Journal of Information & Communication Technology
Vol. 1, No. 2, (Fall 2007) 21-34

Impact of Analyst and Programmer Capability on Software Development Cost

Ghulam Qadir Memon *
Professor and Associate Dean, Dadabhoy Institute of Higher Education, Karachi, Pakistan.
S.M. Ahmed Bari *
Assistant Professor, Dadabhoy Institute of Higher Education, Karachi, Pakistan.

ABSTRACT

Software development cost can be greatly influenced by the capability of the development team, especially, analysts and programmers. This paper intends to investigate the impact of analyst and programmer capability on the software development cost. The task is accomplished by developing various case scenarios involving different levels of analyst and programmer capabilities. Cost drivers and corresponding effort multipliers and average labor rates based on varying analyst and programmer capability are mainly used in determining the software development cost using COnstructive COst MOdel (COCOMO). The percent increase/decrease in software development cost is used as the primary measure of effectiveness for this study. Case Scenario 1 related to nominal ratings of personnel capability is taken as the base case and the results of all other case scenarios are compared against the result of this case scenario. The results of this research indicate that it does not pay off to use lower capability analysts and programmers for software development. This does not mean that low capability analysts and programmers should not be hired at all. Rather, this is indicative of the need for hiring high rating analysts and programmers for complex and large projects. However, for further validation of these results, more data must be collected and results need to be analyzed and verified.

Inspec Classification: C6110B

1) INTRODUCTION

The software development and maintenance are very expensive and costly activities, despite that people are willing to pay for it in the expectation of enhanced value of processed information. In order to benefit from value-based situation, it would be desirable to enhance our understanding and be able to deal with the economic aspects of software development, use and maintenance. Boehm recommends following proper roadmap in software economics, which can lead to fundamental improvements in software design and engineering (B. Boehm, K. Sullivan (2000).

* The material presented by the authors does not necessarily portray the viewpoint of the editors and the management of the Institute of Business and Technology (BIZTEK) or Dadabhoy Institute of Higher Education,Karachi.

* Dr. Ghulam Qadir Memon : gqmemon@yahoo.com
* S.M. Ahmed Bari : ahmedbari@yahoo.com

Software development cost is influenced by the software size, in addition to, various cost drivers and scale factors. The software size is based on software requirements, user interface requirements, diagnostic test code, etc. The most common measure of software size is Source Lines of Code (SLOC). It is also expressed in function points, which are more independent of development platform and language. In the original version of COnstructive COst MOdel (COCOMO) known as COMOCO 81, Boehm recommends the use of 15 cost drivers based on the product, personnel, platform, and project attributes (B. Boehm, 1981). The number of cost drivers is increased to 17 in COCOMO II to take into account the impact of attributes such as software reuse and personnel continuity. In addition, COCOMO II allows the use of two additional user-defined cost drivers, if required (B. Boehm, 2000 B). COCOMO 81 uses 3 modes of software development; namely, organic, semi-detached and embedded; to account for varying development conditions and degree of innovation required to develop the given product ((B. Boehm, 1981, K. Branle, 2004, A. Jayaraman, 2007, B. Boehm, 1984).

On the contrary, COCOMO II uses 5 scale factors to account for precedence of the development environment and project conditions, software development flexibility, architecture/risk resolution, cohesion of development team and process maturity based on the Capability Maturity Model (CMM). Three development modes of OCOMO 81 provide three point estimates always resulting in diseconomy of scale, while five scale factors of COCOMO II give range estimates to choose among various development strategies.

In this study, in order to estimate software development cost, first of all, the nominal development effort is estimated in man-months (MM) or person-months (PM) using the project size and ratings of scale factors. This effort is then converted to adjusted development effort by multiplying the nominal effort with the Effort Adjustment Factor (EAF). EAF is obtained by multiplying all the effort multipliers (EMs) together. These effort multipliers are determined from qualitative ratings of cost drivers. The nominal and adjusted effort values are multiplied by appropriate monthly average labor rates to arrive at nominal and adjusted development cost estimates. From these two estimates, the percent increase/decrease in the software development cost is determined, which is used as the primary measure of effectiveness (MOE) for investigating the impact of analyst and programmer capability on the software development cost.

