Cost estimation in a highly dynamic software business

C. M. Lott
Bellcore
Morristown, NJ 07960
lott@bellcore.com

January 31, 1997

Abstract

Competitive pressures in the custom software contracting business have driven contractors towards the use of staged contracts. The use of this business model demands accurate cost estimates, but existing cost-estimation approaches have not been tailored for this situation. Both light-weight and heavy-weight approaches are needed to support the modern, highly dynamic software business.

1 Background

The current trend towards 75% software solutions that are delivered within 4–8 months has put intense pressure on software contractors. Contractors must estimate accurately if they hope to make a profit, and at the same time they must respond quickly to changes in scope with revised estimates. These market pressures have encouraged many contractors to use a business model called staged contracts. However, use of this business model demands changes to traditional estimation techniques.

Cost estimation refers to the problem of predicting the schedule and human resources necessary to accomplish a software project before work begins. Prediction has been traditionally problematic due to the difficulty of determining precisely what the customer wants based only on relatively short and informal discussions. Additionally, cost estimation is highly dependent on the type of contract that will be offered to the customer. Conventional practice is to offer the customer a bid that includes a single price to deliver a custom software solution. The price is generally understood to include all activities from requirements definition through deployment. Developing an accurate cost estimate for this type of engagement is difficult and exposes the contractor to many risks, primarily loss of revenue due to significant under-estimation of the work to be done. To manage these risks proactively, recent trends in industry indicate an increased use of the *staged contract* model for doing business.

A staged contract for custom software development means that the work is contracted based on significant milestones in the development process [Lot97]. The practical minimum is two stages. Stage 1 of a staged contract commonly involves defining the opportunity, scoping the work, identifying the customer's current and future method of operations, defining some limited ("thin") requirements, and possibly prototyping the user interface either on paper or using a tool such as Visual Basic. Stage 2 then involves the high-level and low-level design of the software, and the system would be implemented and deployed in stage 3. If the work effort is small, stages 2 and 3 are often combined. A proposal for a stage 1 effort can usually be written with a minimum of staff effort, and is often priced at a low rate (in the tens of thousands of dollars) in order to win business. Proposals for successive stages are more similar to traditional work proposals in terms of the amount of effort put into developing them and their price tags.

The staged-contracts approach is novel in several respects. First, it defines a procedure by which either party can discontinue the work at the conclusion of each stage (corresponding to a specific milestone). Second, stage 1 offers a great opportunity for careful estimation of costs required to deliver successive stages. Finally, it permits sophisticated risk management for both the customer and the contractor, even in time-critical, high-pressure projects.

2 Position

Cost-estimation approaches are needed that can utilize the rich information available after a stage 1 contract. For example, stage 1 deliverables may include a prototype graphical user interface (GUI). Aspects of that prototype GUI such as the number of windows, number of interactions with databases, number of report screens, and other measurements might be used in a sizing model to estimate the cost and schedule of delivering the desired functionality. Because of the many factors that can be involved, this would be best labeled a heavy-weight approach.

Approaches are also needed that will support dynamic scope changes. Many recommendations to software practitioners focus on minimizing changes to scope during a project. Although this is a sound practice, it opens the contractor to the risk of building the wrong system. Factors such as explosive growth in Internet use among businesses have made the custom software business in 1997 highly dynamic. Given the short time frames that customers expect of solutions, the risk of building the wrong system has become much larger when compared to traditional contracts in which more time might be available to explore and define requirements carefully. Competitive pressures have also forced contractors to be more open to scope changes. Although staged contracts help contractors and customers manage scope changes, that business model is not a panacea. The contractor must be able to respond to requested changes in scope with an estimate of the impact on both cost and schedule. Because scope changes tend to happen at the worst possible moments, a supporting cost-estimation approach must be as light-weight as possible to gain acceptance among practitioners.

3 Comparison

Existing cost-estimation approaches can be roughly divided into top-down and bottom-up approaches. Top-down approaches such as Barry Boehm's Constructive Cost Model (COCOMO) [Boe81] ask the estimator to supply a size estimate using object points or lines of code, answer a series of questions about product characteristics, and then apply a series of formulas to arrive at a predicted schedule and resource estimate. Bottom-up approaches ask the estimator to define a detailed work-breakdown structure, estimate the cost of the lowest-level tasks, and then sum the estimates (hoping thereby that the errors in the estimates will cancel). There is extensive literature on cost estimation, including [Boe81, DeM82, BP88]. Existing approaches, however, do not take business models such as staged contracts into account, and they offer a one-size-fits-all approach to a problem that demands tailored solutions.

References

[Boe81] Barry W. Boehm. *Software Engineering Economics*. Advances in Computing Science and Technology. Prentice Hall, 1981.

[BP88] Barry W. Boehm and Philip N. Papaccio. Understanding and controlling software costs. *IEEE Transactions on Software Engineering*, 14(10):1462–1477, October 1988.

[DeM82] Tom DeMarco. *Controlling Software Projects*. Yourdon Press, N.Y., 1982.

[Lot97] Christopher M. Lott. Breathing new life into the waterfall model. *IEEE Software*, 14(5), September 1997. To appear.

Biography

Christopher M. Lott received the B.S. degree in computer science from The Ohio State University, and the M.S. and Ph.D. degrees in computer science from the University of Maryland.

Lott is a research scientist in the Applied Research Area of Bell Communications Research in Morristown, New Jersey, USA. Previously, he worked as a faculty research associate with the Department of Computer Science, University of Kaiserslautern, Germany, where he helped establish the Fraunhofer Institute for Experimental Software Engineering in Kaiserslautern.

Lott's research interests include process modeling to support measurement and feedback, controlled experiments to evaluate software engineering technologies, and empirical studies of professional software developers. He is a member of the IEEE Computer Society.