1988

DYNAMIC INVESTMENT STRATEGIES FOR INFORMATION SYSTEMS DEVELOPMENT

Anitesh Barua
Carnegie-Mellon University

DYNAMIC INVESTMENT STRATEGIES FOR INFORMATION SYSTEMS DEVELOPMENT

Anitesh Barua
Graduate School of Industrial Administration
Carnegie-Mellon University

ABSTRACT

This paper presents an analytical model for choosing optimal investment schedules for the development of new systems under various types of risk. Two modes of risk reduction are considered. In the first mode, risk is reduced by gathering information through prototype building or sequential development, where risky parameters are assumed to have unknown but fixed values. The second mode involves an increase in systems development and usage skills through experience and learning, which may reduce the development cost and increase the acceptance of the system among the potential users. The second mode of risk reduction changes the true values of the parameters.

Starting with a conceptual multi-dimensional framework for analyzing systems risk, a dynamic decision-theoretic model for guiding the investment process is developed. The model specifies the level of investment in development activities at any stage, depending on the information gathered from prototypes or parts of the actual system developed to that point. Some properties of global and myopic investment policies are derived. The sensitivity of the level of investment to the accuracy of information is characterized.

Experience and learning effects are considered in a simple two-period setting, where familiarity with the development process in the first period reduces the cost of developing the remaining part of the system in the second period. Extensions, testing, and implementation of the model are discussed.

1. INTRODUCTION

One of the most important problems confronting both practitioners and researchers in the domain of information systems today is systems development. Estimates show that IS and new information technology investments constitute almost 50 percent of capital investment by major firms in the United States of America (Kriebel 1986). Yet, developing a new system often involves many problems, including significant cost overruns, delayed completion and deviation from the desired functionality. Typically, at the beginning of a development project, there is a high degree of risk associated with the development cost and subsequent usage and profitability of the proposed system. From an economic standpoint, there is clearly a need for a model that can guide the investment/development process in the presence of these risky parameters and thereby avoid the problems mentioned above.

A review of the pertinent literature reveals two broad categories of research:[1] project cost/resource estimation and systems development modes. Project cost/resource estimation studies are extremely important because they provide ways for estimating resource requirements for systems development. They do not, however, specify any policy that the development manager can use in making investment decisions. Also, they do not consider the degree of the manager's confidence in the estimation process.

Most of the systems development studies advocate the use of prototyping instead of the classical approach. Unfortunately, these studies do not provide a theory for reducing development risk through prototyping. Thus, a brief overview of the literature reveals that there is no objective basis for making investment decisions for systems development. Also, issues related to the choice of systems development modes have been treated in a rather ad hoc manner in the literature, without much theoretical support.

This paper presents a dynamic decision-theoretic tool for guiding investment in new systems development in the presence of risky success parameters. Two modes of risk reduction are considered. The first is "passive" and involves information gathering through prototyping or sequential development. The second mode may be called "active" and involves improvement of development and usage skills through experience. The increase in skills reduces the development cost and increases the acceptance of the system among potential users. Realistically, at the beginning of a project, the values of relevant success parameters, such as development cost and level of system usage, are not known to the manager. As development activities progress, some information on these parameters becomes available. Simultaneously, there is a learning effect in terms of increased skills of the development personnel and the potential users.[2] This reduces the development cost and also the risk of rejection by users. In this paper, the "active" and "passive" modes will be discussed separately.

The model presented in this paper has the potential for providing a theory-based method for determining the investment schedule of a proposed development project. It should also augment the current understanding of some economic issues related to prototyping and classical development approaches. However, it should be mentioned that the model is not complete in its present form. The section on development skills needs to be enhanced. Also, the two risk reduction modes need to be integrated. Plans for enhancements are outlined in a separate section.

Section 2 presents a conceptual framework for assessing the risk associated with a proposed system. The choice of development modes under various types of risk are discussed in Section 3. The optimal policy for prototype building is derived in Section 4. Section 5 outlines the optimal policy for sequential development. Experience and learning effects are considered in Section 6. Sections 7 and 8 deal with model enhancements and implementation issues. Section 9 contains concluding remarks.

2. A MULTI-DIMENSIONAL MODEL OF SYSTEMS RISK

Research on systems development modes has not addressed the issue of systems risk in an explicit manner. However, it is evident that the investment pattern should be heavily dependent on the type and "amount" of systems risk. Therefore, first it is necessary to identify the relevant parameters with which systems development risk and success may be associated. Caution must be exercised in this choice, because the model may become unduly complex due to the identification of too many parameters. The level of the parameters must be sufficiently high to be meaningful in terms of investment decisions. Three key factors with which systems risk and success may be associated are technological profitability, operational feasibility, and development cost.[3]

Technological profitability, π, of a proposed system (with certain functional characteristics) is the increase in cash flow resulting from the use of the system. For a given context of use, this profit depends on the system functionality and assumes that the system is used by 100 percent of the target population.

Operational feasibility, u, is the level of system use by the target population. It depends on system functionality and the experience level of the users. u may be normalized with respect to the target population to indicate the actual user fraction. Thus, a normalized operational feasibility value of 1 implies that the system is used by 100 percent of the target population.

Total profitability, P, may be written as $P(\pi,u)$. For simplicity, let $P(\pi,u) = \pi u$. This assumption implies that the profitability increases linearly with system use. As seen later, this assumption of functional form is not crucial for the model.

