Empirical Software Engineering, 10, 219–234, 2005.

The Impact of Project Planning Team Experience on Software Project Cost Estimates

JAMES MCDONALD jamesmc@monmouth.edu
Monmouth University, West Long Branch, Monmouth, NJ, USA

Editor: June Verner

Abstract. Data from 135 teams that have participated in a software project planning exercise are analyzed to determine the relationship between team experience and each team's estimate of total project cost. The analysis shows that cost estimates are dependent upon two kinds of team experience: (1) the average experience for the members of each team and (2) whether or not any members of the team have similar project experience. It is shown that if no members of a planning team have had similar project experience then the estimate of cost is correlated with average team experience, with teams having greater average team experience producing higher total cost estimates. If at least one member of the planning team has had similar project experience then there is a weaker relationship between average team experience and cost, and cost estimates produced by those teams with similar project experience are close to those produced by teams with the greatest average team experience. A qualitative examination of the project plans produced by all teams indicates that the primary reasons that teams with less experience of either type produce lower cost estimates are that they have failed to include some tasks that are included by more experienced teams, and that they have estimated shorter task durations than have the more experienced teams.

Keywords: Project management, project planning, project cost estimation, software engineering.

1. Introduction

Models such as COCOMO, CheckPoint, PRICE and SLIM have been developed for estimating software development cost, effort, schedule and quality. See (Boehm, 1981), (Boehm, 2000), (Jones, 1996), (Kemerer, 1987), (Park, 1988) and (Putnam and Myers, 1992). Most of these models are based primarily on the number of lines of code developed or the functionality of the application as measured by function points. In addition, virtually all of them contain factors, derived from analyses of actual project data that are dependent on software development team member experience. These factors usually indicate that, as team experience increases, cost, effort and schedule duration decrease, while the quality of both the software development process, and the quality of the product being developed, increase. They are called "top-down" models.

Despite these consistent empirical results, there has been little empirical research done on how the amount and type of software development team experience impacts the estimation of software project cost. That is the subject addressed in this paper. The most closely related research work has examined the capabilities of expert software developers and maintainers as compared with those of novices. Those studies show that more experienced maintenance programmers create fewer major unexpected maintenance problems. However, the more experienced maintainers were not able to predict maintenance problems any better than maintainers with little experience (Jørgensen and Sjøberg, 2002). Several studies conducted during the 1980s indicate that expert developers behave differently than do novices. They organize information in larger ensembles (McKiethan et al., 1981). They solve problems using more general, principle-based strategies (Weisner and Sherz, 1983). And, under certain conditions, they make changes more efficiently (Soloway et al., 1988). Papers appearing in the psychology literature, which compare the cognitive abilities of experts with those of less experienced individuals, contain similar results (Ericsson and Lehman, 1996). (Doane et al., 1990), using a correlation analysis, conclude that the amount of experience is not necessarily the principle cause of differences between novices and experts. (Ericsson et al., 1993) have found that the type of experience may be more important than the total amount of experience. This author has been unable to find any empirical studies that address the relationship between experience and estimated software project cost.

"Top-down" models are not generally dependent on details of the software architecture. In addition, they do not consider the work breakdown structure of the project, the network schedule that specifies which pieces of software need to be developed, how they will fit together, the tasks that need to be accomplished, or the effort required for the completion of each task.

We call the models based on those additional details "task-based" models or "bottoms-up" models, because they are based on the details of an architectural model, an enumeration of the software objects or modules that need to be developed, the tasks involved in developing those piece-parts, the integration of the parts and, finally, testing and deployment of the software system or product. These details might also include support activities such as creation and maintenance of an appropriate software development environment, and evaluation, procurement and integration of COTS software. In theory, when all of these details have been thought through, and a detailed project plan has been developed, the bottoms-up estimate of effort, schedule and cost should be reasonably accurate. However, in the author's experience, we still sometimes find extremely large differences between task-based estimates and actual results. Frequently, the actual effort and cost to complete a project are significantly greater than estimates based on those details.

The author has had a unique opportunity to gather and analyze data on one aspect of software development projects having to do with the activities that go on during the detailed, bottoms-up planning of a project. This opportunity arose as a result of teaching a software project management course in both industry and in academia for several years (McDonald, 2000). An extensive student exercise, conducted as a part of that course, required students to develop a detailed project plan for a software development project. Based on that detailed plan, students estimated the required effort, cost and schedule that would be required for execution of the plan. Data from a pre-course questionnaire, that was used to place students in work groups, and selected data that were gathered from the completed plans, produced the body of information upon which this analysis is based.

