Breathing new life into the waterfall model

Christopher M. Lott
Bellcore
445 South Street, MCC 1H-331B
Morristown NJ 07960
c.m.lott@ieee.org

Appeared in IEEE Software, September 1997

Imagine that you are in the market for some custom software. You’ve issued an RFP, prohibited any contact between bidders and your organization, and demanded a single price for the whole effort. In response, one of the prospective contractors pitches you the following scenario. “For US$35,000,” the contractor’s account executive says, “an cross-functional team of our best people will work with you for two weeks to define the opportunity, plan your future method of operations, write some thin requirements, and develop a simple prototype of the user interface. At the end of that period, we’ll probably offer you a proposal for doing the next stage of the work at a fixed price and within a fixed schedule. But we do reserve the right not to bid on the next stage.”

Should you be disgusted by the contractor’s insolence or delighted with their savvy approach?

Many customers have come to believe the latter. The approach behind the pitch given above goes by many names, including staged contracts and milestone-based development. Staged contracts are often used in an approach commonly called rapid applications development [1, 3]. (James Martin defined Rapid Application Development (RAD) in [2] as a particular set of tools and methodologies, but the term has since become a watered-down, overused buzzword.) In this column I want to focus on the contracts and pricing policy used by companies such as Bellcore, Cambridge Technology Partners, Keane, and Sapient as part of their rapid applications development methodology. These companies have enjoyed significant, even explosive growth in their rapid apps divisions, and assign much of the credit to the processes that they use. I claim that staged contracts, while clearly inappropriate for some, offer an excellent choice of life-cycle model for many projects.

A staged contract for custom software development means that the work is con-
tracted based on significant milestones in the development process. 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. The novelty here is defining a procedure by which either party can discontinue the work at the conclusion of each contract (corresponding to a specific milestone). This practice might be compared unfavorably with the use of prenuptial agreements, but it has benefits for both sides.

Many readers will already have recognized the waterfall life-cycle model [5] hiding behind a new coat of silicon paint. But things have changed – the old waterfall model has accumulated a series of incremental improvements without anyone really noticing the changes. The success in the 1990’s of this improved version of waterfall can be attributed in large part to different packaging and better technology support. Still, its success goes beyond just clever marketing and Visual Basic. The revised approach offers far better opportunities for risk management when compared to the traditional waterfall model.

Allow me to rehash the traditional way of doing business to illustrate the differences of the staged approach. In a traditional approach, contractors are forced to estimate costs based on limited information. Sources of information may be limited by customer policy (“it’s all in the RFP,” which it never is) or by a contractor’s unwillingness to commit the resources needed to gather information for a contract that it might not win. Developing an accurate cost estimate based on insufficient information is woefully difficult. Worse, this practice exposes the contractor to many risks if the work is underestimated and subsequently delivered late. Under a fixed-price contract, where the contractor has the lion’s share of the risk, a thin profit or a severe loss may result from underestimation. Under a time-and-materials contract, where the customer has assumed most of the risk, the customer may pay far more than expected if the estimate is exceeded, even if penalty clauses subsequently reduce the contractor’s billing rates. Even the spiral process model leaves essentially all risk with the contractor, although that model’s emphasis on risk management and reuse are steps in the right direction.

Staged contracts offer new ways of managing risk for both the customer and the contractor. First, the stage 1 work offers the contractor a chance to develop the information that will be required for accurate cost estimation. Therefore, the risk of failed or cancelled projects due to significant underestimation can be mostly eliminated. The contractor assumes the risk of completing only the preliminary portions of the work at a fixed price, but gains credibility with the customer by
taking on some risk. The stage 1 work also benefits the customer, because the risk of a project that snowballs out of control due to requirements creep can be managed explicitly. In fact, the concentration of up-front activity during stage 1 minimizes requirements creep during later stages. Second, there is no risk to the contractor of investing significant resources in a proposal that will not be accepted. The customer funds much of the work that previously was charged to overhead or special “pre-sales” accounts. Third, both the customer and the contractor minimize the risk of being involved with an intractable partner. Either party can choose to terminate the work – without litigation – after any stage. In short, the contractor sheds some of the risk from a fixed-price contract scenario onto the customer, but the customer doesn’t assume as much risk as with a time-and-materials contract.

