A Survey on Software Cost Estimation in the Chinese Software Industry

Da Yang[1,2], Qing Wang[1], Mingshu Li[1], Ye Yang[1], Kai Ye[1,2], and Jing Du[1,2]

[1] Institute of Software, Chinese Academy of Sciences, Beijing, China 100190

[2] Graduate University of Chinese Academy of Sciences, Beijing, China 100190

{ yangda, wq, mingshu, ye, yekai, dujing }@itechs.iscas.ac.cn

ABSTRACT

Although a lot of attention has been paid to software cost estimation since 1960, making accurate effort and schedule estimation is still a challenge. To collect evidence and identify potential areas of improvement in software cost estimation, it is important to investigate the estimation accuracy, the estimation method used, and the factors influencing the adoption of estimation methods in current industry. This paper analyzed 112 projects from the Chinese software project benchmarking dataset and conducted questionnaire survey on 116 organizations to investigate the above information. The paper presents the current situations related to software project estimation in China and provides evidence-based suggestions on how to improve software project estimation. Our survey results suggest, e.g., that large projects were more prone to cost and schedule overruns, that most computing managers and professionals were neither satisfied nor dissatisfied with the project estimation, that very few organizations (15%) used model-based methods, and that the high adoption cost and insignificant benefit after adoption were the main causes for low use of model-based methods.

Categories and Subject Descriptors

D.2.9 [**Software Engineering**]: Management – *cost estimation, time estimation*; D.2.8 [**Software Engineering**]: Metrics – *performance measures*; K.6.1 [**Management of Computing and Information Systems**]: Project and People Management – *management techniques*

General Terms

Management, Measurement, Economics, Performance

Keywords

Software cost estimation, Estimation method, Estimation model

1. INTRODUCTION

Software cost estimation usually includes effort (the cost of labor) estimation and schedule (the cost of time) estimation [1], and it may be referred to as resource estimation [2], software estimation [3], or project estimation [4] in literature. It is an essential software management activity, and is the basis for project bidding, budgeting, planning, and cost control. The study of software cost estimation started as early as 1960s [2], and the published papers in this field demonstrated an increasing trend during the last two decades [5]. The continuous research effort in this field suggests that software cost estimation is important and still has potential of improvement.

ESEM'08, October 9-10, 2008, Kaiserslautern, Germany.

To identify what to improve and how to improve, it is important to know the current situation of software cost estimation, e.g. the way people make estimations, the estimation accuracy, and the factors influencing the adoption of estimation methods in current software industry. This paper surveyed and analyzed the current situation related to software effort and schedule estimation in China.

There are several reasons that motivated our survey in the Chinese software industry.

- As far as we know, there is still no survey on software cost estimation in the Chinese software industry, though this industry is playing a more and more important role in global software development.
- Most of the previous surveys were conducted in the eighties and early nineties. As we all know, the business environment, development technologies and process have changed a great deal since then.
- The previous surveys lacked research on the factors which influence the adoptions of software cost estimation methods or the factors which lead to the low use of model-based software cost estimation methods.

Section 2 discusses the previous surveys, related research works, and our research questions. Section 3 describes how our research was designed and how the data was collected. Section 4 discusses the results of our research questions. Section 5 discusses the issue of validity, and Section 6 is the conclusions and discussion of future works.

2. BACKGROUND

2.1 The accuracy of software cost estimation

Several studies [3, 6, 7, 8, 9, 10, 11] have surveyed the accuracy of effort and schedule estimation. The results of these survey showed that 59%-76% projects were completed over estimated effort and 35%-80% were over schedule. Most of the previous surveys used mean value to summarize the magnitude of effort and schedule overruns. The mean effort overrun reported is between 18% and 41%, and the mean schedule overrun is between 22% and 25% [3, 6, 7, 10, 11]. We excluded the frequently quoted CHAOS Report published by the Standish Group, because of its validity problems reported by Jørgensen [12].

Because project managers may be comfortable with small cost overruns [13], it is useful to know the distribution of cost overruns and identify the projects that have large cost overruns. Moløkken-Østvold [3] used figures to describe the distribution of cost overruns and the result showed that most projects had small effort overruns (less than 21%) but a few projects had effort overruns greater than 100%. Therefore, the mean value of effort overruns (44%) was much larger than the median value (21%) in his survey research. Moløkken-Østvold then analyzed whether project size affected estimation accuracy and found that large projects may be more prone to under-estimation of effort. However, because of small sample size, it was hard to get statistically significant conclusions.

In this survey research, we surveyed and analyzed both the magnitude and distribution of cost estimation accuracy in the Chinese software industry. We also investigated the possible causes of inaccurate estimations.

2.2 The choice of estimation method

It is reported in previous surveys [3, 4, 8, 14] that most projects used analogy or expert judgment to make estimations, and very few projects (14%-26%) used model-based estimation methods. The model-based methods include formal estimation models such as COCOMO, Use-Case-based estimation, FPA-metrics or other algorithm driven methods.

Researchers discussed many possible reasons for the low adoption rate of model-based methods, e.g. many of these models are not very accurate [15], many organizations do not collect sufficient data to allow the construction of such models [16], organizations feel uncomfortable to use models they do not fully understand [17], etc. There is still no industry survey on why model-based methods are not used extensively. In this survey research, we have tried to explore this question.

2.3 The importance of software cost estimation

There are two previous surveys directly addressing the importance of software cost estimation, and they got similar results. Lederer [18] found that 84% of the respondents described cost estimating as "very important" or "moderately important". Moløkken-Østvold [3] found 78% of the respondents answered that estimation was "very important", "extremely important" or "most important".

