

## Learning in ERP Contracting: A Principal-Agent Analysis

D.J. Wu • Min Ding • Lorin M. Hitt

*DuPree College of Management, Georgia Institute of Technology, Atlanta, GA 30332*

*The Smeal College of Business, Penn State University, University Park, PA 16802*

*The Wharton School, University of Pennsylvania, Philadelphia, PA 19104*

*dj.wu@dupree.gatech.edu • mqd9@psu.edu • lhitt@wharton.upenn.edu*

### Abstract

*We examine IT contracting in one particular segment of the IT outsourcing market - the market for large scale packaged software implementations such as Enterprise Resource Planning (ERP) systems. Using a small sample of actual outsourcing contracts in several industries and a review of the relevant outsourcing literature, we determined the common provisions and structural characteristics of these contracts. We then used this description to develop an analytical model of IT outsourcing, using principal agent techniques. Our model captures key characteristics of these IT contracts including a multi-stage project structure, vendor learning, probabilistic binary outcomes (success/failure), use of incentive contracting, and implementation risks. In addition to deriving the optimal IT contract; we specifically focus on how vendor learning affects the optimal structure of the contract. In particular, we find that with rapid vendor learning at the early stage of the contract, it is often more efficient to use a multi-stage contracting procedure, even if it is not technically required, as this enables stronger incentives to be given to the vendor and creates greater profit and surplus.*

### 1. Motivation

Large scale packaged software systems are typically outsourced to consulting companies for implementation and maintenance. Such information technology (IT) tasks are notorious complicated, and have placed increasing burdens on the design of IT outsourcing relationships and especially the design and negotiation of outsourcing contracts which codify these relationships. Nearly all large firms have or are in the process of implementing large scale operational systems such as Enterprise Resource Planning (ERP) and it is not atypical for these projects to include several contracting opportunities over multiple years.

Contracts are the primary means of codifying the client-vendor relationship in IT outsourcing. Well designed contracts can help manage problems of ex-ante incomplete information about the project (e.g., requirements, client characteristics, vendor capabilities), and provide a framework for the measurement of performance, provision of incentives and the management of technical, business and managerial risks. As a result, firms have been making considerable investments in the development of IT contracts. Despite these investments, there is considerable evidence that many outsourcing agreements have performed less than satisfactorily. There are also numerous documented difficulties with some major ERP implementations, such as those undertaken by Allied Waste, Foxmeyer and Hershey Foods.

Enterprise systems implementation contracts tend to be similar across different projects due to the structure of the enterprise software industry making them amenable to detailed theoretical study. There are a relatively small number of dominant vendors and implementation consultants, and their business practices are similar. This enables a broad characterization of “typical” ERP contracting and implementation practices. These projects also have many of the typical features of many types of IT contracts but on a larger scale. These features include their large size (typically greater than \$15 million), the extensive use of outside consultants (more than half of all expenditure is for external consultants), the use of packaged software (which mitigate some of the systems development risks and place a greater emphasis on the non-technical aspects of vendor management) and the fact that the results are largely observable (the client can either “go live” with the system or not).<sup>1</sup>

<sup>1</sup> Most ERP systems are installed to replace and enhance an existing system. As a result, a major measure of success is whether the client can switch over to the new system and abandon the existing system. This event is easily observable and verifiable. There may be more detailed output metrics (e.g., completeness of system, accuracy of data conversion), but these are unlikely to be verifiable and may even be costly

The emphasis of our analysis is on two additional features that, while generally present in most IT contracts, are particularly important for ERP implementations. First, these projects often have multiple, discrete stages in which a similar set of activities is repeated at different work sites or for different components (i.e., modules) of the system. This enables a well-defined choice between structuring the project as a single stage “big bang” project or a multi-stage incremental “rollout”. Second, because of this structure and the need for close coordination between the vendor and the client to implement idiosyncratic business processes (or redesign them) this creates an important role for client-specific learning during the project by the vendor. This learning process and the choice of single vs. multi-stage project structure will influence both the overall outcome of the project as well as the optimal incentive structure. Our emphasis will be on characterizing optimal incentive contracts and total project value under different project structures (single stage/multi-stage) in the presence of various forms of vendor learning.

Our analysis proceeds as follows. First, using a small sample of actual outsourcing contracts, we identify the salient features of ERP contracts that we will consider in our analytical modeling. Second, we develop a principal-agent model that incorporates the characteristics of this market including the well-known problem of vendor moral hazard (and the need to provide incentives) as well as less commonly modeled characteristics of this market such as vendor learning and probabilistic binary outcomes (success/failure). Third, we provide a set of numerical examples to highlight key managerial insights derived from our analytical model. The paper concludes with a discussion of the relevance and applicability of our model for the design of outsourcing contracts as well as future research in IT contract design.

