# Measure Performance of VRS Model using Simulation Approach by Comparing COCOMO Intermediate Model in Software Engineering

Dr. Devesh Kumar Srivastava
Jaipur India
E-mail: devesh98@gmail.com

**Abstract— Estimation of a software project failure is based on project analogies in the area of software cost estimation. In all software cost estimation approaches important decisions must be made regarding certain approaches in order to obtain reliable estimate. A reliable and accurate estimate can be proved by simulation approach. A simulation modeling approach is proposed for the prediction of software effort productivity indices, such as cost and time-to-market, and the sensitivity analysis of such indices to changes in the organization parameters and user requirements. Simulation provides a controlled environment for evaluating performance. Changes can be made easily to the model and results evaluated quickly. In this paper we measure performance of VRS model through simulation result by comparing standard COCOMO Intermediate-Organic Model. This approach uses a model specification. Results demonstrate the model representativeness, and its usefulness in verifying process conformance to expectations, and in performing continuous process improvement and optimization.**

**Keywords**- Business Process Outsourcing (BPO), Kilo Line of code (KLOC), Constructive Cost Model (COCOMO), Interactive Voice Response (IVR), Voice Response Software (VRS), Effort Adjustment Factor (EAF), Person (P), Month (M).

## I. INTRODUCTION

Simulation is a powerful tool for solving many problems. A Software MAT LAB used as simulator tool which will help in taking a decision on accuracy of VRS Model [1] to estimate the effort and time for development of IVR software. The cost and time estimates are useful for the initial validation and the monitoring of the project’s progress during development [2]. These estimates may be useful for project productivity assessment after completion. Reducing the cost of large scale software projects and shortening cycle time, or time to market, is a major goal of most software development organizations. To pursue such a goal, organizations can set productivity goals for each project, and put in place statistical productivity controls to enable developers and management to take corrective actions when there are deviations from the goal, and to distinguish a random deviation from meaningful deviations. Simulation is one of the methods for performing such control. It can be used at various points in the software life cycle to perform risk analysis, in terms of time to product, and cost, to verify conformance to expectations, and to perform continuous process improvement and optimization. Simulation is a formal and robust technique [3]. It does not rely heavily on mathematical abstraction therefore it is suitable for modeling even complicated environments. Simulation is basically a numerical technique therefore it can be used to generate quantitative output data on various parameters that influence a system performance. Output data analysis, experimental design can be employed to ensure a significant degree of mathematical robustness at every stage of a simulation project. This requires that organizations use metrics and models to evaluate and predict effort and time duration as a function of the user requirement size (in LOC), and the organization parameters [4].The model is parameterized on the basis of measurements and analyses of data coming from IVR software [1]. To meet the modeling and simulation requirements of a particular environment, appropriate simulation tool needs to be selected with the considerations of simulation performance such as efficiency, accuracy, and speed, etc [6]. Popular existing packages are MATLAB which is a PC based system simulation tool with medium to high system complexity, and spreadsheets software. A large number of simulations can be performed situation and gradually refine the model as our understanding of the process improves, which enable us to achieve a good accuracy approximation. Simulation is based on traditional queuing models, using statistical and experimental methods to generate an internal picture of the system from which the statistical data is gathered for performance analysis. The simulation package statically store the simulation configurations, simulation results, results analysis, and dynamically keeps tracks of the parameter modifications of the simulation data set. The initial parameters of a given problem are entered manually. The simulation models and configurations are stored in the input data set file, and from the file, the program extract the whole set of information as the input of the simulation program

[5]. The outputs of the simulation program are stored in the output file in order to conduct analysis, query or visualization.

The remainder of this paper is structured as follows: The second section of the paper describes the existing model of proposed work. The third section describes about research methodology. The fourth describes the simulation result process activities and the fifth section synthetically illustrates the conclusion of simulation experiments.

## II. EXISTING WORK

This section provides some information of COCOMO - software effort estimation models which is used in this research work. The COCOMO model is supposed to be standard model to calculate the effort and development time of software projects.

