

Software Development Outsourcing Contract: Structure and Business Value

WILLIAM B. RICHMOND AND ABRAHAM SEIDMANN

WILLIAM B. RICHMOND is Associate Professor at George Mason University. His main fields of research are outsourcing, economics of information systems and decision support systems. His articles have appeared in such journals as *ACM Transactions on Database Systems*, *Decision Support Systems*, and *Journal of Economic Dynamics and Control*.

ABRAHAM SEIDMANN is an Associate Professor and Areas Coordinator of Computers and Information Systems, Management Science and Operations Management at the William E. Simon Graduate School of Business Administration at the University of Rochester, NY. He is a Department Editor on Manufacturing and Automated Production in *IIE Transactions*, and an Associate Editor for Production Planning and Controls, *Journal of Intelligent Manufacturing* and *International Journal of Flexible Manufacturing Systems*. Dr. Seidmann is the author of many research articles mostly published in such journals as *IIE Transactions*, *Operations Research*, *International Journal of Production Research*, *Management Science*, and *IEEE Transactions*. His current research and consulting activities include management and analysis of automated production systems, computer integrated manufacturing (CIM), management information systems, and stochastic models for scheduling and control.

ABSTRACT: We address the case where a user contracts for the delivery of a new information system from an independent vendor, both of whom are risk-neutral. The delivery task is partitioned into two consecutive stages: system design and software development. The parties can contract for each stage separately or specify an initial contract that covers both stages. We compare the impact of different contracting structures on prices, project value, project completion probability, and the value to the developer of obtaining the first stage of the contract. Specifically, we show that a two-stage contracting can lead to a higher business value than stage-by-stage contracting. When there is competition for the design stage, the vendors bear more of the software development risk, and the probability the system will be completed depends on the contract structure.

KEY WORDS AND PHRASES: contract development of software, outsourcing, software contracting, software development risk.

Acknowledgments: The authors would like to thank Tridas Mukhopadhyay and Robert Kauffman, and the Journal referees for many useful comments. This research has greatly benefited from our discussions with Professor Glenn M. MacDonald of the William E. Simon Graduate School of Business Administration.

THE RAPID EVOLUTION AND DEPLOYMENT OF ADVANCED INFORMATION SYSTEMS (IS) technology have stimulated a new set of management challenges. The strategic role of many business information systems and the complexities of deploying and managing advanced software projects have led to growing concerns about the corporate information systems management function [3]. Many problems involved in the management of information systems resources are organizational, so generating an organizational structure that coordinates the diverse participants and activities is a major management goal. One structure that many IS managers are currently investigating is the outsourcing of software design and development tasks.

Outsourcing is the subcontracting of some or all IS functions by one firm to another. Although much of the media attention is directed at outsourcing facilities management, outsourcing encompasses all information-processing activities, including software development, systems integration, and network management. This paper focuses on contracting issues that arise when firms outsource their software design and development activities.

Research on IS outsourcing and the related contracting issues is relatively new; three recent papers on the subject are [1, 2, 4]. Whang [4] identifies the components common to most software development contracts and develops a model of software contracting to investigate the impact of the user's and the vendor's different goals and how to align them. Richmond et al. [2] investigate the impact potential future enhancements can have during the software development process and on the decision to outsource the system development. Klepper [1] proposes management initiatives aimed at the development of IS partnerships between the user and the vendor and shows that investments made by the user or the vendor in expectation of a deeper and more rewarding relationship may come to naught.

In this paper, we address the case where a *user* outsources the delivery of a new information system to an independent *vendor*, both of whom are risk-neutral. The delivery task is partitioned into two consecutive stages: the system *design* stage and the software *development* stage. The first stage includes system planning, analysis, and design specification, while the second stage comprises the actual programming task, verification, data conversion, and installation. The parties can contract for each stage separately or specify an initial contract that covers both stages.

Our objective is to compare the impact of different contracting structures on prices, project value, project completion probability, and the value of obtaining the first stage of the contract. To obtain meaningful and useful results, we develop a rich analytical model that incorporates key facets particular to software development projects. These include: (1) information asymmetry over the system's costs, business value, and investment level between the user and developer; (2) the multistage nature of software development; (3) the dynamics of the user's and developer's knowledge accumulation during the software design and development stages; and (4) the impact of specific investment in the design stage on business value and the software development cost. For both stage-by-stage and two-stage contracting, we examine two cases. In the first

case, the vendor has market power, or is selected for reasons other than price, such as a previous business relationship with the user. In the second case, there is perfect competition among the vendors. We present evidence that there are significant differences in terms of price, completion probability, and business value between the two contracting frameworks studied here, and the value to the user depends, in part, on the degree of competition among vendors for the design stage. This indicates that management should pay careful attention to the early design of the contracting structure and the compensation policy negotiated with the vendor.