Note: All proofs are omitted due to space limitation; however, they are available from the authors or in the full version of this paper [5].

## 2. Literature review

This work follows Williamson in transaction cost economics, and others in general software development, which emphasizes contracting as

---

to observe. Perhaps as a result, we have not observed variable contract payments conditioned on these outcomes. In practice, contracted parties typically withhold payments until they agree the project is complete at each stage, again suggesting an observable, binary outcome.

the design of governance structures. It also relates to the broader literature on IT outsourcing, which has taken a much more qualitative approach to evaluating IT outsourcing as well as the more specialized literature on ERP. Vendor learning has been previously identified as one of the critical success factors in IT contracting, especially at early stages in a project. Vendor learning is also a key factor in reducing the root causes of high project failure rates such as poor project planning and management, changes in business goals during the project, and lack of business management support [1]. Finally, the use of multi-stage outsourcing contracts has been discussed in the literature on IT outsourcing. A review of this literature can be found in [3].

Our analyses of actual ERP contracts identified at least two major structural dimensions. First, while ERP software is always purchased externally, there is a decision on whether to perform the implementation in-house (“insourcing”) or contract with a service provider for these services (“outsourcing”). Second, these contracts can be grouped into single-stage “big bang” implementations or a series of sequentially interlinked projects where the client has a termination decision at the end of each stage. Beyond these major decisions, the contracts principally differ on the use of fixed-fee versus incentive contracts, the extent to which the vendor makes a relationship specific investment (training, facilities) as part of the contractual commitment. It is the single versus multi-stage decision and the structure of payments that will be the focus of our analysis. We will also compare the performance of optimal outsourcing contract to the relevant insourcing benchmark.

## 3. The base model

### 3.1. Preliminaries, notation and assumptions

Our model considers a risk neutral client considering a contract to engage a risk neutral vendor to exert effort in implementing an information system and providing associated services. The notation for our model is provided in Table 1.

The game has two stages ( $i = 1, 2$ ). In Stage 1 (a “Pilot Project”), a percentage ( $\alpha$ ) of the IT task (for example, a division within a firm or a module of the ERP package such as the human resources module) will be contracted from the principal to the agent. We assume there are only two possible outcomes at the end of Stage 1:

either the project is unsuccessful and the contract terminates or the project is successful. Success or failure is assumed observable and contractible, but the client always has the option to terminate the project after Stage 1.<sup>2</sup> Our model allows for the possibility that due to exogenous events a client will choose to not continue a successful project with probability  $\lambda$ , which is common knowledge to the client and vendor.<sup>3</sup> In Stage 2 (“Implementation”), the principal contracts the agent to finish the rest  $(1-\alpha)$  of the task; outcomes at Stage 2 are also assumed to be binary (success/failure). If the contract sets  $\alpha = 1$ , it is equivalent to a single stage contract.

The strategic decision variables for the principal are the parameters of a linear incentive contract for each stage  $(a_i, b_i)$  where  $a_i$  is a fixed fee, and  $b_i$  is paid contingently on project success, i.e.,  $a_i + b_i R_i \equiv w_i(R_i)$ , where  $R_i$  is the expected revenue attributed to the project at stage  $i$ . These parameters are chosen to maximize the principal’s overall profit at both stages, subject to the behavior of the agent.  $\Pi_i = R_i - w_i$  is the net profit attributed to the project at stage  $i$ . The agent’s decision variables are whether to reject or accept the principal’s contract and if so, how much effort  $(x_i)$  to exert each stage in order to maximize own profit.

The payoffs of the game are as follows. The agent receives contractual payments less own costs (which are assumed linear in  $x_i$ ). The principal’s payoffs depend on the outcome of each stage, which is influenced by both agent effort (and other project parameters) as well as a random component. The random component captures technical, business, managerial or project risks as well as other factors that can cause project failure. Specifically, we assume the agent’s production function or outcome-effort function as follows:

**Assumption 1 (outcome-effort):**

$$p_i = \bar{p}(1 - e^{-\beta_i \frac{x_i}{s_i}}).$$

<sup>2</sup> While it could be conceivable that the ability to commit the client to the project could have some value, a standard provision found in every contract we observed is a clause allowing client cancellation with 30-60 days notice. This type of practice rules out any sort of binding commitment to continue a project.

