The project unit costing method: Constructing a financial justification for the knowledge-based system

by MICHAEL L. MORGAN and GAIL D. WOLF
Magnavox Electronic Systems Company
Fort Wayne, Indiana

ABSTRACT

Although knowledge-based and expert systems are becoming increasingly popular, they are usually constructed incrementally and by ad hoc methods, rather than by conventional software engineering methodologies. The development of such systems often more closely resembles R&D than systems engineering, and the result is often not completely specified until the program is implemented. This makes it difficult to estimate costs and difficult to justify those costs in terms of an uncertain benefit. This paper presents one method of cost justification, which allows a direct comparison and tradeoff of costs and benefits.
INTRODUCTION

The recent media attention on artificial intelligence (AI) often leaves the impression that AI is now a mature discipline, capable of being used in a wide variety of applications and organizations. In fact, AI is still a very new field. It is difficult to find commercially viable examples of programs which equal the complexity of even a modestly-sized conventional application. In part, this is due to the fact that well-trained AI programmers and knowledge engineers are scarce, and that many of the tools are expensive. Another factor is that the exact performance of the program, and consequently its benefits, are difficult to define or quantify until the program has been written. The combination of initial high costs and uncertain benefits has made it difficult to justify AI systems using traditional cost-accounting methods, yet the corporate climate usually dictates that these methods are the only ones which are acceptable.

As part of its ongoing modernization program, Magnavox Electronic Systems Company has prepared detailed cost and technical proposals for 84 projects. Over half of these require advanced information technology methods, including knowledge-based systems, expert systems, and natural language interfaces to databases. Since Magnavox is a defense contractor, and the modernization program is performed under Government supervision, the cost justifications for projects such as these must conform to conventional cost accounting standards, while showing tangible, auditable results. To achieve these goals, Magnavox developed the Project Unit Costing Method. This method offers the following advantages when compared to the ad hoc methods currently in use:

1. Direct comparison between projects, to decide where to place the investment dollar.
2. Costs and benefits are explicit and tangible, allowing tradeoffs to be computed to maximize profit.
3. The method is standardized and readily transferable, so projects may be compared which were prepared by different user groups or subcontractors.

Although this method was developed to evaluate candidate projects for in-house use, it has additional applications in the marketing of AI systems and in steering R&D investments.

The following sections address three areas of AI cost justification:

1. The nature of AI costs, including initial costs, design and development costs, and delivery costs
2. The nature of AI benefits, and
3. The Project Unit Costing Method, which integrates costs and benefits

AI COSTS

Projects based on advanced information technology such as AI differ from their conventional counterparts in five principal ways.

1. AI projects have a higher initial investment. AI projects are frequently prototyped and developed by highly-trained knowledge engineers, working in concert with some of the company’s scarcest experts. Usually, these knowledge engineers must be hired in at great cost, either as employees or as subcontractors. For less complex applications, in-house people may be trained, but training incurs the cost of time and labor. Furthermore, AI work is often done on dedicated workstations, which have a much higher cost per station than the conventional programming terminal. While simpler applications may be built in the conventional environment, there is much to be said for the high-productivity tools available on dedicated workstations.

   These higher initial costs make it more important to do a thorough benefits analysis before beginning an AI project. They may also contribute to management’s hesitation to fund large-scale AI development work.

2. AI design and development tends to become a goal in itself. Conventional projects are normally developed by existing software development groups, such as a Data Processing department. Their cost estimates include only the labor required to build, test, and install the given application. AI projects, on the other hand, are often costed as part of establishing an AI department. If the first few projects must bear the burden of the high initial costs mentioned above, they are not economically justifiable. Therefore, management makes an AI commitment. The company decides to support AI even if immediate benefits are not forthcoming. Unfortunately, this turns into a self-fulfilling prophecy. AI specialists are hired with no immediate expectations of demonstrating profitability and, often, with no immediate projects. These individuals may be from an academic or research environment, and have had little training in identifying or building high-payback projects. They therefore spend many months becoming familiar with their tools and developing small prototypes, but develop little or nothing that can be used in the organization to offset their costs. After some time, management becomes disenchanted with AI, and starves the AI organization of resources until it withers and dies.

3. AI costs are frequently underestimated. This is not unlike the problems which surrounded software development in general during the early years of business computing. For example, each new expert system requires a designer to choose a knowledge representation and control method. It is usually impossible to determine if the choice is the correct one until the knowledge engineer and the human expert have
jointly developed a prototype. Frequently the poor choice does not become apparent until the system is almost complete, since some problems are functions of the size of the knowledge base. If the initial choice of control strategy or knowledge representation must be abandoned, much of the work must be redone. In addition, many of the commercial development tools, known as expert system shells, offer only one means of knowledge representation (usually production rules) and one to two control strategies (usually forward and backward chaining, with blind, depth-first search). Frequently, the designer makes an initial choice of representation and control, and buys a tool, only to be forced by the nature of the application to another methodology and another tool. Experience is of some help here, but even veteran knowledge engineers speak of the "pancake principle;" build the first one to be thrown away.