2) SOFTWARE COST ESTIMATION

Software cost estimation refers to predicting the resources required for a software development process. It includes an approximate judgment of the costs for a project involving many variables. Often software cost estimation is measured in terms of effort. Software project costs consist of hardware, travel, training, and effort costs. Effort costs are generally the largest and least predictable development effort.

Cost estimation should be done throughout the Software Development Life Cycle (SDLC) to allow for refinement. Preliminary estimates help in determining the feasibility of a project, while detailed estimates are essential for assisting with project planning. According to Johnson, the actual effort for individual tasks should be compared with estimated and planned values, enabling project managers to reallocate resources when necessary (K. Johnson, 1998). Software cost estimation is important because without it making hardware-software tradeoff analyses becomes difficult for software analysts and system engineers. This also provides project managers a clue to know what is going on. In this regard, Sommerville raises some very important fundamental questions, which include (I. Sommerville, 2004): How much effort and time is required to complete an activity or a project? What is the total cost of its completion? How much price should be charged to the customer/client?

For estimating software development cost, the important issues include assumptions made,

cost driver ratings including Analyst Capability (ACAP) and Programmer Capability (PCAP), effort distribution, average salaries (labor rates) rates of development personnel, etc.

2.1) Assumptions

Generally, assumptions are made to conduct a research or study. For the sake of this study, the following assumptions have been made:

1. Waterfall distribution of phases and effort is employed.
2. Four phases of software development are considered. These include (1) Product Design (PD), (2) Detailed Design (DD), (3) Code and Unit Test (CUT), and (4) Integration and Testing (IT).
3. The ACAP applies to the first two phases, i.e., PD and DD; while the PCAP is applied to the remaining two phases of CUT and IT.
4. Average salaries (labor rates) of analysts and programmers provided by the participating software houses/organizations based on their ratings are used.
5. ACAP and PCAP ratings range from very low to very high.
6. Test scenario pertaining to the nominal ratings for ACAP and PCAP is used as the base scenario and all other scenarios are compared against this one.
7. Software projects of standard sizes from small to very large are used herein.
8. The ratings for all cost drivers except ACAP and PCAP are assumed to be nominal.
9. The ratings for all five scale factors are also assumed to be nominal.

2.2) ACAP and PCAP ratings

The core functions performed by analyst and programmer include systems analysis, design and development; applications programming and database analysis. The capability of analyst and programmer may not directly affect the quality of a software product (M. Paulk, 2006). However, it may affect the software development productivity, which in turn, directly impacts the cost of software product. ACAP and PCAP ratings can be based on the years of experience, because with the passage of time they develop better understanding about problem domains, programming languages, development platforms, etc. According to Boehm, ACAP and PCAP ratings vary from very low to very high (B. Boehm, 1981 and B. Boehm, 2000 B). For the sake of this study, ACAP and PCAP ratings are based on the experience as per details given in Table1.

Table 1.
ACAP and PCAP rating details

Rating	PCAP	**ACAP**	
	Programming Experience (years)	Programming Experience (years)	Analysis & Design Experience (years)
Very Low	0-1	2	0-1
Low	1-2	2	1-2
Nominal	2-3	2	2-3
High	3-4	2	3-4
Very High	> 4	2	> 4

It is note worthy, normally, analysts are not hired by the software houses in Karachi until they possess about 2 years of programming experience. Furthermore, some software houses prefer to hire programmers who possess at least one year experience.

