ISSN:2320-0790

COMPUSOFT
An International Journal of Advanced Computer Technology

Software Metrics for Identifying Software Size in Software Development Projects

V.S.P Vidanapathirana[1] and K.H.M.R Peiris[2]

[1,2] Faculty of Information Technology, Sri Lanka Institute of Information Technology, Colombo, Sri Lanka

Abstract: Measurements are fundamental any engineering discipline. They indicate the amount, extent, dimension or capacity of an attribute or a product, in a quantitative manner. The analyzed results of the measured data can be given as the basic idea of metrics. It is a quantitative representation of the measurements of the degree to which a system, component, or process possesses a given attribute. When it comes to software, the metrics are a wide scope of measurements of computer programming. The size oriented metrics takes a main role in it since they can be used as the key for better estimations, to improve trust and confidence, and to have a better control over the software products. Software professionals traditionally have been measuring the size of software applications by using several methods. In this paper the researchers discuss about the software size metrics for identifying software size and it is mainly focused on the software development projects in today's Information Technology (IT) industry.

Keywords: Engineering discipline, Software, Software Metrics, Software Size, Programming, Software Development Projects

I. Introduction

Frequently the world has been experiencing a software crisis as a result of the inability in producing high quality and reliable software due to the lack of software management capabilities. These software management capabilities include the accuracy or the improvements of the software metrics and the utilization of such metrics.
Software metrics are tools for anyone involved in software engineering to understand varying aspects of the code base, and the project progress [1]. Using software metrics is different from testing errors, since they can provide a wider variety of information about the aspects like quality, schedule, cost, complexity or the size of a software system. When looking at the current state of software metrics, there are many metrics invented for all the categories in software metrics field such as product metrics, process metrics, resource metrics and project metrics. Each of these metric types have unique purposes, set of specifications and methodologies attached to them, therefore identifying the best and the most accurate resulting software metric has become sort of confusing.

Faced with this situation, the authors has chosen to indicate a great diversity of product software metrics, more specifically the software size metrics used in software development projects in the current IT industry.

Product metrics describes the characteristics of the software product at any stage of its development, from requirements to installed system [2], and software size is one of the most important and common characteristic that comes under product metrics type. It uses the either source or the object code of the particular program in order to measure the software size. It helps to determine and predict the quality of the current software product and identify the adjustments or techniques that can be used to level up or improve the quality of the product as well. Since the results gained from the product metrics are not personal biased, it gives the opportunity of comparing.

The rest of this paper is organized as follows. Section 2 has provided the existing related work. Section 3 describes the objectives of conducting this research and the methodology of this paper is discussed in Section 4. Result and discussion is
Given in Section 5 and finally Conclusion is presented in Section 6.

II. Literature Review

T.Manoharan et.al. have discussed that the source code metrics measure the size of a software program by counting the number of lines in the text of the program's source code and is also used to predict the amount of effort that will be required to develop a program, as well as to estimate programming productivity or effort once the software is produced. [3]

D.Beyer et. al. have discussed that two widespread-used size metrics for object oriented source code are the number of methods as indicator for functional size and the number of attributes as indicator for size of encapsulated data. [4]

A. Abran et al. have mentioned measuring the functional size of software was originally proposed in 1979 by Albrecht in a communication describing the result of an effort started in the mid-seventies in one IBM business unit. The overall goal of the work described in Albrecht's original paper was to measure the productivity of software development, as viewed from an economic perspective. The functional size of software was proposed as a generic measure of the "output" of the development process which allows a comparison of projects in which software was developed using different programming languages. [5]

E. E. Mills et.al. Have described a number of metrics attempt to quantify software "size." The metric that is most widely used, LOC, suffers from the obvious deficiency that its value cannot be measured until after the coding process had been completed. Function points and system Bang have the advantage of being measurable earlier in the development process—at least as early as the design phase, and possibly earlier. Some of stead's metrics are also used to measure software size. [6]