A preliminary analysis of this data, reported by the author in (McDonald, 2002), showed that there was a positive correlation between average team experience and estimated total project cost. Teams having a greater average team experience produced higher cost estimates. A more thorough analysis of the data, reported in this paper, indicates that there are two significant factors that influence the overall cost estimates: (1) Average team experience and, (2) Whether any team members have had similar project experience. It is shown that, while there is a positive correlation between average team experience and estimated cost, if at least one planning team member has had similar project experience there is significantly less positive correlation between average team member experience and the cost estimate. Under those circumstances all teams, regardless of their average team experience, produce total cost estimates, which are similar to those of teams with the greatest average team experience. These findings are consistent with those of (Ericsson et al., 1996), cited above. The exercise is described in Section 2, followed by a discussion of the data collection and analysis in Sections 3 and 4. Section 5 provides a qualitative interpretation of the quantitative results and Section 6 outlines the conclusions, some possible inferences and recommendations resulting from this work.

2. The Project Planning Exercise

In 1994 the author was asked to develop and teach an updated version of a software project management workshop that had been developed in the early 1980s within AT&T. That workshop was delivered during the decade of the 1980s to several hundred AT&T Bell Laboratories research and development employees. The author began to offer the updated version of the workshop in early 1995. By the end of 1997 he, along with another AT&T employee, had delivered the workshop to more than 1600 additional employees of both AT&T and Lucent Technologies. Some of the participants in the updated workshop were the same people who had participated several years earlier in the initial workshop. In 1999 the author moved to a faculty position at Monmouth University and started to teach a one-semester software project management course in Monmouth's software engineering master's degree program. During the workshop and the course students heard a series of lectures on a variety of software project management topics. Participants who had little or no prior project planning experience, and who had not attended the earlier version of the workshop, participated in an exercise, which required them to develop a detailed, bottoms-up, project plan for a make-believe, but real-sounding, software development project. After working on the project plan for approximately 20 hours, they were interviewed by participants who had attended the prior version of the workshop, or who had significant project planning experience, acting as project management auditors. The auditors then read the plan and, after a management review presentation by the developers of the plan, provided a critique of the plan and suggestions for improvement. The instructor recorded the estimated cost for each group's plan and asked if any members of the planning team had ever participated in a similar project. By "similar project" was meant one that had a distributed architecture, with multiple interfaces to legacy systems and which required a graphical user interface.

The project that was used as a basis for the exercise is the development of customized workstation software that will provide the user with an appropriate graphical user interface and communicate with four existing legacy systems using existing proprietary communications protocols. The application is called the Central Control Position System (CCPS). The intended CCPS environment is shown in Figure 1. In the current environment information about transactions is transmitted using handwritten multipart forms. When the CCPS is implemented it will be located at the organization's Billing unit. The Billing unit receives a paper request, and employees who work there enter data from the form into BILLSYS, the Billing system, using the PC as a workstation. The PC forwards data to update the Audits, Credits and Delivery systems (previously this was done manually). The PC receives confirmation that the updates to those systems have been completed successfully. A technician performs the work requested on the initial form and calls the Billing unit by telephone to report completion of the work. The person working at the CCPS workstation enters the data while talking with the technician and the data are forwarded to update all of the systems. The PC receives confirmation from the remote systems, prepares a final confirmation and faxes that confirmation to the originator of the request.

Figure 2 provides a high level architectural view of the software that will be implemented. That information is also provided to the team that is developing the project plan. Then, based on the contents of the lectures which the students have heard, they develop a detailed project plan. That plan must contain answers to the following questions: What will be done? How will it be done? Who will do each task? When will it be done? How much will it cost? The instructor provides a project plan template for the students to follow, which already has some of its sections completed.

Groups of 6 to 8 students work for approximately 20 hours developing the plan. They produce a work breakdown structure, linking tasks through a network diagram, or PERT chart, assigning tasks to team members and estimating duration of each task as well as an overall release schedule. They need to plan for detailed requirements development,

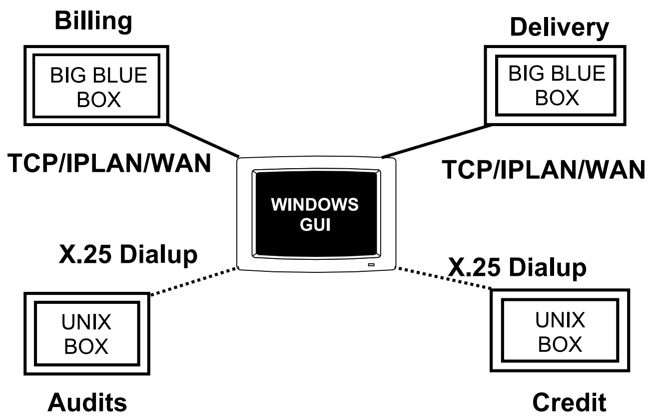


Figure 1. The CCPS environment.

