IJREAT International Journal of Research in Engineering & Advanced Technology, Volume 1, Issue 5, Oct-Nov, 2013
ISSN: 2320 - 8791
www.ijreat.org

Role of Software Metrics on Software Quality

[1]Nivedita Joshi,[2]Pooja Singh

[1]Student, B.Tech (IT), Dronacharya College of Engineering, Maharishi Dayanand University Gurgaon, Haryana,India

[2]Student, B.Tech (IT), Dronacharya College of Engineering, Maharishi Dayanand University Gurgaon, Haryana,India

ABSTRACT

Software metric is a quantitative measure of the degree to which an item possesses a given quality attribute. The main objective of software quality metric is to facilitate management control, planning and managerial involvement and to identify situations for development or maintenance process improvements. This paper focuses on software metrics, its types and different views of software quality. Moreover it examines the realm of software engineering to see why software metrics are important and how they give an insight into the efficiency of software.

Keywords: *Software metric; Source code metric; Functional point metric; Object-Oriented metric;*

1. Introduction

Software metrics are quantitative measures that enable software engineers to gain insight into the efficiency of the software process. They provide measurement for software development including the SRS (Software Requirement Analysis) document, design, testing,integration and maintenance.

Large scaled software evolves complexity that makes the software difficult to control. Thus there is a need of software metrics to ensure the product quality. It is a valuable entity in the entire software lifecycle.

It is vital to introduce definition of software metrics.It can be defined as a function whose inputs are software data and whose output is a single numerical value, which tells the degree to which the software possesses a given quality, attribute.

An optimal metrics should be:

1. Simple and precise
2. Easily obtainable at reasonable cost
3. Robust- changes can be easily accommodated in the process or product
4. Relevant and valid
5. Mutually exclusive

2. Classification

2.1 Process metrics:

It highlights the process of software development. It focuses on duration of the process,the cost incurred and the procedure used. Process metrics can be used to enhance software development and maintenance.

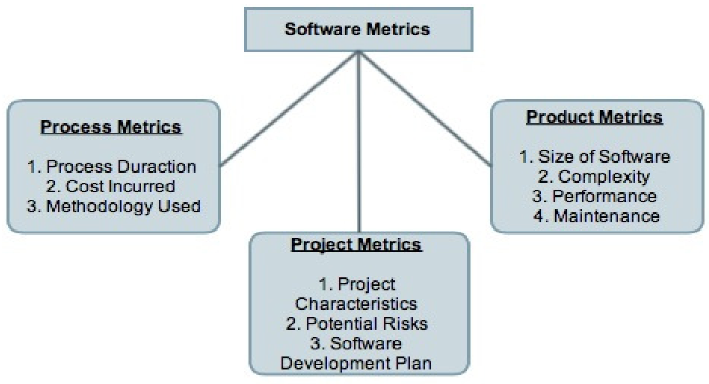

Fig. 1 Software Metrics

2.2 Project metrics

They are used to monitor project state and status. Project metrics prevent the problems of potential risks by standardizing the project. It describes the project characteristics and execution. It helps to elevate the software development plan.

2.3 Product metrics

It describes the attributes of software product at any phase of the development process. Product metrics measures the size and complexity, portability of the product.

3. Software metric as a mathematical function

Mathematically, a metric is a function 'f ' defined by 2 points 'a' and 'b' in coordinate space such that f(a, b) denotes distance between a and b. The function must satisfy the following properties:

1. f(a, a) = 0 = f(b,b) ; the distance of a point to itself is zero.
2. f(a,b) = f(b,a) ; the distance from a to b is same as distance from b to a.
3. f(a,c) <= f(a,b) + f(b,c) ; the distance from 'a' to 'c' is not larger than the sum of the distance from a to b and b to c where b is an intermediate point.

4. Software quality

It canbe defined as the capability of a software product to conform to its requirements. There are two different notions for software quality in software engineering:

1. Software functional quality: it tells how well it fulfills a given design based on functional requirements & specifications and how well it compares to competitors in the marketplace

2. Software structural quality:it reflects thedegree ofrobustness or maintainability to which the software is produced. It is evaluated by analyzing the inner structure of the software, the source code.

It is impossible to have an exact definition or meaning of software quality. The definition will differ according to factors like quality of products and business.

For example, a small word processing error in a student's assignment will not be a vast concern but a slight code error in a space shuttle's computer might endanger human lives.

Software quality can be viewed from different points of visions as follows:

1. User view: This view evaluates the software product against the user's needs. It includes:

   i) Software functionality that is whether the software product is suitable and the result is accurate

   ii) Reliability that is whether the software performs the intended function properly without any failure,

   iii) Fault recovery that is how the system performs during the failures,

   iv) Maturity that is how much work has been done with this technology or the number of versions released of the technologyetc.