Pricing policies, supported by careful cost estimation practices, are crucial for successful use of staged contracts. Each stage requires its own contract with some price tag, which incidentally provides excellent budget visibility for the customer. It is usual practice to quote customers a price for the stage 1 work after extremely little contact (low contractor investment). Work for the follow-on stages is not priced or bid until completion of the stage 1 work. This allows (some would say forces) the customer to finance the cost-estimation activity. Of course, payment schedules that correspond to work milestones are not new [2, p. 98], the concept has simply been extended to provide the customer with more control and better budget visibility over the course of the project. Another wrinkle of pricing policy involves work products. The delivery of intermediate work products such as a requirements document, and the amount of detail in the documents delivered to the customer, is highly dependent on the customer and the contract, and should be reflected in the pricing. Finally, it is common practice to rebate the price charged for the stage 1 contract if the customer commits to later stages. Competition has pushed companies to offer fixed-price, fixed-schedule contracts for the stages instead of more traditional time-and-materials contracts.

Like any approach, the staged contracts approach has a number of drawbacks. At the risk of stating the obvious, the customer must accept the staged approach. Even if a staged contract offers the only possibility for surviving a high-risk project, competitive pressures may prevent a contractor from winning business on this basis. Second, the use of a staged approach results in a fair amount of budget instability for the contractor. Contractors with unstable budgets may have considerable difficulty in retaining the experienced personnel necessary to complete projects successfully. Third, a staged contract may be bad for future business in the case of customers who are detail oriented and want to do work themselves. In other words, a customer may have a hidden agenda of stealing the contractor’s expertise during stage 1 (stealth technology transfer), then completing the work on their own. The dramatic learning that occurs in stage 1 as well as the contract structure make this
possible. Finally, the staged contracts approach should probably be avoided on projects with very short schedules (under about three months) because it does not add significant value. A project of this size involves relatively little risk, and the interval needed to do the staging could easily result in unacceptable delays.

Experience in Bellcore’s Rapid Applications Deployment Department has shown that customers are likely to accept the staged-contract approach when one of the following conditions holds. First, if there is some flexibility in the delivery date, the extra interval added by staging is an acceptable price to pay for the risk-management benefits. Second, if the customer realizes that they lack a coherent understanding of what they should do, then they are willing to fund additional up-front work to define the business opportunity before signing a contract to develop software.

If this article has convinced you that staged contracts are something your organization should investigate, here are some guidelines for making the transition from a conventional contract approach. If you are a software producer, begin by checking your organization’s cost-estimation and resource-management skills. The key to winning contracts while still making money on the stages is staying within budget, so estimates must be very good, and the resources must be managed carefully once the contract has begun. Next, both the customer and the contractor must involve the people who write their contracts. Contractual obligations and ownership of intellectual property must be treated carefully. For example, the customer may demand the right to receive a detailed design document that could be taken to a different contractor. Of course the people who prepare and review technical documents such as the requirements deliverable must be involved. Contractor personnel must walk a fine line between recommending generic and proprietary solutions. A proprietary solution frequently offers the most bang for the buck, and (to the contractor’s delight) locks in a customer for later stages. However, as a customer you may be well advised to demand a generic recommendation in addition so that you understand the functionality of any proprietary solution. Finally, your organization’s method of operations for software work may require tailoring. For example, unlike the traditional proposal phase, stage 1 will yield thin requirements and possibly preliminary design deliverables, and the early availability of these controlled documents must be treated appropriately.

Last but not least, it may be helpful to mention some of the warning signs that might indicate impending trouble during a staged contract, even though they are really no different from a traditional approach. Obviously slipped dates can jeopardize the completion of a stage and lead to significant slippage in later stages. Although legal review and signing of a contract generally involve significant delays, work that proceeds for a significant time based only on a letter of intent may be inviting trouble. During discussions or meetings, the lack of answers or the in-
ability to answer questions by either the contractor or the customer is never a good sign. Finally, defensive attitudes on either side, including blaming, secrecy, or not taking ownership of issues, is a clear signal that something is amiss.

In summary, the staged contracts approach offers a significant improvement over the traditional waterfall life-cycle model. The staged approach permits sophisticated risk management for both the customer and the contractor, even in the high-pressure projects found in companies that practice rapid applications development.

Acknowledgements. Many thanks to the Bellcore staff in Project Aurora, the Quality Management System, and the Rapid Applications Deployment Department for explaining details of staged contracts. Also thanks to Steve McConnell of Construx Software Builders for comments that greatly improved this article.

About the author. Christopher M. Lott is a research scientist in the Applied Research Area of Bell Communications Research in Morristown, New Jersey, USA. He 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. He is a member of the IEEE Computer Society.

References