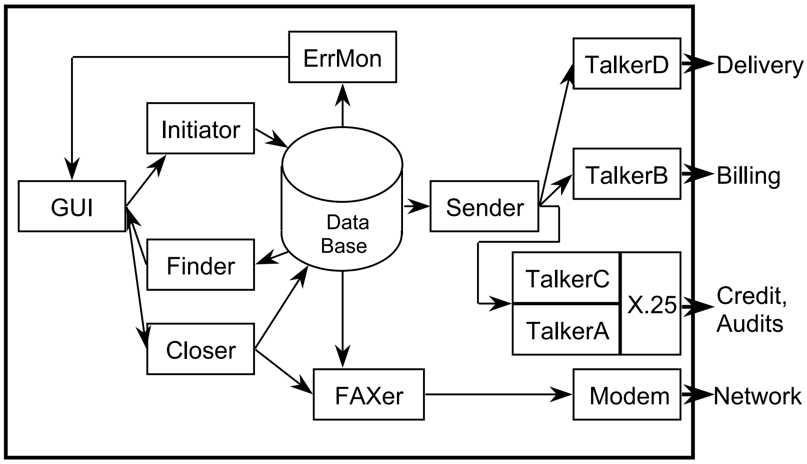


Figure 2. The CCPS architecture.

software design, software development, integration, testing and deployment, as well as for the procurement of equipment, including a small piece of specialized hardware (a card for the PC) necessitated by a proprietary communications protocol used on the dial-up X.25 link.

While they are doing this work the instructor facilitates the technical details of project plan development. However, the groups are encouraged to rely upon their own software development experience to determine what tasks need to be accomplished and the effort required to complete each task. They are not put under any pressure to minimize costs or staffing. They are told what the potential customer has indicated are the desired delivery dates, and that the instructor believed that they were achievable with a modestly sized team. Therefore, the planning groups should feel free to include any and all tasks that might be required for the project, and are encouraged by the instructor to do so.

Teams were provided with a written high level description of the features that would be required of the CCPS system. They were given a short description of the application architecture including the diagram shown in Figure 2. They were also told that each of the software modules, or objects, shown in Figure 2 was intended to be designed and coded by an individual member of the team. Members of the planning teams were told that they would become members of the software development team and that they would be supplemented with additional staff, if necessary. Their responsibility was to develop a detailed work breakdown structure, estimate the duration of individual tasks, schedule those tasks and develop a budget for the project. Prior to working on the project plan participants heard lectures about planning and estimating, during which a variety of techniques for developing the work breakdown structure, estimating and scheduling were described. The estimation techniques discussed included top-down and bottoms-up modeling, delphi techniques for task duration estimation and PERT/CPM methods for scheduling. They were encouraged to use these techniques during their project planning work. The exercise materials told them that the annual cost of team members was $165,000 for non-management team members and $195,000 for management members of the team.

After reviewing the written case materials, planning team members were introduced to the project at a kick-off meeting led by “Mark Money,” one of the instructors playing the role of the manager to whom the project team would report. Mr. Money would be responsible for price and delivery commitment negotiations with the external CCPS customer. Mark Money started the meeting by providing a summary of what he knows about the project, including some information about the customer’s desired delivery dates. Mark then turned the meeting over to “Mr. Big,” who was played by a second instructor. Mr. Big was the top executive of the organization in which the project was being done. He explained his close personal relationship with an upper level executive in the customer’s organization. He told the participants that he was very anxious to get the customer’s business and to build a long term business relationship with the customer. He emphasized the point that CCPS would become very important to the customer and, ultimately, could provide significant income for the organization that he led. He requested the team to do the best job that they could in developing the project plan and asked them not to take undue risks, because he did not want to be embarrassed by not successfully delivering CCPS to the customer’s environment. His, and Mark Money’s, principle interests were in receiving a plan that provided realistic delivery dates and cost commitments. The purpose for this introduction was to encourage participants to feel as though they were working in a real business environment, but one in which they should not take undue risks. Both Money and Big emphasized the importance of a solid, well thought-out plan. They told participants that if they had any questions for the customer, or would like to interview the customer, they should seek-out one of the instructors who would play the role of the customer and provide consistent and straightforward answers to their questions.

During preparation of the plan the instructors provided answers to teams’ questions, facilitation in the use of a software scheduling tool and some assistance in scheduling resources, but generally did not provide significant assistance for developing work breakdown structures or task duration estimates.

3. Data Collection

Before the class started, and before students began to work on the exercise, they were asked to complete a questionnaire, which requested extensive background information. It included 34 questions about the student’s background, interests, years of software experience and current job. Based on their responses they were assigned to a work team for the exercise. The objective in making these assignments was to insure a diversity of technical knowledge in each group. A group consisted of 6 to 8 participants with each having a background or current responsibility in requirements engineering, software development, testing, deployment and customer support and, occasionally, a person or two with some marketing experience and, in some cases, a person or two with experience in digital hardware design. The technical makeup of teams was reasonably homogeneous. A typical team consisted of two or three software developers, one or two testers, one or two requirements engineers, one or two customer support people and, occasionally a hardware design, marketing, sales, product management or manufacturing person. This grouping was intended to simulate the diversity of expertise that would normally need to be assembled on a core planning team for a project of the type described above.