Determining the profitability of an IS has remained a challenge to researchers for many years. Significant progress, however, in determining the economic impacts of Information Technology investment has been recently made by Kauffman and Kriebel (1988a, 1988b) and Banker and Kauffman (1988).[4] Their research indicates that it is currently possible to assess the profitability of investments in Information Systems. But, in the present context, the manager does not know the true values of π and u.[5] At best, he or she can have some crude estimates at the beginning of the project.

Development cost, C_d, is dependent on several factors, including system functionality, desired project duration, technical expertise and experience. While dealing with information gathering techniques, it is assumed that the true (unknown) development cost and operational feasibility are fixed for a given project. In section analyzing of learning/experience effects, this assumption is relaxed, and the impact of the initial investment level on the total development cost is investigated.

As indicated above, very rarely does the systems development manager know the true values of π, u and C_d at the beginning of a project. Actually, the manager may not know the true value of π at the end, either. Therefore, according to this model, risk may be present in one or more of these three factors. This distinction is important, because it suggests different development modes and investment strategies under different types of risk.

3. DEVELOPMENT MODES AND RISK REDUCTION

Prototyping and the classical development cycle are two broad development strategies that have been discussed in the literature. In this paper, a prototype system is considered to be a small-scale version of the real system. It has some of the characteristics of the final system and can be modified or enhanced to provide the desired features. It is assumed that prototyping is strictly an information gathering activity, and that prototypes are discarded once the testing is over. Development of the real system starts after the prototypes have been discarded. Admittedly, a prototype can become a final system, but such a case should more appropriately be termed as evolutionary or heuristic development.

The classical mode proceeds with the development of the actual system from a set of requirement specifications. However, the commitment of resources may take various forms. At one extreme, there is the single-shot or "rifle" approach (Elam 1980), involving a total commitment at the beginning of the project. At the other extreme, the manager may commit minimal resources and make further commitments as more information becomes available.

Intuitively speaking, prototype building appears to be an appropriate development mode when there is risk in technological profitability or operational feasibility. When a prototype is delivered to the potential users, some information (although imperfect) about the technological profitability and operational feasibility is obtained. On the other hand, an actual unfinished system, resulting from a sequential development strategy, does not furnish much information about the unknowns, π and u, since the users cannot put the system to use. But the prototype does give a good indication of the development cost and complexity associated with the proposed system. Due to its relatively small size and scale of operation, a prototype may not provide much information about the complexity and the cost of the real system. Thus, when there is risk in the development cost, a sequential development mode is probably more informative than building prototypes, except for small systems.

It is important to note that prototyping and sequential development provide information on π, u, and C_d only through several intermediate variables. For example, developing a fraction of the system in the sequential mode may provide information on resources, such as computer and personnel time, required for its completion. The values of these variables can then be combined to obtain an estimate of the associated development cost. Similarly, a prototype, after being tested by potential users, may reveal difficulties of use, lack of user skills, and inadequate performance measures, such as turnaround time and reliability. From this data, a manager may be able to obtain imperfect estimates of the operational feasibility of the final system.

4. PROTOTYPE BUILDING POLICY

In this section, a dynamic policy for choosing the prototype building schedule for a proposed system is developed. In general, the implementation of the policy results in a sequence of prototypes. Initially, the manager has to choose the prototype building technology and the "size" and functionality of the first prototype. Note that a given set of functional specifications crudely defines the size of the prototype. However, the converse is not true, since different functional features may be feasible for a fixed size. In this context, technology refers to prototyping tools and their supporting environments. Different technologies may result in different costs for building a prototype with a set of functional specifications and modifying/enhancing an existing prototype. For example, 4GL-based tools have been found to be particularly suitable for the rapid development and modification of prototypes. Nevertheless, the development and modification costs for a given technology also depend on the familiarity of the development personnel with the technology.

From the modifying/enhancing stage onward, the manager is confronted with the problem of choosing the prototype size and functionality. The options at any stage n depend upon the manager's choice of the n-1 prototype, because the latter can render some alternatives infeasible in stage n, making modification/enhancement costs prohibitive.

4.1 Model Assumptions

1 Since a prototype provides some information on π and/or u, it is conceptually equivalent to an information structure.[6] Thus, any prototype induces a partition of the state space of operational feasibility and/or technological profitability. This enables one to associate a "likelihood function" with a prototype. The manager's confidence in the information obtained from prototypes is encoded in this function.

2. For a sequence of prototypes $\{h_1, h_2, ...\}$, it is assumed that $h_1 \leq_i h_2 \leq_i ...$, where "$\leq_i$" stands for "no more informative". In this context, informativeness is determined by Blackwell's (1953) sufficiency criteria. This assumption is realistic because the information derived from a prototype is not lost when a subsequent prototype is built. h_i is a refinement of h_{i-1}, and is obtained through modifications/enhancements of h_{i-1}. Thus, for any i, the state space partition induced by h_{i+1} is finer than that induced by h_i.

In this paper, the optimal prototype-building policy considers risk in u. However, the analysis remains valid even when there is risk in π. Let U = [0,1] denote the set of normalized operational feasibility values. Some definitions related to partitions of U are necessary before another related assumption is stated. While these definitions refer to U as a continuous set, they apply to discrete sets as well.

Definition 1: A sub-interval of U is called "all-favorable" (or all-unfavorable) if the proposed system should be developed (or abandoned) for any u in that sub-interval. A "mixed" sub-interval is one that is neither all-favorable nor all-unfavorable.

