression (SVRs) making use of a rule extraction algorithm.

In this paper (Du, Ho et al. 2015) [10], an approach combining the neuro-fuzzy technique and the System Evaluation and Estimation of Resource Software Estimation Model (SEER-SEM) effort estimation algorithm is described. This implemented model possesses positive characteristics such as learning ability, decreased sensitivity, effective generalization, and knowledge integration for introducing the neuro-fuzzy technique. The performance of the implemented model is accessed by designing and conducting evaluation with published projects and industrial data. The evaluation results indicate that estimation with implemented neuro-fuzzy model containing SEER-SEM is improved in comparison with the estimation results that only use SEER-SEM algorithm.

Bajwas et al. (2014) [11] presented the results of a systematic mapping study on software size estimation metrics and methods. The methods are investigated based on the entity type(s) measured by the method, size attribute type, size measure(s), and theoretical/empirical validation status of the method, and restriction with respect to functional domain type(s) and software development methodology applicability.

This paper (Liu, 2014) [12] takes the excellent Function Point Analysis (FPA) method as the prototype. The development circumstance combines with Object-Oriented (OO) method. In the experiment of the implement section, the Unified Modeling Language (UML), which can integrate with class diagram and FPA, is chosen as a method. As a result, the considered method for the effect of estimating function points (FP) has been improved through the verifying. Furthermore, the automatic estimation model, which is constructed from UML class diagram to FP, is put forward in this paper. The obtained results support the conclusion that automatic estimation of FP is designed and implemented, and a kind of Web Service is developed based on UML.

In a study (Alves, 2014) [13] model based on Function Points Analysis (FPA) to estimate the size and complexity of software system. This paper describes a case study with two teams that developed a software system (Web application) for a real customer. The research uses Function Points Analysis, Use Case Points (UCP) and LOC (Lines of Code) methods.

Among all the software estimation methods, function point method, COCOMO II, Pert, Delphi and Analogy are widely used. A research paper (Ren, 2013) [14] gives advice on how to choose estimation methods. If software enterprises have enough capital to hire high-level estimate experts, Delphi can be considered. Function point method should be chosen first if the software project is in early life cycle. Pert is relatively suitable for the latter life cycle as a supplementary means. COCOMO II uses source lines of code or function points as input and calculates effort, duration and cost, which provides the most direct guidance for companies. When estimating a software product, a variety of estimation methods may be used at the same time to improve the accuracy of software estimation.

Table 1. Comparative analysis of existing techniques

Author & year	Considered Research Work	Remarks
Sholiqa et al., 2017	Compared the concepts of Function Points and Use Case Points - Activity Based Costing models	Authors have only discussed about the considered comparison based on deviation, complexity factors, process and parameters but there is no proper explanation of concepts.
Zhou and Huang, 2016	Space multi-band Variable Object Monitor (SVOM) approach	Need further improvement of work in terms of software size estimation. Also need to develop the SVOM for the real time size estimation.
Sabahat et al. 2016	Step forward regression based parametric model for the prediction of software size estimation and For the validation of project, K-fold cross validation framework has been considered.	Estimation process is limited to desktop game size. Not investigated for the size estimation of mobile games. Dataset of only board games is used.
Nassif et al., 2016	Fuzzy logic and neural networks have been used to improve the accuracy of the use case points method for size estimation.	Need of more transactions per use case and weight complexity need further enhancement.

Ayyildiz, 2016	Pearson's correlation coefficient and applied multiple regression analysis on data obtained from 12 commercial software projects	Domain specific results obtained. Diversity of software projects is required for actual software size estimation.
Kiewkanya, 2016	Research investigates metrics captured from class diagram to determine which metrics can be used to accurately estimate C++ software size. The estimation model is built using Multiple linear regression analysis	Small code line samples used for testing of size estimation with only class diagram.
Moeyersoms, de Fortuny et al. 2015	In this research Software fault and effort prediction are considered to minimize costs of a software project using trees (using C4.5 and REPTree).	Scope of more improvement in the result accuracy.
Du, Ho et al. 2015	In this paper, a hybrid intelligent model combining a neural network model integrated with fuzzy model (neuro-fuzzy model) has been used to improve the accuracy of estimating software size and cost.	Tested on limited no. of projects
Bajwas et al. 2014	Systematic mapping study on software size estimation metrics and method. Parameters evaluation with respect to functional domain type(s) and software development methodology applicability	Work reviewed was limited to some specific parameters of functional and software domains.
Liu, 2014	Designed Function Point automatic estimation model based on UML class diagram. The development circumstance combines with Object-Oriented (OO) method.	Lacking the adjustment of industry data; extracting the history empirical data; the feedback processing work to do the further careful research, in order to strengthen the serviceability of this model.
Alves, 2014	Function Points Analysis (FPA) to estimate the size and complexity of software system. This paper describes a case study with two teams that developed a software system (Web application) for a real customer	Work is need to evaluate with some standard parameters like MRE, MMRE
Ren, 2013	Presented a review approach on the existing software estimation techniques and presented the evaluation parameters on the aspects that how to choose estimation method.	Not implemented on real time project.

