Software Change Effort Estimation at Design Level using Flow Chart

Priyadarshini
Department of Computer Applications
Mangalayatan University
Aligarh

Sushant Kumar
Department of Computer Applications
Mangalayatan University
Aligarh

Aprna Tripathi, PhD
Department of Computer Applications
Mangalayatan University
Aligarh

ABSTRACT
The success of a software depends on it exact estimation of its effort, cost and duration. Most of the software fails just because of wrong estimation. Efficient Cost and effort estimation at early stages of software development is one of the big challenges for software engineers as well as software scientists. There are a number of estimation metrics to estimate effort but the metrics to estimate cost of software change in early phases of software development is rare. In this paper, we are proposing a metric to estimate the effort of software change at design level using flow chart. The results are analyzed with ten flow charts of various program of length 50-200 line of codes. Also results are compared with existing metrics.

Keywords
Effort, cost, flow chart, software change

1. INTRODUCTION
The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software, and the study of these approaches; that is, the application of engineering to software. The study and application of methodologies to develop quality software that fulfill customer needs.

Software engineering is the establishment and use of sound engineering principles in order to obtain economically software that is reliable and works efficiently on real machines.

- Software engineering is an engineering discipline that is concerned with all aspects of software production.
- Software engineers should adopt a systematic and organised approach to their work and use appropriate tools and techniques depending on the problem to be solved, the development constraints and the resources available.

To estimate effort required to implement requested changes is a challenging area of research.

In this paper proposed a method to estimate effort using flowchart in the early phase of software development.

Different authors proposed different set of symbols to design the flowchart. In our proposed approach we considered the following set of flow chart symbols.

Also, we assigned different weights of the symbols based on their complexity in the program.

Symbol	Name	Weight age
	Input Box	1
	Subscript Box	6
	Process Box	2
	Decision Box	3
	Loop Box	4
	Document Box	5
	Data Box	7

Fig. 1. Flowchart Symbols along with their weights

2. RELATED WORK
During the last few decades, various models, methods and techniques have been developed for the estimation of software maintenance effort. The work presented in this paper is based on SRS. The paper also considers technical and environmental factors. F. Niessink and H. van Vliet [1] proposes a variant of the function-point analysis to remedy some of its shortcomings. Albrecht [2] discusses the function-points calculation for size of the software. Boehm [3] discusses Constructive Cost Model (COCOMO) and computes annual maintenance effort using software development effort and annual change traffic. Leung [5] discusses a method to estimate project effort using analogy. Sharma and Kushwaha [4] discuss the improved requirement based complexity

(IRBC). Gill and Kemerer [6] discuss cyclomatic complexity density and gives relation between productivity and size of the software system in source lines of code. Aprna et al[7] computes the requirement based maintenance effort estimate based on IEEE Std 830-1998 of requirement engineering document, soon after freezing the requirement.

3. PROPOSED APPROACH
In this paper we proposed a method to estimate program change effort in the early phase i.e. at design level.

The proposed method has following steps:

Step 1. To fetch the initial change requirement
Step 2. Design the flow chart of original and modified program
Step 3. Find the difference between these two flowcharts
Step 4. Estimate the effort using proposed formula

Step 1. To fetch the initial change requirement
First we have initial request like that we take reverse number of program, in the reverse number

of program we need to two variables, with the help of variable we create logic.

Step 2. Design the flow chart of original and modified program.

Step 3. Find the difference between these two flowcharts
For the initial request we create flowchart, in flowcharts for original and modified program.

Step 4. Estimate the effort using proposed formula
Find the Difference of Flowchart: - By comparing both the flowcharts differences are identified.

Effort required for Change = ∑ (Weight of symbol * Difference of number of Symbol)

If difference of symbol is zero then we will consider difference of operators used.

Difference of symbol will be computed by modified and original flowcharts

4. CASE STUDY

A program to find the reverse number of a given number is considered.

Changing Request: - In initial flowchart we will change the some features as per request come palindrome program.

Original Flowchart: - For the initial request we create flowchart, in flowchart we work step by step format that the reverse number of program work that is shown in fig 2.

Modified Flowchart: - We will modify the initial flowchart as per changing request here changing request is modify reverse flowchart to palindrome flowchart that is shown in fig 3.