2.3) Cost drivers

COCOMO II uses 17 cost drivers, in addition to, 2 user defined cost drivers, if required. In this study, all cost drivers except ACAP and PCAP are assumed to be nominal. The qualitative ratings of ACAP and PCAP vary from very low to very high. These are converted to corresponding quantitative effort multiplier values as described in section 3.2. These effort multipliers are multiplied by the nominal effort to estimate the adjusted effort for determining the adjusted cost so that the appropriate price could be charged to the customer.

2.4) Average labor rates

To estimate software development cost, the effort is multiplied by average salaries (labor rates) of development team. Since the focus of this study is on the impact of analyst and programmer capability, hence, average labor rates for them will be used. Based on the ratings of ACAP and PCAP in section 2.2, the data about analyst and programmer salaries was collected from various software houses in Karachi and their average rates along with minimum and maximum are shown in Table 2.

Table 2.
Average labor rates based on ACAP and PCAP ratings.

Rating	Average Labor Rate (Rs./Month) Based on					
	PCAP			ACAP		
	Min.	Max.	Avg.	Min.	Max.	Avg.
Very Low	15000	25000	19333	18000	25000	23667
Low	18000	30000	21833	21000	31000	29000
Nominal	22000	40000	27000	27500	42500	34333
High	28000	47500	32083	35000	50000	39667
Very High	32000	52500	37917	43000	55000	46000

It must be noted that for the same rating, usually analysts are paid more than programmers because the formers possess more experience and have more bearing on the development effort.

2.5) Software development cost estimation

To estimate software development cost, first of all, the development effort is estimated. There are many techniques and models for software effort estimation. However, due to openness and in-depth coverage, COCOMO/COCOMO II are most widely used. In this study, COCOMO II will be used for effort estimation (B. Boehm, 2006). The equation for nominal effort estimation is given by:

$$E_n = a\ S^e \qquad (1)$$

Where E_n is the nominal effort and a is a constant that can be calibrated according to the local conditions. In the absence of the calibrated value, Boehm et al. suggest a value of 2.94 (B. Boehm, 2006), which is used herein. S stands for the size of the software product to be developed in Kilo Source Lines of Code (KSLOC). The products can be classified as small, intermediate, medium, large, and very large based on their sizes as per details \ given in Table 3 (B. Boehm, 1981).

Table 3.
Classification of software products based on size (B. Boehm, 1981).

Classification	Size (KSLOC)
Small (S)	2
Intermediate (I)	8
Medium (M)	32
Large (L)	128
Very Large (VL)	512

In Eq. 1, e represents the exponential constant to be determined from the ratings of a set of 5 Scale Factors (SF) that determine the economies/diseconomies of scale for the software product under development (B. Boehm, 2006, S. Chulani, 1999, Softstar Systems, 2005). These scale factors replace the development modes in original COCOMO, which always give diseconomies of scale. The ratings for these scale factors vary from very low to extra high, for details see Boehm et al. (B. Boehm, 2006). Mathematically, e is determined by the formula:

$$e = b + 0.01 \sum_{j-1}^{5} SF_j \qquad (2)$$

In Eq. 2, b is a constant and a value of 0.91 is recommended by Boehm et al. [(B. Boehm, 2006). The term $\sum SF_j$ represents the summation of scale factors. Since nominal ratings are assumed in this study, the sum thereof is given in Table 4 (B. Boehm, 2006).

Table 4.
Summation of nominal scale factors.

Scale Factor	Value
Precedentedness (PREC)	3.72
Development Flexibility (FLEX)	3.04
Architecture/Risk Resolution (RESL)	4.24
Team Cohesion (TEAM)	3.29
Process Maturity (PMAT)	4.68
$\sum SF_j$	**18.97**

Substituting values in Eq. 2, we get:

$$\mathbf{e} = 0.91 + 0.01\ (18.97) = \mathbf{1.0997}$$

For example, for a medium size project (32 KSLOC), putting the values in Eq. 1, we get:

$$\mathbf{E_n} = 2.94\ (32)1.0997 = \mathbf{132.91\ MM}$$

Table 5 shows nominal effort computation for standard size products. These nominal efforts can be adjusted by multiplying with appropriate effort multipliers. In this study,all cost drivers are assumed to be nominal except ACAP and PCAP. Their phase-wise ratings will be used to estimate the adjusted effort. Finally, this adjusted effort will be multiplied by respective average labor rates to estimate software development cost.