A.COCOMO Basic Model

COCOMO Basic model is proposed by B.W.Boehm. COCOMO model have three sub-models i.e. basic, intermediate and detailed model [11], [12] .

| **Development Mode** | **Basic Effort Equation** | **Time Duration (D)** |
|---|---|---|
| Organic | Effort = 2.4 KLOC $^{1.05}$PM | D = 2.5 * (Effort )$^{0.38}$M |
| Semi Detached | Effort = 3.0 KLOC $^{1.12}$PM | D = 2.5 * (Effort )$^{0.35}$M |
| Embedded | Effort = 3.6 KLOC $^{1.20}$PM | D = 2.5 * (Effort )$^{0.32}$M |

B. COCOMO Intermediate Model

The Intermediate COCOMO is an extension of the basic COCOMO model. Here we use the same basic equation for the model. But coefficients are slightly different for the effort equation [11], [12].

| **Development Mode** | **Intermediate Effort Equation** | **Time Duration (D)** |
|---|---|---|
| Organic: | Effort = EAF * 3.2 * ( KLOC)$^{1.05}$PM | D = 2.5 * (Effort )$^{0.38}$M |
| Semi Detached | Effort = EAF * 3.0* ( KLOC)$^{1.12}$PM | D = 2.5 * (Effort )$^{0.35}$M |
| Embedded | Effort = EAF * 2.8* (KLOC)$^{1.20}$PM | D = 2.5 * (Effort )$^{0.32}$M |

C. COCOMO II model

This model is an extension of COCOMO Intermediate model and specific one among COCOMO-II series .It has 17 cost drivers [11], [12].

$$\text{EFFORT} = \text{EAF} * 2.9\ \text{KLOC}^{1.10}$$

$$\text{Duration} = 3.67 * (\text{Effort})^{.31}$$

D. SEL – Model

The Software Engineering Laboratory (SEL) of the University of Maryland has established a model i.e. SEL Mode[l11[12] ]for estimation. Estimation of effort according to SEL model is defined as follows.

$$\text{EFFORT} = 1.4 * (\text{Size})^{0.93} \quad \backslash$$

$$\text{Duration D} = 4.6\ \text{KLOC}^{0.26}$$

Effort (Person-Months) and lines of code (size in thousands of lines of code i.e. KLOC) are used as predictors.

E. Walston-Felix Model

Walston and Felix had developed their model in 1977 to estimate effort from a various aspects of the software development environment such as sixty projects collected in IBM's Federal Systems division. It provides a relationship between delivered lines of source code. This model constitutes participation, customer-oriented changes, memory constraints etc. According to Walston and Felix model, effort is computed by [11[12] :-

$$\text{EFFORT} = 5.2\ \text{KLOC}^{0.91},$$

$$\text{Duration D} = 4.1\ \text{KLOC}^{0.36}$$

F. Bailey-Basil Model

This model developed by Bailey-Basil between delivered lines of source code and formulates a relation [11][12]

$$\text{EFFORT} = 5.5\ \text{KLOC}^{\mathbf{1.16}}$$

E. Halstead Model

This model developed by Halstead between delivered lines of source code and formulates a relation [11[12]

$$EFFORT = 0.7\ KLOC^{1.50}$$

G. Doty (for KLOC > 9)

This model developed by Doty between delivered lines of source code and formulates a relation [11[12]

$$EFFORT = 5.288\ KLOC^{1.047}$$

## III. Proposed Work

Simulation is defined as the process of creating a model of an existing or proposed system e.g. a project in order to identify and understand those factors which control the system and/or to predict the future behavior of the system. The model is simulated and many experiments are carried out. We begin simulations with a simple approximation of the situation and gradually refine the model as our understanding of the process improves, which enable us to achieve a good accuracy approximation. The client / customer demands cost and time duration before development the software. In such way VRS model will be useful to predict the estimated effort and time duration of IVR software. We took seventeen assumed dataset in terms of line of code metric as shown in table 1 to observe the simulation results using MATLAB version 2.0. The COCOMO Intermediate organic model is supposed to be standard model to estimate the reliable effort and schedule. The author found that the parameter values of VRS model are close to COCOMO Intermediate -organic Model therefore the VRS-model is being compared with COCOMO intermediate organic model to represent the simulation result in graphical way.

