Software complexity Methodologies Comparisons

Masoud Rafighi[1], Nasser Modiri[2]
[1]Taali, Qom, Iran; [2]Zanjan Azad University, Zanjan, Iran;
Paper Reference Number: 6
Name of the Presenter: masoud rafighi

Abstract

Studies and researches result show that complexity is one of the software natural features. Software natural complexity and software requirement functionality are related with each other and they could not be lower than special value. in this paper have defrayed to measurement complexity with using the MacCabe and Halstead models and with an example discuss about software complexity and then compared and peruse flow metric information Henry and Kafura, complexity metric system Agresti-card-glass, design metric in item's level then categorized object oriented and present a model with 4 level of software complexity..

Key words: measurement software complexity, McCabe model, Halstead model

1. Introduction

Due to high cost of software, software organization are trying to find away to make it lower. Because of this the researcher are trying to find the relation of software feature and problem of extended software. Hard works need more time to do, in this time we need more source, that it means more cost. One of the reasons for proceeding to software's complexity and its measurement is controlling the expenditure of software's life time, because software complexity is one of the basic agents in increasing cost of extended and maintenance. Software complexity is an item that is not identify and it's not easy to measure and describe and usually disregarded in planning project process. So we are looking for a way to predict how hard is maintenance, change and understanding software. That with measurement and control decrease the cost on software's life time

2. complexity measure

Basic of complexity describe is quality of connection between different part of software system, the most simplest metric for structure complexity is measure. The measure determine with LOC or functional point.

- ✓ LOC

One of the most famous balance software is line counter with LOC unit or about big program with KLOC which is used for quantity of software complexity. Unfortunately there is no agreement on every part of LOC. most of the researcher come to an agreement to not calculate the distance of lines. But yet there is no agreement about comment, sign, structure like BEGIN in Pascal and...

Another problem in free format language is different structure are in one textual line or one executive structure is broken to more than one line executive code.

LOC metric is simple, understandable, it used in every program language and it has wide usage. Also we can use it for evaluation programmer although it needs attention because of the style of programming it can has effect on values, a programmer it can has effect on values, a programmer may produce many lines and another one be success to compress that function in lower space. Also extender, work on different thing except producing more code, like document, programming test and... also the time of wage payment to code line need more attention because there is many way to make the program massive.[1]

Function point metrics

Quantity metric which are base on the number of code line program are not satisfy. from the user point of view function points are a group of measurable code. A huge program may have millions LOC. But a program with 1000 function points is a huge application program or a real system. A function as a collection of programmable structure, with definition of formal parameter and local variable that change with this structure is defined.

A metric of functionality point, in IBM is a weighted total of five items that characterize a application program.

Function point is coming from a tentative relation base on metric countable from software information domain and evaluation of software complexity.

Function point will caulk with a complete table. five feature of domain will determine. There are count in suitable place of table. To determine the values of information domain flow this sentences:

The number of incoming user: every incoming user that has different application data from software will count. Entrance should count different from requests.

The number of outgo user: every outgo user that bring information for user will count. In this paper, outgo is reports, monitor, error massages and ...

Sporadic ingredient data in a text report, won't count differently.

The number of user's requests: the request will define as a online entrance which produce answer without any pause every one of the requests will count.

the number of files: every main logical files is a logical group of data which can be part of a big information bank or a separate file, and will count.

The number of outgo interface: all of the machine reading (like data file on thin tape) which use to transfer the information to another system, will count.

	Weighted coefficient						
Measurement parameter	Count		simple	average	complex		
Number of entrance user		*	3	4	6	=	

Number of out go user		*	4	5	7	=	
Number of requests user		*	3	4	6	=	
Number of files		*	7	10	15	=	
Number of out go interface		*	5	7	10	=	
Total count $\longrightarrow$							

Fig 1: find the function point.

One complex value will determine for every count when the data has assembled. The organization which use this way will develop determination simple, average or complex portal evidences. For function point(FP) use this frame:

FP = total count x[0.65+0.01x $\sum(F_i)$]

Total count: sum all FP portals which is in fig.1

F_i (I =1 to 14) <<value of complexity conduction>> base on answer of this questions:

1. Does system need support and retrieval?
2. Does it need connection data?
3. Is there any parcel processing operation?
4. How important is efficiency?
5. Does system work in a operational environment?
6. Does system need online data portal?
7. Does online data online need to make input transaction on operation or multi job monitor?
8. Does main files update online?
9. Are the entrances, outgoes, files and requests complex?
10. Is the internal process complex?
11. Are the codes usable again?
12. Is there any reduction or installation in design?
13. Is it designed for installing in different organization?
14. Does the application program make the changes simple and use easily by user?