Table 5.
Nominal effort computation for standard size projects.

Classification	Size (KSLOC)	Nominal Effort (MM)
Small	2	6.30
Intermediate	8	28.94
Medium	32	132.91
Large	128	610.44
Very Large	512	2803.70

3) DEVELOPMENT OF CASE SCENARIOS

In order to facilitate investigating the impact of analyst and programmer capability on the software development cost, various case scenarios have been developed. These are based on phase-sensitive effort multipliers reflecting capability of analysts and programmers, phase-wise effort distribution, and average labor rates pertaining to capability of analysts and programmers. Keeping in view, the nature of the problem under consideration and solution required, these cases cover reasonable range of variation regarding analyst and programmer capability.

3.1) Phase-wise effort distribution

In order to estimate software development cost, nominal effort will be distributed phase-wise and adjusted effort will be estimated by summing up phase-wise adjusted effort. In this study, four distinct phases of waterfall development are considered. These include:

- Product Design (PD)
- Detailed Design (DD)
- Coding and Unit Test (CUT)
- Integration and Testing (IT)

Based on recommended values of phase-wise waterfall distribution of effort by Boehm (B. Boehm, 1981), Table 6 shows details of distribution scheme used in this study. Table 6 further shows that the distributed development effort varies with development phase as well as the project size resulting in variation of percent increase/decrease in development cost. For larger projects, the percentage distribution of programming phase is less as compared to smaller projects. The case is reverse in the case of integration and testing phase.

Table 6.
Phase-wise effort distribution details (based on B. Boehm, 1981).

	Development Phase			
Project	**PD**	**DD**	**CUT**	**IT**
Small	17	27	37	19
Intermediate	17	26	35	22
Medium	17	25	33	25
Large	17	24	31	28
Very Large	17	23	29	31

3.2) Phase-sensitive effort multipliers

In this study, all other cost drivers are assumed to be nominal except ACAP and PCAP. Phase-sensitive effort multipliers suggested by Boehm (B. Boehm, 1981) will be used in this study as given in Table 7. According to the assumption made herein, the analyst capability will be applied to PD and DD phases, whereas the programmer capability includes CUT and IT phases of software development. As such, these phase-sensitive effort multipliers will be used in conjunction with phase-wise effort distribution given in Table 6. It must be noted that the effort multiplier value for nominal rating is 1, which means the effort will remain unaffected.

Table 7.
Phase-sensitive effort multipliers for ACAP and PCAP (B. Boehm, 1981).

Rating	**ACAP**		**PCAP**	
	PD	D D	CUT	IT
Very Low (VL)	1.80	1.35	1.50	1.50
Low (L)	1.35	1.15	1.20	1.20
Nominal (N)	1.00	1.00	1.00	1.00
High (H)	0.75	0.90	0.83	0.83
Very High (VH)	0.55	0.75	0.65	0.65

3.3) Complete list of case scenarios

ACAP and PCAP each has five ratings from very low to very high, giving a total of (5! =) 25 or (5 x 5 =) 25 case scenarios. Thus, a complete list of all the case scenarios is given in Table 8. Furthermore, each of the 25 case scenarios is tested for a cluster of five software project sizes. In Table 8, C.S. stands for case scenario and ALR represents average labor rate in Rs./MM (rupees per man-month).

4) TESTING CASE SCENARIOS

Each of five ratings of ACAP was tested for five ratings of PCAP. Furthermore, each case scenario was tested for five project sizes. This section focuses on various aspects of testing these case scenarios under identical conditions.