VRS Model:

$$\text{Initial Effort } E = 3.4 * (\text{Project Size})^{1.15},$$

$$\text{Time Duration } D = 2.2 * (\text{Effort})^{0.31}$$

COCOMO Intermediate -organic Model

$$\text{Initial Effort} = 3.2 * (\text{Project size})^{1.05},$$

$$\text{Time Duration} = 2.50 * (\text{Effort})^{0.38}$$

**Life Cycle of IVR Software**

A Systems development life cycle (SDLC) is a process that includes Software Analysis, Requirements, Software Design, Software Coding, implementation, testing and installation, call testing, and operation phase.

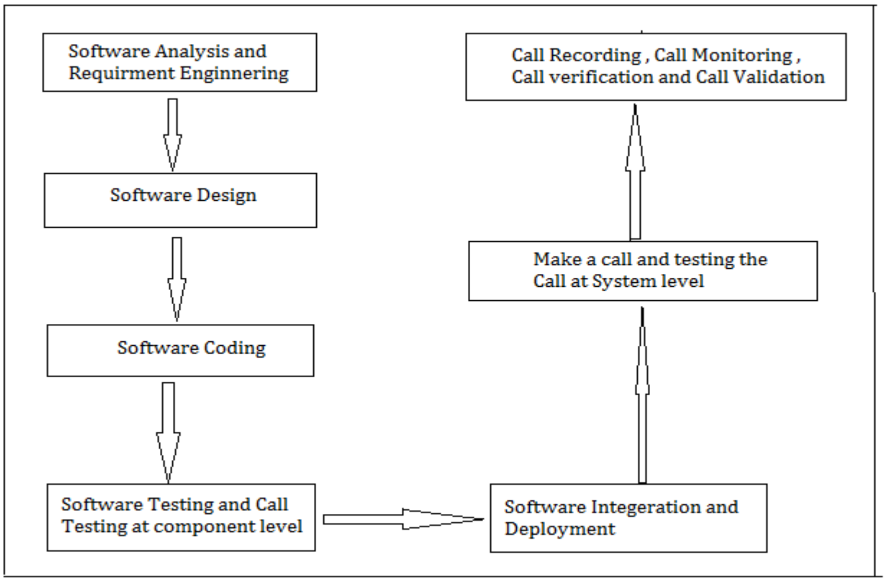


Figure 1: IVR software life cycle

**Comparison of parameter value**

| Model | a | b | c | d |
|---|---|---|---|---|
| VRS Model | 3.4 | 1.15 | 2.2 | 0.31 |
| CoCoMo Intermediate Model | 3.2 | 1.05 | 2.5 | 0.38 |

Here it is observed that parameter value of VRS Model is slightly vary than CoCoMo intermediate Model due to phases of IVR life cycle as shown in the Figure 1 , an additional phase i.e. call testing is added than tradition software life cycle.

Table-1: Calculated data used to get simulated result (‘c: \data_simulation.xls)