<sup>3</sup> Alternatively,  $\lambda$  may also be interpreted as the discount factor or a measure of the value of a long-term relationship between the client and the vendor.

Assumption 1 asserts that the probability of success at stage  $i$  is a function of the agent’s effort  $x_i$ , as well as other factors. Note the saturation property implied by the production function - even if an infinitely large amount of effort has been put into the project, there is still a probability of  $(1-\bar{p})$  that the project might fail.<sup>4</sup> This unavoidable risk caps the business value of any contract to be  $Q\bar{p}$ . We define  $\beta_i$  as the project team’s knowledge/expertise at a given stage in the project. By way of learning, we mean that when  $\beta_i$  changes (increases) over project size or length  $(\alpha)$ .

The principal receives  $\alpha Q$  revenue for a successful project in the first stage and  $(1-\alpha)Q$  revenue for a successful project in the second stage. The principal’s overall payoff is this expected revenue less transfers to the agent under their incentive contract.

**Table 1: Notation**

$w_i$	: payment to the agent in stage $i$
$R_i$	: IT investment revenue at stage $i$
$\Pi_i$	: IT investment profit at stage $i$
$p_i$	: production function (bounded by $\bar{p}$ )
$\beta_i$	: knowledge/expertise (bounded by $\bar{\beta}_i$ )
$n$	: speed of learning
$s_i$	: stage $i$ project size
$x_i$	: effort in stage $i$
$\alpha$	: proportion of IT task in stage 1
$t_1(\alpha)$	: learning by training at stage 1
$Q$	: total revenue of the entire IT project
$\lambda$	: continuation probability of stage 2
$\Delta(\beta)$	: total IT payoff if do-it-alone
$\pi_2$	: agent’s stage 2 profit
$\pi_2^0$	: agent’s stage 2 profit reservation
$\pi$	: agent’s overall profit from both stages
$\pi^0$	: agent’s overall profit reservation

<sup>4</sup> The outcome-effort function assumed also implies the property of decreasing returns to scale (DRTS) – supported empirically by previous literature on software development and maintenance (e.g., [2]). In addition, this type of production means that the problem is well defined even when the agent cost function is linear (thus obviating the need to assume any sort of scale diseconomies in the cost side such as a quadratic cost function).

$V^I(\alpha)$ : total business value if do-it-alone

$V(\alpha)$ : total business value if outsourced

In addition, we assume satisfaction of agent individual rationality (IR) and incentive compatibility (IC), standard in the principal-agent literature.

### 3.2. Solution and discussion

We solve for the optimal two-stage contract by backward induction (as a special case, the one-stage contract solution can be recovered from this analysis by setting  $\alpha = 1$ ). In Stage 2, the principal adjusts the incentive contract  $(a_2, b_2)$  to maximize own profit:

$$\max_{a_2, b_2} \Pi_2$$

s.t.

(i) Agent IR

$$a_2 + b_2[(1-\alpha)Q\bar{p}(1-e^{-\beta_2 \frac{x_2^*}{1-\alpha}})] - x_2^* \geq \pi_2^0$$

(ii) Agent IC

$x_2^*$  solves

$$\max_{x_2} [a_2 + b_2[(1-\alpha)Q\bar{p}(1-e^{-\beta_2 \frac{x_2}{1-\alpha}})] - x_2]$$

The expected profit is

$$\Pi_2 = (1-b_2)(1-\alpha)Q\bar{p}(1-e^{-\beta_2 \frac{x_2}{1-\alpha}}) - a_2.$$

The optimal solution  $(a_2^*, b_2^*, x_2^*)$  is given by:

$$a_2^* = -(1-\alpha) \left[ Q\bar{p} - \frac{1}{\beta_2} - \frac{\ln(Q\bar{p}\beta_2)}{\beta_2} \right] + \pi_2^0 \quad (1)$$

$$b_2^* = 1 \quad (2)$$

$$x_2^* = (1-\alpha) \frac{\ln(Q\bar{p}\beta_2)}{\beta_2}$$

In interpreting these results it is useful to define  $Q\bar{p} - \frac{1}{\beta} - \frac{\ln(Q\bar{p}\beta)}{\beta} \equiv \Delta(\beta)$ , which represents the IT payoff should the principal insource the project (assuming they have internal expertise  $\beta$  and there are no agency issues). Note that  $\Delta(\beta)$  has three terms: 1)  $Q\bar{p}$ , the maximum expected revenue given project risk,

2)  $-\frac{1}{\beta}$ , the value of vendor expertise, and 3)

$-\frac{\ln(Q\bar{p}\beta)}{\beta}$  which is the deduction from

maximum benefit that arises from the tradeoff of cost and benefits. The principal benefits from a more knowledgeable project team as  $\frac{\partial \Delta(\beta)}{\partial \beta} > 0$  (assuming  $\beta > \frac{1}{Q\bar{p}}$ ), indicating

the importance of learning (increasing of  $\beta$  over time). This will be our key focus later.