It is also important to know whether organizations are satisfied with the current software cost estimation. If organizations are satisfied with the software cost estimation, they will have fewer incentives to improve the estimation methods and process; and if organizations are dissatisfied, they can invest more to improve the software cost estimation. However, the answer to the above question is not so direct and simple. Lederer [18] reported that although systems development cost estimation is important and not accurately done, computing professionals are neither particularly satisfied nor dissatisfied with their estimation. The average rating was 3.02 on the one-to-five point scale (1=very unsatisfactory 5=very satisfactory). He proposed that when contrasted with the high importance of cost estimation and lack of accuracy, the satisfaction figures suggest that computing professionals may have accepted the inaccuracy of cost estimation as a fact of life.

Moores and Edward [13] found that 91% of the responding mangers answered 'yes' to the question 'do you see estimation as a problem?', while only 9% answered 'no'. If both Lederer and Moores are right, then computing professionals may have accepted the problems of cost estimation as a fact of life.

In this survey research, we have surveyed on the importance of software cost estimation and people's satisfaction level about the current software cost estimation.

2.4 For what purposes are cost estimations used

It is discussed in literature [15, 19, 20] that software cost estimation can be used for a number of purposes, e.g. budgeting, tradeoff and risk analysis, project planning and control, and software process improvement analysis. Different usage purposes of estimation may require different properties of estimation methods, and can produce different values to organizations. For example, when estimations are used for software process improvement analysis, the cost estimation model used should have good transparency so that we can analyze the factors influencing the productivity [21]. Therefore, the usage purposes of cost estimation are useful information for the improvement of software cost estimation methods.

We find only one existing survey [22] that addressed the uses of estimations. The top usage purpose reported was project planning and scheduling. However, the previous surveys have not addressed the usage purposes such as software process improvement analysis or short-term project planning.

2.5 When do organizations usually make cost estimations?

An interesting question is "When do organizations usually make cost estimations". Such information is good for us to get insights on the needs of organizations and potential improvement of estimation methods.

As project lifecycle proceeds, organizations usually have different amount of information for software cost estimation and may apply different estimation technologies at different development phases. It is illustrated in [1] that the uncertainty ranges of cost estimations present a decreasing trend as the software development proceeds. This phenomenon is named as the cone of uncertainty [23]. Gryphon added that the Cone of Uncertainty does not reduce itself, and it may be reduced by the improved estimation methods that become available as the project progresses [24].

Lederer [18] addressed this question and found that 77% projects prepared estimation during initial project proposal stage, 64% projects prepared (or revised) estimation during feasibility study, 51% during systems analysis, and 48% during systems design. However, this survey was conducted in the early 1990s, and the development technologies and process have changed a lot since then. In this study, we have also surveyed on the development phases when organizations usually make estimations.

2.6 Research questions

As discussed in the above subsections, this survey study tries to investigate the following research questions related to software cost estimation.

RQ1: What is the accuracy of effort and schedule estimation?

RQ2: Does project size affect effort and schedule estimation accuracy?

RQ3: To what extent are different estimation methods used in the industry?

RQ4: For what purposes are cost estimations used?

RQ5: How important do people think estimation is, in comparison with other aspects of development?

RQ6: How satisfied are people with the current software cost estimation?

RQ7: When do organizations usually make cost estimations?

RQ8: What are the causes of inaccurate estimations?

RQ9: What are the barriers and difficulties in the application of software cost estimation models?

3. METHODS

The research questions we try to investigate require various types of information, e.g. RQ1 requires objective historical project data and RQ6 requires subjective opinions. Only one survey method such as analyzing historical project data, face-to-face interviews, or questionnaire surveys cannot provide enough information we need. So our research consists of two parts with different data collecting methods.

The first part of our study was investigating RQ1 and RQ2. These two research questions require large amount of detailed quantitative project data, e.g. the planned effort and the actual effort. We found that such information is hard to collect by conducting interviews or questionnaires, because not all organizations have collected such data and recalled information would not be reliable. In this research, we have analyzed the Chinese Software Benchmarking Standards Group (CSBSG) project data set, which provided us with reliable historical project data for investigating RQ1 and RQ2.

The second part was investigating RQ3 to RQ9. The CSBSG data set does not have the information required by RQ3 to RQ9. Because most respondents can provide such information winthin relatively short time without difficulties, we have conducted questionnaire surveys to elicit answers to RQ3 to RQ9. The questionnaire survey method can cover more organizations compared with face-to-face interviews.

3.1 Analyzing the CSBSG data set

The Chinese Software Benchmarking Standards Group (CSBSG) was initiated by the Chinese Systems and Software Process Improvement Association (CSSPI), a branch of the China Software Industry Association. The goal of the CSBSG is to build a large software project benchmarking data set that can reflect the status and best practices of the Chinese software industry. In addition, the CSBSG data set is the first Chinese benchmarking data set with the support from government and industry association.

The CSBSG data contains project metrics similar to the ISBSG data, e.g. size, effort, schedule, defect, etc. Since 2006, the CSBSG has collected 1012 project data from 141 organizations.

Among the 1012 CSGSG projects, 112 have recorded complete information of planned and actual values of project development effort and schedule. And we used all the available 112 projects to investigate RQ1 and RQ2. These 112 projects came from 61 organizations and covered wide spectrum of business area such as communication, finance, manufacturing, government, etc. Figure 1 shows the distribution of the 112 projects among business areas.