4.1) Cost estimation for ACAP and PCAP

In order to compute adjusted estimated cost for each case scenario, first adjusted cost is estimated for ACAP and PCAP separately. Since there are two phases each related to ACAP and PCAP, the formula for estimating this effort is:

$$C_{A/P} = 3\sum_{i-1}^{2} \frac{E_n P_i}{100} ALR_{A/P} \qquad (3)$$

Where A/P refers to ACAP or PCAP based respectively and $\mathbf{C_{A/P}}$ is adjusted cost based on ACAP or PCAP. $\mathbf{E_n}$ is nominal effort, $\mathbf{P_i}$ percent effort distribution for phase i, $\mathbf{EM_i}$ effort multiplier for phase i and $\mathbf{ALR_{A/P}}$ average labor rate for ACAP or PCAP.

Table 8.
Complete list of case scenarios.

C.S.	ACAP				PCAP			
	EM		Rating	ALR	EM		Rating	ALR
	PD	DD			CUT	IT		
1	1.00	1.00	N	34333	1.00	1.00	N	27000
2	1.80	1.35	VL	23667	1.50	1.50	VL	19333
3	1.80	1.35	VL	23667	1.20	1.20	L	21833
4	1.80	1.35	VL	23667	1.00	1.00	N	27000
5	1.80	1.35	VL	23667	0.83	0.83	H	32083
6	1.80	1.35	VL	23667	0.65	0.65	VH	37917
7	1.35	1.15	L	29000	1.50	1.50	VL	19333
8	1.35	1.15	L	29000	1.20	1.20	L	21833
9	1.35	1.15	L	29000	1.00	1.00	N	27000
10	1.35	1.15	L	29000	0.83	0.83	H	32083
11	1.35	1.15	L	29000	0.65	0.65	VH	37917
12	1.00	1.00	N	34333	1.50	1.50	VL	19333
13	1.00	1.00	N	34333	1.20	1.20	L	21833
14	1.00	1.00	N	34333	0.83	0.83	H	32083
15	1.00	1.00	N	34333	0.65	0.65	VH	37917
16	0.75	0.90	H	39667	1.50	1.50	VL	19333
17	0.75	0.90	H	39667	1.20	1.20	L	21833
18	0.75	0.90	H	39667	1.00	1.00	N	27000
19	0.75	0.90	H	39667	0.83	0.83	H	32083
20	0.75	0.90	H	39667	0.65	0.65	VH	37917
21	0.55	0.75	VH	46000	1.50	1.50	VL	19333
22	0.55	0.75	VH	46000	1.20	1.20	L	21833
23	0.55	0.75	VH	46000	1.00	1.00	N	27000
24	0.55	0.75	VH	46000	0.83	0.83	H	32083
25	0.55	0.75	VH	46000	0.65	0.65	VH	37917

Next, cost for ACAP and PCAP is summed up to find the adjusted estimated cost for each case scenario. That is:

$$C_k = C_{A(k)} + C_{P(k)} \qquad (4)$$

Where $\mathbf{C_k}$ is the total adjusted software development cost, $\mathbf{C_{A(k)}}$ the adjusted development cost based on ACAP and $\mathbf{CP_{(k)}}$ the adjusted development cost based on PCAP for case scenario k and a particular size of the project.

4.2) Number of cost computations

First of all, three computations for nominal cost for each of 5 sizes of software product (small to very large) were performed giving a total of (5 x 3 =) 15 computations. To test each case scenario, all 5 sizes are considered. For each size, 3 adjusted cost computations are performed in addition to one percent increase/decrease in software development cost. This means a total of (5 x 4 =) 20 cost computations were performed for each case scenario. In other words, on the aggregate the number of cost computations for all 25 case scenarios and nominal cost was (20 x 25 = 500 + 15 =) 515.