In the next section we present a new model for stage-by-stage contracting. The model assumes that the user will sign separate contracts for the design and development stages. Switching vendors after the design stage is possible, but it may result in more expensive software development. After that, we develop a two-stage contracting scheme in which the vendor's bid specifies a total price for both the system design and the system development stages. The contract also specifies a cancellation penalty the user must pay to the vendor if the user cancels the software development stage. We then present an example and compare the impact of the two contracting frameworks. We end the paper with several concluding remarks.

Stage-by-Stage Contracting

THIS SECTION DISCUSSES THE CASE WHERE THE USER HIRES AN EXTERNAL VENDOR, and they contract on a stage-by-stage basis. The sequence of decisions is shown in figure 1. The user company wants to select a design vendor and a development vendor so as to maximize the company's net business value. The user company solicits bids for the design stage from a set of homogeneous vendors. In responding to the request for proposal, the design vendor must select a price for the design stage, P_{des} . Because the vendors are homogeneous, the price is the same across all vendors. This allows us to focus on the characteristics of stage-by-stage contracting versus two-stage contracting, while omitting complicating factors associated with selection among heterogeneous vendors.

Based on the bids received from the design vendors, and on expectations about the business value of developing the system, the user company decides whether to hire a design vendor and start the project or cancel the project. Both the user and the vendor believe the system's value is uniformly distributed between $BV_H(I_d)$ and $BV_L(I_d)$. Once the design contract is signed, the design vendor makes a specific investment, I_d , and completes the design work. The cost of the investment is $C_{inv}(I_d)$. The investment lowers the cost of the development stage by $C_{save}(I_d)$, and it raises the system's expected business value $(BV_H(I_d) + BV_L(I_d))/2$, where $BV_H(I_d)$ is the highest business value the user expects from developing the system and $BV_L(I_d)$ is the lowest business value the user expects from developing the system. The specific functions for the cost of investment and the savings from that investment depend on the vendor's production function. The relationship between the investment and the system's business value depends on the system under development. We assume that the vendor faces diminishing returns from its

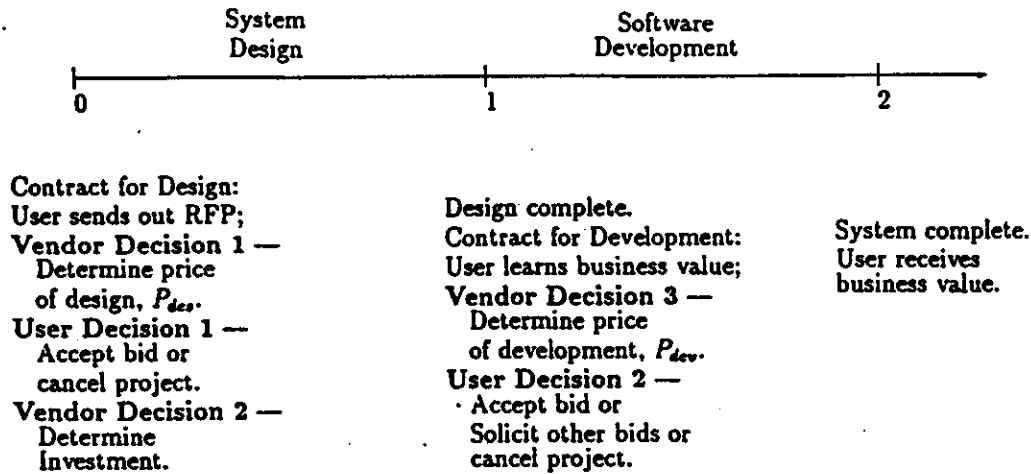


Figure 1. Timeline of Decisions for Stage-by-Stage Contracting

investment (i.e., that the added cost of increasing investment eventually exceeds the added value from the additional investment).

When the design is completed, the user observes the actual business value the system will generate, BV , and solicits a bid for the development stage from the original design vendor. If this bid is rejected (because it is too high), the user company solicits bids from other firms in the market. After comparing these external bids with the observed business value, the user will either cancel the project or hire a vendor from the market for the software development stage. To determine the optimal prices and investment level, we start with the user's second decision and work backward.