Unsurprisingly, in a risk neutral setting, the optimal solution involves the principal "selling" the project to the agent – the sign of the fixed component of the incentive is negative ( $a_2^* < 0$ ), while the variable component is set at  $b_2^* = 1$ , which transfers all the ex-post project benefit to the agent. The principal gains a profit of  $\Pi_2^* = -a_2^* = (1-\alpha)\Delta(\beta_2) - \pi_2^0$  and the agent obtains a net profit of  $\pi_2^* = \pi_2^0$ , exactly the agent's reservation profit for stage 2. This contract also achieves first best as  $\Pi_2^* + \pi_2^* = (1-\alpha)\Delta(\beta_2)$ , the maximum possible benefit of Stage 2 given vendor cost and capability.

Rolling back to Stage 1, the principal's problem at Stage 1 is similar to Stage 2:

$$\max_{a_1, b_1} V(\alpha) \equiv \Pi_1 + p_1 \lambda \Pi_2^*$$

$$= (1-b_1)\alpha Q\bar{p}(1-e^{-\beta_1 \frac{x_1}{\alpha}}) - a_1$$

$$+ \bar{p}(1-e^{-\beta_1 \frac{x_1}{\alpha}})\lambda \left[ (1-b_2^*)(1-\alpha)Q\bar{p}(1-e^{-\beta_2 \frac{x_2^*}{1-\alpha}}) - a_2^* \right]$$

s.t.

(i) Agent IR

$$\{a_1 + b_1[\alpha Q\bar{p}(1-e^{-\beta_1 \frac{x_1}{\alpha}})] - x_1^*\}$$

$$+ \bar{p}(1-e^{-\beta_1 \frac{x_1}{\alpha}})\lambda \pi_2^*(x_2^*) \geq \pi^0$$

(ii) Agent IC

$x_1^*$  solves

$$\max_{x_1} \left\{ [a_1 + b_1[\alpha Q \bar{p}(1 - e^{-\beta_1 \frac{x_1}{\alpha}})] - x_1] + \bar{p}(1 - e^{-\beta_1 \frac{x_1}{\alpha}}) \lambda \pi_2^*(x_2^*) \right\}$$

The expected profit

$$\Pi_1 = R_1 - w(R_1) = (1 - b_1) \alpha Q \bar{p} (1 - e^{-\beta_1 \frac{x_1}{\alpha}}) - a_1.$$

The optimal solution, denoted as  $(a_1^*, b_1^*, x_1^*)$ , can be shown to be:

$$a_1^* = -[\alpha \Delta(\beta_1) + \bar{p} \lambda (1 - \alpha) \Delta(\beta_2)] + \frac{\alpha}{\beta_1} \ln \frac{\alpha Q + \lambda (1 - \alpha) \Delta(\beta_2)}{\alpha Q} + \pi^0 \quad (3)$$

$$b_1^* = 1 + \frac{\lambda}{\alpha Q} [(1 - \alpha) \Delta(\beta_2) - \pi_2^0] \quad (4)$$

$$x_1^* = \frac{\alpha}{\beta_1} \ln \frac{[\bar{p} \beta_1 (\alpha Q + \lambda (1 - \alpha) \Delta(\beta_2))]}{\alpha}$$

The interpretations of the above optimal solutions at Stage 2 are the following. After rearranging equation (4) as

$$b_1^* \alpha Q p_1 = \alpha Q p_1 + p_1 \lambda [(1 - \alpha) \Delta(\beta_2) - \pi_2^0]$$

it is apparent that the solution is to have the agent's variable compensation be the benefit of the first stage, plus the expected benefits of the second stage (again, "selling the project to the agent"). Net of their costs and the negative fixed component, the agent obtains an overall profit of  $\pi^* = \pi^0$  for the entire project. Second, the principal obtains an overall expected payoff for the entire project of  $V^*(\alpha) = -a_1^*$  (note that  $a_1^* < 0$ ). The negative of equation (3), the principal's overall payoff  $V^*(\alpha)$  consists of three parts: the expected income stream (or benefit) from both stages (first two terms), the agent's cost of effort (third term), and payment to the agent (fourth term).