There may be some threshold value of u above which it is optimal to build the system. This implies that the total profitability just equals the development cost for this threshold value of u. For illustration, let this value be .5. Then the sub-intervals [0,.5] and (.5,1] are all-unfavorable and all-favorable respectively. [.25,.75] is an example of a mixed sub-interval.

Definition 2: A partition $\{u_\alpha\}$ of U is called the "action-relevant" partition if each sub-interval of the partition is either all-favorable or all-unfavorable.

The name is derived from the fact that once this partition is achieved, only one action (i.e., either develop or abandon) is optimal for all values in any sub-interval of the partition.[7] In the above example, {[0,.5],(.5,1]} is the least refined action-relevant partition of U. It is assumed that in any sequence of prototypes, there is a prototype h_n that induces the action-relevant (or finer) partition. If n > 1, then the partition induced is finer than the action-relevant partition. This concept is used later to derive an important property of prototype sequences.

3. If the prototype sequences $\{h_1, h_{12}\}$ and h_2 are such that $h_{12} \equiv h_2$, then $c_1 + c_{1,2} = c_2$, where c_i is the cost of building h_i, i = 1, 2, and $c_{1,2}$ is the incremental cost of building h_{12}, when h_1 has already been developed. Note that h_{12} is obtained through modifications/enhancements of h_1.

4.2 Notation

Let $\{e_i\} = E_i$ be the information set of h_i. In the present context, e_i is an estimate of π or u, after the prototype h_i has been built. For risk in u, let $\phi(e_i|u)$ denote the probability of the estimate being equal to e_i, given that u is the true operational feasibility. Let f(u) denote the prior probability density function of the operational feasibility states {u}. The marginal probability of e_i is

$$p(e_i) = \int_{u \in U} \phi(e_i|u)f(u)$$

where the integral sign represents a general summation operator and is valid for discrete sets as well. The conditional probability density function of the operational feasibility u, given that an estimate e_i has been obtained, is given by

$$\phi(u|e_i) = \phi(e_i|u)f(u)/p(e_i).$$

Let $\alpha \epsilon \{a,d\}$ be a decision variable, where a ≡ "abandon the project" and d ≡ "develop the system". Let C_d be the development cost of the system when it is known with certainty. Define

$$v_i = \sum_{e_i \epsilon E_i} \max_{\alpha \epsilon \{a,d\}} [\theta p(e_i)] - v_0 \text{ where } \theta$$

$$= 0 \text{ if } \alpha = a \text{ (if the project is abandoned)}$$

$$= \int_{u \epsilon U} \pi u \phi(u|e_i) - C_d \text{ if } \alpha = d \text{ (if the system is developed).}$$

The variable v_i is the value associated with prototype h_i. It depends on the possible values of π, u and C_d and on the "informativeness" of h_i as encoded in the conditional probability density $\phi(u|e_i)$. The variable v_o is the expected payoff without building any prototypes and is given by

$$\max \{0, \int_{u \epsilon U} \pi u f(u) - C_d\}.$$

The value of θ is the expected payoff from choosing an action after an estimate of u is made. Define $\delta v_{i+1} = v_{i+1} - v_i$, which is the incremental value of building h_{i+1}, given that h_i has already been built. Note that this is the ex ante incremental value, since no estimate has yet been received from h_i. The ex post incremental value of building h_{i+1}, given that some estimate e_i has been received from h_i, is denoted by $\delta v_{i+1}(e_i)$.

Definition 3: A feasible prototype h_i is one for which $\delta v_i > 0$.

Proposition 1: Any sequence of feasible prototypes is finite.

Proof:

Let $\{h_1, h_2, ...\}$ be any sequence of prototypes. By assumption 2, there exists some prototype h_n in the sequence that induces the action-relevant (or finer) partition of the state space. Then, $\delta v_{n+1} = v_{n+1} - v_n = 0$, since the next prototype h_{n+1} does not provide any action-relevant information, although it induces a finer partition than h_n. The ex post incremental value $\delta v_{n+1}(e_n)$ can also be shown to be equal to zero.

4.2.1 Implications of Proposition 1

a) Determining the optimal policy is considerably simpler because of the finiteness property. Due to the dependence of the value of information gathering at any stage on the possible actions in subsequent stages, it might have been rather difficult to find the optimal policy for an unknown number of stages.[8]

b) This proposition provides a stopping rule for drawing the graph structure of the sequential prototyping scheme: for any path, stop whenever a prototype inducing the coarsest action-relevant (or finer) partition is encountered. The set of prototypes obtained in this manner is called the *initial feasible set*. This does not imply that all the prototypes in this set are actually developed. First, at any stage, some all-favorable or all-unfavorable estimate may be obtained, whereby the ex post incremental values of the remaining prototypes become zero. Secondly, the cost of prototyping, which is considered in the optimal policy in section 4.3, may eliminate some of the prototypes in the initial feasible set. Prototyping costs have purposefully not been considered in determining the initial feasible set. For obtaining this set, it may make intuitive sense to stop whenever the ex ante incremental value, δv_{i+1}, is less than the incremental cost, $c_{i,i+1}$. Unfortunately, this may not lead to the optimum number of prototypes that are actually developed. This is because that the ex post incremental value of a prototype, given that a certain estimate has been obtained, may be greater than its ex ante incremental value.

c) Since the coarsest action-relevant partition involves only two sub-intervals, a path probably does not contain more than four to six nodes (prototypes). Thus, the overall structure is not very large.