User Decision 2: Develop or Cancel System

The user's second decision is made after the design is complete. The user must decide whether to cancel the project or hire a vendor to develop the system. If the user continues the project, it must decide whether the system development stage should be contracted to the original system design vendor or to another vendor. These vendors are assumed to be homogeneous and charge a market price of C_{dev} for the software development stage. This price is the cost the vendor will incur to develop the software, since it has made no investment ($I_d = 0$) during the system design stage. It is also the cost the original vendor would face if it made no investment during the system design stage. Soliciting bids from other vendors entails a market search cost, C_{search} . In addition, bringing in another vendor entails costs associated with making the new vendor familiar with the project, $C_{learning}$. These costs together are denoted as C_s , $C_s = C_{search} + C_{learning}$, and are referred to collectively as switching costs. The user must consider these added costs when deciding whether to switch vendors.

When making this decision, the user knows the system's value, BV , the price quoted by the design vendor for development, P_{dev} , its cost to locate a new vendor, C_s , and the price the new vendor would quote, C_{dev} . The user's three alternatives and their payoffs are listed in Table 1. The user will select the alternative with the maximum value.

Table 1 User's Options at the Completion of the Design Stage

Alternatives	Payoffs
Cancel the project	$-P_{des}$
Continue with the design vendor	$BV - P_{des} - P_{dev}$
Continue with a different vendor	$BV - P_{des} - C_{dev} - C_s$

Vendor Decision 3: Determine Development Price

Working backward, we now consider the vendor's third decision, that of setting a price for the system development work, P_{dev} . The vendor knows the price paid for design, P_{des} , the investment it made, I_d , and the cost of developing the system $C_{dev} - C_{save}(I_d)$. It has prior expectations about the business value, which depend on its previous investment level. Given the level of investment, I_d , the business value is uniformly distributed between a low of $BV_L(I_d)$ and a high of $BV_H(I_d)$. The structure of this distribution function is known to both the user and the vendor and is denoted by $P(BV)$.

The vendor's investment increases the business value by enabling it to better identify the user's requirements and to determine the functionality that will improve the user's decision making. How the investment affects the business value depends on the system. For a typical transaction processing system, such as an accounts payable system, many of the requirements are standard. Investing to identify the user's unique requirements will improve the business value, but even if no investment is made, the users will benefit from automating (or improving) their current system. For a decision support system, however, understanding the decision-making environment and the decision maker's unique requirements is essential to the system's acceptance, usefulness and value. A lack of investment here can easily lead to the system's failure and result in a business value of zero.

The vendor wants to maximize its expected profits from the software development stage. Its expected profit is the probability the user will hire the vendor, given the price quoted for development, times the net value of developing the software at that price. Formally, it is:

$$(1) \quad \begin{cases} \int_{P_{dev}}^{BV_H(I_d)} P(BV) (P_{dev} - C_{dev} + C_{save}(I_d)) dBV & \text{for } P_{dev} > BV_L(I_d) \\ P_{dev} - C_{dev} + C_{save}(I_d) & \text{for } P_{dev} \leq BV_L(I_d) \end{cases}$$

$$(2) \quad = \frac{BV_H(I_d) - P_{dev}}{BV_H(I_d) - BV_L(I_d)} (P_{dev} - C_{dev} + C_{save}(I_d)) \quad \text{for } P_{dev} > BV_L(I_d).$$

When $P_{dev} > BV_L(I_d)$, to determine the optimal price of development, take the derivative of equation (2) and set it to zero. This results in an optimal development price of

$$(3) \quad P_{dev} = \frac{BV_H(I_d) + C_{dev} - C_{save}(I_d)}{2}.$$

Presenting the user with a lower price increases the probability that the user will continue with software development, but it also lowers the developer's revenue. This price represents the optimal trade-off between a higher probability of developing the system and a lower price for the development work. The optimal price for the system development work, however, is bounded from above by the market price plus switching costs and below by the minimum possible business value. Therefore, we get

$$(4) \quad P_{dev} = \min \left(\max \left(\frac{BV_H(I_d) + C_{dev} - C_{save}(I_d)}{2}, BV_L(I_d) \right), C_{dev} + C_s \right).$$

The vendor wants to charge at least the low business value, since charging less does not increase the probability that the user will continue with the software development stage, but it does lower the value to the vendor. The possibility that the user could hire another developer, however, limits the vendor's price to $C_{dev} + C_s$, the price another developer would charge plus the user's cost of switching to the other developer.