V. Tiwary et.al. Have discussed the complexity has a direct impact on overall quality, so complexity of the modules should be less. Size of the software reflects the complexity, development effort and reliability of the software. LOC (Lines of Code) or KLOC (Lines of Code in thousands) is an intuitive initial approach to measuring software size. LOC or KLOC is depends upon the factors such as blank lines, comments, executable statements etc. The reliability will decrease if modules have a combination of high complexity and large size. High complexity and small size will sometimes also decrease the reliability because; the smaller size results in a short code which is difficult to alter. For the object oriented code additional metrics are required to evaluate the quality of the software. Weighted method per class (WMC) is one of them such metrics which is used to predict how much time and effort is required to develop and maintain the class. [2]

N. E. Fenton et.al have mention that the Size Metrics are represented by a number of metrics attempt to quantify software "size". For a software application is easy to measure the number of lines of codes for quantify software size. We discuss here a little bit about some aspects of software size. Each product of software development is a physical entity. In this acceptation, it can be described in terms of its size. The most commonly used measure for the length of a code source of a program is the number of lines of code. [7]

M.H. Sherif et.al have discussed that code size and code complexity are two essential metrics to predict the development effort, fault-proneness analysis, and other risk management activities during coding phase. However, it is found that size and complexity metrics only become inadequate to describe the attributes of modern software. [8]

Software quality metrics are a subset of software metrics that focus on the quality aspects of the product, process, and project. In general, software quality metrics are more closely associated with process and product metrics than with project metrics. Nonetheless, the project parameters such as the number of developers and their skill levels, the schedule, the size, and the organization structure certainly affect the quality of the product. [9]

A. J. Albrecht et. al. proposed a function-oriented metric which has subsequently gained wide currency: the function point. Function points are computed using the experimental relationship between the direct measures of the software's information domain and estimation of its complexity on a weighted scale. The information domain values are based on the following criteria: number of user inputs, number of user outputs, number of user inquiries, number of files, and number of external interfaces. Once they are computed, function points are used in a manner similar to lines-of-code to normalize measures for software productivity, quality, and other attributes such as, errors per function point, defects per function point etc. [10]

Software size estimation is a critical issue in the project management area. Good early estimations are essential for a reliable prediction of project effort and cost as well as for an efficient planning and scheduling. Software size represents one of the most interesting internal attributes of a software product. Internal attributes can be measured in terms of the product itself, separate from its behaviour while external product attributes can be measured only with respect to how the product relates to its environment. The first ones, such as software size, are easier to measure than external ones. Software measurement presents the problem that it can only be carried out when the product is finished and then it is not very useful. For that reason, many metrics provide functional measures of the software size such as metrics of functions points, functions blocks, object points or Bang metrics Those variables obtained from software specifications should have a correspondence with the final product size expressed, for example, in lines of code. The recent class point approach based on design documentation is an alternative for object oriented products. [11]

III. OBJECTIVES

In order to identify and understand the specific objectives based on the topic "Software Metrics for identifying Software Size in Software Development projects", the authors have several objectives as follows.

- Identify the current usage of software size metrics for software development projects.
- Identify the current methodologies of software size metrics used in software development projects in IT industry.
- Analyse the advantages and disadvantages of software size metrics in software development projects.
- Identify the knowledge about software size metrics among the industrial individuals.
- Find the best and the most accurate software size metrics in software development projects.

IV. METHODOLOGY

Since questionnaire is one of the best methods that can be used to gather large amounts of information from a large crowd in a short period of time and in a relatively cost effective, practical manner the team decided to conduct an online questionnaire which includes questions to lead the way of finding the best software size metrics which are available in current IT industry. The questionnaire was consist of simple, clear and objective set of questions where the responsive parties can give their personal ideas about the software size metrics that they use and identify as the best metrics that estimate the software size in software development projects.

In order to become participants in the survey, the team contacted 50 selected individuals, who has experience in the roles of Project Managers, Quality Assurance Engineers, Software Engineers, Developers, Business Analysts, Interns, and as well as undergraduates in the IT related field. The online questionnaire that includes 11 questions were emailed to them to gather the response.