4.3) Measure of effectiveness (MOE)

To evaluate and compare results, certain criteria or measures of effectiveness are required. In this case, the percent increase or decrease in software development cost was used as the primary MOE. The objective was to investigate how different ratings of analysts and programmers impact the development cost. The percent increase/decrease rather than absolute increase/decrease was chosen because of varying project sizes. Mathematically, the percent increase/decrease (change) in software development cost for case scenario k and a particular project size (PC_k) can be given by the following formula:

$$PC_K = \frac{C_k - C_n}{C_n} 100 \qquad (3)$$

Where $\mathbf{C_k}$ stands for adjusted software development cost for case scenario k corresponding to a particular project size and $\mathbf{C_n}$ represents the nominal cost pertaining to the same size of the project.

5) EVALUATION AND INTERPRETATION OF RESULTS

This section focuses on the evaluation and comparison of results so that the impact of analyst and programmer capability on the software development cost could be analyzed. The results are evaluated on the basis of the percent increase/decrease in the software development cost. To make the process of evaluation and comparison of results more comprehensive, the results have been tabulated in Tables 9 and 10 and plotted in Figures 1 and 2.

Case scenario 1 pertaining to nominal values for all cost drivers and corresponding effort multipliers is used as the base case. Since the nominal and adjusted software costs for this case scenario are the same, hence, there is no percent increase/decrease (change) in the cost. Table 9 and Figure 1 indicate that out of remaining 24 case scenarios, 10 showed increase in development cost, whereas other 14 case scenarios displayed decrease in the development cost. Furthermore, case scenario 2 involving the lowest rankings (very low) for both ACAP and PCAP gave the worst performance in terms of percent cost increase. On the contrary, the case scenario 25 possessing the highest ratings (very high) for both ACAP and PCAP showed the best performance based on the same MOE.

In order to evaluate and interpret results further, let apply a simple valuation scheme. Under this scheme, the ratings of very low, low, nominal, high and very high carry a value of 1, 2, 3, 4 and 5, respectively, for both ACAP and PCAP. For example, the case scenario 1 involving nominal ratings for both analysts and programmers will have a value of (3 + 3 =) 6. Similar values for all the case scenarios are given in Table 10. The negative values in Tables 9 and 10 indicate percent decrease in development cost and difference in percent decrease in development cost, respectively. Table 10 clearly indicates that the case scenarios having total value of 6 or more showed decrease in cost and the remaining ones showed

an increase in development cost.

Table 9.
Comparison of results for all case scenarios in terms of % increase/decrease in software development cost (all sizes).

Case Scenario	% Increase/Decrease in Software Development Cost				
	Project Classification				
	S	I	M	L	VL
1	0.00	0.00	0.00	0.00	0.00
2	51.17	51.34	51.51	51.68	51.85
3	34.90	34.75	34.61	34.46	34.32
4	21.36	21.01	20.66	20.31	19.96
5	8.46	7.95	7.45	6.95	6.45
6	-6.25	-6.90	-7.54	-8.18	-8.81
7	35.25	35.62	35.99	36.37	36.75
8	21.39	21.45	21.52	21.58	21.64
9	8.53	8.38	8.23	8.08	7.94
10	-0.50	-0.82	-1.14	-1.46	-1.77
11	-13.33	-13.81	-14.29	-14.77	-15.24
12	20.87	21.37	21.87	22.38	22.89
13	8.95	9.15	9.35	9.56	9.76
14	-9.24	-9.41	-9.58	-9.75	-9.92
15	-20.45	-20.80	-21.14	-21.48	-21.83
16	9.39	9.95	10.51	11.08	11.66
17	-1.05	-0.77	-0.49	-0.21	0.08
18	-7.13	-7.03	-6.93	-6.83	-6.73
19	-16.41	-16.48	-16.56	-16.64	-16.72
20	-26.34	-26.59	-26.84	-27.09	-27.35
21	-3.90	-3.24	-2.58	-1.90	-1.21
22	-12.87	-12.47	-12.07	-11.67	-11.25
23	-15.97	-15.73	-15.48	-15.23	-14.98
24	-25.33	-25.27	-25.20	-25.13	-25.06
25	-33.89	-34.00	-34.11	-34.22	-34.33

