Proceedings of the 2010 International Conference on Industrial Engineering and Operations Management
Dhaka, Bangladesh, January 9 – 10, 2010

Demand Based Pricing Model for Outsourced Software Application Maintenance

Jagadish Kundu
Cognizant Technology Solutions
GN-34/3, Salt Lake Electronic Complex
Kolkata, India 700 091

Abhik Mukherjee
Department of Computer Science & Technology
Bengal Engineering and Science University
Shibpur, Howrah, India 711 103

Abstract

The maintenance of the software applications are being outsourced across countries. There are increased concerns from the customers regarding productivity of the personnel of the vendor working out of a different country. Traditional methods of estimating effort and cost for outsourced software application maintenance are proving to be inadequate. Companies are looking for a flexible model where they will pay for the exact amount of service they are consuming and the cost should not be dependent on productivity of the individual programmers, familiarity of the programmers with the application, organization climate of the vendor etc. The Demand Based Pricing model suggested by the authors provides a solution which is addresses productivity concerns of the programmers of the vendor as well as provides the flexibility required by the customer in terms of the production capacity.

Keywords
IT Management, Software Maintenance, Effort Estimation, Cost Estimation, Demand Based Pricing

1. Introduction
There is a consensus that application maintenance takes up the majority of the effort [1] in the life cycle of the application software and the percentage of efforts spend on maintenance has increased over the years [2]. Driven by the factors like continuous pressure to reduce cost of maintenance year after year and increased intention of the companies to focus on core competencies, large companies across the world are outsourcing the maintenance of the application software to IT vendors. All major IT vendors are leveraging their global software development centers in low-cost countries like India, China, Russia, Brazil, etc. (called offshore centers) to get most of the maintenance work done, remaining work is getting done at the customer location (called onsite centers) or the IT vendors' offices near the customer location (called near-shore centers).

In most of these endeavors, in the area of application maintenance, IT vendors use Fixed Price (FP) or Time and Material (T&M) based pricing models. These methods of pricing are easy to understand and easy to transact with. IT vendors are quite conversant and comfortable with these pricing models. In the T&M model, IT vendors share very limited project risk with the customer and they get rewarded for doing the work efficiently or adding any intellectual capital to reduce the cost or improve quality or schedule of the project. The problems of this model become more aggravated when the maintenance projects get executed in the Global Software Delivery Model (GSDM) [3] of the vendors. In the GSDM, customer managers do not see most of the programmers who are working at offshore centers thousands of miles away. Not being able to meet the programmers working in the projects on a day to day basis, it is very difficult for the customer's IT managers to assess the commitment, efficiency and productivity of the offshore programmers. There is number of factors [4] which might cause inefficiencies in the projects getting executed in GSDM. Customer IT managers working far away from the offshore locations often feel uncomfortable in handling these factors. Also there are differences amongst individual IT vendors – due to variations in the organizational climate [5] which contribute significantly to the overall performance of the outsourced projects. Companies with several years of experience of working in GSDM are looking for better methods than the T&M, where they do not have to bother about the lesser productivity of any programmer, loss of productivity due to attrition, addition of fresh programmers into the team. There is significant work going on in the area of 'Transaction Based Pricing model' [11] which addresses these issues of vendor productivity. However, this model works better where the work is repeatable and the work processes are matured, like in the BPO or Infrastructure related services.

Fixed Price is another commonly used pricing model. In the model, customers has to commit on the scope of the work (a certain volume work etc.) and do not enjoy flexibility which is needed to cater to the changing business dynamics. Also, many a times, especially for the maintenance projects, there is ambiguity in the definition of the scope, resulting in conflict between the customer and the vendor. Another 'Outcome Based Pricing Models' [12] can solve many of these issues; however the pre-requisite like deep appreciation of client's business model and knowledge client's operational nuances make it difficult for offshore-based IT service providers to adopt this model.

Due to these shortcomings, many customers are looking for pricing models better than the traditional models. In this model, customer will pay for each of the service requests ('demand'). Let us call this 'Demand Based Pricing' model. In this model, customer gets the flexibility while they do not pay for any inefficiency on the part of the vendor. Though this model poses significant operational challenges to the IT vendors, this model provides the customers with required flexibility needed for outsourced maintenance projects and enables them to meet the changing business objectives, without being concerned about the efficiency of the IT vendor. Of course, the model will not provide absolute flexibility (i.e. no work at all for a period, or sudden surge in volume of work to very high percentage of the current volume of work) to the customer since a core team always needs to be there for knowledge continuity, as well as ramp-up/knowledge transition needs a definite lead time. In this paper, the authors have discussed the challenges of implementing such a model and how to overcome two of the fundamental challenges (effort and cost) and excluded the minimum/maximum levels of demands needed to support this model.