| **Sr. No** | **Project Size(KLOC)** | **Effort( PM) ( VRS Model)** | **Duration (M) ( VRS Model)** | **Effort (PM) (CoCoMo)** | **Duration (M) (CoCoMo)** |
|---|---|---|---|---|---|
| 1 | 2.2 | 8.419087 | 4.258482 | 7.32308 | 5.327502 |
| 2 | 2.4 | 9.305117 | 4.392648 | 8.023646 | 5.515708 |
| 3 | 2.5 | 9.752364 | 4.457042 | 8.375041 | 5.606284 |
| 4 | 2.6 | 10.202304 | 4.51979 | 8.727141 | 5.865608 |
| 5 | 2.7 | 11.10991 | 4.640801 | 9.433349 | 5.865608 |
| 6 | 2.8 | 11.10991 | 4.640801 | 9.433349 | 5.865608 |
| 7 | 3.0 | 12.02730 | 4.756361 | 10.142086 | 6.029320 |
| 8 | 3.1 | 12.48949 | 4.812287 | 10.497351 | 6.108721 |
| 9 | 3.3 | 13.42053 | 4.920749 | 11.209587 | 6.263023 |
| 10 | 3.5 | 14.36009 | 5.025062 | 11.92399 | 6.411803 |
| 11 | 3.6 | 14.83292 | 5.075782 | 12.28196 | 6.484279 |
| 12 | 3.8 | 15.78447 | 5.174567 | 12.99938 | 6.625683 |
| 13 | 4.0 | 16.74356 | 5.270059 | 13.71870 | 6.762681 |
| 14 | 4.2 | 17.70987 | 5.362527 | 14.43981 | 6.895620 |
| 15 | 4.3 | 18.19565 | 5.407700 | 14.80102 | 6.960667 |
| 16 | 4.6 | 19.66303 | 5.539293 | 15.88714 | 7.150516 |
| 17 | 4.7 | 20.1554 | 5.581925 | 16.24998 | 7.212138 |
| 18 | 4.8 | 20.64935 | 5.623978 | 16.6132 | 7.272978 |
| 19 | 5.2 | 22.64033 | 5.786772 | 18.06981 | 7.509004 |
| 20 | 5.4 | 23.64459 | 5.865155 | 18.80024 | 7.622933 |
| 21 | 5.5 | 24.14883 | 5.903648 | 19.16597 | 7.678948 |
| 22 | 5.7 | 25.16141 | 5.979303 | 19.89842 | 7.789168 |
| 23 | 5.8 | 25.66972 | 6.01649 | 20.26513 | 7.843407 |
| 24 | 5.9 | 26.17934 | 6.053268 | 20.63215 | 7.897088 |
| 25 | 6.0 | 26.69026 | 6.089646 | 20.99949 | 7.950223 |
| 26 | 6.1 | 27.202462 | 6.125636 | 21.36713 | 8.002830 |
| 27 | 6.2 | 27.71592 | 6.161249 | 21.73508 | 8.054921 |
| 28 | 6.3 | 28.23063 | 6.196494 | 22.10332 | 8.106509 |
| 29 | 6.4 | 28.74656 | 6.231381 | 22.47186 | 8.157608 |
| 30 | 6.5 | 29.2637 | 6.265919 | 22.84068 | 8.208228 |
| 31 | 6.6 | 29.78204 | 6.300116 | 23.20978 | 8.258383 |
| 32 | 6.7 | 30.30156 | 6.333982 | 23.57917 | 8.308083 |
| 33 | 6.8 | 30.82224 | 6.367524 | 23.94883 | 8.357343 |
| 34 | 7.1 | 32.39112 | 6.466284 | 25.05943 | 8.502548 |
| 35 | 7.2 | 32.91632 | 6.498606 | 25.43016 | 8.550129 |
| 36 | 7.4 | 33.96999 | 6.562393 | 26.17238 | 8.644113 |
| 37 | 7.5 | 34.49843 | 6.593871 | 26.54387 | 8.690533 |
| 38 | 7.6 | 35.02793 | 6.625080 | 26.915611 | 8.736583 |
| 39 | 7.7 | 35.55848 | 6.656027 | 27.287593 | 8.782270 |
| 40 | 7.8 | 36.09007 | 6.686716 | 27.65982 | 8.827602 |
| 41 | 8.0 | 37.15629 | 6.839351 | 28.40497 | 8.917228 |
| 42 | 8.1 | 37.69091 | 6.777289 | 28.77790 | 8.961537 |
| 43 | 8.2 | 38.22652 | 6.807000 | 29.151071 | 9.005518 |
| 44 | 8.3 | 38.76312 | 6.836479 | 29.52446 | 9.049179 |
| 45 | 8.4 | 39.30068 | 6.86573 | 29.89808 | 9.092524 |