2) Product view: It looks at the internal characteristics of the product. The idea behind this is that if the product is sound in terms of features and functionality, it will be favorable from user's viewpoint. That is the internal features will affect the external product behavior.

3) Value based view: Value based view are held by different departments such as marketing, finance, technical departments inan organization. Such viewpoints helps to develop a 360-degree product with different viewpoints for example the marketing department takes user's view and the technical department takes a product based view.

4) Manufacturing view: This view point emphasizes on building the product without any defects, getting it right the first time rather than making a defective product and wasting valuable project time on finding bugs at later stage. ISO 9001 standards are used by the organization for quality management systems.Indian Organization gives these standards for Standardization.

5. Software Metrics: Strengths and weaknesses

Most of the software metric has multiple definitions and rules.

5.1 Source code metrics

The SLOC was originally developed to estimate man-hours for a project. It is used to measure the size of a program by counting the number of lines in the source code. The lines of code (LOC) can be defined as:

a) **Physical line of code**: sets of instructions terminated by hitting the enter key of a keyboard.

b) **Logical line of code**: are the number of a program's commands

Physical line of code and logical line of code are considered identical for some languages but for some languages they can be considered differently. For example in the below code:

```
/* This is a comment */

For(inti=1;i<10;i++)

{

printf("this is a line");

}
```

This code has 5 physical line of code, 2 logical lines of code(for statement and printf) and 1 comment.

Where as the code below has only 1 Physical line of code,2 logical lines of code and 1 comment:

```
For(inti=1;i<10;i++)

{

printf("this is a line");

} /* This is a comment */
```

- **Strengths of physical line of code:**
  - Easy to measure
  - Used in many software project quality estimation
  - Scope for automation of counting
- **Weaknesses of physical line of code:**
  - Includes white spaces and comments
  - Functions imperfectly for certain visual languages
  - Includes dead code and is unreliable for direct conversion to logical statements
- **Strengths of logical line of code:**
  - Does not include white spaces and comments
  - Used in many software project quality estimation
  - Possible to perform direct mathematical conversion to logical statements
- **Weaknesses of logical line of code:**
  - Functions imperfectly for certain visual languages
  - Difficult to measure
  - The metric is imprecise for software use

5.2 Function Point Metrics

Allan Albrecht first defined function points in Measuring Application Development Productivity at IBM. The functional user requirements can be categorized into one of the five types: outputs, inputs, inquiries, internal files and external interfaces.

The first three elements (output, inputs and inquiries) are of Transactional Function Types and the last two (internal files and external interfaces) are of Data Function Types. Once the function is defined it is then assessed for complexity and assigned a number of function points.

Function point can be applied to Development projects, Enhancements projects etc.

- **Strengths of function point metrics:**
  - ➢ Direct mathematical to logical conversion of statements is very easy
  - ➢ These are used by many software cost estimating tools
  - ➢ These are stable for all programming languages
  - ➢ Useful for software reuse analysis

- **Weaknesses of function point metrics:**
  - ➢ These are unreliable for projects which are below 15 function point in size
  - ➢ It requires good deal of experience
  - ➢ It can be delayed and costly

5.3. Object-Oriented Metrics

Object oriented (OO) metrics are units of measurement that are used to characterize OO software engineering products (source code), OO software engineering processes (analysis, designing) and OO software engineering people (efficiency of individual tester). OO metrics provides four functionalities:

1. Localization: process of placing items in close physical proximity to each other.
2. Encapsulation: packaging of a collection of items
3. Information hiding: suppression of details
4. Abstraction: focusing on essential details of an item

- **Strengths of OO metrics:**
  - ➢ OO metrics are simple as compared to others
  - ➢ OO metric attractive in the OO community

- **Weaknesses of OO metrics:**
  - ➢ The OO metric have no conversion rules to function point metrics
  - ➢ They are difficult to enumerate
  - ➢ They have no conversion rules to lines of code metrics

6. Conclusion

With the advancementin software industries, software metrics have also developed fast. The software metric has become crucial in software development process. By using software metric the overall rate of progress in software quality and productivity will improve. Through this research paper we are thus able to tell how software metric plays a vital role in ensuring the quality of the product.

References

[1]Ho-Won Jung, Seung-Gweon Kim, and Chang-Sin Chung. Measuring software product quality: A survey of ISO/IEC 9126

[2]Roland Petrasch, "The Definition of, Software Quality': A Practical Approach", ISSRE, 1999

[3]B. Kitchenham and S. Pfleeger, "Software quality: the elusive target", Software, IEEE, vol. 13, no. 1, pp. 12–21, 1996.

[4]S. H. Kan, "Metrics and Models in Software Quality Engineering", 2nd ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2002

[5]Binstock, Andrew. "Integration Watch: Using metrics effectively". SD Times. BZ Media. Retrieved 19 October 2010

[6]Robert L. Glass. Building Quality Software. Prentice Hall, Upper Saddle River, NJ, 1992