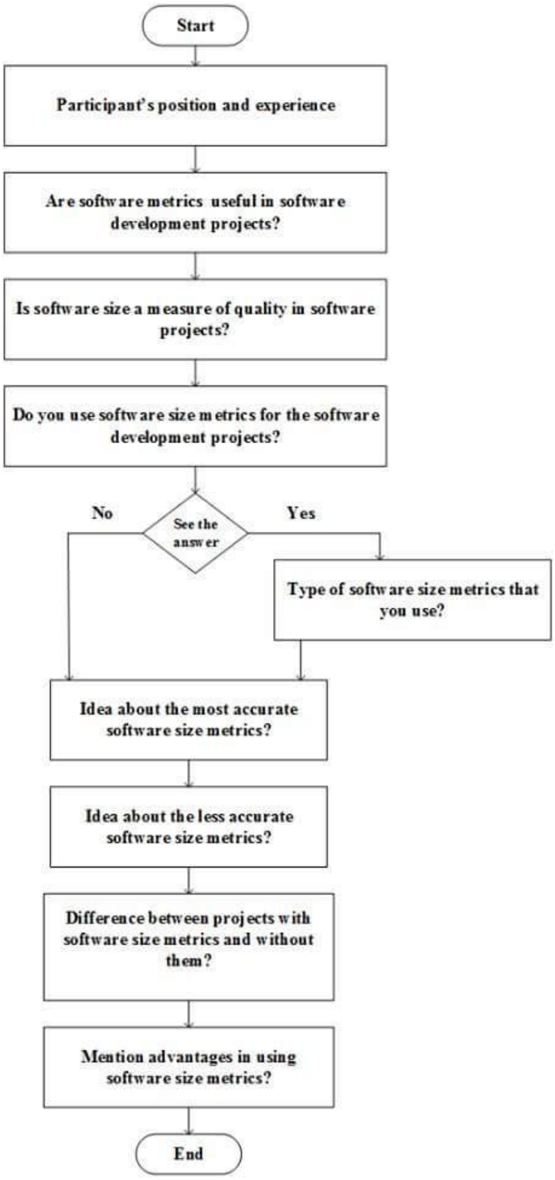
Start
Participant's position and experience
Are software metrics useful in software development projects?
Is software size a measure of quality in software projects?
Do you use software size metrics for the software development projects?
No
See the answer
Yes
Type of software size metrics that you use?
Idea about the most accurate software size metrics?
Idea about the less accurate software size metrics?
Difference between projects with software size metrics and without them?
Mention advantages in using software size metrics?
End

Figure 1: Flow chart of the initial study

Out of the 50 participants, 40 of them responded. Therefore the survey makes a success rate of 80%. After gathering the responses, the team analyzed and discussed those information to come up with a meaningful result, as well as a clear conclusion.

V. RESULTS AND DISCUSSIONS

The audience of the survey was purely IT industry oriented, because it was conducted among participants who are currently working on and worked with software development projects. Most of them were having experiences in the industry between 2 to 5 years, which makes a percentage of 42.5%. A percentage of 40% can be identified as the next highest, the participants who has industry experiences less than one year. The rest of the 7% have experiences in the industry for more than 5 years.

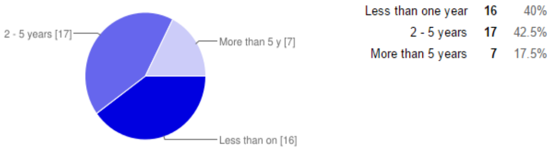

Figure 2: Years of experience

92.5% of industry participants responded that they think that software metrics are important in software development projects. The idea of “maybe” was given by 7.5% of them. No one responded that the software metrics are not important in software development projects.

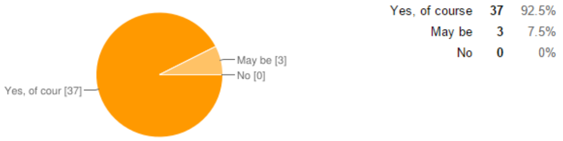

Figure 3: Importance of software metrics

Among the participants, 57.5 % of them responded that the idea about software metrics that measures the software size is important in software projects. 30% of them mentioned that it depends, and 12.5% of them responded that it does not important in software development projects.