This solution converges to the single stage solution as  $\alpha \rightarrow 1$ , yielding

$$a_1^* = -\Delta(\beta_1) + \pi^0, \quad b_1^* = 1, \quad x_1^* = \frac{\ln(Q \bar{p} \beta_1)}{\beta_1}.$$

It is straightforward to show that the negative of the first three terms combined in equation (3)

$$\alpha \Delta(\beta_1) + \bar{p} \lambda (1 - \alpha) \Delta(\beta_2) - \frac{\alpha}{\beta_1} \ln \frac{\alpha Q + \lambda (1 - \alpha) \Delta(\beta_2)}{\alpha Q} \equiv V^1(\alpha)$$

is the overall IT project payoff should the principal choose to conduct the project in-house. Since  $V^1(\alpha) = V(\alpha) + \pi^0$ , the optimal contract eliminates vendor moral hazard and achieves first best. This is summarized as follows:

*Proposition 1: The linear contract defined by Equations (1), (2), (3), (4), implements first best.*

We now consider the rationale for dividing the project into two stages. The presence of the second stage allows the firm to discount the first-stage expertise requirement, provided they can "catch up" in the second stage.

*Proposition 2. Given the presence of a second stage, the first stage project team expertise discount, defined as*

$$\frac{1}{Q \bar{p}} - \frac{1}{\bar{p}} \frac{\alpha}{[\alpha Q + \lambda (1 - \alpha) \Delta(\beta_2)]}, \quad \text{increases}$$

*monotonically as  $\lambda$ ,  $\beta_2$  increases and decreases monotonically as  $Q$ ,  $\bar{p}$ ,  $\alpha$  increases.*

Finally, we state additional propositions, the first focusing on project value as a function of exogenous project parameters, and second focusing on the value of vendor capability and learning.

*Proposition 3a. The total value of the implementation project  $V(\alpha)$  increases monotonically as  $\lambda$ ,  $Q$ ,  $\bar{p}$  increases, regardless of whether the agent learns or the speed of agent learning.*

*Proposition 3b. The total value of the implementation project  $V(\alpha)$  increases monotonically as  $\beta_1$ ,  $\beta_2$  increases.*

These two propositions suggest that the comparative statics on project value of the project parameters are in the expected direction and independent of vendor learning. Moreover, Proposition 3b shows that the monotonicity of project value in vendor capability is not contingent on the other project parameters. This enables us to direct our attention to

characterizing different learning conditions without concern that the model will yield implausible results for some parameterizations.

#### 4. Learning and project structure

In our model, learning is when project team expertise changes between stages. We consider a setting where the amount of learning is related to the size of the pilot phase, and pilot phase expertise is fixed. We later extend this to include learning that occurs from project effort and explicit investment. Strictly speaking, this says that capability improvement only occurs after the pilot phase is completed, although variable values of  $\beta_1$  can be readily accommodated by a variable transformation. While we will consider a variety of potential learning functions, we restrict attention to learning processes that are increasing and concave. This form plausibly describes a “discovery” learning process in which firms learn additional detail as a project progresses until they know essentially “everything” and can learn little more. Nearly all learning functions employed in the related literature satisfy this condition (e.g., [4]), including two common forms we will use in our numerical examples: linear and power law [both accommodated by the flexible form  $\beta_2(\alpha) = \beta_1 + (\bar{\beta}_2 - \beta_1)\alpha^{\frac{1}{n}}$  when  $n \geq 1$ ]. This leads to our second key assumption.

**Assumption 2 (Concave Agent Learning):** Learning at both stages is concave in project size  $\alpha$ .

##### 4.1. Baseline learning model

Recall that we have defined  $\Delta(\beta)$  as the value for an insourced project (no agency issues) given the team’s expertise  $\beta$ . Denote

$$\Delta \equiv \Delta(\beta_2), \Delta' \equiv \frac{\partial \Delta}{\partial \alpha}, \Delta'' \equiv \frac{\partial^2 \Delta}{\partial \alpha^2}. \quad \text{Our}$$

central result on agent learning is summarized as Theorem 1 and associated corollaries:

*Theorem 1. Given any learning function satisfying  $-2\Delta' + (1 - \alpha)\Delta'' < 0$ , the total value of the implementation project  $V(\alpha)$  is concave in  $\alpha$ , if:*

(i)  $\beta_1$  is sufficiently large (i.e.,  $\beta_1 \rightarrow \infty$ ) or

(ii)  $Q$  is sufficiently large (i.e.,  $Q \rightarrow \infty$ ) and  $\bar{\lambda p}$  is low (i.e.,  $\bar{\lambda p} \rightarrow 0$ ) or

(iii)  $Q$  is sufficiently large (i.e.,  $Q \rightarrow \infty$ ) and  $\bar{\lambda p}$  is high (i.e.,  $\bar{\lambda p} \rightarrow 1$ ).