| | | | | | |
|---|---|---|---|---|---|
| 46 | 8.5 | 39.83921 | 6.894757 | 30.27191 | 9.13556 |
| 47 | 8.6 | 40.37868 | 6.923566 | 30.64597 | 9.178292 |
| 48 | 8.7 | 40.91913 | 6.95216 | 31.02024 | 9.220727 |
| 49 | 8.9 | 42.00272 | 7.008719 | 31.76943 | 9.304726 |
| 50 | 9.3 | 44.18087 | 7.119432 | 33.27033 | 9.469383 |
| 51 | 9.5 | 45.27527 | 7.173641 | 34.02199 | 9.550118 |
| 52 | 9.6 | 45.82377 | 7.20047 | 34.39813 | 9.590102 |
| 53 | 9.7 | 46.37313 | 7.22712 | 34.77445 | 9.629837 |
| 54 | 9.8 | 46.92334 | 7.253594 | 35.15097 | 9.669326 |
| 55 | 10.7 | 51.9123 | 7.484391 | 38.5481 | 10.01431 |
| 56 | 11.0 | 53.5896 | 7.558535 | 39.68371 | 10.12541 |
| 57 | 11.6 | 56.96467 | 7.703009 | 41.95955 | 10.34227 |
| 58 | 11.8 | 58.0956 | 7.750096 | 42.71949 | 10.41305 |
| 59 | 12.1 | 59.79737 | 7.819773 | 43.8606 | 10.51788 |
| 60 | 12.2 | 60.36604 | 7.842751 | 44.24129 | 10.55248 |
| 61 | 12.5 | 62.07624 | 7.910967 | 45.38428 | 10.65526 |
| 62 | 12.7 | 63.21981 | 7.955861 | 46.14704 | 10.72296 |
| 63 | 13.0 | 64.94022 | 8.022356 | 47.29231 | 10.82332 |
| 64 | 14.4 | 73.04589 | 8.320269 | 52.65391 | 11.27415 |
| 65 | 14.7 | 74.79867 | 8.381655 | 53.80631 | 11.36728 |
| 66 | 14.8 | 75.38412 | 8.401938 | 54.1907 | 11.39807 |
| 67 | 15.8 | 81.27081 | 8.600078 | 58.04167 | 11.69934 |
| 68 | 16.0 | 82.45498 | 8.63873 | 58.81336 | 11.7582 |

**PROPOSED PROGRAM WRITTEN IN MATLAB 2.0**

```
clc;
clear all;
N= xlsread('c:\data_simulation.xls',1);
%VRS method
for i=1; size (N)
        effort1(i)=3.4 * (N(i))^1.15;
        duration1(i) = 2.2 * (effort1(i))^0.31;
%COCOMO
effort2(i)=3.2 * (N(i))^1.05;
duration2(i)=2.50 * (effort2(i))^0.38;
end
figure(1);
plot(effort1, effort2);
figure(2);
plot(duration1, duration2);
```

## IV RESULT THROUGH SIMULATION APPROACH

Obtained results are expressed by curves that give the effort and time duration values over VRS model and COCOMO Intermediate -organic model as shown in Figure 2 and Figure3 respectively.

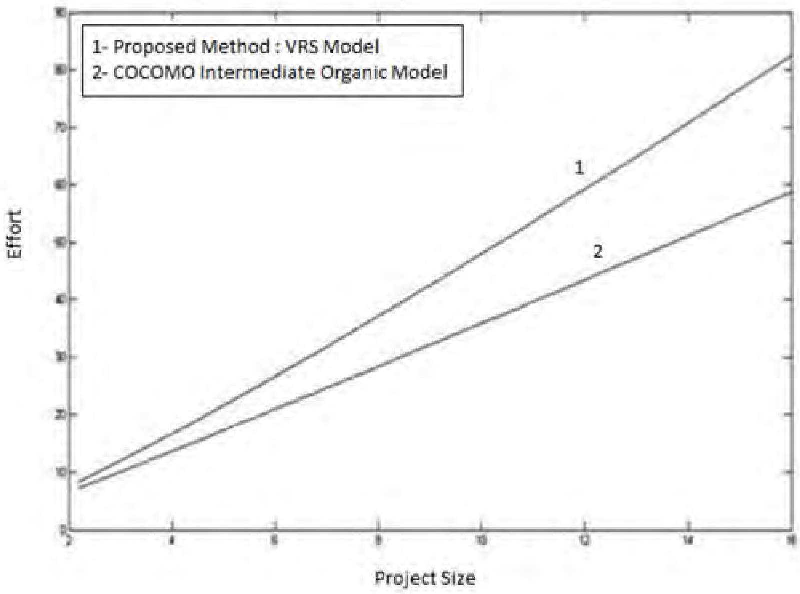


Figure 2: Graph shows effort effects.

