Simulator for Time Estimation of Software Development Process

P. K. Suri[1], Bharat Bhushan[2]

[1]*Department of Computer Science & Applications, Kurukshetra University, Kurukshetra (Haryana), India.*
[2]*Department of Computer Science & Applications, Guru Nanak Khalsa College, Yamuna Nagar (Haryana), India*

Summary
Estimating the time is a critical activity. Not only does this time impact on the technology solution but also impacts the project management solution. Keeping in view an attempt has been made to design a simulator for time /effort estimation of software development process using Erlang-6 distribution [9]. It is assumed that there is a long term fixed average time required to complete an activity and it is assumed that there are six major activities of a standard system development process in which each activity is mutually independent and follows a negative exponential distribution with average time of completion say β. The value of β depends on the size of the process. The input for the simulator has been derived by using an algorithm for generating pseudo random numbers which follows Erlang-6 distribution. The total mean time of software development has been found and computed by varying the average time of completion of an activity. This simulator will be an asset to affordably keep track of time during the process of development and thus to satisfy the client in this era of competitive market of software.

Keywords:
Erlang distribution, Simulation, Activity, Time estimation, Software development process.

1. Introduction

As the role of software is expanding rapidly in many aspects of modern life, quality and customer satisfaction become the main goal for software developers and an important marketing consideration for organizations. However, quality by itself is not a strategy that will ensure a competitive advantage. There are also other factors like budget and delivery time that must be considered in relation to quality [1].

"Software crisis", a crisis with software being delivered over budget and much later than its original delivery dead line, or not been delivered at all, or a product delivered on time, but with significant quality problems. To discuss critical issues in software development [3] the NATO conference was held in Germany, led to the birth of software engineering & new theories for understanding of software development. Software development process has an associated life cycle e.g. Waterfall model, which was developed by Royce [4] and Spiral life cycle model developed by Boehm [8]. Generally System Development Life Cycle has following activities:

1. System/information engineering and modeling
2. Software Requirements Analysis
3. Systems Analysis and Design
4. Code Generation
5. Testing
6. Maintenance

Doban, O.; Pataricza, A[2] investigated cost efficiency and development time for software development process and found that these are the most important factors for the software product quality and emphasized that there must be strict adherence to the optimum strategy being followed during the design of the software.

A. Hosalkar and B. Bowonder [10] emphasized to reduce software development cycle time. If more & more pressure is induced to reduce the cycle time, the firms have to anticipate and adopt an approach to prevent, what we call, a software development failure and therefore software risk assessment becomes an integral part of software quality.

Ali Arifoglu[5] applied a process to find out the efforts and development time for the software product to be developed. The process starts with planning the activity and that is continuously refined throughout the development process.

Victor A. Clincy [6] proposed that software development process is a major area impacting organizations' ability to increase development cycle time

Since time estimation is stochastic in nature so here simulator [7] has been designed to evaluate time estimation for various stages of system development process. There are six major activities for software development process, requires specific time period for their completion. Each process is assumed to be mutually independent and each of these times may reasonably be assumed to follow a negative exponential distribution. These activities are performed in a sequence. Suppose each activity is taking an average of β units of time then

Manuscript received July 5, 2007
Manuscript revised July 25, 2007

the total time for the process will be Erlang distributed with mean value of mβ units of time. m is the no. of activities. Further if the average of each of these six times is the same, the sum is said to be Erlanged-6 distributed. An Erlang-m distributed random variable is the sum of m-independent and identically distributed exponential distribution and its density function.

$$h_m(t) = 1/(m-1)! * (1/\beta)^m * t^{m-1} * e^{-t/\beta}$$

To generate Erlang variants with a mean time of mβ units, we have to generate m-random observations from an exponential distribution, with a mean time of β, and add them.