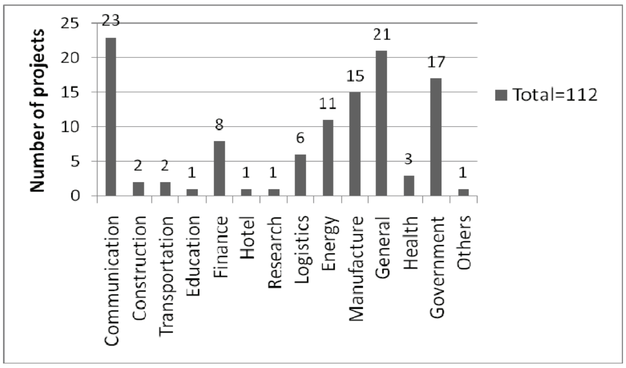

Figure 1. Projects distribution among business areas

The size (source line of code), effort (man-hours), and duration (days) of the 112 projects are summarized in Table 1.

Table 1. Summary of the 112 projects

	Mean	Median	Min	Max
Size (SLOC)	123788.1	46116.5	1480	2339728
Effort (Man-Hours)	7883.6	4102	160	115816
Schedule (Days)	175.1	150	10	851

We measured the accuracy of estimation using the percentage difference between the actual value and the estimated value, as recommended by Conte et al., [25]. Bergeron and St-Arnaud [10] argued that this measure is more meaningful, since profit or loss should be calculated on the basis of expected cost by most project managers. The measure Relative Error to the Estimate (REE) was defined in the formula below [25]:

$$REE = \frac{x-y}{y} \quad x = \text{actual}, \quad y = \text{estimate}$$

3.2 Conducting questionnaire survey

Because the CSBSG data set did not have the information required for RQ3 to RQ9, we have conducted questionnaire surveys to collect data. We got support from the CSSPI, a branch of the China Software Industry Association, during the survey research, because the research results may contribute to the software process improvement in Chinese. With support from industry association, we got access to and trust from a large number of organizations. The questionnaires were distributed during the "2007 Chinese Systems and software process improvement conference". The participant organizations covered different size, business area, and maturity level. Most of these organizations were also CSBSG member organizations, which made the two parts of our surveys more consistent.

We have followed the seven-step process defined in the SEI Guideline of Survey Design [26] during the questionnaire surveys. Below are the seven-step survey process we followed and some related information:

1) Identify the research objectives

Explore the status of software cost estimation. Get insights on potential improvements of cost estimation methods and process.

2) Identify and characterize the target audience

The organizations that are performing or are concerned about software process improvement.

3) Design the sampling plan

The organizations participating the "2007 Chinese Systems and software process improvement conference" came from various provinces of China, and most of them are CSBSG member organizations.

4) Design and write the questionnaire

Most of the questionnaire questions are designed as close-ended, so that it is easy to be completed within limited time.

5) Pilot test the questionnaire

We performed a pilot study within a CMMI-4 software development organization and validated the feasibility of this questionnaire survey. During the pilot study, we asked the respondents about how they understood and answered each questions. The respondents felt the questions were easy to understand and answer, except for the questions asking about the performance of estimation. The respondents found it very hard to recall historical projects and quantitatively describe the estimation performance. Besides, existing researches [3, 17] show that people are often too optimistic about estimation performance. So we removed the questions about estimation performance from the final questionnaire and kept the questionnaire simple.

6) Distribute the questionnaire

We distributed 400 questionnaires during the "2007 Chinese Systems and software process improvement conference".

7) Analyze the results and write a report

During the final questionnaire survey, we got 171 respondents from 116 organizations. The sizes of the organizations, as summarized in Table 2, were higher than the average level of Chinese software companies. And Table 3 shows that most of these organizations were performing software process improvement under CMM, CMMI, or ISO-9000 standards.

Table 2. Size of the 116 respondent organizations

Organization size (Persons)	# of organizations
< 51	9
51-100	11
101-200	26
201-500	32
501-1000	19
> 1000	19

Table 3. Information about software process improvement

Software Process Improvement Standards	# of organizations
CMM	5
CMMI	43
ISO-9000	15
CMM&CMMI	1
CMM&ISO-9000	8
CMMI&ISO-9000	25
CMM, CMMI, ISO-9000	4
No SPI assessment	15
Total	**116**

Research questions RQ5, RQ6, RQ8, and RQ9 used individuals as unit of analysis, while research questions RQ3, RQ4, RQ7 used organizations as unit of analysis. For more than one response from one organization, we need to aggregate data from respondents of such organization when analyzing RQ3, RQ4 and RQ7.

Because RQ3, RQ4 and RQ7 were all multiple-choice questions, we used the following simple rule to aggregate the response: for organization as unit of analysis, the selected responses are the answers that were chosen by one or more respondents from the same organization.

4. RESULTS AND ANALYSIS

4.1 What is the accuracy of effort and schedule estimation? (RQ1)

Figure 2 and Table 4 demonstrate the cost estimation accuracy of the 112 projects from the CSBSG data set. The REE is measured by "(actual – estimation)/estimation" as described in section 4.1.

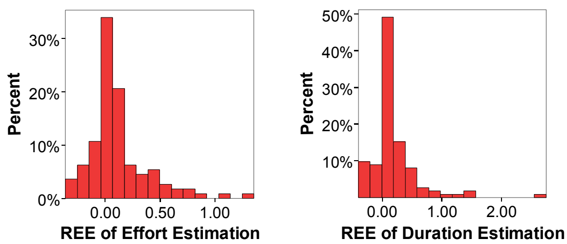

Figure 2. Distribution of the cost estimation REE

We find that very few (3 and 3%) projects have exactly the same estimated effort and actual effort, but many (35 and 31%) projects have small effort estimation REE between -0.05 and 0.05. However, relatively larger numbers (19 and 17%) of projects have the same estimation schedule and actual schedule. We think that these results are reasonable. Because the actual effort data was usually collected from the weekly personal effort report, there was usually small chance that actual effort was the same as the estimated effort. And the schedule was usually documented in contract or set as a goal, so organizations had more incentives to keep the projects on schedule.