The answer of this questions is between 0 to 5 the constant values in this frame have found tentative.

When function points were calculated, they are used in a way like LOC method. For normalization of software implement qualification, quantity and another qualification.

3. Other complexity metrics

✓ Cyclic number McCabe

Cyclic complexity is the most usage member of static software metric. Cyclic complexity measure the number of liner independence way in a yardstick. It shows a number which can compare with other programs complexity. cyclic complexity is program complexity or McCabe complexity. it's easy to understand this complexity and you can get useful result.

This measure is independent from language and format language. Cyclic number is a simple way to compare software.

Cyclic complexity measure is coming from connection graph to measure.

CC = E - N + p	E: number of edge graph
N: number of disconnect nod	P: number of disconnect part of graph

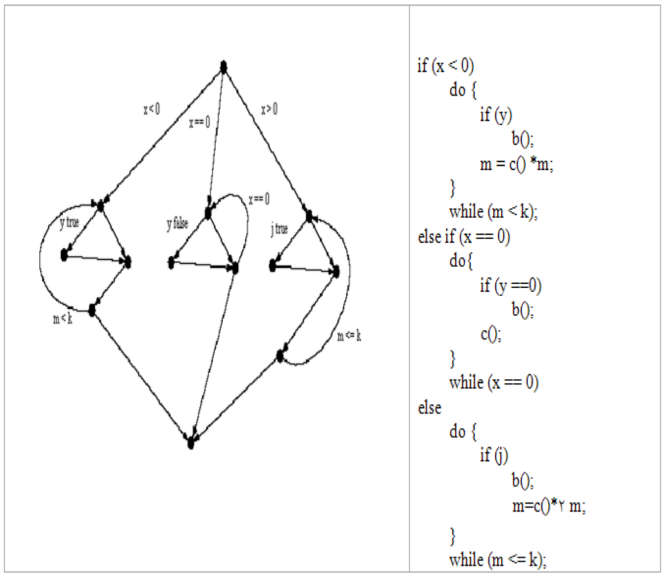
Countable treaties are needed for real count this item. For example some tools which get cyclic complexity have this treaty. this complex number give you a better measure to calculate the program complexity. this figure show a part of code and connection graph with cyclic number 9.

Nodes which has more than one way increase the cyclic complexity.

Another way to calculate cyclomatic complexity is:

Cc= number of decision +1

So, what’s the decision? Decisions come from conditional predicate. The cyclomatic complexity of a procedure without any decision is 1.there is no maximum value for cyclomatic complexity because one procedure can have many decision. Conditional predicate, include for, case, if ... then .... else..., while, do and ...[2]



```
if (x < 0)
      do {
            if (y)
                  b();
            m = c() *m;
      }
      while (m < k);
else if (x == 0)
      do{
            if (y ==0)
                  b();
            c();
      }
      while (x == 0)
else
      do {
            if (j)
                  b();
                  m=c()*٢ m;
      }
      while (m <= k);
```

Fig 2: example of cyclic complexity graph.

It’s merit to mention that cyclic complexity is not sensitive about unconditional junction like go to, return and break-statement, however they increase complexity. The complexity of many programs are measure and determine a confine for complexity that help software engineers to find the natural risk and perpetuity of a program.[3]

+1	If..Then
+1	Else...If..Then
+1	Case
+1	For [Each] ..
+1	Do..
+1	While

Table 1. effect of conditional predicate in cyclic complexity.

Criterion which is regulated for development and maintenance and for estimate this risk, coast and perpetuity program in reengineering can use. Studies show that the cyclic complexity program and errors frequency are dependent. The low complexity help out to understand program easier. Having changes in programs which are low cyclic complexity have lower risk than programs which are high cyclic complexity. Also cyclic complexity of yardstick is a powerful measure to test it. One common cyclic complexity usage is compare it with a collection threshold value. You can see one of this collection in table. 2.