Simulation of Time Estimation for Software Development Process

m: no. of activities
beta: activity time
runs: no. of times the simulation process is repeated
r: random number
Sndy: a procedure to generate random number
p: product of random numbers
erlang: composite time for completion of software development process
avg: mean time for an activity
sd: deviation of composite time from its mean

Algorithm to compute composite time estimation for software development process

```
1) Read beta, m
2)  Read runs
3) For j =1 to runs
   [begin]
      i) p =1
      ii)  for i = 1 to m
           [begin]
              r = Sndy (dum)
             [generate random no.]
              p = p*r
           [end]
      iii) erlang [j] = beta * alog (p)
          [generate erlang variate for jth interval]
       iv) write (erlang[j])
                [end]
```

Algorithm to compute mean time estimation for an activity of software development process

```
1) s = 0
2) for k = 1 to runs
   s = s + erlang(k)
3) avg = s/ runs
4) write (avg)
5) end
```

Algorithm to compute the variability among estimated composite time and mean time for software development process

```
1) s = 0
2) for k=1 to runs
3)  s = s + (erlang[k] – avg )**2
4) sd = sqrt (s/ runs)
5) write (sd)
6) end
```

1. Implementation

2.

3. On executing with

A.) No. of runs = 20000

4. Sample input data

1 no. of activities m = 6
2 Length of time interval for each activity (β) = 5 units

5. Sample output

Composite Time for completion of the Software Development Process = 30.47968

B.) No. of runs = 20000

6. Sample input data

3 no. of activities m = 6
4 Length of time interval for each activity (β) = 10 units

7. Sample output

Composite Time for completion of the Software Development Process = 60.95935

C.) No. of runs = 20000

8. Sample input data

5 no. of activities m = 6

6 Length of time interval for each activity (β) = 15 units

9. Sample output

Composite Time for completion of the Software Development Process = 91.43903

D.) No. of runs = 20000

10. Sample input data

7 no. of activities m = 6

8 Length of time interval for each activity (β) = 20 units

11. Sample output

Composite Time for completion of the Software Development Process = 121.9187

E.) No. of runs = 20000

12. Sample input data

9 no. of activities m = 6

10 Length of time interval for each activity (β) = 25 units

13. Sample output

Composite Time for completion of the Software Development Process = 152.3984

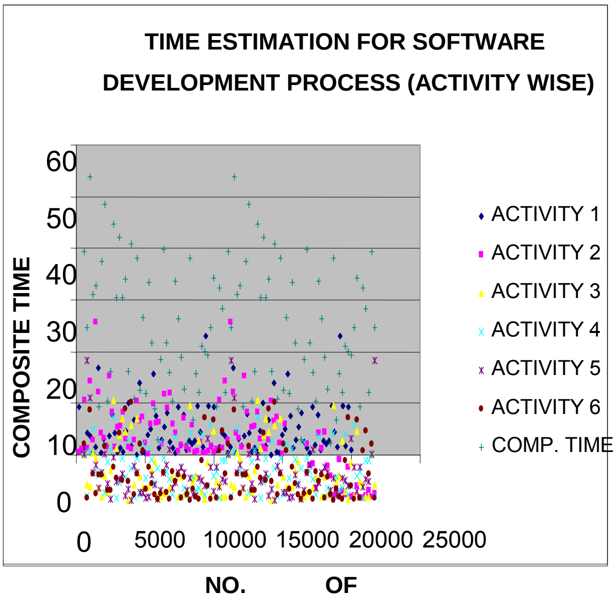
13.1

13.2 Discussion and conclusion

13.3 The motivation for incorporating simulation into software development process for time estimation with simulated data lead to useful information for the completion of the project well in time. Timing being very crucial in software development, if a delay happens in the development activity, the market could be taken over by the competitors. Also if a 'bug' filled product is launched in a short period of time (quicker than the competitors), it may affect the reputation of the company. So, there should be a tradeoff between the development time and the quality of the product. Customers don't expect a bug free product but they expect a user-friendly product. That results in Customer Ecstasy.