Two points are immediately noticeable about the price the vendor will charge for development. First, the price charged by the design vendor is always low enough to ensure that the user will not switch to another vendor. Second, as the switching costs and the potential low and high business values increase, the price for development also increases.

Vendor Decision 2: Determine Investment Level

Continuing to work backward, the next decision is the vendor's investment. Based on the different values of C_{dev} and C_s , and the functional forms of $BV_H(I_d)$, $BV_L(I_d)$, C_{Inv} , and C_{save} , the vendor will make different investment decisions. The vendor estimates these based on its experience and information contained in the request for proposal (RFP). The investment lowers the cost of completing the development stage and raises the business value (and therefore, either the price of the development stage, P_{dev} , or the probability that the user will continue with the development stage, or both). The equations specifying the investment level are complex and have been relegated to an appendix that is available upon request from the authors.

The investment decision is *independent* of competition among the vendors and the vendor's market power at the design stage. To understand this, remember that the vendor makes its investment decision *after* it has been awarded the design contract. The vendor's goal when determining the investment level is to maximize its expected value from the *development* stage (see equation [2]) minus the cost of making the investment, C_{Inv} . Both the expected value of the development stage and the cost of investment are the same under competition and when the vendor has market power.

Competition for the development stage can affect the level of investment. If the user can easily switch vendors after the design stage, then the maximum the design vendor can charge for the development stage is $C_{dev} + C_s$ (see equation [4]). This limits the

vendor's expected value from the development stage, which limits the investment it is willing to make. In this case, increasing the maximum value of $BV_H(I_d)$ or $BV_L(I_d)$ decreases the level of investment. This occurs because the vendor can lower its effort without changing the price it can charge for development or reducing the probability that the user will contract for the development stage. Simultaneously, it increases its expected net value by lowering the cost of its investment.

User Decision 1: Design or Cancel System

The next step is determining how the user will make its first decision, whether or not to start the system design, given the price of the design. The user knows only the price quoted by the vendor for the design. It has expectations over the investment the developer will make and the price of the development stage, and we assume that the expectations are accurate. The user can arrive at these expectations by playing the role of a vendor and determining the price it would charge and the investment level it would make. The user also expects the business value of the completed system to depend stochastically on the investment level, I_d , made by the design vendor in the design stage. The user's expectations are the same as the vendor's, and are uniformly distributed between a low of $BV_L(I_d)$ and a high of $BV_H(I_d)$. At this point, the user has two options. It can cancel the project, which will result in a net surplus of 0. Or it can continue with the design, and will do so if the expected value is greater than or equal to zero. Continuing with the design results in an expected net value to the user of:

$$(5) \quad \int_{BV_L(I_d)}^{P_{dev}} P(BV) (-P_{des}) dBV + \int_{P_{dev}}^{BV_H(I_d)} P(BV) (BV - P_{des} - P_{dev}) dBV,$$

where the first term is the expected payoff associated with canceling the system after design, and the second term is the expected payoff from completing the system. The total expected value equals

$$(6) \quad \frac{(BV_H(I_d) - P_{dev})^2}{2(BV_H(I_d) - BV_L(I_d))} - P_{des},$$

with a limit of

$$(7) \quad \frac{BV_H(I_d) + BV_L(I_d)}{2} - P_{des} - P_{dev},$$

when $P_{dev} \leq BV_L(I_d)$. This limit is reached when the user believes the minimum business value will be higher than the price the vendor will quote for the development stage.

Vendor Decision 1: Determine Design Price

Finally, the vendor must choose a price for design. To do this, it must estimate P_{dev} and I_d . These estimates are based on the vendor's experience and from working

backward through the problem as we are doing now. Since the vendor wants to maximize its expected profits, the price for design is limited by the user's expected business value (equation [6]) and is given by:

$$(8) \quad P_{des} = \frac{(BV_H(I_d) - P_{dev})^2}{2(BV_H(I_d) - BV_L(I_d))}.$$

This price will reduce the user's expected payoff to the minimum level for which the user will start the project (which we have normalized to zero). This price is the maximum at which the user will start the design, and it is the price that a design vendor will charge when it has market power. This happens when the user selects the design vendor for reasons other than price, such as a preexisting business relationship or special domain expertise, but wants to avoid price gouging for the development stage, which typically requires less company-specific expertise.