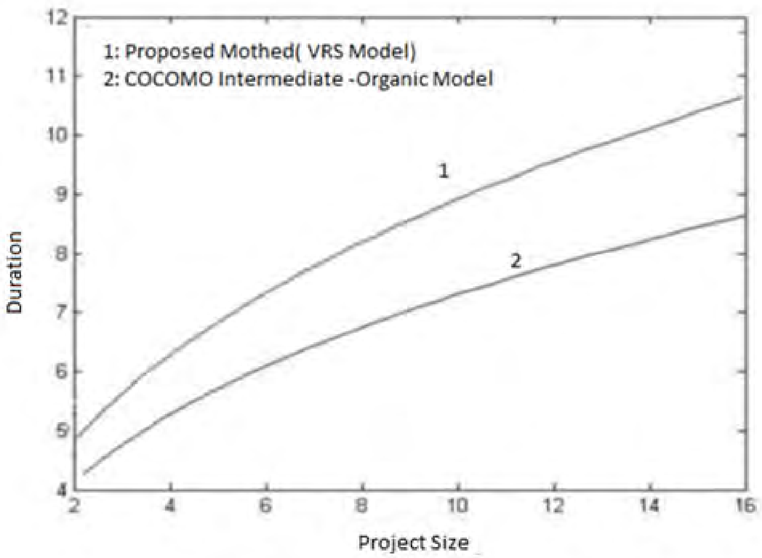


Figure 3: Graph shows duration effects.

## V CONCLUSION

Simulation is one of the productivity control methods that enable software developers to take corrective actions and perform risk analysis in terms of effort cost and time to perform continuous process improvement and optimization. This paper has introduced a software process simulation modeling approach for the prediction of IVR software production cost and delivery times. In both the figure the curve of both models are closed to each other with respect of effort and duration. Therefore VRS model is accurate and may be used to estimate the effort and time duration for only IVR software.

## REFERENCES

[1] Devesh Kumar Srivastava, Durg Singh Chauhan, Raghuraj Singh, "VRS Model: A Model for Estimation of Efforts and Time Duration in Development of IVR Software System" Vol 5, Issue 1 pp. 27-46 January 2012.
[2] L. Angelis, I.Stamelos, "A simulation tool for efficient Analogy based cost estimation" Empirical Software Engineering Vol. 5, pp. 35-68,2000, Kluwer Academic Publishers, Boston.
[3] Walt Scacchi, "Experience with software process simulation and modeling", The Journal of System and Software, Elsevier Vol.46, pp. 183-192, 1999.
[4] Benediktsson, O. and Dalcher, D,"Effort Estimation in Incremental Software Development". IEEE Proc. Software., Vol. 150, no. 6, pp. 351-357, December 2003.
[5] Steinman J. SPEEDES, "A multiple synchronization environment for parallel discrete-event simulation", International Journal in Computer Simulation Vol.2, pp. 25 l- 286, 1992.
[6] Nicol, D.M, "The cost of conservative synchronization in parallel discrete-event simulations", Journal of the ACM 40(2): 304- 333, 1993.
[7] Martin Shepperd, Chris Schofield, "Estimating Software Project Effort Using Analogies," *IEEE* Transactions on Software Engineering, vol. 23, no. 12, pp.736-743, 1997.
[8] Pidd, M. Computer Simulation in Management Science. Wiley, Chichester, 3rd edition. 1992.
[9] Helsgaun, K, Discrete Event Simulation in Java, Writings on Computer Science, Roskilde University 2000.
[10] Pankaj Jolte, An Integrated Approach to Software Engineering Third edition Narosa Publishing house New Delhi.
[11] Yogesh Singh, K.K.Aggarwal "Software Engineering" Third edition, New Age International Publisher Limited New Delh
[12] Pressman, R.S. and D. Software Engineering: A Practitioner's Approach. 5 ed. 2000, Maidenhead: McGraw-Hill.
[13] Parvinder S. Sandhu, PorushBassi, and Amanpreet Singh Brar, "Software Effort Estimation Using Soft Computing Techniques", World Academy of Science, Engineering and Technology 46 2008.
[14] Ceric, V., and R. Paul. Diagrammatic representation of the conceptual model for discrete eventsystems. Mathematics and Computers in Simulation Vol.34, pp.317- 324, 1992.
[15] Mitat Uysal, Estimation of the Effort Component of the Software Projects Using Simulated Annealing Algorithm World Academy of Science, Engineering and Technology 412008.
[16] Saleem Basha, Dhavachelvan P, "Analysis of Empirical Software Effort Estimation Models", International Journal of Computer Science and Information Security, Vol. 7, No. 3, 2010.