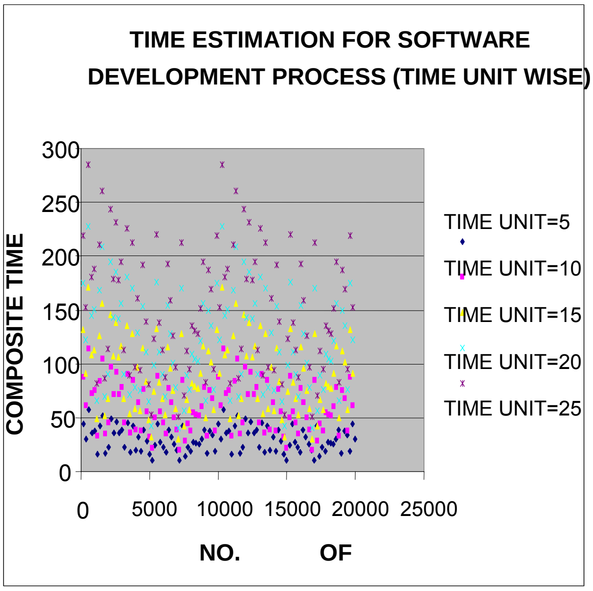
The above simulator was validated with the help of pseudo random numbers for the time to be taken to complete the six activities of software development process. The result shows that the composite time for the completion of project is very-2 close to the estimated values.

The graph 1 is plotted between no. of runs and composite time estimation for software development process activity wise. The total composite time is found to be the sum of the time of each activity.



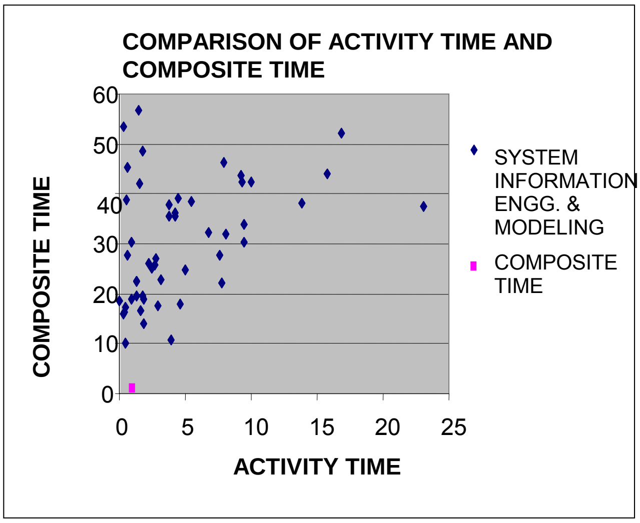
Graph No. 1

The graph 2 shows the relationship between no. of runs and composite time estimation for software development process time unit wise. it is found that if the units of time is increased then the overall time for completion of the software process will be proportionally increased and the total time for completion is again the sum of the times for each activity.



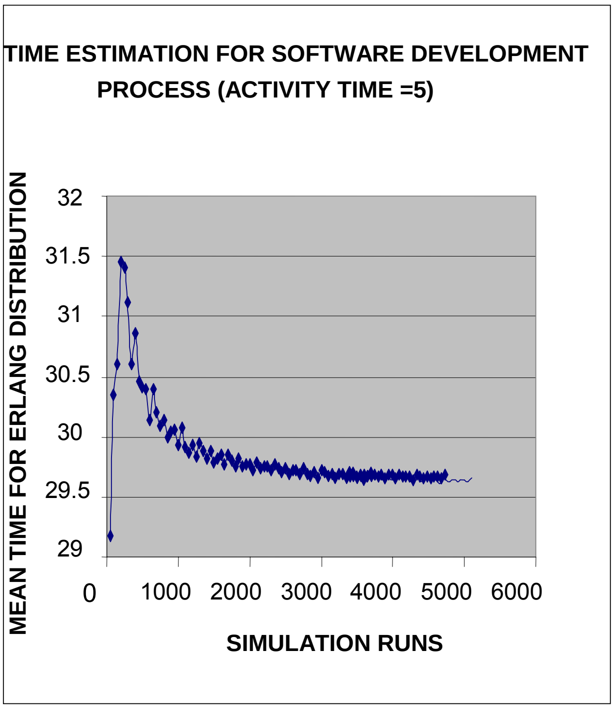
Graph No. 2

The graph 3 shows a comparison between activity time and composite time.



