ISSN (Online): 2230-7893
www.IJCEM.org

Software Effort and Cost Estimation using Software Requirement Specification

Ayman Hussein Odeh

Software Engineering Department, College of Engineering and IT,
Al Ain University for Science and Technology, Al Ain, UAE

Abstract
Process of estimating the effort required for the software development process is difficult and very complicated. There are many techniques to estimate the effort and cost of producing software, this article is based on a correct reading and mathematical analysis of the software problem, extract the variables needed as input and output for the required software, and applying the Software Science Metrics on the variables and their relationship to estimate unique and used operators and operators. Finally depending on these metric calculate the size and the time and effort ... etc. expected to produce software.
Keywords: ECE: Effort and Cost Estimation, SMSCQA: System for Measurement Source Code Quality Assurance

1-Introduction

Software engineering is a branch of computer science, aims to produce high-quality software [1]. To check the quality of the software, the measurements must be performed on the software product itself. Generally this is in the testing phase of software product life cycle, where a large list of metrics is prepared. This thing is very helpful in producing a high quality product. This article focuses on the estimation of the predicted values for some software metrics, especially software science metrics (Halstead's metrics), before starting coding (implementation) phase depending on the specifications and requirements of the software product in order to determine the metrics in the early stage of the life cycle. so the development team can define, and prepare the suitable plan to start development process.
IEEE Standard 610.12 defines Software Engineering as: "The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software". Therefore the software metrics are very important for software industry to quantify and evaluate the development process, effort, cost, and maintenance of software.
Cost and effort required to produce any product are very important factors. Especially if this is related to intellectual work such as programming and software engineering in all its stages.

2-Overview of Halstead Metrics:

Halstead metrics used to evaluating and measurement source code, these metrics are based on the operators, and operands [2], generally operands consist of variables, numbers, and constants used as part of expressions. Operators may consist of separate symbols like (+) plus operator, or contiguous symbols like (sqrt) square root operator. These metrics measured directly from the source code itself as following:

1. n1: Number of unique operators
2. N1: Number of all operators
3. n2: Number of unique operands
4. N2: Number of all operands

Table 1 : Indirect Metrics

Metrics	Name: description	Formula	Unit
n	Vocabulary	$n_1 + n_2$	word
N	Size (Length)	$N_2 + N_2$	word
D	Difficulty	$\frac{n_1}{2} X \frac{N_2}{n_2}$	
V	Volume	$N * \log_2 n$	bits
E	Effort	$V * D$	discriminations
B	Errors	$V / 3000$	bugs
T	Time	$\frac{E}{\beta}$	second
$\hat{N}$	Estimated Length	$\hat{N} = n_1 \log_2 n_1 + n_2 \log_2 n_2$	

But depends on operators an operands Halstead derived very important group of metrics. like: Length of source code, Vocabulary, Difficulty, Volume, Effort, Bugs, Time, and many other metrics were indirectly measured by derivation directly metrics using formulas as shown in the following table, Table(1) shows the indirectly metrics with their formulas are used in Software Science [2][3].

where: β is the programmer's mental discrimination, or it called (S): the Stroud number *, defined as the number of elementary discriminations performed by the human brain per second.

• The S or β value for software scientists is set to 18 [4].

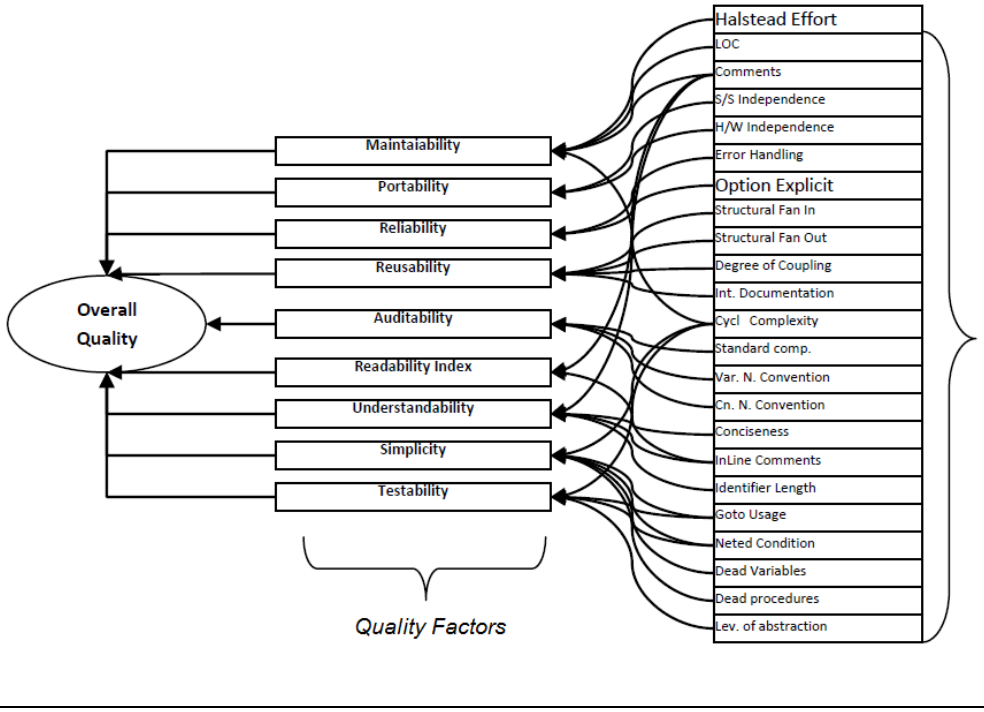

Fig 1: The SMSCQA Model [5]

---

*: In 1967, psychologist John M. Stroud suggested that the human mind is capable of making a limited number of mental discrimination per second (Stroud Number), in the range of 5 to 20.