The summary of estimation accuracy in Table 4 shows that 53 (47%) projects overran effort larger than 5%, and 25 (22%) projects overran effort larger than 20%. The mean effort overrun was 11.2%, and the median effort overrun was 5.2%.

Table 4. Summary of the estimation accuracy

Estimation REE %	Effort (n)	Schedule (n)
-21 and less	7	11
-6 to -20	14	3
-5 to 0	12	9
0	3	19
0 to 5	23	13
6 to 19	28	24
20 to 39	10	14
40 to 59	9	10
60 to 79	2	2
80 to 99	2	2
100 and more	2	5
Mean	12%	17%
Median	5%	7%
Std. Dev.	0.27	0.41

Most projects (70 and 63%) overran their intended schedule, and 33 (29%) projects overran their intended schedule larger than 20%. The mean schedule overrun was 17%, and median overrun was 7%. The accuracy of schedule estimation was worse than that of effort estimation.

Table 5 summarizes the estimation accuracies reported in previous researches and the result in our study of the Chinese Software Industry, by extending the literature review by Moløkken-Østvold [3]. The estimation accuracy in our study is similar to that reported in previous research, especially the percentage of projects that underestimate effort or schedule.

Table 5. Comparison of the estimation accuracy

Sources	[6]	[7]	[8]	[9]	[10]	[11]	[3]	**New**
Year	1984	1988	1989	1991	1992	2003	2004	**2007**
Cost overrun	34% median	33% mean			33% mean	18% mean	21% median 41% mean	**5% median 12% mean**
Effort Act. > Est.	61%		70%	63%		59%	76%	**68%**
Effort Act. < Est.	10%			14%		15%	19%	**29%**
Schedule overrun	22% mean					23% mean	9% median 25% mean	**7% median 17% mean**
Schedule Act. > Est.	65%		80%			35%	62%	**63%**
Schedule Act. < Est.	4%					3%	2%	**21%**

The mean and median values of effort or schedule overrun in our study are smaller than those in previous research. One reason for the difference is that the percentages of projects that underestimate (Act. > Est.) effort or schedule in our study are similar to those in previous research, but the percentages of projects that overestimate (Act. < Est.) effort or schedule in our study are larger than those in previous research. Only using the mean and median values cannot reflect the estimation performance, so we have provided the distribution of accuracy in Table 4. Another reason for the difference in accuracy may be that the sample organizations in our study have higher software process maturity levels, so the projects developed in these sample organizations are more predictable. However, most of the sample projects in our study still suffered from schedule and effort overruns, and 29% projects overran schedule larger than 20%.

4.2 Does project size affect effort or schedule estimation accuracy? (RQ2)

To analyze whether project size affects effort or schedule estimation accuracy, we used a similar analysis method used by Moløkken-Østvold in [3]. We divided the 112 projects into halves based on the project size. Because the median size of the 112 project is 46116, the 56 projects smaller than 46116 SLOC are marked as SMALL Group, and the other 56 projects larger than 46116 SLOC are marked as LARGE Group.

Table 6 shows that the mean and median values of effort estimation REE in the group LARGE are both higher than those in the group SMALL. This indicates that the large projects have more effort overruns than the small ones. We performed t-test to examine the difference in mean accuracy between the LARGE and SMALL projects [27]. The p value of t-test for equality of means is 0.057. We find that the large projects also have larger variance of accuracy (higher standard deviation of estimation REE). The Levene's test [27] for equality of variances gets very small p value as 0.004, and we can conclude that the difference in variance of accuracy between the two groups is significant at the 0.05 level.

Table 6. Compare estimation accuracy between groups

REE of Estimation	Groups	Mean	Median	Std. Dev.	Levene's Test for Equality of Variances (p)	t-test for Equality of Means (p)
Effort	SMALL	0.07	0.04	0.20	0.004	0.057
	LARGE	0.16	0.08	0.33		
Schedule	SMALL	0.06	0.00	0.27	0.033	0.003
	LARGE	0.28	0.15	0.49		

When comparing the schedule estimation accuracy between the large and small projects, we get similar results. The large projects have higher schedule overrun and variance of accuracy. The t-test (for equality of means) and Levene's test (for equality of variances) both produce p values smaller than 0.05, which indicate that the differences are significant at the 0.05 level.

In [3], Moløkken-Østvold also used t-test to examine the difference in mean effort estimation accuracy between the 21 largest and the 21 smallest projects. The p-value in that previous research was 0.11, which was larger than what we got in our study. The difference of the estimation accuracy between the large and small projects was more significant in our study. One reason may be that we used more sample projects (112 vs. 42).

We can conclude that large software projects had lower effort and schedule estimation accuracy, were more prone to effort and schedule overruns, and had higher variance of estimation accuracy than small projects.

4.3 To what extent are different estimation methods used in the industry? (RQ3)

During our questionnaire survey, 115 organizations reported the estimation methods they were using. It is essential to note that the organizations may use a combination of two or more different methods. The estimation methods reported can be classified into four categories: expert judgments, analogy, capacity related or price-to-win methods, and model-based methods. The capacity-related methods regard software cost estimation as a capacity, and directly take the personal or schedule's constraints as estimates. The price-to-win methods also directly take some constraints, e.g. customer's cost and schedule budgets, as estimates. The model-based methods include formal estimation models such as COCOMO, Use-Case-based estimation, FPA-metrics or other algorithm driven methods.

Table 7 shows the research results in our survey, and Table 8 compare our research results with those reported in previous researches.