4. CONCLUSIONS

The process of estimating the size of software projects is extremely important, so that each organization before starting the production or development of a software first evaluates the required parameters to evaluate the software size according to human resources and existing facilities. In this research paper, different available techniques and methods are discussed. Software size estimation is crucial for the evaluation of parameters like cost estimation. There are a variety of size estimation models and methods that each of them has its strengths and weaknesses. Some of the available can be used for the estimation of size along with software cost and effort but the existing concepts can be further improved with the use of optimization based meta-heuristic concepts. In future meta-heuristic concepts can be used to optimize the results.

REFERENCES

[1] Kotonya, G., & Sommerville, I. (1998). *Requirements engineering: processes and techniques*. Wiley Publishing.

[2] Laranjeira, L. A. (1990). Software size estimation of object-oriented systems. *IEEE Transactions on software engineering*, *16*(5), 510-522.

[3] Dewi, R. S., & Subriadi, A. P. (2017). A Comparative Study of Software Development Size Estimation Method: UCPabc vs Function Points. *Procedia Computer Science*, *124*, 470-477.

[4] Zhou, L. X., & Huang, M. (2016). Elementary research on software functional testing size estimation for space astronomical optical payload data processing system. In *14th International Conference on Space Operations* (p. 2323).

[5] Sabahat, N., Malik, A. A., & Azam, F. (2017). A Size Estimation Model for Board-Based Desktop Games. *IEEE Access*, *5*, 4980-4990.

[6] Nassif, A. B., Capretz, L. F., & Ho, D. (2016). Enhancing Use Case Points Estimation Method Using Soft Computing Techniques. *arXiv preprint arXiv:1612.01078*.

[7] Ayyildiz, T. E., & Koçyigit, A. (2016, August). A Case Study on the Utilization of Problem and Solution Domain Measures for Software Size Estimation. In *Software Engineering and Advanced Applications (SEAA), 2016 42th Euromicro Conference on* (pp. 108-111). IEEE.

[8] Kiewkanya, M., & Surak, S. (2016, July). Constructing C++ software size estimation model from class diagram. In *Computer Science and Software Engineering (JCSSE), 2016 13th International Joint Conference on* (pp. 1-6). IEEE.

[9] Moeyersoms, J., de Fortuny, E. J., Dejaeger, K., Baesens, B., & Martens, D. (2015). Comprehensible software fault and effort prediction: A data mining approach. *Journal of Systems and Software*, *100*, 80-90.

[10] Du, W. L., Ho, D., & Capretz, L. F. (2015). Improving software effort estimation using neuro-fuzzy model with SEER-SEM. *arXiv preprint arXiv:1507.06917*.

[11] Bajwa, S. S., Gencel, C., & Abrahamsson, P. (2014, October). Software Product Size Measurement Methods: A Systematic Mapping Study. In *Software Measurement and the International Conference on Software Process and Product Measurement (IWSM-MENSURA), 2014 Joint Conference of the International Workshop on* (pp. 176-190). IEEE.

[12] Liu, L., Yang, A., Tao, Q., Zhu, L., & Wu, D. (2014, July). Study of the software size estimation model based on UML. In *System Science and Engineering (ICSSE), 2014 IEEE International Conference on* (pp. 138-142). IEEE.

[13] Alves, L. M., Oliveira, S., Ribeiro, P., & Machado, R. J. (2014, June). An empirical study on the estimation of size and complexity of software applications with function points analysis. In *Computational Science and Its Applications (ICCSA), 2014 14th International Conference on* (pp. 27-34). IEEE.

[14] Ren, A., & Yun, C. (2013, November). Research of Software Size Estimation Method. In *Cloud and Service Computing (CSC), 2013 International Conference on* (pp. 154-155). IEEE.