CC	**Kind of procedure**	**Risk**
1-4	**One simple procedure**	**Low**
5-10	**One perennial procedure with good structure**	**Low**
11-20	**A complex procedure**	**Average**
21-50	**A complex warning procedure**	**High**
>50	**A susceptible of error and changeable procedure**	**Very high**

Table 2. cyclic complexity.

cyclic complexity is usage in different precinct like:

- ✓ Analysis code development risk
- ✓ Analysis changes in maintenance risk
- ✓ Test planning
- ✓ Halstead 's metric

4- Halstead metric

Professor Maurice Halstead separate the software knowledge and computer knowledge. Criterion of Halstead complexity for measurement the range of yardstick program complexity is coming from source code. Halstead's criterions were for determine a quantative criterions from yardstick's values. these criterions were the most powerful typical determine the code complexity between primary metrics. this metric use as a maintenance metric duo to applying these metrics to code. There is many different idea about value of Halstead criterion which is in the range of "complexity... and unreliable " to " the most powerful maintenance criterion ". one thing which is so important is reliable to tentative document in typical maintenance, but it's clear that this Halstead criterion are useful even in development state for estimate the quality of code in programs which have high calculative density [1].Halstead's criterions are based on four value which are from code source.[4]

n_2 : number of different values which are in program.

N_1 : total number of operator

N_2 : total number of values

This numbers cause 5 criterion:

Criterion	**Symbol**	**Frame**
Length of program	**N**	**N= N1 + N2**
Collection of word program	**N**	**n= n1 + n2**
Bulk	**V**	**V= N * (LOG2 n)**

Difficulty	**D**	**D= (n1/2) * (N2/n2)**
Effort	**E**	**E= D * V**

Table 3.Halstead metric.

If one time a rule for calculating the value be specified, it's easy to calculate this criterion. Derivation of number of code items need a sensitive scanner which is a simple program for most of the languages. Halstead's criterions are operational in operational system and for development effort one time after writing the code. Code maintenance at development time have to attend, Halstead's criterions should use during code development the pursuit the complexity. They were criticized duo to difference reasons. This is a claim which says these criterions measure lexical and textual complexity not structural or logical flow complexity. However that the most powerful measure criterions is maintenance. specially , estimate the complexity with Halstead's criterions for code which has high rate of logic calculations instead of logic junction is more tender. Cyclic complexity is one of the structural complexity criterion. Another metrics express other aspect of complexity, include structural and calculative complexity as what you see on table. 4.

Criterion of complexity	**Usual criterion**
Halstead's Criterion of complexity	**Algorithmic complexity will measure by counting values**
Henry and Kafura metrics	**Connection between yardsticks(parameters, public, values, calling)**
Bowles metrics	**System and yardstick complexity, connecting by parameters and public values**
Troy and Zweben metrics	**Connection or to be yardstick, structure complexity (maximum depth structure chart) call to, call by**
Ligier metrics	**To be yardstick structure chart**

Table 4. an example of criterion of complexity.

5. object-oriented complexity model

Paradigm OO by using a better way to analysis problem, plan and implement solution, is basic change in software engineering. Most of the software engineering purpose are accessible like maintenance, reliable, usable.

Some advantages of OO system is fast development, high quality, easy maintenance, decreasing coast, better informational structure and increasing compatibility. one of the main reasons of this claims is OO methods with support of data secession hierarchy analysis.

Some important question which should be answered:

What is the difference between OO paradigm and primary paradigm?

How this difference make access to software engineering purpose easier?

Are this purpose really as they were claimed?

To answer this questions we need to have ability measurement and suitable criterion.
software metrics have many cohort as a basic rule in a engineering way for design and OO software development control like software complexity level.[5]

complexity of OO system can express with a collection of criterion which define in deferent level. A model of complexity system with four level has suggested for OO system: values, method, object, system.