Using proposition 1, the optimal prototyping policy can be derived. There are three possible decisions at the end of each stage: abandon the project, stop prototyping and develop the system, and gather more information through further prototyping.

4.3 Optimal Policy

Assume that the true operational feasibility of a proposed system is not known with certainty. Let α_i be the action taken at the end of stage i. The initial action α_0 denotes the choice of the first prototype. This action is taken at the end of stage 0, which may be considered as the planning stage. For choosing the first prototype, evaluate the initial prototyping options $h \epsilon \{h_1\}$ as follows and build the one with the highest net value. For any h_1, the value is given by

$$V_1 = \sum_{e_1 \epsilon E_1} \max_{\alpha_1 \epsilon \{a_1, d_1, g_1\}} [\theta_1 p(e_1)] - v_0 - c_1$$

where $\theta_1 = 0$ if $\alpha_1 = a_1$ (if the project is abandoned at the end of stage 1, since the profitability of an abandoned project is zero).

$$= \int_{u \epsilon U} \pi u \phi(u | e_1) - C_d \text{ if } \alpha_1 = d_1$$

(if the system is developed at the end of stage 1.)

$$= \max_{h \epsilon \{h_2\}} [\sum_{e_2 \epsilon E_2} \max_{\alpha_2 \epsilon \{a_2, d_2, g_2\}} \{\theta_2 p(e_2 | e_1)\} - c_{1,2}]$$

if $\alpha_1 = g_1$ (i.e., if more information is gathered at the end of stage 1). The variable V_1 depends on the prototypes that can be built by starting with h_1. The variable of θ_1 is the expected payoff realized through the action taken at the end of stage 1, depending on the estimate e_1. The variable of θ_2 may be defined similarly for different values of α_2. At the end of stage 1, the manager can abandon the project ($\alpha_1 = a_1$), develop the same ($\alpha_1 = d_1$), or gather more information ($\alpha_1 = g_1$). For $\alpha_1 = g_1$, the manager must evaluate the alternatives $\{h_2\}$, assuming that h_1 has already been developed. Thus, a recursive relation is established for every path.

For a path terminating in the n_{th} prototype,

$$\theta_n = 0 \text{ if } \alpha_n = a_n$$

$$= \int_{u \epsilon U} \pi u \phi(u | e_n) - C_d \text{ if } \alpha_n = d_n$$

Since no more prototyping is economically feasible along this path after stage n, g_n is not an element of the choice set in stage n. At the beginning of stage 1, the prototype with the highest path value is chosen. For any other stage i, if estimate e_i is obtained at the end of i, then the decision α_i for stage i+1 is determined by calculating

$$\max_{\alpha_i \epsilon \{a_i, d_i, g_i\}} [\theta_i] - \sum_{s=1}^{i} c_{s-1,s}$$

At the end of stage i, the analysis is partially ex post with respect to e_i, because stages 1 through i are matters of the past. Note that the sunk cost (represented by the second term in the above expression) does not affect the decision α_i. However, it is included in order to calculate the net value at the end of any stage i.

The interpretation of the policy is as follows. Building a larger prototype (with more functional features) provides more information on technological profitability or operational feasibility. At one extreme, the prototype may incorporate all the functions of the real system and thereby provide accurate information on the parameters. However, generally this alternative is economically infeasible, especially when the initial risk is high. Under such conditions, there is a tradeoff between the level of investment in prototypes and the accuracy of information obtained. For a given degree of risk, as manifested in the prior distribution of the manager, the above policy determines the optimal investment level in each stage.

4.4 Delay Costs and Incremental Prototyping

To this point, no constraint has been placed on the time taken to complete the prototypes. If delay costs are absent (or not taken into account), then the development of prototypes becomes more gradual.

Proposition 2: Let $\{h_1,h_{12}\}$ and h_2 be two sequences such that $h_{12} = h_2$. Under risk, it is optimal to choose the first sequence, even if the opportunity cost of capital is zero.

Proof:

For opportunity cost of capital > 0, the result follows from the concept of time value of money. Let this be equal to zero. As before, let $c_1 + c_{1,2} = c_2$. Let

$$V_1 = \sum_{e_1 \epsilon E_1} \max_{\alpha_1 \epsilon \{a_1,d_1,g_1\}} [\theta_1 p(e_1)] - v_0 - c_1$$

and

$$V_2 = \sum_{e_2 \epsilon E_2} \max_{\alpha_1 \epsilon \{a_1,d_1\}} [\theta_2 p(e_2)] - v_0 - c_2$$

The variable V_i is the net value associated with the sequence i, i = 1,2. The value V_1 cannot be less than V_2 because the manager can always decide (beforehand) to build h_{12}, irrespective of the estimate received from h_1. This option is considered in the calculation of V_1 (through g_1). The two sequences are ex ante equivalent (i.e., $V_1 = V_2$), if for all $e_1 \epsilon E_1$, the maximum value of θ_1 is given by

$$\sum_{e_{12} \epsilon E_{12}} \max_{\alpha_2 \epsilon \{a_2,d_2\}} [\theta_{12} p(e_{12}|e_1)] - c_{1,2}$$

However, there is at least one all-favorable or all-unfavorable estimate in E_1. For such an estimate, the maximum value of θ_1 corresponds to $\alpha_1 = d_1$ or a_1. Thus, $V_1 > V_2$ and $\{h_1,h_{12}\}$ is preferred to h_2. This may be generalized to sequences of arbitrary length.