*Corollary 1:  $-2\Delta' + (1 - \alpha)\Delta'' < 0$  holds whenever  $Q\bar{p}\beta_2 > \sqrt{e}$  and Assumption 2 (Concave Agent Learning) holds.*

The intuitions behind the conditions of Theorem 1 are the following. *Condition (i)* suggests the principal uses two-stage contract to harvest the benefit of extremely knowledgeable project team where the risk terms dominate project cost issues. The tradeoff is to balance the first-stage payoff  $\Delta(\beta_1)$  and the expected second stage payoff  $\bar{p}\lambda\Delta(\beta_2)$  by setting a weight ( $\alpha$ ), so that the overall project value  $\alpha\Delta(\beta_1) + \bar{p}\lambda(1 - \alpha)\Delta(\beta_2)$  is maximized. *Condition (ii)* treats a “high risk – high value” project environment where failure is likely – a staged contract enables early termination. *Condition (iii)* identifies a “low risk – high value” project environment as  $\bar{p}\lambda\Delta(\beta_2) \rightarrow \Delta(\beta_2)$ , in which the principal can exploit agent learning. All these characterize conditions under which a staged contract is desirable, given  $\bar{p}\lambda\Delta(\beta_2) > \Delta(\beta_1)$ .

Before further analyzing the implications of this result, we consider the required assumption:  $-2\Delta' + (1 - \alpha)\Delta'' < 0$ . As noted in Corollary 1, this is nearly equivalent to concave agent learning, except in projects with very low overall value due to risk, limited benefit or (lack of) vendor capability. When the assumption  $-2\Delta' + (1 - \alpha)\Delta'' < 0$  is violated, payoffs become convex in project length ( $\alpha$ ) irrespective of the three conditions stated in Theorem 1. This yields a boundary solution either involving a situation when a single stage contract is optimal or, alternatively, a degenerate two-stage contract in which the initial phase is as small as possible. Conditions where these outcomes can arise are described in the following Corollary:

*Corollary 2: The total value of the implementation project  $V(\alpha)$  is convex in  $\alpha$ , if:*

- (i) *There is no learning over time, i.e.,  $\beta_2 = \beta_1$  or*
- (ii) *Learning is instant in that  $\beta_2 > \beta_1$ , i.e., independent of project length.*

The first condition describes a situation in which there is no incremental value of learning (or the speed of learning is zero), so the optimal contract sets  $\alpha = 1$ . The second condition describes a situation in which a first stage project of any size yields the learning benefit (or the speed of learning is infinite), but this is tempered by the possibility of early project termination. Thus, when the learning benefit offsets the project risk  $\Delta(\beta_1) < \overline{p\lambda\Delta(\beta_2)}$ ,  $\alpha \rightarrow 0$  is optimal, otherwise a single stage project is optimal.

Finally, we consider optimal project sizing and speed of learning. First, Theorem 1 establishes the existence and uniqueness of the optimal project sizing and any hill-climbing numerical optimization method can readily identify the solution. Second, a reasonable approximation of project sizing<sup>5</sup> can be obtained as

$$\alpha \rightarrow 1 - \frac{\overline{p\lambda\Delta(\beta_2)} - \Delta(\beta_1)}{p\lambda\Delta'(\beta_2)} \equiv \tilde{\alpha}. \text{ In this}$$

approximation, if the incremental payoff of learning is relatively small, then the role of the second stage will be limited, even to the extreme of a single stage. Third, speed of learning plays a significant role as a faster learner improves the total value of the project and reduces the scale of the initial phase. These are summarized below:

*Corollary 3: Given  $\overline{Qp\beta_2} > \sqrt{e}$  and assume*

$$\frac{\partial \beta_2(n)}{\partial n} > 0 \text{ where } n \text{ measures the speed of}$$

*learning, the faster the speed of learning:*

(i) *The larger the total value of the implementation project  $V(\alpha)$ ,  $\frac{\partial V(\alpha)}{\partial n} > 0$ ;*

(ii) *The smaller the size of the pilot project phase,  $\frac{\partial \tilde{\alpha}(n)}{\partial n} < 0$ .*

#### 4.2. Capability improvement through effort or training

<sup>5</sup> Obtained by solving FOCs for  $V(\alpha)$  ignoring the cost terms, since the conditions in Theorem 1 guarantee that the cost term is dominated by the benefit terms.