In addition to questions about the technical backgrounds of the participants, a question was asked about each participant's years of experience in working in a software development and support environment, which was:

> "For how many total years have you worked on software requirements, software design or development, software testing or software deployment and customer support? (If you have never worked in any of these positions please enter zero.)_____"

This question was included when the questionnaire was initially designed because the author thought that it might provide useful information that could be helpful in assigning teams members. Although ultimately not used for this purpose, the data have proven useful in conducting the analysis described in the next section in which responses to that question have been combined with estimates of cost contained in completed project plans to determine the relationship between average team experience and estimated cost.

4. Data Analysis

In (McDonald, 2002) the author showed, that there was a positive correlation between average team experience and the cost estimates produced by the teams. The relationship developed there is shown in Figure 3.

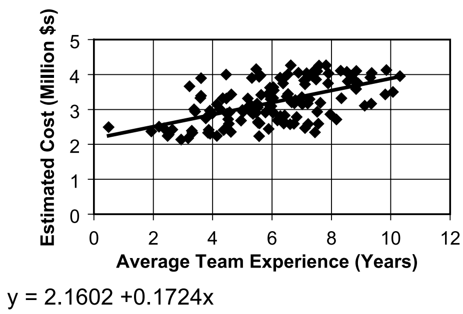


Figure 3. Relationship between average team experience and estimated cost.

While preparing the analysis presented in that paper several different measures of team experience and several different relationships between experience and cost were tried. In addition to the linear relationship, upon which the published results were based, both logarithmic and higher order polynomial relationships were tried. For a measure of team experience the author also considered using the experience of the team member reporting the greatest experience and eliminating data for team members reporting zero experience before calculating the average team experience. For each of these alternative assumptions none of the results were significantly different than the results found using the average team experience and a linear relationship. For that reason those alternatives have not been considered in this analysis. Further diagnosis of the residuals associated with these data and the linear regression model, which was fit to the data, indicated an anomaly for those teams on which there was a team member who had previous similar project experience. Figure 4 shows the residuals associated with each data point. It is visually obvious that the residuals associated with teams having at least 1 member with similar project experience are towards the right on the experience axis and the residuals for those points are, on the average, smaller than those for the remainder of the population. The actual mean square error for teams having no members with similar project experience is 0.139. For teams having at least one member with similar project experience the mean square of the residuals is 0.024. The data points associated with each team are listed in Table 1. The first column shows the team number. The column labeled Y_i shows the cost estimate produced by the team in millions of dollars. The columns labeled X_{i1} and X_{i2} show the average team experience in years and whether any team member has had similar project experience (indicated by $X_{i2} = 1$) or not (indicated by $X_{i2} = 0$), respectively.

In order to investigate the relationships among average team experience, whether any member of the team has had similar project experience, and the cost estimate produced

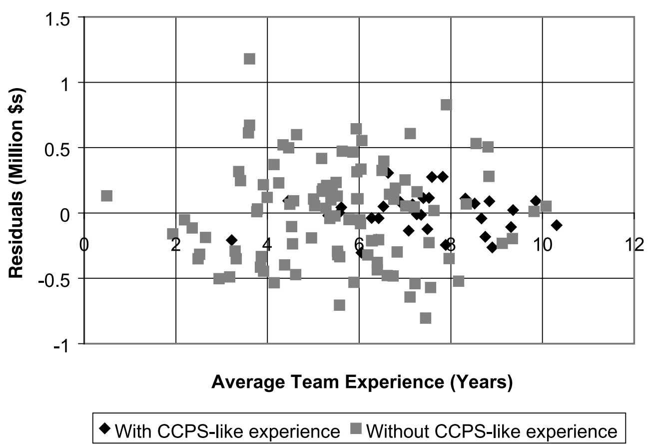


Figure 4. Plot of residuals.

by the team, we will use the following linear regression with interaction model described in (Neter et al., 1996):

$$Y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \beta_3 X_{i1} X_{i2} + \varepsilon_i$$

Where:

X_{i1} = Years of average team experience for the i^{th} team,

$$X_{i2} = \begin{cases} 1 \text{ if any member of team } i \text{ has had similar project experience,} \\ 0 \text{ if no member of team } i \text{ has had similar project experience,} \end{cases}$$

Y_i is the cost estimate developed by the i^{th} team,
β_0, β_1, β_2 and β_3 are constants, and
ε_i = a random error term with mean, $E\{\varepsilon_i\} = 0$ and variance, $\sigma^2\{\varepsilon_i\} = \sigma^2$.