2. Challenges

In the Demand Based Pricing model, each maintenance task is viewed as a demand. Each demand will have a priority associated with it. Based on the priority, the demand has to be fulfilled within a stipulated time period as agreed upon previously for that category of priority. IT vendors can start working on the demand irrespective of the priority as and when the demand arrives, but has to complete the demand within a stipulated time period associated with the priority of the demand under the SLA ('Service Level Agreement') for the predefined price for the demand. The size of each demand varies and the arrival pattern of the demands also varies, depending upon the business needs and changing environment. However, since the IT vendor has to complete the task within the SLA time period, the vendor has to keep enough resources prepared to work on it. Deciding the optimum number of resources needed is a key challenge, since keeping higher number of resource will result in higher cost. In the competitive market place, accurately predicting the optimum number of resources required (using sophisticated forecasting techniques) will make the difference between a success and a failure.

Another challenge is how to arrive at the right price for a demand. The price should not be based on the actual effort spent on it, since this is very subjective and will vary on the productivity of the programmer and the environment provided to the programmer by the IT vendor. The price also should be independent of the ratio of work done between onsite and offshore, which influences the price. Customer should not be bothered about where the work is getting done (onsite or offshore) as long the quality requirements and SLA is met. It is a challenge for the IT vendors to decide the onsite-offshore ratio for the purpose of quoting the right price. The ratio will vary depending upon the nature of the task, urgency of the task, phase of the SDLC etc. Combining so many varying parameters and arriving at the right price for a demand is difficult. A good grasp of the expected amount of effort needed for a category of demand is the key to determining resource need. In this paper, we will try to address this challenge.

There are many other challenges which include categorizing each demand, defining prices for each category of demand, catering to the sudden change of volume of demand ('what to do with the resources?' or 'where to get additional resources as productive as the resources already working on the application?) etc. However, a mechanism to determine the 'most likely' effort for a demand is the essential first step.

3. Solution Approach

Pricing individual demands based on the actual effort spent is dependent on the productivity of the worker, hence violates the essence of demand based pricing. So, the basis cannot be the actual effort as prevalent in the T&M. In the same time, the estimated effort also is not a good basis. Despite of several years of research activity in the area of effort estimation for software maintenance, it still remains an elusive and challenging goal to accurately predict effort [7]. Though there are several methods prevalent (like Complexity Point, Function Point, Work Breakdown Structure, etc), they are subjective, erroneous and work within a narrow application domain. In the proposed model, the authors propose to continue the established estimation procedure between the vendor and the customer. To reduce the error, the authors propose to categorize each demand into a bucket ('demand category'). Instead of focusing on the accuracy of estimating the effort for the demand, the author focuses on the accuracy of finding the right bucket where the demand would fit in. The price for the bucket should be fixed. Since the price for a range of effort (any value between the lower limit and upper limit of the bucket) is fixed, the variations due to inaccuracy in the effort estimation will not have significant effect, as long as the demand category of the demand does not change. We find categorizing demands into demand categories is less challenging and time-consuming than estimating exact effort.

Based on our experience in the outsourced software maintenance, we observe that the size of more than 90% of the maintenance requests is less than 20 Person Days (PD). For the purpose of this paper, let's assume that all maintenance requests above 20 PD can be handled differently than Demand Based Model.

3.1 Categorizing the demand

Typically IT vendors maintain a database for keeping data related to all the maintenance demands (also called tasks) received from the customers. We have collected actual effort (less than 20 PD) spent on maintenance tasks by 50 customers across the world from one IT vendors for three consecutive years. We observe that maximum number of tasks needed less than 2 PD of effort. Also, we observe that as there is lesser and lesser number of tasks as the effort increases. Since the pattern uniform across customers and over the years, we have decided to categorize the demands based on the actual effort as follows.

Table 1: Categorization of the Demand

Actual Effort	%-age of total annual demands	Categorization of the Demand
0-2 PD	60 to 70%	Very Small
2-5 PD	20 to 25%	Small
5-10 PD	5 to 10%	Medium
10-20 PD	1 to 5%	Large

3.2 Weibull Distribution

The Weibull distribution [8] is a continuous probability distribution which is widely used to describe the behavior of random phenomena in applied science and engineering problems. The Weibull distribution has been given in a variety forms by different authors. We have choosen to use Weibull distribution model data collected for the maintenance tasks. We modeled the distribution of efforts for the demands into 3-parameter generalized Weibull distribution as given by Equation (1) below.

$$f(T) = \frac{\beta}{\eta}\left(\frac{T - \gamma}{\eta}\right)^{\beta-1} e^{-\left(\frac{T-\gamma}{\eta}\right)^{\beta}} \tag{1}$$

Where, η = scale parameter, β = shape parameter (or slope), γ = location parameter. We observe a consistent pattern in the values of η, β, γ (based on historical maintenance data for three consecutive years for 50 Fortune 500 customers). The values obtained within a tolerance of ±10% from the mean values of η, β, γ for each year for a customer. Thus we have concluded that Weibull distribution can be rightly used to model the actual effort data for maintenance tasks. In the subsequent section, we would explain how we have calculated the 'most likely effort' for each category of demand, since the data fits well into Weibull distribution.