3- Effort Estimation using Source Code

Given any source code, and using the SMSCQA system [5] nine quality factors, and more than 30 source code metrics can be measured, including all Software Science Metrics [4], see Figure (1).

So the Effort (E) also can be defined, the following example shown in figure(2), will be used to illustrate that

```
double a,b,c,x1,x2,delta;
String x3="";
do
{ a=kb.nextDouble();
 b=kb.nextDouble();
 c=kb.nextDouble();
}while (a==0.0);
delta=b*b-4*a*c;
if(delta>=0)
 {
  x1=(-b+Math.sqrt(delta))/(2*a);
  x2=(-b-Math.sqrt(delta))/(2*a);
 }
 else
 {  x3="Solution using Complex Numbers";}
```

Figure (2): The quadratic equation program

As a result of passing the above source code to the SMSCQA the following results shown in table (2) will be achieved

Table 2: Halstead's Metrics Values

No	Metric	Value
1	n_1	9
2	N_1	26
3	n_2	10
4	N_2	27
5	Vocabulary	19
6	Size	53
7	Difficulty	12.15
8	Volume	225.14
9	Effort	2735.45
10	Errors	0.075
11	Time	151.9

The Effort required for this source code is E=2735.45 discriminations, related to Halstead's metrics

4- Proposal method: Effort, and Cost Estimation using Requirements and Design (ECE)

Using requirements specification for any problem – specially the problem that can be described mathematically – all requirements factors and variables can be estimated, the same problem (Quadratic equation) will be taken. The algorithm shown in Figure (3) will be used to describe a problem and extracting specification and required operators and operand needed to solve a problem.

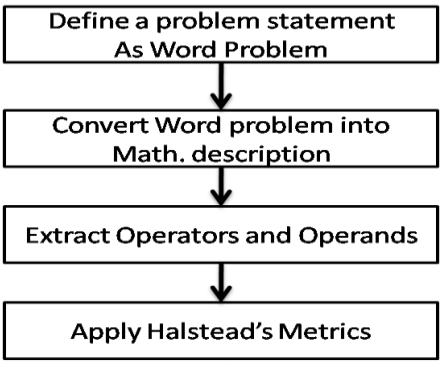

Fig 3: Steps of ECE

Step1: Define a problem as Word Problem: for the quadratic equation, the problem is to find roots of this equation when y=0.

Step2: Math. Description: The general form for this equation is : $y = ax^2 + bx + c$, where: a,b, and c are constants with known values , with constrain that (a) is not equal to zero. To solve this problem the following steps needed:

1. Define a,b, and c.
2. (a) is not equal to zero (a!= 0).
3. Calculate Delta () , using this equation : Delta=b*b-4*a*c.
4. if Delta>=0 then the roots of quadratic equation will be calculated using following formulas:

$$x_1 = \frac{-b - \sqrt{Delta}}{2a}$$, and

$$x_2 = \frac{-b + \sqrt{Delta}}{2a}$$

5. otherwise (when Delta <0) the solution will be in complex numbers

Step3: Extracting Operators and Operands: By analyzing the previous steps the used (unique) operators and operands can be extracted as following: Operands are: (***a,b,c,0,Delta,4,x1,x2, and 2.)***, Operators are: !=,=,*,-,>=, $\sqrt{}$,/, and <.

Step4: Applying Halstead's metrics: Again the main Software Science Metrics can be calculated depending on the estimation operators, and variables used in the description of the problem as shown in the Table(3) :

Table 3: Estimated of operators and operands

Metric	Description	Val
n_1	Number of unique operators	8
N_1	Number of all operators	19
n_2	Number of unique operands	9
N_2	Number of all operands	22

By using data existing in Table (3) all Halstead's metrics can be derived, and the following table shows the indirect metrics derived from the requirements of the software problem:

Table 4: Estimated group of metrics

No	Metric	Value
1	Vocabulary	17
2	Size	41
3	Difficulty	9.7
4	Volume	167.5
5	Effort	1638.6
6	Errors	0.055
7	Time	91.03

There is a direct relationship, as is known in software engineering [6], between the cost and effort, and this

relationship is a direct incremental correlation: $Cost = f(Effort)$ the more effort will increase the cost, and thus by estimating effort the cost needed to produce software can be estimated also.

5- Conclusion

Using the proposal method is very important for software development teams, and for the market and industry, because it helps to

1. Estimate effort, and cost needed to develop software before staring of the implementation of the software.
2. Using the estimated time, the more efficient (suitable) programmer can implement the source code [7].
3. Enhance of the overall quality of the software.
4. Increment of productivity.

References

[1]. Roger S. Pressman, "Software Engineering a practitioner's approach fifth edition", McGraw HILL, 2001.

[2] U.S. Department of Transportation Federal Aviation Administration," Software Quality Metrics", National Technical Information Service, Springfield, Virginia 22161, 1991.

[3] Gurdev Singh. *et al.*," A Study of Software metrics", IJCEM International Journal of Computational Engineering & Management, Vol. 11, January 2011.

[4]: Hamer, P. G. Halstead's software science- a critical examination. In proceedings of the 6th International Conference on software Engineering, IEEE Computer Society Press, New York, pp. 197-206, 1982

[5]: SMSCQA: Ayman H. Odeh,"SMSCQA: System for Measuring Source Code Quality Assurance", *International Journal of Computer Applications (0975 – 8887) Volume 60– No.8, December 2012*

[6]: Jyoti G. Borade," Software Project Effort and Cost Estimation Techniques", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 8, August 2013

[7] Ayman H. Odeh, Tariq R. Somro" Recruiting Quality Software Developers", WULFENIA journal, *Vol 20, No. 5;May 2013*