Table 7. Estimation methods used by organizations

Methods	Organizations (n)	Percentage
Capacity-related, price-to-win	61	53%
Expert judgments	80	70%
Analogy	80	70%
Model based methods	17	15%
Total	**115**	**100%**

Table 8. Estimation methods used by organizations

Sources	[14]	[8]	[4]	[3]	**New**
Year	1987	1989	1995	2004	**2007**
Percentage used each methods (more than one method possible)					
Expert consultation		26%	86%	100%	**70%**
Intuition and experience	85%	62%			
Analogy		61%	65%		**70%**
Software cost models	13%	14%	26%	28%	**15%**
Price-to-win		8%	16%		**53%**
Capacity related		21%	11%		
Top-down			13%		
Bottom-up			51%		
Other	12%	9%	0%		

Table 7 shows that 80 (70%) out of the 115 organizations used expert judgment for software cost estimation and the same number of organizations used analogy. These percentages are similar to those reported in the previous surveys [3, 4, 8], in which 86%-100% organizations used expert judgment, and 61%-65% organizations used analogy.

Only 17 (15%) organizations used model-based methods. This percentage is similar to the results (13%-14%) reported in 1987 [14] and 1992 [8], but smaller than the results (26%-28%) reported later in 1995 [4] and 2004 [3]. The low use of model-based method in the Chinese software industry deserves further analysis.

We find that 61 (53%) organizations used capacity-related or price-to-win methods. This percentage is much larger than that reported in the previous surveys. In the survey by Heemstra [8] and Wydenbach [4], only 11%-21% organizations used capacity-related methods and 8%-16% used price-to-win methods. Boehm argued that the capacity-related methods reinforce poor practices and the price-to-win methods generally produce large overruns, so these methods are unacceptable [1]. Researches [1, 20] suggested that organizations should estimate the most likely cost, and organizations can adjust some objectively defined cost driver like software size when making trade-off or feasibility analysis.

Our conclusions are that the Chinese software organizations should improve estimation process and reduce the usage of capacity-related or price-to-win methods.

4.4 For what purposes are cost estimations used? (RQ4)

Table 9 shows the frequency of the estimation usage purposes reported by 115 organizations.

Table 9. The usage purposes of software cost estimation

Where are estimates used?	Organizations (n)	Percentage
Project proposal evaluation	52	45%
Contract negotiation	53	46%
Making budget	85	74%
Project-level planning and control, e.g. effort or schedule distribution among development phases	97	84%
Short-term planning and control, e.g. weekly or monthly team work plan	58	50%
Software process improvement, e.g. assess new process, improve productivity	43	37%
Total	**115**	**100%**

The most frequently selected purpose is project-level planning and control. Other usage purposes (project proposal evaluation, contract negotiation, making budget, short-term planning and control, and software process improvement) were all selected by more than one third of the organizations. Though we could not survey on all possible usage purposes, the current research results indicate that software cost estimations are used for various purposes. Software cost estimation methods may be improved to better support the various usage purposes and bring more value to organizations.

4.5 How important do people think estimation is, in comparison with other aspects of development? (RQ5)

As Table 10 shows, most of the respondents (145 and 87%) rated software cost estimation as "important", "very important", or "most important", which is similar to the result reported by Lederer [18] and Moløkken-Østvold [3].

Table 10. The importance of software cost estimation

Importance	Persons (n)	Percentage
Most unimportant	1	1%
Very unimportant	2	1%
Unimportant	3	2%
Neutral	15	9%
Important	63	38%
Very important	77	46%
Most important	5	3%
Total	**166**	**100%**

4.6 How satisfied are people with the current software cost estimation (RQ6)

When asked "How are you satisfied with the software cost estimation in your organization?", most respondents (62%) selected "neutral", 36 (22%) respondents selected "very unsatisfactory" or "unsatisfactory", and 26 (16%) selected "satisfactory". We find similar phenomenon reported by Lederer [18], that is: the computing professionals were neither particularly satisfied nor dissatisfied with the software cost estimation.

Table 11. The satisfaction level of respondents

Satisfaction Level	Persons (n)	Percentage
Very unsatisfactory	2	1%
Unsatisfactory	34	21%
Neutral	100	62%
Satisfactory	26	16%
Very Satisfactory	0	0%
Total	**162**	**100%**

Considering the frequent effort or schedule overrun and the high importance of software cost estimation, Lederer proposed that the satisfaction figures suggest that computing managers and professionals may have accepted the inaccurate cost estimation as a fact [18]. The hypothesis by Lederer needs further research. However, the neutral satisfactory attitude by computing professionals may result in fewer incentives and less investment for improving software cost estimation methods and process.

4.7 When do organizations usually make cost estimations? (RQ7)

113 organizations provided feedback to our question "At what software development phases does your organization usually make cost estimations?". Table 12 shows that the organizations usually made cost estimations at early phases of software development lifecycle.

Table 12. When cost estimations are usually made

Software Development Phases	Organizations (n)	Percentage
Initial project proposal stage	64	57%
Feasibility study	76	67%
Requirement	84	74%
Design	41	36%
Implementation	31	27%
Integration and testing	20	18%
Transition	13	12%
Total	**113**	**100%**

The result is reasonable, because project feasibility analysis, budgeting, and project-level planning and control all require early estimation. There are many risks and uncertainties about user requirements and solutions at the early development phases, which makes the software cost estimation more difficult [20]. We suggest that software cost estimation methods should be improved to provide better support for early estimation, e.g. to take the uncertainties into account, to estimate the probability distribution of cost, and to assess the risk of cost overrun, etc.

4.8 What are the causes of inaccurate estimation? (RQ8)

We investigated similar potential causes of inaccurate estimations and used the same rating levels of "extent of responsibility" as those used by Lederer in previous survey [18], so that it is easier for us to make comparisons.