4.4.1 Implications of Proposition 2

For a given technology and a set of functional features, if the prototyping options at any stage differ only in terms of size, as in the two sequences in proposition 2, then the manager is better off by committing a smaller amount of resources at any given stage. This assumes that there is no delay cost or time constraint. Generally, delay (opportunity) costs are incurred by the users of the system, while the development manager may be from the IS department in case of centralized development. Therefore, the objectives of the two sides may not be fully compatible.

When delay costs are present (or taken into account), the optimal policy must be modified. For the initial action, calculate $V_1 - \Gamma(\tau)$ instead of just V_1, where $\Gamma(\tau)$ is the delay cost of τ, the expected duration of prototyping with h_1 as the first prototype. For a given graph structure, the expected duration is obtained by using the probabilities of those estimates for which any path with h_1 as the first prototype may be chosen. Since there is always a considerable backlog of applications development in an IS department, the investment at any stage should be larger than that determined by the optimal policy without delay costs.

4.5 Myopic Investment Policy

Definition 4: A myopic investment policy with respect to sequential prototyping is one that does not consider prototyping options in the subsequent stages.

At any stage, a manager using a myopic investment policy chooses an action as though the system is either developed or abandoned at the end of the stage. A manager may follow a myopic policy for several reasons. First, such a policy is simple and does not require evaluation of the entire path. Secondly, the options in stage n may not be fully known in stage n-1. This is often true if the proposed systems or technologies are new to the firm.

Proposition 3: Let the prototyping options differ only in terms of their sizes, as in proposition 2. For stage 1, let $\delta v_1 - c_{0,1}$ be > 0 and maximum for the prototype that is least costly (i.e., the smallest prototype). For any stage i > 1 and any mixed estimate e_i, let $\delta v_{i+1}(e_i) - c_{i,i+1}$ be > 0 and maximum for the smallest prototype in stage i+1. Under these conditions, a myopic, global policy results in the same sequence of prototypes.

Proof:

Since the prototypes differ only in terms of their size, the global policy invests in the smallest prototype at each stage, according to proposition 2. With a myopic strategy, the first prototype is the one corresponding to

$$\max_{h \epsilon \{h_1\}} \left[\sum_{e_1 \epsilon E_1} \max_{\alpha_1 \epsilon \{a_1,d_1\}} \{\theta_1 p(e_1)\} - c_{0,1} \right] - v_0$$

$$= \max_{h \epsilon \{h_1\}} [\delta v_1 - c_{0,1}]$$

By hypothesis, this corresponds to the smallest prototype in stage 1. If a mixed estimate is obtained at the end of stage 1, the myopic policy still chooses the smallest prototype, since the incremental value is maximum for this action. By similar argument, the smallest prototype is chosen according to the myopic policy, whenever a mixed estimate is obtained from the previous stage. Thus, the two policies result in identical sequences.

4.5.1 Implications of Proposition 3

Building larger prototypes may not always result in a proportionally larger "amount" of information. On the other hand, the cost of building prototypes increases proportionally with size. Under this condition, a larger prototype at any stage may result in an increase in value that is less than the corresponding increase in cost. This corresponds to the situation described in proposition 3, where the smallest prototype at each stage results in the maximum net value. Whether or not this is a general situation is an empirical issue. For the present context, the implication is that this situation ensures the optimality of a myopic policy.

Proposition 4: If $\delta v_1 < c_{0,1}$ for all prototypes in stage 1, then no prototype is built according to a myopic policy. However, a global policy may still build one or more prototypes under this condition.

Proof:

Since the myopic policy does not consider prototyping in subsequent stages, the initial prototype, according to this policy, corresponds to the maximum of $\delta v_1 - c_{0,1}$. If this expression is negative for all prototypes in stage 1, then no prototype is built, and an action is taken on the basis of the null-system (with expected value v_0).

The global policy, however, considers options in later stages for making current decisions. Thus, if the ex post incremental values of prototypes in later stages are significantly greater than the corresponding incremental costs, then the overall value associated with a given path may be positive. It may then be optimal to build prototypes with the global policy.

4.5.2 Implications of Proposition 4

With new technologies, large initial investments may be necessary in order to cover acquisition and learning/training costs. Also, due to lack of experience, the initial prototypes may not be very informative for a given cost. Therefore, the initial prototyping activity may turn out to be rather costly. At later stages, prototyping may not remain as costly as before due to an increase in skills and the fact that the acquisition cost is incurred only at the beginning. The myopic policy does not look beyond the barrier created by the initial setup cost and lack of familiarity with the technology. In proposition 4, if $v_0 = 0$, then the project is abandoned by the myopic policy. However, through prototyping the global policy may find the project to be a profitable one. In this case, there is a possibility of abandoning a profitable project with a myopic policy. Of course, this situation should not arise with familiar technologies.

4.6 Sensitivity of Investment to Accuracy of Estimates

The accuracy of the estimates depends on the size and the functionality of the prototypes. The investment policy outlined in section 4.3 is sensitive to this accuracy. Accuracy differences may be attributed to various reasons. A development team with higher skills may be able to develop prototypes with more functional features for a given cost. Then with new technologies, prototypes in later stages may be more accurate (informative) than the initial ones. Whatever maybe the cause of an increase in accuracy, the net effect of such an increase is a finer partition of U. A related proposition is stated next.