Competition among the vendors for the design stage will drive this price down, possibly to the point where the vendor's total expected profit, over both the design and development stages, is zero. Thus, the design price equals the cost of the design stage, C_{des} , plus the cost of investment, $C_{Inv}(I_d)$, plus the expected net cost of development, $P(\text{design and development})(C_{dev} - C_{save}(I_d) - P_{dev})$. Therefore, under perfect competition,

$$(9) \quad P_{des} = C_{des} + C_{Inv}(I_d) + P(\text{design and development})(C_{dev} - C_{save}(I_d) - P_{dev})$$

where $P(\text{design and development}) = P(\text{development} | \text{design})$

$$(10) \quad = \begin{cases} 1 & \text{for } P_{dev} \leq BV_L(I_d) \\ \cdot (BV_H(I_d) - P_{dev}) / (BV_H(I_d) - BV_L(I_d)) & \text{for } BV_L(I_d) < P_{dev} < BV_H(I_d) \\ 0 & \text{for } BV_H(I_d) \leq P_{dev} \end{cases}.$$

It is possible that this price will be negative—that the vendor would pay the company to design the software—with the expectation of charging enough for the development stage to raise its total expected profits to zero. The vendor will be willing to pay the user to design the system when the optimal price for the development stage is much greater than the cost of development. This will only occur when the user faces large switching costs.

The price for design depends greatly on the degree of competition, and different parameters affect the design price differently. For example, when the market price plus switching costs limit the development price, P_{dev} , increasing the switching cost lowers the price of the design stage. This occurs under competition and when the vendor has market power. Increasing the expected business value, however, increases the price of the design stage when the developer has market power, but has no effect when there is competition.

Two-Stage Contracting

THIS MODEL IS SIMILAR TO THE PRECEDING MODEL. We examine cases where there is competition among the vendors and when the vendor has market power or is selected for a reason other than price. The parameter definitions are the same as in the previous section, as is the impact of investment and the distribution of the business value. In two-stage contracting, the user contracts for both the design and development stages in the first period. The user wants to select a design and development vendor who will maximize the company's net surplus. Before selecting the vendor, the user solicits bids for the total cost of the system. The vendor must select a total price, P_{Total} , for the system and a penalty payment, $P_{penalty}$, that the user must pay if it cancels the system after the design stage. For convenience, we define $P_{Total} = P_{penalty} + P_{dev}$. Based on the bids received from the vendors, the user company decides whether or not to start the project. The selected vendor then makes a specific investment and completes the design work. Once the design is completed, the user observes the actual business value the system will generate and determines whether to cancel the project or use the vendor to complete the software development stage.

In two-stage contracting, the vendor has only two decisions instead of three. After the design stage, the vendor does not have to determine an optimal development price. The development price is part of the total price, which is determined as part of the vendor's first decision. To find the optimal investment level and prices, we start with the user's second decision and work backward.

User Decision 2: Develop or Cancel System

The second decision is made after the design is complete. The user must decide whether to cancel or continue with the project. At this point, the user organization knows the total price of the system, the penalty it must pay for canceling the system, and the system's business value. The user's two alternatives and their payoffs are listed in Table 2. The user will select the alternative with the maximum value, where P_{dev} is the difference between the total system price and the penalty.

The set of parameter values for which the user contracts with a vendor are the same under competition as when the vendor has market power. Only the distribution of the gains changes. It is not surprising that, *ceteris paribus*, increasing the maximum business value, $BV_H(I_d)$, the minimum business value, $BV_L(I_d)$, or the value of the investment, $C_{save}(I_d)$, increases the system's net value and the probability of the user finishing the project; whereas increasing the cost of the investment, $C_{inv}(I_d)$, lowers the system's net value and the chance the user will contract for the system design and development.

Vendor Decision 2: Determine Investment Level

The next step is determining the optimal investment level. Investing will raise the system's expected business value, thus increasing the probability that the system will

Table 2 User's Alternatives at the Completion of the Design Stage

Alternatives	Payoffs
Cancel the project	$-P_{penalty}$
Continue with the project	$BV - P_{Total} = BV - P_{penalty} - P_{dev}$

be completed. Making an investment in the design also decreases the cost of developing the system in the next stage. Determining the optimal investment requires determining the vendor's expected payoff, given that the user hired it to design and develop the system. The vendor's expected payoff from making an investment and completing the development work is:

$$(11) \quad P(\text{development} | \text{design}) (P_{dev} - C_{dev} + C_{save}(I_d)) - C_{Inv}(I_d).$$

Based on the different values of C_{dev} and C_p , and the functional forms of $BV_H(I_d)$, $BV_L(I_d)$, $C_{Inv}(I_d)$, and $C_{save}(I_d)$, the vendor will make different investment decisions. Again, the vendor estimates these based on its experience and the information contained in the RFP. The equations specifying the investment level are complex and are developed in the appendix. As in the stage-by-stage contracting approach, the investment decision is endogenous to the developer. This makes it *independent* of competition among the vendors and the vendor's market power.