Table 13 shows several potential causes of inaccurate estimations surveyed by us, 167 respondents rated these causes according to their "extent of responsibility" for inaccuracy (Rating 1=very low responsibility and 5=very high responsibility).

The two most severe causes are volatile requirements and unclear requirements. They have high ratings of 3.82 and 3.7 for "extent of responsibility". As Lederer [18] proposed, the volatile requirements may make computing managers and professionals view systems estimating and development as an effort to hit a moving target.

Table 13. Causes of inaccurate estimations

Causes		Extent of Responsibility Mean Rating (1-5 scale)
1	Requirements are volatile	3.82
2	Requirements are unclear	3.70
3	Pressure from senior manager and client to set or change the estimation results	3.22
4	Not enough resource for estimation	3.17
5	Not efficient historical projects	3.13
6	Lack of appropriate software cost estimation methods and process	3.10
7	Lack of stakeholder collaboration	3.07
8	Lack of risk assessment and management	3.06
9	Lack of cost control according to plan	3.01
10	Lack of estimation tools	2.97
11	Hard to assess the ability of developers	2.90
12	Lack of product risk assessment	2.87
13	Estimation lack involvement of developers	2.72
Other causes proposed by respondents		
14) The project bidding requirements predefined the project cost		
15) The survival pressure and business pattern of company		
16) Lack of training and appropriate application of estimation methods		

The third cause with high rating (3.17) is "Pressure from senior manager and client to set or change the estimation results". It is the same problem we find in RQ3, that many organizations used capacity-related or price-to-win methods. These methods reinforce poor practices and generally produce large overruns.

The fourth cause with high rating (3.13) is "Not enough resource for estimation". This cause may be related to the problems we discussed in RQ6. Computing managers and professionals are neither particularly satisfied nor dissatisfied with the software cost estimation, so they do not have great desire to improve software cost estimation methods and process.

4.9 What are the barriers and difficulties in the application of software cost estimation models? (RQ9)

During our survey, 153 respondents reported the barriers and difficulties in applying cost models.

The top three selected factors are the high adoptions cost of estimation models (selected by 58% respondents), insufficient investment for improving software cost estimation (selected by 45% respondents), and the insignificant benefit from using cost estimation models (selected by 40% respondents). These three factors are correlated. When adopting cost estimation models costs a lot and provides insignificant benefit, the return on investment of using cost estimation models will be low, and organizations may be reluctant to invest and use model-based methods. Considering that most of the computing managers and professional were neutral about software cost estimation, the low ROI could result in even fewer incentives of using cost estimation models.

Table 14. Barriers or difficulties of applying cost models

Barriers or difficulties	Persons (n)	%
1) Software cost estimation models cost a lot of effort to collect data, configure parameters, calibrate models, etc	89	58%
2) Organization do not have sufficient investment for improving software cost estimation	69	45%
3) The software cost estimation models cannot bring significant benefit	61	40%
4) Lack corresponding tools which are easy to use	56	37%
5) Haven't found appropriate software cost estimation models or tools	52	34%
6) Software cost estimation models are hard to learn and use	34	22%
Total	**153**	**100%**
Other causes proposed by respondents		
7) Client didn't require using software cost estimation models		
8) Estimation models were not accurate and effective		
9) The culture didn't favor using software cost estimation models		
10) Senior managers only looked at the result and didn't care for the estimation process		
11) Schedule pressure was very high, and the estimation could not help		
12) Estimation models haven't well defined what kind of data need to be collected		

When examining the 17 organizations that reported using model based methods, we found they were all medium-to-large size and high process maturity organizations. We found that none of the 17 organizations was a small one with less than 50 employees, and most of them had more than 200 employees. Out of the 17 organizations, 16 were CMMI/CMM level 3 or above. The only organization that did not perform CMMI assessment was Microsoft China Research and Development Group, which was also a large size and high maturity organization. This observation suggests that it may be more difficult for small organizations or organizations with not much investment on process improvement to use model-based estimation methods.

5. THREAT TO VALIDITY

The main threat to validity of the study is that our samples may be biased to organizations above average size and process maturity level of the industry. Our sample organizations are those performing or being concerned about software process improvement. These organizations are more willing to cooperate with us on improving the software development. Perfect probability sampling is hardly possible, because there are always many organizations unwilling to cooperate or not having the ability to collect historical project data. The sample organizations in the study are consistent with our target audience and conforms to our research goal of exploring the potential improvement of software cost estimation methods and process.

An important factor when analyzing the accuracy of cost estimations is that we can only analyze the projects that properly recorded the estimated and actual values of development effort and schedule. These projects may be recorded by organizations with higher maturity levels than those who did not record project data. And the projects developed in higher maturity organizations may be more predictable and have higher estimation accuracy.

When evaluating the generalizability of the results, one must consider that this is a survey of Chinese software organizations. There may be cultural issues that reduce the generalizability of the results.

6. SUMMARY AND DISCUSSION

The goal of our survey research was to investigate current situation related to software cost estimation in China, identify areas of potential improvement, and try to provide suggestions on how to improve the software cost estimation methods (especially the model-based methods) and process.

The acceptance and use of cost estimation technology is an important aspect of current situation, and also critical for future improvement of software cost estimation. To show a better picture of current situation, we find the Unified Theory of Acceptance and Use of Technology (UTAUT) [28] can help us organize our various observations about technology usage, barriers of technology transfer, estimation performance, potential improvement, etc. In the UTAUT model, four constructs play significant roles as direct determinants of user acceptance and usage behavior: i) performance expectancy (the degree to which an individual believes that using the system will help to bring gains in job performance), ii) effort expectancy (the degree of ease associated with using the system), iii) social influence (the extent to which the individual believes that 'important others' believe that he or she should use the new system), and iv) facilitating conditions (the degree to which an individual believes that an organizational and technical infrastructure exists to support use of the system).