Furthermore, case scenario 2 relating to the lowest ratings for analysts and programmers showed more increase in development cost for larger projects as compared to smaller ones. On the contrary, case scenario 25 involving the highest ratings for both more decrease in development cost for larger projects as compared to smaller ones. However, the difference in percent increase or decrease in cost between the small and very large projects for same case scenarios remained within 3%. For case scenarios having a total value of 6 or greater,

the general trend was that as the total value was increasing (ratings were improving), more decrease in cost was being achieved in larger projects. On the other hand, for case scenarios possessing total value of less than 6, the general trend was that more increase in cost was shown for larger projects as compared to smaller ones.

This indicates that experienced analysts and programmers even though charge much more than less experienced ones, but still they can display tremendous cost savings, especially, for larger projects. The reason for this is that even though analysts and programmers with very high ratings charge almost double than those with very low ratings, but they require less than half the effort.

Table 10. Comparison of results for all case scenarios in terms of difference in % increase/decrease in software development cost.

Case Scenario	ACAP		PCAP		Total Value	Difference in % Increase/ Decrease in Cost
	Rating	Value	Rating	Value		
1	N	3	N	3	6	0.00
2	VL	1	VL	1	2	0.68
3	VL	1	L	2	3	0.58
4	VL	1	N	3	4	1.40
5	VL	1	H	4	5	2.01
6	VL	1	VH	5	6	-2.57
7	L	2	VL	1	3	1.50
8	L	2	L	2	4	0.25
9	L	2	N	3	5	0.60
10	L	2	H	4	6	-1.27
11	L	2	VH	5	7	-1.91
12	N	3	VL	1	4	2.02
13	N	3	L	2	5	0.82
14	N	3	H	4	7	-0.69
15	N	3	VH	5	8	-1.37
16	H	4	VL	1	5	2.26
17	H	4	L	2	6	-1.12
18	H	4	N	3	7	-0.40
19	H	4	H	4	8	-0.31
20	H	4	VH	5	9	-1.01
21	VH	5	VL	1	6	-2.68
22	VH	5	L	2	7	-1.62
23	VH	5	N	3	8	-0.99
24	VH	5	H	4	9	-0.27
25	VH	5	VH	5	10	-0.44

Figure 1.
Comparison of results for all case scenarios in terms of % increase/decrease in software development cost (all sizes)

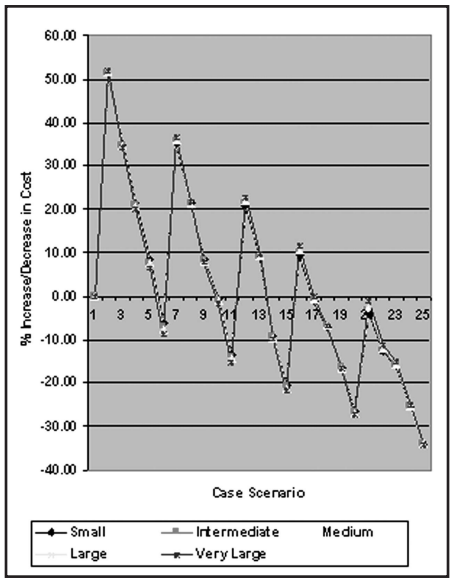
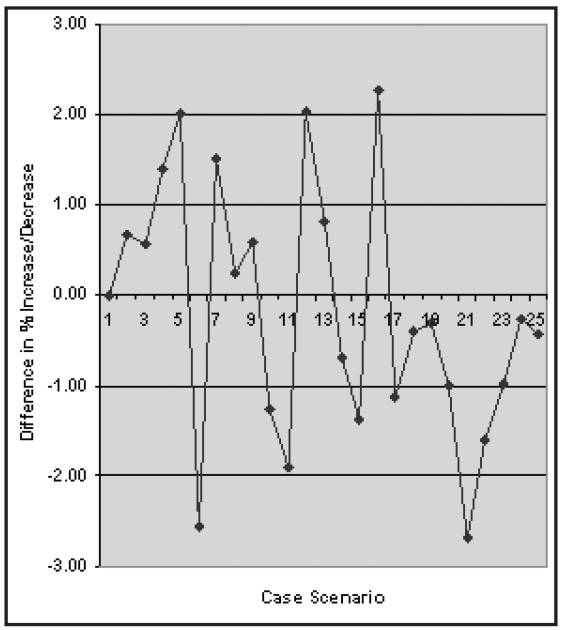