Graph No. 3

The graph no. 4 shows the mean time for an activity to be completed using Erlang Distribution for β=5 and it is found that the mean time becomes constant as the number of runs increases.



Graph No. 4

References:

[1] Boehm, Barry W. and Hoh In, "Identifying Quality-Requirements Conflicts", *IEEE Software*, pp. 25-35 (1996).

[2] Doban, O; Pataricza, "A Euromicro Conference- 2001, Proceedings", Vol. 27, pp. 208-213 (2001).

[3]J.N. Buxton, P. Naur and B. Randell, Software Engineering, Petrocelli, "Report on two NATO Conferences held in Garmisch, Germany (October 1968) and Rome, Italy (October 1969)", (1975).

[4] Royce, "The Software Life Cycle Model (Waterfall Model), Managing the development of large software systems: Concepts and Techniques", W. Royce in: Proc. WESCON (1970).

[5] Ali Arifoglu, "ACM SIGSOFT Software Engineering Notes" Vol. 18, Issue 2, pp. 96 – 105 (1993).

[6] Victor A Clincy, "Software Development Processes" , Journal of Computing Sciences in Colleges, Vol. 19, Issue 2 , pp. 278 – 287(2003).

[7] A.M. Law, W.D. Kelton, “Simulation Modeling and Analysis”, McGraw Hill Int. Series, New York, (2000).

[8] Boehm, “A spiral Model for Software Development and enhancement”, Computer, (1988)

[9] J.S. Hunter, and T.H. Naylor, “Experimental Design for Computer Simulation Experiments”, Management Science, Vol. 16, No. 7, pp. 422-434 (1970).

[10] A. Hosalkar and B. Bowonder, “Software Development Management: Critical success factors”, Academy Street, Jersey City, NJ 07306, USA.

Dr. P.K. Suri received his Ph.D. degree from Faculty of Engineering, Kurukshetra University, Kurukshetra, India and master’s degree from Indian Institute of Technology, Roorkee (formerly known as Roorkee University), India. He is working as Professor in the Department of Computer Science & Applications, Kurukshetra University, Kurukshetra - 136119 (Haryana), India since Oct. 1993. He has earlier worked as Reader, Computer Sc. & Applications, at Bhopal University, Bhopal from 1985-90. He has supervised five Ph.D.’s in Computer Science and thirteen students are working under his supervision. He has more than 100 publications in International / National Journals and Conferences. He is recipient of 'THE GEORGE OOMAN MEMORIAL PRIZE' for the year 1991-92 and a RESEARCH AWARD –“The Certificate of Merit – 2000” for the paper entitled ESMD – An Expert System for Medical Diagnosis from INSTITUTION OF ENGINEERS, INDIA. His teaching and research activities include Simulation and Modeling, SQA, Software Reliability, Software testing & Software Engineering processes , Temporal Databases, Ad hoc Networks, Grid Computing , and Biomechanics.

Bharat Bhushan received the MSc. (Physics), Punjab Univ. Chandigarh in and M.Sc. (Comp. Sc.), MCA degrees from Guru Jambeshwar University. 2001 respectively. Presently working as Head, Department of Computer Science and Applications, Guru Nanak Khalsa College, Yamuna Nagar (affiliated to Kurukshetra University, Kurukshetra- Haryana, India) and senior most teacher of computer science in Haryana since 1984. He is a member of Board of Studies of Computer Science, Kurukshetra University. His research interest includes Software engineering, Digital electronics and Simulation Experiments.