3.3 Calculation of 'Most Likely' Effort for each Category

Approximation of a representative value ('Most Likely') in each Demand Category is taken from the median of the Weibull distribution within the each demand category. The median is given as by Equation (2) below.

$$T = Y + \eta[\ln 2]^{\frac{1}{\beta}} \tag{2}$$

There are several methods for finding out the Weibull parameters, out of which we chose Maximum Likelihood Estimation (MLE) [10] because it has many large sample properties that make it attractive. It is asymptotically consistent, which means that as the sample size gets larger, the estimates converge to the right values. It is asymptotically efficient, which means that for large samples, it produces the most precise estimates. It is asymptotically unbiased, which means that for large samples one expects to get the right values on average. The distribution of the estimates themselves is normal, if the sample is large enough. These are all excellent large sample properties. Based on the above methodology, we have developed a software program to estimate β, η and 'most likely' effort of any range of efforts. Using the software program which is based on MLE, we have analyzed project data for a customer for three years and arrive at the following 'most likely' effort (Table 2).

Table 2: Effort based on demand category for a customer

Demand Category	Average β	Average η	Most likely effort (PD)
Very Small	2.38	1.46	1.29
Small	1.56	4.5	4.04
Medium	1.14	9.2	8.75
Large	1.05	16.43	16.09

3.4 Calculation of Cost for each Category

In general, the wages for resources at onsite locations are significantly higher than similar resources at offshore locations. So, the cost of work done at onsite is much higher for same amount of work done at offshore. Though IT companies are devising different techniques to get more work done at offshore, still there are certain percentages of work needs to be done at onsite. Percentage of work done at onsite depends upon number of factors which include the nature of work, size of work, experience level of people at onsite compared to offshore, available turnaround time, current work load of the resources and many other factors. It is very difficult to use any deterministic algorithm to come out with the cost.

To obtain a most probabilistic estimate, we used Monte Carlo Simulation (MCS) [9] technique. This is a problem-solving technique used to approximate the probability of certain outcomes by running multiple trial runs, called simulations, using random variables. Its purpose was to help us use a range of possible values for our estimates, instead of using a fixed number. To come out with the cost, instead of fixing percentage of work to be done at onsite, we have given a range of percentages of work that can be done at onsite. The percentages can be modified based on the situation of the project and the historical data. Table 3 below illustrates the data used for MCS for the Medium category demand.

Table 3: Monte Carlo Simulation for a Medium category demand for a customer

	Category: Medium (5-10 PD)			Mean effort (PD) =		6.88
SDLC Phase	Requirement / Analysis	HLD / LLD	Coding	Testing	Documentation / QA	PM/On-Off coordination
Mean Phase effort (% of total effort)	20	10	25	30	5	10
Onsite Min %	50	10	0	0	0	40
Onsite Max %	100	100	10	25	25	60

Table 4 below illustrates the cost for each SDLC activities for different iterations of the MCS. We have used RANDBETWEEN() function on MS-Excel to come out with random values in the iterations.

Table 4: Cost for each SDLC activities for different iterations of the MCS

	Requirement / Analysis	HLD / LLD	Coding	Testing	Documenta tion / QA	PM/Off-onsite coordination	Total Cost ($)
Min	506.368	147.5072	302.72	363.264	60.544	226.7648	1607.168
Max	770.56	385.28	368.768	561.408	93.568	279.6032	2459.187
1	511.65184	295.4547	335.744	434.5958	75.07456	261.10976	1913.631
2	759.99232	313.9482	315.9296	418.7443	81.67936	276.96128	2167.255
3	633.18016	292.8128	368.768	474.2246	93.568	266.3936	2128.947

(Only 3 rows are shown, there were 657 rows)

Based on an assumptions regarding daily onsite & offshore rates (which will vary), our model determines the average cost ((Max Cost+Min Cost)/2.0). The standard deviation is calculated using the Excel function Standard Deviation = STDEVP (Min Cost, Max Cost, Average Cost). The total error (ε) is given by ε = 3* Std Deviation/ Sqrt(N), where N is the number of iterations required. We have used the formula to obtain the number of iterations required N assuming an error less than 2%. Table 5 below illustrates the input data used and the output obtained in the model.