The investment is sensitive to P_{dev} , the difference between the total price and the penalty. In stage-by-stage contracting, determining P_{dev} occurs after the investment decision; therefore, when working backward, P_{dev} is known when determining the optimal investment level. In two-stage contracting, the vendor determines P_{dev} before it selects its investment level. When working backward to determine the optimal investment level, P_{dev} is unknown. Too large a P_{dev} (i.e., too low a penalty, $P_{penalty}$) increases the probability that P_{dev} will exceed the realized business value, and the user will cancel the project after the design stage—thus lowering the vendor's incentive to invest. If P_{dev} is too low (i.e., the penalty is too large), the marginal expected value from increasing investment is low, since it will not greatly affect either the probability of the user canceling after the design stage or the vendor's total payoff.

User Decision 1: Design or Cancel System

The next step is determining how the user will make its first decision, the decision to start or cancel the project, given the total price of the design and development stages. To do this, it must estimate the vendor's investment level. As stated earlier, the user can do this by playing the role of the vendor. At this point, the user has two options. It can cancel the project, which will result in a net surplus of zero. Alternatively, it can start the project, and will do so if the expected value is greater than or equal to zero. Starting the project results in an expected net value given by:

$$(12) \quad \int_{BV_L(I_d)}^{P_{dev}} P(BV) (-P_{penalty}) dBV + \int_{P_{dev}}^{BV_H(I_d)} P(BV - P_{Total}) dBV,$$

where the first term is the expected payoff associated with canceling the system after the design stage, and the second term is the expected payoff from completing the system. The total expected value equals

$$\frac{(BV_H(I_d) - P_{dev})^2}{2(BV_H(I_d) - BV_L(I_d))} - P_{penalty},$$

with a limit of

$$(13) \quad \frac{BV_H(I_d) + BV_L(I_d)}{2} - P_{Total},$$

when $P_{dev} \leq BV_L(I_d)$. This limit is reached when the user believes the minimum business value will be higher than the price the vendor will quote for the development phase. Note that this is the same as in stage-by-stage contracting.

Vendor Decision 1: Determine Total Price and Penalty Payment for Cancellation

Finally, we examine the vendor's decision in setting the total price for the system and the penalty payment for canceling the system. When making this decision, the vendor has an expectation about the system's business value and knows the cost of developing the system. The vendor wants to maximize its expected profits from designing and developing the system. When the vendor has market power, or when it is selected for reasons other than price, the vendor will charge a total price that will drive the user's expected value from the system to zero:

$$(14) \quad P_{total} = P_{dev} + P_{penalty} \text{ is such that } \frac{(BV_H(I_d) - P_{dev})^2}{2(BV_H(I_d) - BV_L(I_d))} - P_{penalty} = 0,$$

with a limit of

$$(15) \quad P_{Total} = P_{dev} + P_{penalty} = \frac{(BV_H(I_d) + BV_L(I_d))}{2}.$$

Competition among the vendors can drive the vendor's expected profits to zero, resulting in a total price such that:

(16)

$$\frac{BV_H(I_d) - P_{dev}}{BV_H(I_d) - BV_L(I_d)} (P_{dev} - C_{dev} + C_{save}(I_d)) + P_{penalty} - C_{des} - C_{Inv}(I_d) = 0.$$

Note that the vendor's investment, and therefore $BV_H(I_d)$ and $BV_L(I_d)$, depend on P_{dev} . For a given investment level, lowering P_{dev} increases P_{Total} when the vendor has market power. Lowering P_{dev} may also lower investment, thus lowering P_{Total} , so the way the vendor breaks down P_{Total} between $P_{penalty}$ and P_{dev} affects the total price and the project's expected value.

Under competition, the vendor is indifferent to how it partitions the total price, but the user organization is not. How the vendor partitions the price affects the user's expected value. We assume that competition drives the vendors to partition the price to maximize the user organization's expected value.

Example and Analysis

IN THIS SECTION WE PRESENT AN EXAMPLE and calculate the optimal investment, prices and business value, and compare the results for stage-by-stage contracting and two-stage contracting.