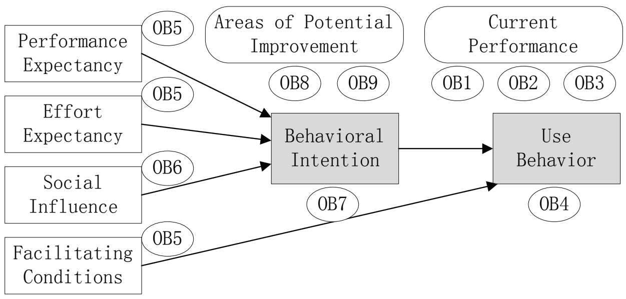

Figure 3. Summary of observations about current situation

Figure 3 shows the key elements of the UTAUT model (in rectangles), the two additional elements related to our survey results ('Current Performance' and 'Areas of Potential Improvement' in rounded-rectangles), and the nine observations about software cost estimation in the Chinese software industry (marked as OB1 to OB9 in ovals).

Related to the current performance of cost estimation we have the following observations:

- OB1: More than half of the software projects suffered from effort or cost overruns. The distribution of estimation accuracy showed that 22% projects overran effort larger than 20%, and 29% projects overran schedule larger than 20%.
- OB2: Large software projects had lower effort and schedule estimation accuracy, were more prone to effort and schedule overruns, and had higher variance of estimation accuracy than small projects.

- OB3: Requirements volatility, vagueness, and "pressure from senior manager and client to set or change the estimation results" are the top three rated causes for inaccurate cost estimations.

The estimation accuracy may be related to the estimation methods used. We also investigated the "Use Behavior" of estimation methods and found:

- OB4: Few (15%) organizations used model-based methods, and many (53%) organizations used capacity-related or price-to-win methods in the Chinese software industry.

More projects used capacity-related or price-to-win methods in our study than those reported in previous surveys. We suggest that the Chinese software organizations should improve the estimation process and reduce the use of capacity-related or price-to-win methods. Two pieces of evidence can support this suggestion: 1) existing literature suggests that the capacity-related and price-to-win methods reinforce poor practices and generally produce large overruns [1]; 2) OB3 indicates that 'set or change estimation results according to pressure like capacity or price' is one of the top three rated causes for inaccurate cost estimations.

As an explanation of the low use of cost estimation models, the UTAUT model suggests that "performance expectancy", "effort expectancy", "social influence", and "facilitating conditions" are the direct determinants of user acceptance and usage behavior. We obtained findings related to this explanation:

- OB5: The high cost of adoptions (related to ' effort expectancy'), insignificant benefit (related to 'performance expectancy'), and insufficient investment for improving software cost estimation (related to 'facilitating conditions') were the top three selected factors for low use of software cost estimation models.
- OB6: Three out of the six additional factors for low use of software cost estimation models proposed by participants were related to "social influence" as: 1) client didn't require using software cost estimation models; 2) the culture didn't favor using software cost estimation models; and 3)senior managers only looked at the result and didn't care about the estimation process.
- OB7: Computing managers and professionals were neither particularly satisfied nor dissatisfied with the software cost estimation. (We think that OB5 and OB6 may be some of the reasons for OB7. And OB7 may indicate that mangers or professionals do not have much desire to introduce new cost estimation technologies in such a situation.)

The UTAUT model and our findings suggest that: to introduce new cost estimation technology and change the current situation, we should also pay attention to the "performance expectancy", "effort expectancy", "social influence", and "facilitating conditions". Only providing accurate cost estimation is not enough to guaranty the acceptance and usage. The research work on the challenge of introducing a new software cost estimation technology into a small software organization [29] provides further evidence. In this case study [29], despite the Web-CoBRA cost estimation method was considerably more accurate than the company's estimation process [16], the company did not fully adopted the new methods.

Related to the areas of potential improvements we found that:

- OB8: Software cost estimations were used for various purposes.
- OB9: Organizations usually made cost estimations at early phases of software development lifecycle.

OB8 and OB9 indicate two needs of software cost estimation, and cost estimation methods may deliver more benefits by better supporting various estimation goals [19] and early lifecycle cost estimation [30].

According to the UTAUT model, the acceptance and usage of cost estimation technologies can be improved by improving the performance expectancy, effort expectancy, social influence, and facilitating conditions. Then, we may identify more potential improvements of cost estimation technologies in these areas. However, improving the estimation technology is not the only way to improve software cost estimation, and we may also improve software development process to better cope with the uncertainty of cost estimation [31].

We hope that our observations and related analysis can stimulate more discussion, more research, and improvement of software cost estimation. In the future, we will continue to conduct surveys and experiments in more detail to investigate the factors influencing the transition of software cost estimation techniques and methods. We will try to design measures to evaluate cost estimation methods based on performance expectancy, effort expectancy, social influence, and facilitating conditions. We will also monitor the software development in China over time, to see the variance and changes in the performance of software cost estimations, related factors, and potential improvements.

Acknowledgements

We would like to thank: the Chinese Systems and Software Process Improvement Association for the support of this research; the participants of our survey for their time and effort; and the anonymous reviewers for their constructive suggestions.

This work is supported by the National Natural Science Foundation of China under grant Nos. 60573082，90718042; the National Hi-Tech Research and Development Plan of China under Grant No. 2006AA01Z182， 2007AA010303; the National Key Technologies R&D Program under Grant No. 2007CB310802.