This model provides an intuitively appealing and a practical description of what might be going on with our data. For example, for a team without similar project experience, $X_{i2} = 0$, so the product $X_{i1}X_{i2} = 0$ and the expected value of the cost estimate would be:

$$E\{Y_i\} = \beta_0 + \beta_1 X_{i1}.$$

For teams with similar project experience $X_{i2} = 1$ and, therefore, $X_{i1}X_{i2} = X_{i1}$. The expected value of the cost estimate is then:

$$E\{Y_i\} = (\beta_0 + \beta_2) + (\beta_1 + \beta_3) X_{i1}.$$

Both of these expressions are simple linear expressions, albeit with different intercepts and slopes, depending upon the values of the parameters.

Finding the values of the parameters β_0, β_1, β_2 and β_3 in the model described above requires only doing a standard computerized regression fit to determine the values of these four parameters, so that the sum of the error terms over all 135 sample points is minimized. That regression was done using MS Office Excel's® Solver capabilities and yielded the results shown in Table 2.

To test for interaction effects we formulate the following hypotheses:

$$\left.\begin{aligned} H_{0i}&: \beta_i = 0 \\ H_{ai}&: \beta_i \neq 0 \end{aligned}\right\} \quad \text{For } i = 0, 1, 2, 3.$$

The t* statistics for testing these hypotheses are:

$$|t_i^*| = \frac{|b_i|}{s\{b_i\}}$$

If $|t_i^*| \leq t\ (0.975;\ 135) = 1.97$ (from a table of the t distribution) we would conclude H_{0i}. However, we conclude H_{ai}: $\beta_i \neq 0$ for all four parameters because, from Table 2 above, all of the t-statistic values are > 1.97.

Table 1. Data for each team.

Team #	Y_i	X_{i1}	X_{i2}	$X_{i1}*X_{i2}$	Team #	Y_i	X_{i1}	X_{i2}	$X_{i1}*X_{i2}$	Team #	Y_i	X_{i1}	X_2	$X_{i1}*X_{i2}$
1	3.5	10.1	0	0.0	46	3.6	8.8	0	0.0	91	3.3	6.0	0	0.0
2	3.1	5.2	0	0.0	47	4.3	7.8	1	7.8	92	2.6	5.5	0	0.0
3	3.2	7.6	0	0.0	48	2.9	3.4	0	0.0	93	4.1	8.3	1	8.3
4	4.0	9.4	1	9.4	49	2.7	6.2	0	0.0	94	3.3	7.3	0	0.0
5	4.0	6.9	1	6.9	50	4.1	7.4	1	7.4	95	2.6	7.2	0	0.0
6	2.6	5.6	0	0.0	51	2.7	4.5	0	0.0	96	3.3	5.2	0	0.0
7	2.9	6.0	0	0.0	52	3.4	6.5	0	0.0	97	2.7	3.8	0	0.0
8	4.1	8.8	1	8.8	53	4.3	7.6	1	7.6	98	2.6	6.6	0	0.0
9	3.1	5.2	0	0.0	54	4.0	7.4	1	7.4	99	2.6	5.5	0	0.0
10	3.9	5.6	1	5.6	55	2.8	6.4	0	0.0	100	3.1	5.3	0	0.0
11	3.1	5.4	0	0.0	56	3.7	8.9	1	8.9	101	2.4	4.4	0	0.0
12	3.6	6.0	1	6.0	57	3.8	8.8	1	8.8	102	2.3	7.4	0	0.0
13	2.9	5.0	0	0.0	58	2.2	3.2	0	0.0	103	3.8	7.1	1	7.1
14	4.0	5.6	1	5.6	59	3.9	3.6	0	0.0	104	2.9	5.3	0	0.0
15	3.0	3.9	0	0.0	60	3.1	5.4	0	0.0	105	3.7	7.9	1	7.9
16	2.9	6.0	0	0.0	61	2.3	3.9	0	0.0	106	3.5	6.1	0	0.0
17	2.4	1.9	0	0.0	62	3.7	6.4	1	6.4	107	4.0	7.9	0	0.0
18	2.9	5.8	0	0.0	63	2.5	7.1	0	0.0	108	2.4	3.9	0	0.0
19	3.1	4.1	0	0.0	64	3.2	6.8	0	0.0	109	2.8	6.3	0	0.0
20	2.6	6.7	0	0.0	65	2.4	5.9	0	0.0	110	2.7	8.2	0	0.0
21	3.9	6.3	1	6.3	66	3.3	8.3	0	0.0	111	3.4	4.6	0	0.0

Table 1. (continued).