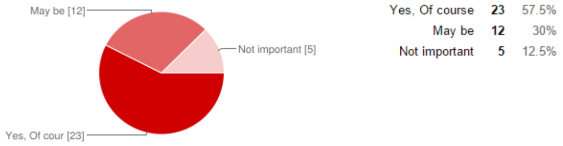

Figure 4: Importance of software size metrics

The usage of the software size metrics among the participants was 90%. 10% of them do not use software metrics that measures the software size in their software development projects.

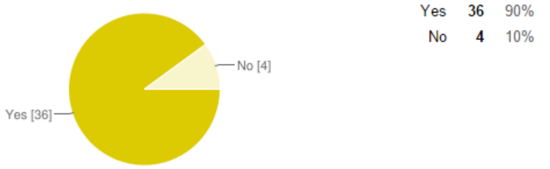

Figure 5: Usage of software size metrics

Among the participants who use the software metrics to measure the software size, the highest percentage of 72.5% goes with the software metric, Function point analysis. The next highest goes with a percentage of 67.5%, towards the Lines of code metric. A few of them, a percentage of 15% use Number of classes in a design diagram, and 12.5% of them use Object points. Feature points metric is used by 10% of the participants. None of them use number of boxes in a data flow diagram as a software metric, in the development of software products.

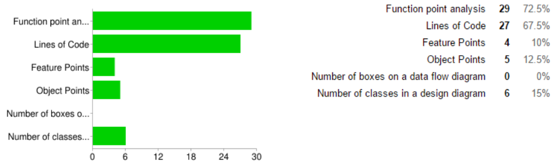

Figure 6: Types of software size metrics in use

As the most accurate software size metric, the participants have selected Function point analysis, which can be represented in a percentage of 62.5%. Lines of code were selected by 30% which makes it the next highest. Feature points made a percentage of 17.5% and for number of classes in a design diagram it was a percentage of 5%. A very few participants has selected number of boxes in a data flow diagram as the most accurate software size metric, and that percentage was 2.5%.

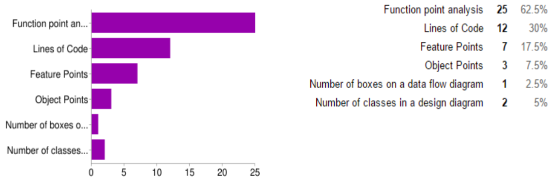

Figure 7: The most accurate software size metrics

From the participants, 55% think the number of boxes in a data flow diagram as the least accurate software size metric. 30% of them think it is lines of code, and for 17.5% it is object points. Feature points were selected as the next less accurate software size metric by making a percentage of 15%. A percentage of 12.5% think that the less accurate software metric is the number of classes in a design diagram. A very few of the participants think it is function point analysis, which can be represented in a percentage of 5%.

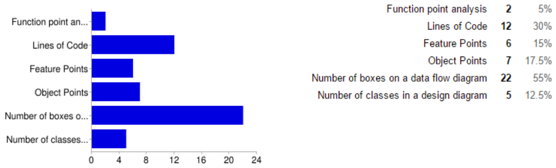

Figure 8: Least accurate software size metrics

Among the participants, 85% of them think that there is a difference in software products there are with software size metrics and 15% think there is no any difference in software products when they are without software size metrics.

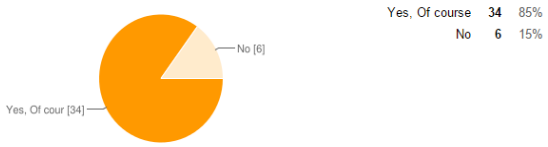

Figure 9: Difference between software products with metrics and without metrics

According to the responses of the industry participants for the questionnaire, software metrics play a huge role in software development projects and among them; there are metrics that can be used to measure the software size, to ensure the quality of the software products. At the end of the project they can be used to deploy a project with less bugs and errors so it makes them easy to maintain as well.

Based on the answers provided by the industry participants, the software size metrics the scope of the project can be calculated, and they are important to have a baseline measure about the code's efficiency. It makes easier when it comes to comparison between projects, and also is useful to estimate project resource estimations. When it comes to calculating the costs and financial purposes, the size of the software is important since it allows for better time and cost estimates for repetitive projects.