Proposition 5: The number of prototypes in the initial feasible set may decrease (and cannot increase) with an increase in the accuracy of the estimates.

Proof:

Let $\{h_1, h_2, .., h_n\}$ be a sequence of prototypes such that h_n corresponds to the action-relevant (or finer) partition. With an increase in accuracy, h_{n-1}, which previously provided a partition coarser than that of h_n, may now induce the action-relevant (or finer) partition. Under this condition, h_n does not remain feasible and is deleted from the initial feasible set. The proof of the second part of the proposition is obvious.

While the initial feasible set reduces or stays the same with an increase in accuracy, the number of prototypes actually developed may increase. Consider a situation where a mixed estimate e_1 is received at the end of stage 1. If the cost of building a prototype in stage 2 is greater than the corresponding value, then no prototype is developed, and an action is taken on the basis of e_1. Now consider a learning/experience effect, which results in more accurate prototypes for the same cost from stage 2 onwards. In the presence of such an effect, the new incremental value associated with a prototype in stage 2 may exceed its cost, and therefore it may be optimal to gather more information through this prototype. This phenomenon of gathering more information may be observed with new technologies.

5. SEQUENTIAL DEVELOPMENT

When a high degree of risk is associated with the development cost of a proposed system, it is important to obtain information on the magnitude of the cost before making a commitment to develop the system. For complex systems, prototypes may not provide much information on development cost due to their relatively small scale of operation. Under such situations, building a part of the actual system may be more informative. This mode of developing a fraction of the actual system and making further commitments on the basis of the cost information obtained is called *sequential development*. The decision variable related to this problem is the fraction of the project that should be pursued at any given stage.

5.1 Optimal Sequential Development Policy

Assume that the proposed system can be divided into n modules numbered 1,2,...,n. Each module performs a set of functions. The modules must be built in sequence. If one or more module(s) is (are) built, then some information about the development cost of the other modules is obtained. Building more modules at a time gives more accurate information about the remaining modules. Let there be a time constraint of T periods. Also, assume that the profitability of an incomplete system is zero.

With these assumptions, there are initially n+1 options, including that of not undertaking the project. Let $x_1 \epsilon \{0,1,...,n\}$ be the decision variable at the beginning of the first period. If $x_1 = i$, then the first i modules are built in period 1, $0 \leq i \leq n$. Let $\{c_{di}\} = C_{di}$ be the set of costs for module i over which the manager may define a prior distribution $f(c_{di})$. Let $\mu(c_{di}|e)$ denote the conditional probability density of the true cost, given that an estimate e has been received. If $x_1 = 0$, then the net value realized is also zero. The rest of the options in period 1 are evaluated as follows:

$$\max_{x_1 \epsilon \{1,2,..,n\}} \Phi(x_1), \text{ where } \Phi(x_1)$$

$$= \pi u - \sum_{i=1}^{n} \int_{c_{di} \epsilon C_{di}} c_{di} f(c_{di}) \text{ if } x_1 = n$$

(if the entire system is to be developed in one shot), with expected cost of module i given by

$$\int_{c_{di} \epsilon C_{di}} c_{di} f(c_{di}).$$

Otherwise, $\Phi(x_1)$

$$= \sum_{e_1 \epsilon E_1} \max_{x_2 \epsilon \{0,..,n-x_1\}} [\Omega_1 p(e_1)] - \sum_{i=1}^{x_1} \int_{c_{di} \epsilon C_{di}} c_{di} f(c_{di})$$

$$\Omega_1 = 0 \text{ if } x_2 = 0$$

(if the project is abandoned at the end of period 1),

$$= \pi u - \sum_{i=x_1+1}^{n} \int_{c_{di} \epsilon C_{di}} c_{di} \mu(c_{di}|e_1) \text{ if } x_2 = n-x_1$$

$$= \sum_{e_2 \epsilon E_2} \max_{x_3 \epsilon \{0,1,...,n-x_2-x_1\}} [\Omega_2 p(e_2|e_1)] -$$

$$\sum_{i=x_1+1}^{x_1+x_2} \int_{c_{di} \epsilon C_{di}} c_{di} \mu(c_{di}|e_1) \text{ (in all other cases).}$$

The value of e_1 is a vector estimate (at the end of period 1) of the costs of the modules comprising the system. This estimate is perfect for modules $1,2,...,x_1$, since these modules have been completed by the end of period 1. The decision variables x_1, x_2 and x_3 must satisfy the conditions $n-x_2-x_1 \geq 0$ and $n-x_3-x_2-x_1 \geq 0$. $\{e_1\} = E_1$ and $\{e_2\} = E_2$ are determined by choices of x_1 and x_2 respectively. For a time constraint of T periods, Ω_{T-1} is equal to

$$0 - \sum_{i=k}^{l} \int_{c_{di} \epsilon C_{di}} c_{di} \mu(c_{di}|e_{T-1}) \text{ if } 0 < x_T < n-x_{T-1}-...-x_1$$

If the system is not completed in the last period. The limits of the sum, k and l, are given by

$$k = \sum_{j=1}^{T-1} x_j + 1, \text{ and } l = \sum_{j=1}^{T} x_j$$

The first term of Ω_{T-1}, with $0 < x_T < n-x_{T-1}-...-x_1$, is equal to zero, since the profitability of an incomplete system has been assumed to be zero. Thus, the alternative $0 < x_T < n-x_{T-1}-...-x_1$ is always dominated by $x_T = 0$ and $x_T = n-x_{T-1}-...-x_1$. If estimate e_s is obtained at the end of s, then x_{s+1} is chosen as follows:

$$\max_{x_{s+1} \epsilon \{0,...,n-x_s-...-x_1\}} [\Omega_s] - \sum_{i=1}^{x_s} \int_{c_{di} \epsilon C_{di}} c_{di} \mu(c_{di}|e_s)$$

As in the prototyping policy, note that the second term in the above expression represents the sunk cost after s periods and does not affect the decision x_{s+1} for period s+1.