Team #	Y_i	X_{i1}	X_{i2}	$X_{i1}*X_{i2}$	Team #	Y_i	X_{i1}	X_{i2}	$X_{i1}*X_{i2}$	Team #	Y_i	X_{i1}	X_2	$X_{i1}*X_{i2}$
22	4.1	8.5	1	8.5	67	4.0	4.5	1	4.5	112	2.7	5.0	0	0.0
23	3.3	6.8	0	0.0	68	2.9	4.6	0	0.0	113	3.9	7.5	1	7.5
24	2.8	6.8	0	0.0	69	4.3	6.6	1	6.6	114	3.4	5.6	0	0.0
25	4.0	7.2	1	7.2	70	3.0	4.2	0	0.0	115	3.3	3.6	0	0.0
26	2.4	4.6	0	0.0	71	3.9	5.3	1	5.3	116	2.5	2.4	0	0.0
27	3.4	6.5	0	0.0	72	4.0	7.3	1	7.3	117	2.9	5.5	0	0.0
28	3.4	7.0	0	0.0	73	3.8	8.8	0	0.0	118	3.8	8.5	0	0.0
29	3.9	6.4	1	6.4	74	3.2	6.7	0	0.0	119	3.4	5.9	0	0.0
30	3.2	9.3	0	0.0	75	3.6	5.9	0	0.0	120	2.9	8.0	0	0.0
31	3.1	5.5	0	0.0	76	2.4	2.6	0	0.0	121	2.9	5.4	0	0.0
32	2.2	5.6	0	0.0	77	3.3	4.3	0	0.0	122	2.6	4.5	0	0.0
33	2.9	4.5	0	0.0	78	4.1	9.9	1	9.9	123	2.7	6.4	0	0.0
34	4.2	5.5	1	5.5	79	2.1	2.9	0	0.0	124	4.1	7.5	1	7.5
35	3.2	5.5	0	0.0	80	4.0	6.5	1	6.5	125	3.7	3.2	1	3.2
36	2.3	3.8	0	0.0	81	2.9	7.5	0	0.0	126	2.5	2.2	0	0.0
37	3.1	6.0	0	0.0	82	3.2	7.2	0	0.0	127	3.1	9.1	0	0.0
38	3.3	6.0	0	0.0	83	2.2	4.1	0	0.0	128	3.1	5.9	0	0.0
39	3.4	3.6	0	0.0	84	2.9	4.0	0	0.0	129	2.6	6.4	0	0.0
40	2.2	2.5	0	0.0	85	2.8	6.3	0	0.0	130	2.6	7.6	0	0.0
41	2.3	3.3	0	0.0	86	2.5	0.5	0	0.0	131	3.3	4.5	0	0.0
42	3.0	5.0	0	0.0	87	3.2	7.0	0	0.0	132	3.7	7.1	0	0.0
43	2.3	2.5	0	0.0	88	3.0	5.4	0	0.0	133	2.8	3.8	0	0.0
44	3.9	9.3	1	9.3	89	2.4	3.3	0	0.0	134	4.0	8.7	1	8.7
45	3.0	3.4	0	0.0	90	4.0	10.3	1	10.3	135	3.4	9.8	0	0.0

Table 2. Regression results.

Regression coefficient	Estimated regression coefficient	Estimated standard deviation	t*
β_0	+2.3092	0.1071	21.5536
β_1	+0.1130	0.0183	6.1594
β_2	+1.4846	0.3015	4.9247
β_3	−0.0884	0.0421	2.1001

Our best estimates, then, for the cost developed by the teams are:

$E\{Y_i\} = 2.31 + 0.113\ X_{i1}$ for teams with no similar project experience, and
$E\{Yi\} = 3.79 + 0.0246\ X_{i1}$ for teams with similar project experience.

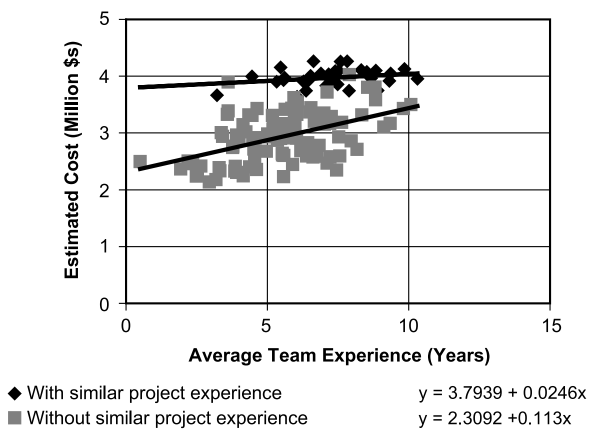


Figure 5. Scatter plot with both types of experience included.

Plots of these regression lines on the scatter chart of the data in Figure 5 show that cost estimates produced by teams that have similar project experience are almost independent of the teams' average years of experience. For teams without similar project experience the cost estimate is driven quite strongly by average team experience. For teams with lower amounts of experience the estimate is typically lower than that developed by both teams with greater average team experience and teams with similar project experience.