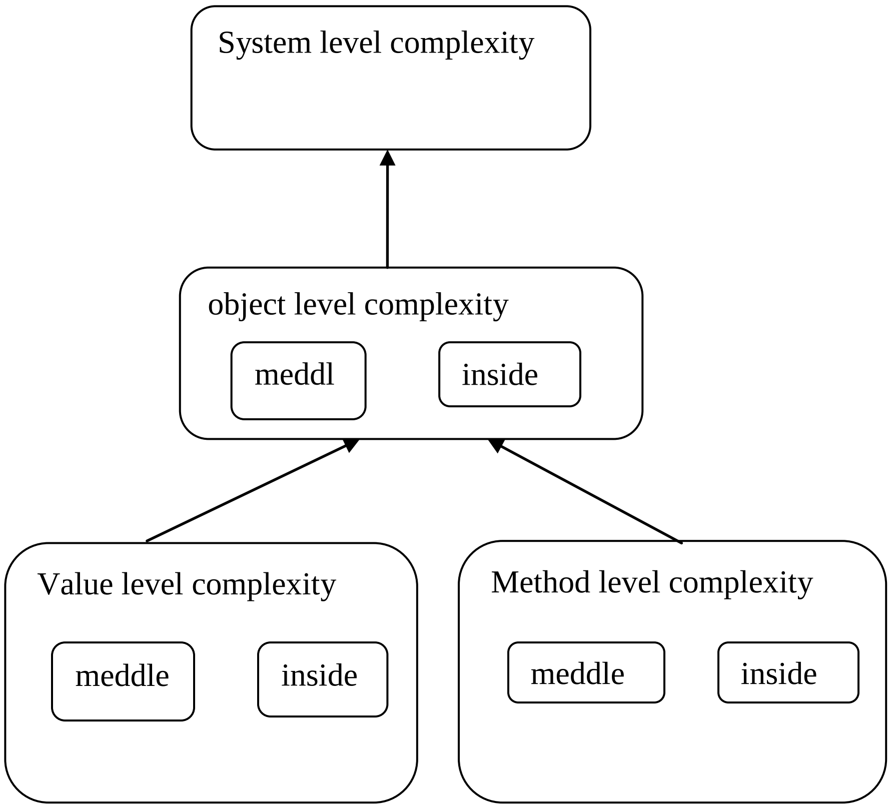


Fig 3:a model of complexity in object-oriented system with 4 level.

Value level complexity have relation with definition of values in system method level complexity have relation with definition of method in system object level complexity is a combination of value and method complexity with inheritance structure criterions. System level complexity give you a performance from high level of organization and size of OO system. There are some criterions to make system connection acceptable in every level. Criterions are usable in every part of systems life OO metrics can be calculated in different levels. We can have some metrics in level of system which assemblage structural feature of all part of system. In class level we can calculate the structural feature of class like union and depth of inheritance. We can determine some metrics on method levels.[6]

6. **Conclusions**

Software metrics are useful technique. To improve quality we have to find a method to measure the complexity of software for control and supervision on it. In this paper, the algorithms and methods of measurement the software complexity are compared. Studies and researches show that we can find the complexity by using algorithms and different methods as the high level of complexity cause many errors, need to test it and high coast of development and maintenance. so, software complexity has directly relation with coast of development and maintenance. so it's not logical to disregard it. As result to decrease the coast of maintenance and repairing software you should measure and restrain the complexity of software. It is suppose that the present ways to measure the software complexity has wide domain that we should guide it to requirement complexity if we remove complexity sooner. We will have less coast so it's logical to looking for methods to measure the complexity in first phase of software production (requirements phase, analysis and design phase).

References

[1]sylvia B. sheppard, phil milliman, M. A. borst, and tom love."
Measuring the Psychological Complexity of Software Maintenance Tasks with the Halstead and McCabe Metrics" IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. SE-5, NO. 2, MARCH 1979. Pp.96-104
[2]Yas Alsultanny." Using McCabe Method to Compare the Complexity of Object Oriented Languages" IJCSNS International Journal of Computer Science and Network Security, VOL.9 No.3, March 2009.pp.320-326
[3]Paul. D. Scott." Measuring Software Component Reusability by Coupling and Cohesion Metrics" JOURNAL OF COMPUTERS, VOL. 4, NO. 9, SEPTEMBER 2009,797-805
[4]Yingxu Wang and Jingqiu Shao," Measurement of the Cognitive Functional Complexity of Software" Proceedings of the Second IEEE International
Conference on Cognitive Informatics (ICCI'03)0-7695-1986-5/03 2003 IEEE
[5]Jitender Kumar Chhabra, K.K. Aggarwal, Yogesh Singh," Code and data
spatial complexity: two important software understandability measures" Information and Software Technology 45 (2003) 539–546
[6]S. R. Chidamber and C. F. Kemerer, "A Metrics Suite for Object Oriented Design," IEEE Trans. on Software Eng., vol. 20, no.6, 1994, pp. 476-493.