6. EXPERIENCE AND LEARNING EFFECTS

So far, risk reduction through sequential information gathering has been considered. It was assumed that π, u and C_d had true values that did not change with an increase in development and usage skills. For new technologies and/or application systems, however, there is always a considerable amount of learning on the part of development personnel, leading to an increase in development skills and a subsequent reduction in development cost. Similarly, with the prototyping approach, users gain experience and skills through use of the prototypes. This increases the acceptance of the final system. In this paper, only the increase in development skills is considered.

At the beginning of a project, if the manager decides to build the entire system in a single pass, then he or she will not be able to take advantage of the learning effect. If the manager decides to build a fraction σ (consisting of a certain number of modules) initially, then the remaining $1-\sigma$ fraction can be built with increased skills at a lower cost. It is similar to writing small or medium-sized programs with a new language to gain familiarity with the subtleties before undertaking a highly complex project. A simple two-period setting is considered next.

Let computer time and development personnel time be the only resources for development. Then the development cost is $C_d = (K+P)t$, where K and P are computer and personnel cost per unit time respectively, and t is the time taken to complete the project. This cost function assumes that the personnel time is equal to the computer time. The subsequent analysis is valid for other costs functions as well. Let the time taken to develop fraction y be given by $t_y = y[L_1e^{-bE} + L_2]$, where E is the experience, measured by the time spent by development personnel with the technology; b is the learning rate; L_1+L_2 is the time taken to complete the project without any prior experience; and L_2 is the estimated minimum completion time with "considerable" experience. Let $L = L_1+L_2$. Some estimates of L_1, L_2 and b may be obtained from data on previous projects undertaken by the IS development personnel.

Let σ be the fraction to be developed in the first period. Then the total cost is given by $C_d = (K+P)[\sigma L + (1-\sigma)L_1e^{-\sigma bL} + (1-\sigma)L_2]$. From the first order condition, the optimal σ satisfies the equation

$$1 - (1-\sigma)bLe^{-\sigma bL} - e^{-\sigma bL} = 0$$

Note that since b or L $\neq 0$, $\sigma \neq 0$, .5, or 1. The above equation can be solved numerically to find the optimal σ.

Proposition 6: If the learning rate increases, ceteris paribus, then the investment in the first period decreases if the original $\sigma > .5$. If $\ln(1-\sigma) - \ln(\sigma) > \sigma bL$, then the initial investment increases.

Proof:

Using the implicit function theorem, $\delta\sigma/\delta b < 0$ if $1-\sigma < \sigma e^{\sigma bL}$. Taking a natural log transformation, the results are obtained.

Proposition 7: $\delta\sigma/\delta L < 0$ if $\sigma > .5$, and > 0 if $1-\sigma > \sigma e^{\sigma bL}$.

Proof:

Follows from the implicit function theorem.

6.1 Implications of Propositions 6 and 7

Propositions 6 and 7 indicate that the change in the optimal value of σ with changes in b and L depends on whether the original period-1 investment is greater than .5. If two development teams 1 and 2, with learning rates b_1 and b_2 respectively ($b_1 > b_2$), develop the same system independently, and if it is optimal for team 1 to develop a fraction $\sigma > .5$ in the first period, then team 2 should develop a fraction $< \sigma$ in period 1. A similar interpretation of proposition 7 may be given with respect to two teams differing in development skills and hence in L, the time taken to complete a given project without prior experience.

7. MODEL ENHANCEMENTS AND EXTENSIONS

As mentioned in the introduction, the model presented in this paper is not complete in its present form. The incompleteness is perhaps not very glaring, considering that it is an attempt at formal modelling in an area that has primarily been dominated by rules of thumb. Enhancements and extensions of the basic model are discussed below.

1. In this paper, gathering information and increasing skills through learning and experience have been treated as being disjoint. These two issues need to be integrated into a single coherent model.

2. More often than not, projects are chosen from a portfolio of interrelated projects. Under such conditions, a project can no longer be considered in isolation from the other items in the portfolio. The fact that the risk associated with a project affects (and is affected by) other projects has to be incorporated in the model.

3. The model developed in this paper deals with a single IS with a given set of functional characteristics. Quite often, system functionality may take a range of values, and the most profitable combination may not be known at the outset. An important extension of the model is to include this feature in the optimal development policy.

4. Typically, managers use heuristics for managing the development process. The usefulness of the proposed model may be demonstrated by comparing different managerial heuristics with the optimal policy for various projects (with different risk factors, as manifested in the prior distributions). A simulation set-up appears to be the most promising and feasible approach to this testing. Development projects,

characterized by various ranges of profitability, development cost, operational feasibility and prior distributions on these factors will be used as cases. For each case, commonly used strategies and the optimal policy will be simulated. The actual differences will give an indication of the usefulness of the model. The optimal policy may also serve as a benchmark for comparing and identifying good heuristics.