While we treat learning as occurring ex post through initial project sizing, it is equally plausible that learning can be a function of pilot phase effort (“learning by doing”) or through explicit effort devoted to learning in the first stage (“learning by training” - in addition to project effort considered before). We now consider whether the results of Theorem 1 continue to hold under this more general setting of learning.

In this formulation, consider a more general second stage capability function  $\beta_2(\alpha, x_1(\alpha), t_1(\alpha))$  where we introduce an additional investment in learning  $t_1(\alpha)$  in addition to first stage effort  $x_1(\alpha)$ . This formulation requires an extension of the concave agent learning assumption over these additional parameters:

**Assumption 3 (Generalized Concave Agent Learning):** Both  $x_1(\alpha)$  and  $t_1(\alpha)$  are concave in  $\alpha$ ; further more,  $\beta_2(\alpha, x_1(\alpha), t_1(\alpha))$  is concave in  $t_1$ .

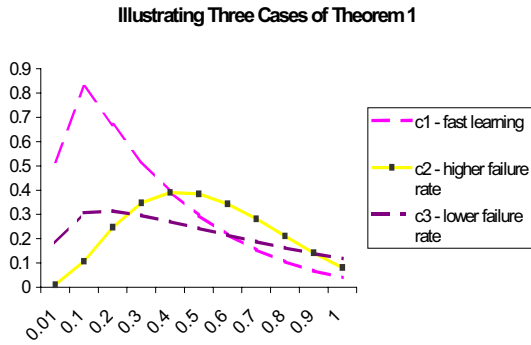
*Corollary 4: Given the two Concave Agent Learning Assumptions (Assumptions 2 & 3) and  $\overline{Qp\beta_2} > \sqrt{e}$ . Theorem 1 holds under the more general setting when second stage capability is a function of project and training effort:  $\beta_2(\alpha, x_1(\alpha), t_1(\alpha))$ .*

#### 5. Numerical examples and managerial implication

In this section, we illustrate managerial insights derived from our analytical model using numerical examples. These analyses are summarized by Figures 1 & 2 in which we plot project value as a function of project sizing. Note that to conserve space, the various conditions shown in each graph may have different scaling on the value axis.

Figure 1 shows project value as a function of project sizing under the three conditions of Theorem 1. Condition 1 (C1 on the graph), considers a highly capable vendor with fast learning, shows the benefit of a relatively small initial stage to capture vendor learning, even on an unfavorable project with a high failure and non-continuation probability. The next condition (C2 - higher failure rate) shows the benefits of subdivision for high risk - high reward projects. In contrast, the last case (C3) illustrates the value

of subdivision for low risk – high reward projects.



**Figure 1: Illustrating three cases of Theorem 1.**

**Parameterization:**

$$\beta_2 = \beta_1 + (\bar{\beta}_2 - \beta_1)\alpha^n.$$

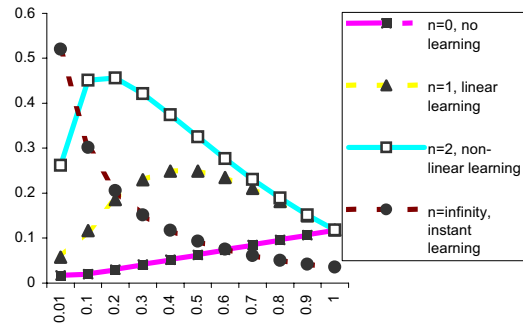
**C1:**  $(\beta_1 = 5, \bar{\beta}_2 = 30, n = 2, Q = 0.4,$   
 $\bar{p} = 0.51, \lambda = 0.25, \text{Y-scaling } \times 1000);$

**C2:**  $(\beta_1 = 1.5, \bar{\beta}_2 = 5, n = 1, Q = 1,$   
 $\bar{p} = 0.7, \lambda = 0.75, \text{Y-scaling } \times 100);$

**C3:**  $(\beta_1 = 1.5, \bar{\beta}_2 = 5, n = 2, Q = 1,$   
 $\bar{p} = 0.8, \lambda = 0.75, \text{Y-scaling } \times 10).$

Figure 2 illustrates how the speed of learning affects optimal project sizing. In the extreme cases, no project learning or infinitely fast learning, project value is convex and a boundary solution attains. Otherwise, the faster the rate of learning the smaller the pilot project size, illustrating Corollary 3.