Software metrics which measure the software size make it easier to manage the risks in software developing projects, since they provide a good idea about cost estimations and financial states. In complex projects, they ensures the efficiency of the code. Therefore the quality of the software product is well assured.

Some of the industrial workers consider that the software size is not an important fact in software development projects. They consider it as a time consuming process which needs additional effort and time other than the developing time. And as the team realized, it requires more knowledge since some of the industrial workers are not much familiar with the metrics that can be used to measure the software size. On the other hand, since the software size is just a small factor in a software product, they think it does not make any difference even if they use software metrics to estimate the software size.

Some of the well experienced industrial workers do not believe that the software size metrics are important for the process of implementing quality software products. It is mostly because of the confidence they have by being in the IT industry for a long time. Based on their experience on development projects they believe software size estimation is not a considerable and important factor in achieving the success in those projects.

VI. Conclusion

The results gained from the conducted online survey revealed the usage of software metrics that are currently in industrial use, which organizations use to measure the software size in software development projects. It also showed the most accurate and least accurate software size metrics among them.

Clearly Function point analysis and Lines of code plays a massive role in software size metrics but when considering the accuracy, function point analysis method can be considered as the most accurate metric that can be used to measure the software size. Function point analysis does not consider about the programming language is being used or the platform in the software development project runs, therefore the assessing the productivity of them can be done in an easy and accurate manner. Lines of code metric is mostly suitable for the projects with very tightly constrained environments, but not for complex projects that involves many resources and efforts.

According to the results gathered, number of boxes in the data flow diagram can be considered as the least accurate software metric when measuring the software size in software development projects. The main drawback in that method is that it takes a long time to create a data flow diagram, and difficult to construct as well. On the other hand, since the physical considerations are left out in the data flow diagram, the programmers may get confused towards the system. Therefore a software product cannot be assured its quality by a metric with such issues.

Acknowledgment

Our sincere gratitude goes all the people who contributed to make this research a success. Finally we would like to thank all the anonymous reviewers for their comments and suggestions.

References

[1] R. Kumar and R. Verma, *"Software Quality Metrics: Concept and Significance"*, 2nd ed. Jhunjhunu, 2014.

[2] V. Tiwari and R. Pandey, *"Open Source Software and Reliability Metrics"*, vol. 1, Dec. 2012.

[3] T.Manoharan, *"Metrics Tool for Software Development Life Cycle"*, vol. 2, pp. 1-16, Jan. 2014.

[4] D. Beyer, C. Lewerentz, and F. Simon, *"Impact of Inheritance on Metrics for Size, Coupling, and Cohesion in Object-Oriented Systems"*, vol. 2006, pp.1-17, Mar.2001.

[5] A. Abran, S. Oligny, C. Symons, D. St-Pierre and J.M. Desharnais, *"Functional Size Measurement Methods – COSMIC- FFP: Sesign and Field Trials"*, 2000.

[6] E. E. Mills, *"Software Metrics, SEI Curriculum Module SEI-CM-12-1.1"*, resources.sei.cmu.edu, 1988. [Online]. Available: http://resources.sei.cmu.edu/asset_files/CurriculumModule/1988_007_001_15608.pdf.[Accessed: March 20, 2015].

[7] N. E. Fenton and S. L. Pfleeger, *"Software Metrics: A Rigorous & Practical Approach"*, 2nd Edition, London: International Thomson Computer Press, 1997, pp. 638.

[8] M. H. Sherif, *"Handbook of Enterprise Integration"*,CRC Press, 2009. [Online]. Available: http://goo.gl/JUh9Gz. [Accessed: March 20, 2015].

[9] S. H. Khan, *"Metrics and Models in Software Quality Engineering"*, Addison-Wesley Professional, 2003. [Online]. Available: http://goo.gl/TxXUSx. [Accessed: March 20, 2015].

[10] A. J. Albrecht,*"Measuring Application Development Productivity"*,vol. SE-9 ,pp. 648-652, Nov. 1983.

[11] M. N. M. García and F. J. G. Peñalvo, *"Improving Estimations in Software Projects with Data Mining Techniques"*, vol. 18, pp. 265-278, April 1992.