8. IMPLEMENTATION ISSUES

The implementation of the model is not as difficult as it may seem at the outset. Consider the data requirements for successful implementation:

- Prior probability distributions over C_d, u and π.
- Enumeration of prototyping alternatives.
- Identification of functional modules of a proposed system.
- Partitions of C_d and u induced by the modules and prototypes.

The last requirement deserves further clarification. Consider development cost as an example. If cost can vary between \$100,000 and \$300,000, the manager may believe that by building a fraction of the system (e.g., the first three modules), he or she will get an indication of whether the cost lies in the \$100,000 to 200,000 range or in the 200,001 to 300,000 range. Thus, the partition induced by the first three modules has two states, C_d^1 = 100,000 - 200,000 and C_d^2 = 200,001 - 300,000. Once the manager specifies his or her prior probabilities over these two states, the option of building the first three modules in the first period can be evaluated.

One commonly cited criticism of decision theory is that it requires the decision maker to specify his or her beliefs over a large number of states. In contrast, the development manager is not interested in exact values of the parameters. In fact, it is probably impossible to know them with certainty. The manager is only concerned with favorable and unfavorable ranges of values and is interested in knowing the range in which the relevant parameter lies. Thus, the decision-theoretic formulation of the problem is not very complex.

9. CONCLUSIONS

In this paper, the problem of choosing dynamic investment strategies for systems development has been addressed. A conceptual multi-dimensional model of systems risk has been proposed. Using this risk assessment model, the optimal investment policies for prototyping and sequential development under different types of risk have been determined. Some features of global and myopic policies, such as delayed investment and abandoning of profitable projects, have been characterized. The optimal investment policy in the presence of learning effects has also been dealt with, albeit in a simple two-period setting.

The model makes some assumptions, one of which is that prototypes and sequential modules are equivalent to information structures. This is the only crucial assumption of the paper. It is not very restrictive in principle but is definitely subject to the familiar problem of eliciting the manager's beliefs and prior distributions.

Systems development is undoubtedly one of the key issues in the domain of information systems. Surprisingly enough, very little attention has been given to the formal characterization of investment strategies for systems development. The model presented in this paper is an attempt to structure the problem in an analytical framework. It is hoped that further studies along these lines will help throw more light on issues relevant to this area.

10. ACKNOWLEDGEMENTS

The author wishes to thank Professor Charles Kriebel for providing the motivation for this research. Special thanks are due to the author's colleagues, Vasu Krishnamurthy and Suman Basuroy, for discussions that generated important insights into the problem. The comments and suggestions made by Professor Andrew Boynton and the anonymous referees are gratefully acknowledged.

11. REFERENCES

Banker, R. D., and Kauffman, R. J. "Strategic Contributions of Information Technology: An Empirical Study of ATM Networks." Working Paper, CRIS #177, GBA #88-25, New York University, New York, NY 10006, March 1988.

Blackwell, D. "Equivalent Comparison of Experiments." *Annals of Mathematical Statistics*, 1953, pp. 265-272.

Elam, P. G. "Choosing Between System Development Alternatives." *Journal of Systems Management*, Vol. 31, No. 9, September 1980, pp. 36-40.

Kauffman, R. J., and Kriebel, C. H. "Identifying Business Value Linkages for Information Technology: An Exploratory Application to Treasury Workstations." Working Paper, CRIS #182, GBA #88-47, New York University, New York, NY 10006, April 1988a.

Kauffman, R. J., and Kriebel, C. H. "Modeling and Measuring the Business Value of Information Technology."

Forthcoming in Measuring the Business Value of Information Technologies, ICIT Research Study Team #2, ICIT Press, 1988b.

Kriebel, C. H. "Understanding the Strategic Investment in Information Technology." Working paper, #48-85-86, Graduate School of Industrial Administration, Carnegie-Mellon University, Pittsburgh, PA 15213, July 1986.

Marschak, J., and Radner, R. "Economic Theory of Teams." New Haven, CT: Yale University Press, 1972.

McKinsey & Company Research Report "Unlocking the Computer's Profit Potential: A Research Report to Management." New York: McKinsey & Company, Inc., 1968.

Miller, A. C. "The Value of Sequential Information." *Management Science*, Vol. 22, No. 1, September 1975, pp. 1-11.

12. ENDNOTES

1. Software productivity is a third category, involving ex post productivity comparisons of completed projects. In contrast, the analysis in this paper is ex ante and dynamic in nature.

2. The usage skills of the potential users increases primarily with the prototyping approach.

3. These are similar to the feasibility factors discussed in the McKinsey & Company report (1968). However, the operational definitions are more precise in the current context.

4. For example, Kauffman and Kriebel (1988a) have quantified the impact of a bank's "treasury workstation" system on its ability to increase demand balances from customers. Banker and Kauffman (1988) have estimated the contribution of an automated teller machine to increasing a bank branch's local deposit market share.

5. In fact, if the manager knows the true values, then there is no need for the proposed model.

6. The concept of information structures is discussed in detail by Marschak and Radner (1972).

7. This name was first suggested by Vasu Krishnamurthy of the Graduate School of Industrial Administration, Carnegie-Mellon University.

8. Miller (1975) provides a general formulation of sequential information gathering problems, where the value of information depends on the decisions that can be taken in later stages.