**Impact of Learning to Optimal Sizing**



**Figure 2: Subdivision and speed of learning. Parameterization:**

$$Q = 1, \bar{p} = 0.8, \beta_2 = \beta_1 + (\bar{\beta}_2 - \beta_1)\alpha^n,$$

$\beta_1 = 1.5$ , for all cases;

**No learning** ( $n = 0, \bar{\beta}_2 = 5, \lambda = 0.5, \text{Y-scaling } \times 10$ ); **Linear learning**

( $n = 1, \bar{\beta}_2 = 5, \lambda = 1, \text{Y-scaling } \times 10$ ); **Non-**

**linear learning** ( $n = 2, \bar{\beta}_2 = 5, \lambda = 1, \text{Y-scaling } \times 10$ ); and **Instant learning**

( $n \rightarrow \infty, \bar{\beta}_2 = 10, \lambda = 1, \text{Y-scaling } \times 3$ ).

Table 2 below summarizes our results on optimal sizing as a result of learning and project characteristics, using

$$\beta_2 = \beta_1 + (\bar{\beta}_2 - \beta_1)\alpha^n \text{ as an example.}$$

**Table 2: Learning and optimal sizing**

Speed of learning	No Learning	Dynamic Learning	Instant Learning
	$n = 0$	$n \geq 1$	$n \rightarrow \infty$
Decision	Grouping	Breaking	Mix
When	Always	Top Team; Talent Team, High Value, High/Low Risk	Pending on Value of Learning

## 6. Conclusions and further research

In this paper we have considered the optimal contract structure for a multi-stage project in which vendor learning is a critical component. We argue

that this type of structure is particularly salient for the analysis of enterprise software projects which can be naturally broken into multiple stages and where client-specific vendor learning is likely to be very important. While the baseline principal agent model replicates standard results for risk neutral actors (for our specific parameterizations), our analysis yields new insights into the general literature on contract design. First and foremost, we have examined the role of agent learning in the design of contracts. In general, multi-stage contracts will tend to be favorable in the presence of vendor learning. The gains of subdivision and optimal project sizing are dependent on the rate of learning and can be large compared to the value of the project. In the absence of learning, single stage contracts will generally be preferable. Second, these results are robust to situations where learning is determined not only by pilot project sizing but also by investments in training or agent effort in the pilot phase.

While we have made specific assumptions about the cost and benefit functions and assumed agent risk neutrality, these assumptions have enabled us to focus on the role of learning as opposed to the already well known issues of risk sharing in the presence of risk averse agents or various types of scale effects, which are already well understood in the principal-agent context.

While our results provide some initial insights into one aspect of contracting for enterprise software projects, there is a tremendous opportunity for using these and related models to support empirical and experimental research on contractual performance. In particular, our analysis highlights the importance of learning and its relationship to optimal project sizing, as well as the more commonly understood issues of project risk and cost structure. While we have highlighted the role of incentives and vendor learning, for future work there may be other factors that may be relevant.<sup>6</sup>

## 7. References

[1] R. Banker, G. Davis, and S. Slaughter, "Software Development Practices, Software Complexity, and Software Maintenance", *Management Science*, 44 (4), 1998, pp. 433-450.

[2] E. Barry, T. Mukhopadhyay, and S. Slaughter, "Software Project Duration and Effort: An Empirical

Study", *Information, Technology and Management* 3 (1), 2002, pp. 113-136.

[3] E. Clemons, L. Hitt, and E. Snir, "A Risk Analysis Framework for IT Outsourcing", Wharton Working Paper, University of Pennsylvania, 2001.

[4] G. L. Lilien, P. Kotler, and K.S. Moorthy, *Marketing Models*, Prentice Hall, New Jersey, 1992.

[5] D.J. Wu, M. Ding, and L. M. Hitt, "Learning in ERP Contracting: A Principal-Agent Analysis", Dupree Working Paper, Georgia Institute of Technology, September 2003.

---

<sup>6</sup> We would like to thank Eric Clemons, Rob Kauffman, Kunsoo Han, Rui Dai and two anonymous referees for their constructive criticisms. This work was funded in part by NSF Grant IIS-9733877.