Example

In a long-range system plan, Telco, a local telecommunications company, identifies a need for a billing system to support the new services, such as caller ID, video phones, and video on demand, that it plans to offer. In addition to billing, the system will provide information for marketing and support basic operating functions, such as turning on new services.

The internal information system group is overburdened with maintenance work, and the company wants to use a client-server architecture, with which its internal IS group has no expertise. Therefore, it has decided to hire an outside vendor to design and develop the billing system. Before sending out RFPs, the company uses industry contacts to determine which software developers have experience with the communications industry. These prescreened vendors are sent an RFP along with a detailed description of the proposed billing system that was developed during the system planning process. Initially, assume that the RFP specifies a stage-by-stage contracting scheme.

In response to the RFP, the vendor must determine a price to charge for the design work. To determine the appropriate price for the design stage, the vendor must first estimate what it would charge for the development stage. The development price, P_{dev} , depends on:

- The cost of development, which the vendor can estimate from its experience and the specifications outlined in the RFP.
- The switching cost, which depends on how easily another vendor can use the design documentation, its estimate of Telco's cost of searching for a new vendor, and its estimate of the delay cost to Telco from bringing in a different vendor.

- The functional form specifying how the system's business value changes with investment. The vendor estimates this based on the RFP and its experience. For simplicity, assume $BV_H(I_d) = BV_H \cdot I_d$ and $BV_L(I_d) = BV_L \cdot I_d$. This implies that the more the vendor invests identifying Telco's unique requirements, the higher the expected business value. This functional form also implies that as the investment increases, the uncertainty surrounding the system's business value also increases. This is reasonable since the vendor does not know how the investment will create business value—that depends on how Telco uses the system. If we view the investment as identifying functions and features specific to Telco, the higher the investment, the more functions and features identified, each with its own uncertain business value. As more functions and features are identified, the uncertainty is compounded.
- The functional form of the cost savings in the development stage resulting from the investment. The vendor should be able to estimate this based on the design and development work it has done for similar systems. Here, we use a linear approximation, $C_{save}(I_d) = k_d I_d$.
- The functional form of the cost of investment. Again, by keeping careful records of previous work, the vendor should be able to estimate the cost of investment. Because the increase in the system's expected business value and the decrease in development costs are directly proportional to investment, the cost of investment must be a convex function. This is necessary to ensure that, eventually, there are diminishing returns to increased investment. For simplicity, we use $C_{inv}(I_d) = k_d I_d^2$.
- Specific parameter values for the example are given in Table 3.

The optimal price for development, P_{dev} , depends on the investment level, which depends on the price of development. The vendor can jointly determine the development price and investment level in two ways, both of which are easily supported by computer.

The vendor can assume the development price will take on each of the three cases identified in equation (4). For each case, there is an optimal investment level (the equations are specified in the appendix). Given the optimal investment level, the vendor calculates the optimal development price, determines the associated expected value from the development stage, and chooses the investment level and development price resulting in the highest expected value from the development stage.

Alternatively, the vendor can find the optimal investment level numerically. For each investment level, there is an optimal development price and associated expected value from the development stage. By enumerating through the set of investment levels, the vendor can determine which one leads to the highest payoff.

Now that the vendor has estimated the optimal development price and investment level, it is ready to determine the design price. If the vendor has market power, for example if it developed Telco's system plan and Telco wants it to design the billing system, then the design price is 222.31. If there is competition among the vendors, the design price reduces to 215.71.

Table 3 Parameter Values for Example

Parameter	Value
BV_H^*	3,000
BV_L^*	500
C_{des}	300
C_{dev}	600
C_s	500
C_{inv}	500
C_{save}	100

Telco must now determine whether to accept a bid or cancel the billing system. To do this, it must first estimate the price the vendor will charge for development, and the investment it will make. These estimates are made in the same way the vendor made its estimates. Telco bases its estimates on the data in its system plan and its IS group's experience. Telco may create the estimates by relying on experience, or it may try to reason like a vendor to determine the investment level and development price.

We assume that the development price and investment level Telco estimates is the same that the vendors arrive at. Although it is unlikely the estimates will be exactly the same, they should be close if Telco provides enough information in its RFP.

Based on the design price and its estimates of the investment level and development price, Telco calculates the expected value of developing the system. When the vendor has market power, the expected value is zero, and when there is competition among the vendors the expected value is 6.6. In both cases, Telco will develop the system, but it will only reap the benefits when there is competition among the vendors.