5. Reasons for these Results

As indicated in Section 4, analysis of the data makes it clear that estimated total project cost is related to both average team experience and to whether any member of the planning team has had similar project experience. In general, higher development costs are estimated by teams having more experience of both types. Two hundred seventy one experienced software project planners, who served as project management auditors for the completed project plans in the workshop, have generally concluded that the higher cost estimates were probably most realistic. The obvious next question becomes why this happens. Unfortunately, the author did not retain any quantitative data, beyond the measurements of experience outlined above, that would help answer this question. At the end of each course the instructor did have a brief opportunity to inspect the documented project plan and listen to readouts from both the project planning team and the audit team that reviewed the contents of the plan and interviewed team members. After gathering, recording and reporting cost information and hearing a brief presentation by the team that developed the plan and a critical readout from audit teams for a large number of planning groups it became clear that, for those teams that produced cost estimates towards the lower end of the cost distribution, the most frequently mentioned recommendations from the auditors could be summarized as:

1. You might want to reconsider the methods that were used to estimate the duration of individual tasks. Some of the task durations seem unduly short.

2. It appears that you have forgotten to include the tasks required for [various functions]. The forgotten tasks were typically: (a) Tasks required to obtain and verify the accuracy of interface requirements between CCPS and the four legacy systems; (b) Those associated with obtaining customer signoff, or agreement, on requirements; (c) Tasks associated with insuring that all team members understood and complied with guidelines for the application architecture; (d) Activities required for inspection of requirements, designs, code test plans and test cases; (e) Integration of software components and modules; and, (f) Tasks associated with deployment, such a site surveys and customer training.

For teams whose estimates were closer to the upper end of the cost distribution these comments were almost never heard. It should be pointed out that these conclusions regarding why this happens have been derived only by a qualitative evaluation of the individual project plans and the associated presentations.

These qualitative conclusions should not be interpreted with the same level of integrity as the results outlined in Section 4. This qualitative analysis is similar to some of classification methods described in (Miles and Huberman, 1994). However, it is much less formal. Therefore, the rigor of these results is substantially less than they would be if any of the more rigorous qualitative methods described in that reference were applied.

The conclusions reached from these informal, qualitative findings are that less experienced teams are likely to forget to include certain kinds of tasks in their work breakdown structures and are likely to estimate shorter durations for the tasks that they have included.

There is no definitive actual cost for completion of the CCPS project because the CCPS product has never been developed. Therefore, it is virtually impossible to say what the correct cost estimate should be. However, this author, and the numerous experienced project planners who have served as auditors for the workshop plans, have concluded that it is likely that the higher estimates are better than the lower estimates. Therefore, these results might serve as an indicator that, when a project plan is developed by a relatively inexperienced team, the managers and customers to whom the results are delivered might be wise to have the plan carefully reviewed by either themselves or by an independent group of reviewers to insure that all necessary tasks have been included and that appropriate task duration estimates have been made.

6. Limitations of this Analysis

While the preceding analysis reached several appealing conclusions, it must be pointed out that it does have several limitations:

- The work was not based on an experiment that was designed to determine the relationship between experience and cost estimates. It is an analysis of observational data which was initially gathered for another purpose, namely to stimulate discussion at the end of the project management course or workshop.

- The teams were not selected randomly from a well defined population. Participants in the workshop and the course were usually self-selected or were selected by their management for participation. Therefore, the population from which sample points were selected is not a well defined population. Consequently, it is not appropriate to conclude that the findings apply to any specific population.

- The statistical analysis presented in Section 4 made some assumptions about the error terms that appear in the model used in that section. Those assumptions may or may not be true. The model was used to develop estimates of the model's parameters and to test hypotheses subject to the assumptions, so that reasonable, if not statistically rigorous, conclusions could be drawn about the impact of experience on cost estimates.

- The analysis reported in this paper addressed only the relationship among average team experience, similar project experience and cost estimates. It is likely that there were other factors that influenced the cost estimates, such as team dynamics and assumptions made by teams about the meaning of the requirements. However, no quantitative data that would characterize other differences among teams regarding these factors were recorded. Therefore, their impact could not be analyzed as part of this work.

7. Conclusions and Recommendations

When relatively inexperienced software development teams are given responsibility for planning a new software development project the cost estimates that they produce might be lower, as compared with the actual cost, than those that would be produced by more experienced teams. This could happen for two reasons: (1) Inexperienced teams are typically less productive than teams with greater experience (as indicated in this paper's earlier references to cost estimation models) and, if so, are likely to spend more time and cost developing the product than would more experienced teams, and (2) Inexperienced teams tend to produce lower cost estimates than do more experienced teams unless they have at least one planning team member who has had similar project experience (shown by the analysis in this paper). Therefore, when managers, or customers, are provided with cost estimates for new software projects they should determine how much and what kind of experience the planning team has had. If the experience is relatively shallow, in average years of team experience, or with similar projects, they should thoroughly review the details of the plan that produced the estimate, either personally, or by using an independent team of reviewers or auditors to insure the integrity of the plan and the cost estimates included in that plan.