Effort Estimation = (W * O) + (W* S)= 2+ 5=7 units

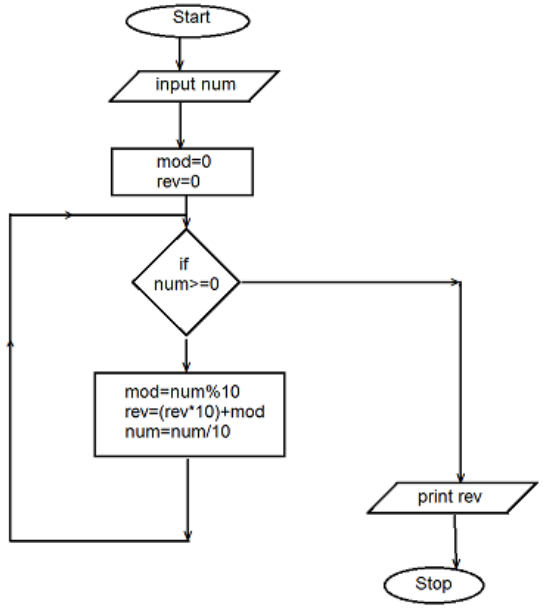

Fig. 2. Flowchart for Reverse a Number

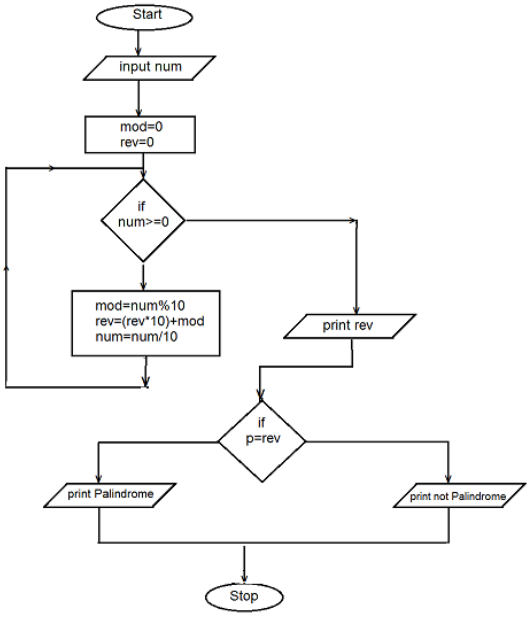

Fig. 3. Flowchart for finding a number as Palindrome

5. EXPERIMENTAL STUDY

To know the correctness of the proposed method, we held an experiment session in the computer lab. Nine programs are considered for this. Three different persons with almost having same programming skills were selected. As an input original program, and requested change is given to the programmer and the time required modifying that program is counted. Fig. 4 shows the time required to change the program.

S No.	Flowchart A	Flowchart B	B		
			Starting Time	Ending Time	Duration
1	Sum of number	Find even/odd	11:00	11:02	2M
2	Series 1+1/2+1/3+.....1/n	Series 1!+1/2!+1/3!+.....1/n!	11:05	11:15	10M
3	Reverse	Palindrome or Not	11:20	11:23	3M
4	Prime or Not	First 100 prime	11:25	11:30	5M
5	Bubble short	Selection short	11:35	11:45	10M
6	Bill unit	Internet bill	11:50	11:59	9M
7	Find even/odd or +ve/-ve	Sum of even/odd or +ve/-vesn	12:05	12:12	7M
8	Series 1,3,6,10,15,....n	Series 1,4,9,16,25,....n	12:15	12:19	4M
9	Triangle of pattern	Middle triangle of pattern	12:25	12:40	15M

Fig. 4. Experimental Results

Table 1 shows the detailed computation of effort required for considered program through proposed method.

Table 1. Estimated effort for the considered program

Type of Sybmol	**Weight of Sybmol (W)**	**Original Flow Chart**		**Modified Flowchart**		**Difference**		**W * O**	**W* S**
		No. of Symbols	**No. of Operators**	**No. of Symbols**	**No. of Operators**	**No. of Symbols(O)**	**No. of Operators(S)**		
Process Box	2	2	9	2	10	0	1	2	0
Data Box	1	2	-	4	-	2	0	0	2
Decision Box	2	1	2	2	2	1	0	0	3

6. RESULTS AND COMPARISION

Above flowchart we can see the that where effort is increase then time is increase, with the help of initial flowchart and modified flowchart we will compute the effort for the difference of how many number of operator is occur in initial flowchart and how many number of symbol is occur in modify flowchart. Table 2 is showing the comparative results.

Table 2. Time required applying the requested change in the existing program.

Program ID	**Effort Computed**	**Time Consumed in Implementation in Lab (In Minutes)**
P1	12	2
P2	26	10
P3	14	3
P4	17	5
P5	24	10
P6	20	9
P7	19	7
P8	16	4
P9	35	15

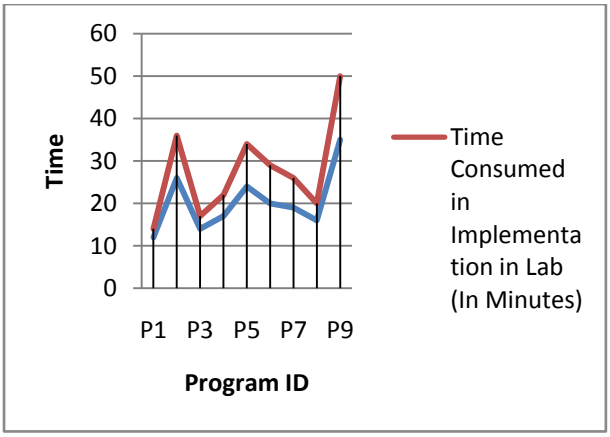

Table 3. Comparison between proposed and experimental values

As shown in fig 5 the trend of the estimated time and the time required in actual implementation is almost similar.

7. CONCLUSIONS

In this paper the effort required for change is estimated at early stage of the software development. Results shows that the trend of estimated values and the real time required to implement the program is very closed to one another. In future we are planning to automate this proposed approach.

8. REFERENCES

[1] Bowman, M., Debray, S. K., and Peterson, L. L. 1993. Reasoning about naming systems.

[2] Ding, W. and Marchionini, G. 1997 A Study on Video Browsing Strategies. Technical Report. University of Maryland at College Park.

[3] Fröhlich, B. and Plate, J. 2000. The cubic mouse: a new device for three-dimensional input. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems

[4] Tavel, P. 2007 Modeling and Simulation Design. AK Peters Ltd.

[5] Sannella, M. J. 1994 Constraint Satisfaction and Debugging for Interactive User Interfaces. Doctoral Thesis. UMI Order Number: UMI Order No. GAX95-09398., University of Washington.

[6] Forman, G. 2003. An extensive empirical study of feature selection metrics for text classification. J. Mach. Learn. Res. 3 (Mar. 2003), 1289-1305.

[7] Brown, L. D., Hua, H., and Gao, C. 2003. A widget framework for augmented interaction in SCAPE.

[8] Y.T. Yu, M.F. Lau, "A comparison of MC/DC, MUMCUT and several other coverage criteria for logical decisions", Journal of Systems and Software, 2005, in press.

[9] Spector, A. Z. 1989. Achieving application requirements. In Distributed Systems, S. Mullender