Figure 2.
Comparison of results for all case scenarios in terms of difference in % increase/decrease in software development cost



6) CONCLUSION

Results of this research indicate that the analysts and programmers having better capability ratings performed better than those with lower ratings in terms of reduction in software development cost. The case scenario involving the highest ratings for analysts and programmers showed the most decrease in software development cost. On the contrary, lowest ratings case scenario displayed the maximum increase in development cost. More experienced analysts and programmers show better results for larger projects as compared to less experienced ones who show even worse performance for larger projects. Furthermore, the results of this research are in agreement with general perception that it does not pay off to hire analysts and programmers with lower experience and capability.

7) RECOMMENDATION AND FUTURE RESEARCH

Based on results discussed above, this study recommends hiring more experienced and capable analysts and programmers for large and complex projects. Depending upon the nature and size of the project, less experienced/capable personnel may be hired. This does not mean that programmers with little or no experience should not be hired at all. They may be hired for smaller and less important projects so that requisite numbers of personnel could be produced in the market.

This study may be considered a beginning in the area of investigating the impact of development team capability on the software development cost with special reference to software houses in Karachi. The results of the research hold good at least in this particular case. However, for widespread application and acceptance under varying circumstances at different places including Pakistan, more research will have to be carried out to analyze and validate data for making appropriate recommendations.

8) ACKNOWLEDGEMENT

The cooperation of participating software houses in providing necessary data regarding analysts and programmers is greatly acknowledged.

REFERENCES

B. BOEHM, AND K. SULLIVAN (2000 A), SOFTWARE ECONOMICS: a roadmap. In Proc. 22nd International Conference on the Future of Software Engineering, A. Finkelstein, Ed., Limerick, Ireland, June 2000.

B. BOEHM (1981). Software engineering economics. Prentice-Hall, Englewood Cliffs, New Jersey, USA, October 1981.

B. BOEHM (2000 B), et al. Software cost estimation with COCOMO II. Prentice-Hall, Upper Saddle River, New Jersey, USA, 2000.

K. BRANLEY (2004). Software estimation. March 2004. http://home.iprimus.com.au/kbranley/Papers/Cost_Estimation.pdf

A. JAYARAMAN (2007). Characteristics that make one estimation technique better than others. Institute of Software Research, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA, May 2007.

B. BOEHM (1984). Software engineering economics. IEEE transactions on software engineering, 10(1): 4-21, January 1984.

K. JOHNSON (1998). Software cost estimation: metrics and models. Department of Computer Science, University of Calgary, Alberta, Canada, February 1998.

I. SOMMERVILLE (2004), Software engineering: chapter 26 - Software cost estimation. 7th Edition, Addison Wesley, May 2004.

M. PAULK (2006). Factors affecting personal software quality. The journal of defense software engineering. (electronic version), March 2006.

S. CHULANI (1999). Bayesian analysis of software cost and quality models. PhD

dissertation, Faculty of the Graduate School, University of Southern California, USA, May 1999.
SOFTSTAR SYSTEMS (2005). Overview of COCOMO. Revised April 18, 2005. http://www.softstarsystems.com/overview.htl