Table 5: Output Table for medium category demand for a customer

Assumption: Onsite PD Rate (€)	560
Assumption: Offshore PD Rate (€)	176
Average Cost (€)	2033.178
Standard Dev	347.8354
error limit	2%
Total error count	40.66355
No of iterations required	658.5358
Median Cost (€)	2080.072

The above process illustrates how to compute the cost for a particular category of demand for customer. This is based on the historical data available with the customer for three consecutive data. Since the data will change every year, hence the cost needs to be revised every year and the data used to calculate the cost include last three years data.

4. Implementation

We had calculated β and γ for a European customer using three consecutive years' historical data available. We have applied the model in 4 software maintenance projects for the customer started from the beginning of 2009. The work was getting done in the customer location and also in the vendor location in India. Based on the Demand Based Pricing model, the vendor has given the customer the prices in terms of Euro for each the four categories of service. The customer decided to use the T&M method for the 1st year of the service. However, the customer and the vendor have decided to collect the data throughout the year and observe if this model is applicable contractually from the next year. Based on the data collected in last 6 months, we find that vendor would have lost about 5% revenue if the model was applied from the beginning. However, this would have resulted in lesser number of conflicts between the customer and the vendor regarding the productivity of the vendor employees, thus resulted in requiring lesser management overhead and increased customer satisfaction.

5. Conclusion

Though it will be easier to implement this model commercially, there will be significant operational challenges for the IT vendors to implement such a model. IT vendors have to schedule the maintenance tasks better in accordance to their corresponding SLAs. They have to manage the productivity of its resources, especially in situation like attrition, slower learning curve of new resources, etc. More importantly, they have to use forecasting techniques to predict arrival pattern of the demands, so that they can plan for addition/release of resources. Unlike manufacturing industries, the application of techniques in the areas of forecasting, scheduling, control, resource planning is uncommon in IT industry. IT industry needs to apply the techniques used in the areas of Operation Research for other industries to improve its processes. We believe that application of those techniques will be needed if IT vendors have to provide service in the Demand Based Pricing Model. This will also increase the overall efficiency of the IT industry. There are certain limitations in the current version of the model. One of the prominent limitations is it does not prescribe the minimum and maximum level of work volume that the customer has to provide. Without a certain minimum volume guaranteed, the vendors will not be able to manage the application specific knowledge and the sudden ramp-down in the team. Also there has to be limit on the maximum level work, because IT vendors needs a definite lead time for ramping-up the team and provide necessary application-specific knowledge. The current practice of deciding the team size in consultation with the customer also has to be incorporated into the model. For future work, fuzzy logic could be used to overcome the current limitation of classifying one task into a specific demand category. By introducing partial membership of a demand into two categories can reduce the effect of misclassification of a demand.

Acknowledgements

The authors would like to thank Saikat Lahiri, Indranil Chakraborty, Sanjay Mandal and Amitava Bit for their help and support for this research.

References

1. Piattini, M., Polo, M., Ruiz, F., 2003, Advances in Software Maintenance Management: Technologies and Solutions, Idea Group Publishing, 2003.
2. Glenn, L. H., and Ira, R. W., Application software maintenance: can it be controlled?, December 1984, SIGMIS Database, vol. 16, no. 2.
3. Santonu, S., Renuka S., and Krishnakumar, P., January 2008, A collaborative platform for application knowledge management in software maintenance projects, Compute '08: Proceedings of the 1st Bangalore annual Compute Conference.
4. Charalambos, L. I., and Robbie N., June 2008, A Risk profile of offshore-outsourced Development projects, Communication of the ACM, vol. 51 no. 6, pp. 89-94.
5. Bhatt, P., Shroff, G., Anantaram, C., and Misra, A. K., 2006. An Influence Model for Factors in Outsourced Software Maintenance, Journal of Software Maintenance and Evolution: Research and Practice, vol. 18, no. 6, pp. 385–423.
6. Ahn, Y., Suh, J., and Kim, S., 2003. The software maintenance project estimation model based on function points, Journal of Software Maintenance and Evolution: Research and Practice, vol. 15, pp. 71–85.
7. Maya, D., 2008, Complementing approaches in ERP effort estimation practice: an industrial study, Proceedings of the 4th international workshop on Predictor models in software engineering.
8. http://www.weibull.com.
9. http://www.decisioneering.com/monte-carlo-simulation.html.
10. http://www.weibull.com/LifeDataWeb/estimation_of_the_weibull_parameter.htm
11. www.sqs-conferences.com/de/vortraege/palanisamy_ab.pdf
12. http://www.siliconindia.com/shownews/IT_firms_prefer_outcomebased_pricing_model-nid-50701.html