Once the contract is signed, the vendor makes the investment determined in the process of calculating the price for the design stage. After the design stage, the vendor quotes a price for the development stage. The price quoted is the development price determined initially in the process of responding to the RFP (i.e., when determining the design price). The investment level and development price do not change in this model, because the vendor's information does not change. Based on the system's identified business value and the price the vendor quotes for development, Telco contracts for the development stage.

The process that both Telco and the vendors go through is essentially the same when the contract is a two-stage contract. As shown in Table 4, however, the results, are markedly different. We explore some of those differences below.

Analysis

The example and analytical results reveal that the form of the contract affects the business value. Although the equation specifying the system's expected value is the same for both contracting schemes (see equations [6] and [12]), the contracting scheme

Conclusion

This paper views software development as a two-stage process—design and programming—and models the development process to examine the relationship between a proposed system's business value and the structure of the contract used to direct the system's development. We show the type of contracting framework used may have a significant impact on the price paid, on the resulting business value, and on the completion probability of the project. In particular, we show that the linkages between the stages must be explicitly recognized. Recognition of these linkages profoundly affects the user organization's willingness to commit to the more beneficial two-stage contracting.

Although the models presented in this paper capture many key features of the outsourcing decisions, we make several limiting assumptions, and there are certain extensions that we hope future research will address. These include having heterogeneous developers, making the cost of development known only to the developer and only after the design phase, including an investment by the user that will increase the business value, and incorporating a more detailed mechanism by which the user can switch vendors after the design phase. Additionally, we believe that a further interesting extension of the two-stage contracting case would allow the design vendor to refuse to proceed with the software development phase. In this case an appropriate penalty structure needs to be designed. Finally, we believe that it would be interesting to couple heterogeneous developers with the ability of the design vendor to retain some rights to its work, so that another (more specialized) software development vendor would have to pay to use the results of the design phase. This may increase the vendor's incentive to invest in the design phase.

REFERENCES

1. Klepper, R. The management of partnering development in IS outsourcing. In *Proceedings of the Twenty-sixth Annual Hawaii International Conference on System Sciences*, vol. 4, 1992, pp. 518-527.
2. Richmond, W.B.; Seidmann, A.; and Whinston, A.B. Incomplete contracting issues in information system development outsourcing. *Decision Support Systems*, 8 (1992).
3. Sullivan-Trainor, M.L., and Maglitta, J. Competitive advantage fleeting. *Computerworld*, 24, 41 (1990): 1-4.
4. Whang, S. Contracting for software development. *Management Science*, 38, 3 (1992): 307-324.

Table 4 Results for Example

	Stage-by-Stage		Two-Stage	
	Market power	Competition	Market power	Competition
Value to user	0	6.6	0	110.4
Value to vendor	6.6	0	110.4	0
P_{des}	222.31	215.71	282.91	172.50
P_{dev}	1,100	1,100		
$P_{Total} - P_{penalty}$			1,690.8	1,690.8
I_d	0.65	0.65	0.95	0.95
Probability of completion	0.52	0.52	0.488	0.488

affects the level of investment the vendor makes, and this affects the expected business value. Additionally, the contract structure affects P_{dev} (the development price for stage-by-stage contracting and the difference between the total price and the cancellation penalty in two-stage contracting), which affects the probability of completing the system development stage. The example indicates that developing the system under a two-stage contract leads to a higher expected business value.

The price the user pays for the system depends on the form of the contract and the degree of competition among the vendors, with lower prices associated with more competition. In addition, the distribution of the price between design and development, and therefore the level of risk associated with the project, depends on both the level of competition and the form of the contract. Under both stage-by-stage and two-stage contracting, when the developer has market power, it takes less risk than under competition.

The degree of competition does not affect the probability of completing the system once the design work has started. The probability of completion depends on the price of development, P_{dev} , the investment, I_d , and the high and low business values, $BV_H(I_d)$ and $BV_L(I_d)$, none of which is affected by the degree of competition. As the example shows, the contract structure does affect the probability of completion by affecting the price of development and the investment level.

In stage-by-stage contracting, the design vendor has an advantage over other vendors in obtaining the software development work. It knows the user's organization, since the user does not have to incur search costs to find a new vendor for software development, and does not have to incur switching costs to familiarize the new vendor with the project. This enables the incumbent vendor to take advantage of its position and raise its price on the software development work above the market price. The vendor, however, does not gain from this increase, since the increase in price for software development is coupled with a corresponding decrease in price for the system design.