This author feels that the differences between cost estimates produced by experienced and inexperienced teams could be even greater when there are pressures on the planning team to commit to deliveries with very tight time or budget constraints, although this has not been analyzed in this paper. Perhaps the impact of such additional constraints could be analyzed in future research work.

Acknowledgments

The author wishes to acknowledge the contributions of Mr. Glenn A. Secor to this work. He developed and maintained the extensive case study materials that have been used as the basis for the exercise during which students developed the detailed project plan. He closely managed and facilitated the work of many of the teams that developed project plans for which data are reported in this paper. I would also like to thank the numerous students who have participated in the exercise. All worked hard to produce credible plans and I hope they have benefited from that experience.

References

Boehm, B. 1981. *Software Engineering Economics*. Englewood Cliffs, NJ: Prentice-Hall.

Boehm, B. W., et. al. 2000. *Software Cost Estimation with COCOMO II*. Upper Saddle River, NJ: Prentice-Hall.

Doane, S. M., Pellegrino, J. W., and Klatsky, R. L. 1990. Expertise in a computer operating system: Conceptualization and performance. *Human Computer Interaction* 5: 263–304.

Ericsson, K. A., and Lehman, A. C. 1996. Expert and exceptional performance: Evidence of maximal adaptation to task constraints. *Annual Review Psychology.* 47: 272–305.

Ericsson, K. A., Krampe, R. T., and Tesch-Römer, C. 1993. The role of deliberate practice in the acquisition of expert performance. *Psychological Review* 100(3): 363–406.

Jones, T. C. 1996. *Applied Software Measurement, Assuring Productivity and Quality*. New York: McGraw-Hill.

Jørgensen, M., and Sjøberg, D. I. K. 2002. Impact of experience on maintenance skills. *Journal of Software Maintenance and Evolution: Research and Practice* 142(2): 123–146.

Kemerrer, C. F. 1987. An empirical validation of software cost estimation models. *Communications of the ACM* 30(5).

McDonald, J. 2000. Teaching project management in industrial and academic environments. *Proceedings of the 13th Conference on Software Engineering Education and Training (CSEE&T 2000)*, Austin, TX, pp. 151–160.

McDonald, J. 2002. Relationship between planning team estimates and project cost estimates. *Proceedings of the IASTED Software Engineering Applications Conference*, Cambridge, MA, pp. 718–724.

McKeithan, K. B., Reitman, J. S., Reuter, H. H., and Hirtle, S. C. 1981. Knowledge organization and skill differences in computer programmers. *Cognitive Psychology* 13(3): 307–325, July 1991.

Miles, M. A., and Huberman, A. M. 1994. *Qualitative Analysis: An Expanded Source Book*. London: SAGE Publications.

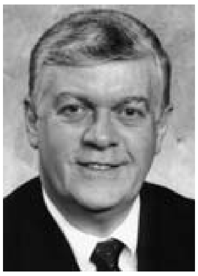
Neter, J., Kutner, M. H., Nachtsheim, C. J., and Wasserman, W. 1996. *Applied Linear Statistical Models*. Boston, MA: WCB/McGraw-Hill, pp. 455–496.

Park, R. E. 1988. *The Central Equations of the PRICE Software Cost Model*. Mt. Laurel, NJ: Lockheed-Martin PRICE Systems.

Putnam, L. H., and Myers, W. 1992. *Measures for Excellence: Reliable Software on Time, Within Budget*. Englewood Cliffs, NJ: Prentice-Hall.

Soloway, E. B., Adelson, B., and Erlich, K. (eds.) 1998. Knowledge and process in the comprehension of computer programs. The Nature of Expertise: Lawrence Erlbaum Associates.

Weisner, M., and Sherz, J. 1983. Programming problem representation in novice and expert programmers. *International Journal of Man-Machine Studies* 14: 391–396.



James McDonald, Associate Professor and Chair of the Department of Software Engineering at Monmouth University, holds a bachelor's degree in Electrical Engineering from New Jersey Institute of Technology, an MSEE degree from Massachusetts Institute of Technology and a Ph.D. from New York University. Dr. McDonald has an industrial background in both software development and digital hardware design. He is a senior member of the IEEE, and a member of the IEEE Computer Society, the Association for Computing Machinery, the American Society for Engineering Education and the Project Management Institute. His primary research interests are in the areas of software project management and testing.