REFERENCES

[1] Beohm, B.W. 1981. Software Engineering Economics. Prentice Hall PTR, Upper Saddle River, NJ.

[2] Briand and Wieczorek, L.C. 2002. Resource estimation in software engineering. In: J.J. Marcinak, Editor, Encyclopedia of Software Engineering, John Wiley & Sons, New York.

[3] Moløkken, K., Jørgensen, M., Tanilkan, S.S., H, Gallis, Lien, A.C. and Hove, S.E. 2004. A survey on software estimation in the Norwegian industry. In Proceedings of the 10th International Symposium on Software Metrics.

[4] Wydenbach, G. and Paynter, J. 1995. Software Project Estimation: a Survey of Practices in New Zealand. New Zealand Journal of Computing. 6, 1B, 317-327.

[5] Shepperd, M.J. 2007. Software Project Economics: A Roadmap. International Conference on Software Engineering: Future of Software Engineering. 304-315.

[6] Jenkins, A.M., Naumann, J.D. and Wetherbe, J.C. 1984. Empirical Investigation of Systems Development Practices and Results. Information & Management, 7: p. 73-82.

[7] Phan, D. 1990. Information Systems Project Management: an Integrated Resource Planning Perspective Model, in Department of Management and Information Systems. Arizona: Tucson.

[8] Heemstra, F.J. 1992. Software cost estimation. Information and Software Technology. 34, 10, 627-639.

[9] Lederer, A.L. and Prasad, J. 1995. Causes of Inaccurate Software Development Cost Estimates. Journal of Systems and Software. 31, 125-134.

[10] Bergeron, F. and St-Arnaud, J.-Y. 1992. Estimation of Information Systems Development Efforts: A Pilot Study. Information & Management. 22, 239-254.

[11] Sauer, C. and Cuthbertson, C. 2003. The State of IT Project Management in the UK 2002-2003. Templeton College, University of Oxford.

[12] Jørgensen, M. and Moløkken, K. How large are software cost overruns? A review of the 1994 Chaos Report. Information and Software Technology. 48, 4 (Apr. 2006).

[13] Moores, T.T. and Edwards, J.S. 1992. Could Large UK Corporations and Computing Companies Use Software Cost Estimating Tools? – A Survey. European Journal of Information Systems. 1, 5, 311-319.

[14] McAulay, K. 1987. Information Systems Development and the Changing Role of MIS in the Organisation. First New Zealand MIS Management Conference, Wellington.

[15] Briand, L. C., Emam, K. E. and Bomarius, F. 1998. COBRA: a hybrid method for software cost estimation, benchmarking, and risk assessment. Proceedings of the 20th international conference on Software engineering. (Kyoto, Japan, April 19-25, 1998). 390-399.

[16] Ruhe, M., Jeffery, R. and Wieczorek, I. 2003. Cost estimation for web applications. Proceedings of the 25th International Conference on Software Engineering. (Portland, Oregon, May 03-10, 2003).

[17] Jørgensen, M. 2004. A review of studies on expert estimation of software development effort. Journal of Systems and Software. **70**, 1–2, 37–60.

[18] Lederer, A.L. and Prasad, J. 1992. Nine Management Guidelines for Better Cost Estimating. Communications of the ACM. 35, 2 (Feb, 1992), 51 - 59.

[19] Boehm, B., Abts, C. and Chulani, S.2000. Software Development Cost Estimation Approaches—A Survey. Annals of Software Engineering. 10, 177-205.

[20] Stutzke, R.D. 2005. Estimating Software-Intensive Systems. Addison Wesley.

[21] Boehm, B., et al, Software Cost Estimation with COCOMO II. 2000. Prentice Hall.

[22] Park, R.E., Goethert, W.B. and Webb, J.T. 1994. Software cost and schedule estimating: A process improvement initiative. Special Report CMU/SEI-94-SR-3, Software Engineering Institute, Carnegie Mellon University. URL = http://www.sei.cmu.edu/publications/documents/94.reports/94.sr.003.html.

[23] McConnell, S. 1996. Rapid Development: Taming Wild Software Schedules. Microsoft Press.

[24] Gryphon, S., Kruchten, P., McConnell, S.and Little, T. 2006. Letters: The Cone of Uncertainty. IEEE Software. 23, 5, (September/October, 2006), 8-10.

[25] Conte, S.D., Dunsmore, H.E. and Shen, V.Y. 1986, Software Engineering Metrics and Models. Menlo Park: Benjamin-Cummings.

[26] Kasunic, M. 2005. Designing an Effective Survey. Technical Handbook. Pittsburg: Software Engineering Institute. Carnegie Mellon University. (September, 2005).

[27] Bryman, A. and Cramer, D. 2005. Quantitative Data Analysis with SPSS, Routledge.

[28] Venkatesh, V. and et. al, User Acceptance of Information Technology: Toward a Unified View. MIS Quarterly. Vol.27 No. 3, 425-478

[29] Keung, J., Jeffery, R. and Kitchenham, B. The Challenge of Introducing a New Software Cost Estimation Technology into a small Software Organization. Proceedings of the 2004 Australian Software Engineering Conference (ASWEC'04)

[30] Yang, D., et al, 2006. "COCOMO-U: An Extension of COCOMO II for Cost Estimation with Uncertainty", In: Wang, Q., Pfahl, D., Raffo, D.M., Wernick, P. (eds.) Software Process Change. LNCS, 3966, 132–141. Springer, Heidelberg (2006)

[31] Yang, D., Boehm, B., Yang, Y., Wang, Q. and Li, M. 2007. Coping with the cone of uncertainty: an empirical study of the SAIV process model. In Proceedings of International Conference on Software Process. (May, 2007) 37–48