The most common defense against these redundant costs is rapid prototyping, the development of an initial version of the finished system within a few weeks. This offers the advantages of keeping the expert's interest, and of allowing the knowledge engineer to get a feel for the domain and the problem very quickly. It also allows management to see progress quickly. Another defense against wasting money on tools, is to choose the most general tool available. Unfortunately, these tools are usually the most expensive, and often require more sophistication and more work on the part of the knowledge engineer than a more tightly focused, but more powerful, tool. Furthermore, many of the most general tools do not lend themselves well for transitioning into the delivery environment.

4. AI programs are never completed. Unlike conventional programs, which are usually tightly specified, many AI application have such broad specifications that no one can agree whether they are done. There are always cases on which the human expert can outperform the machine. As those cases are discovered, there is a temptation to add "just one more" rule to improve the system. This often upsets some other, previously satisfactory, part of the system, so a bit more fine-tuning is required. This process continues until money or patience runs out.

5. Finally, the problems of transitioning an AI system into the user's environment are not well understood. Many AI systems have been built to run on specialized hardware and in a special support environment. The transition onto machines which are more accessible to the user has not been thought through, so users are forced to come to the developers to use their application. While this may have some serendipitous value in keeping developers attuned to users needs, it is not a suitable solution to the delivery problem in most domains. As an alternative, several of the manufacturers of AI workstations and tools are now offering "delivery machines" and run-time environments. These are suitable for a large class of problems. For still other problems, however, the system must end up on the machines that users have on their desktops. This may include personal computers, or access to departmental minicomputers or company mainframes. The impact of recording the application into a conventional programming language, or the cost of a run-time copy of an expert system shell, have often been left out of the original estimate.

AI BENEFITS

AI benefits are frequently thought to be intangible. Frequently, the first expert systems are built by a company to "keep up with the technology." Due to the fatal loop described above, the initial experience with AI often does nothing to dispel the notion that AI's benefits are not readily quantifiable.

AI-based systems are frequently used to augment functions whose costs are presently captured as part of indirect and overhead expenses. For many manufacturers, direct labor represents less than 10% of total cost incurred, yet detailed accounting of the labor dollar is kept only for direct labor. This means that costs for the white-collar and non-touch functions which are reduced by AI are not visible to the accounting system. Where the costs are not visible, the reduction of those costs is not seen to be a benefit.

THE PROJECT UNIT COSTING METHOD

Based on the above observations, any method of estimating the costs and benefits of AI-based systems must have the following characteristics:

1. The in-house AI group must be run on a business basis. That is, they must generate discernable products with demonstrable benefit. While the company may serve as the investment banker, they have a right to reap a reasonable return on investment. In the case of an independent business, these returns flow from tangible benefits called sales. In any in-house business, an equally tangible measure of benefits must be found.

2. The system must be specified in terms of cost reduction. Any new (To-Be) system will grow out of existing (As-is) procedures and costs. By using the new system, those costs may be reduced. If they are not, then the quality, reliability, or timeliness of the output is increased. These non-cost benefits contribute profitability by increased market share, or by allowing the company to penetrate new markets. They can usually be translated into cost reduction terms by comparing the To-be costs with the costs of achieving a similar increase in profitability by As-is methods. This approach allows designers and management to make tradeoffs between cash flow, capital investments, and the ultimately reduced costs. This also elevates the financial side of development from a "trust-me" basis to one in which benefits are clearly visible on the bottom line.

Methods for evaluating investments based on cost reduction are well-known in the production environment, where the cost reduction is applied directly against the cost of goods sold on a per-unit basis. In that environment these methods are known as Unit Cost methods. Magnavox's adaptation is to apply these methods to the indirect and white-collar functions which do not directly contribute to product costs. The Magnavox method is described below.
Cost Analysis

The cost analysis must capture all aspects of current (As-is) costs. These may include: (1) labor, (2) burden, (3) subcontracts, (4) material, (5) capital, (6) travel, and (7) expenses (e.g. software, furniture, office supplies).

Benefit Analysis

Savings and avoidances

For cost/benefit analysis purposes, savings and avoidances are discriminated. Savings are a reduction in cost as a result of the implementation of the project, given that the business volume remains constant. Avoidances are attributable to an increase in business volume. Cost reduction from savings and avoidances come from two sources. First, there are those cost elements, such as material costs, which are directly reduced. The dollar savings of these cost elements may be captured directly. Second, there is a reduction in current or anticipated headcount attributed to the project. These are ultimately translated into dollars using the labor grades and categories, and the average cost of labor in those grades and categories for each year of the analysis.

Project unit descriptions

To define the scope and benefits of the projects, it is necessary to define a unit of the project output. Units can be defined for both touch and non-touch (i.e. process-oriented and non-process-oriented) projects. A unit must be related to the function of the project. Examples of project units include:

1. A printed wiring board
2. A work order
3. A purchase order
4. A bid

Examples of unacceptable unit definitions include:

1. Any time-related unit, such as a man-month (units must be functionally oriented)
2. A decision

As-is versus To-be cost per unit

The As-is cost per unit is defined as the dollar amount required to generate one project unit using current methods. The To-be cost per unit is similarly defined as the cost to generate one project unit, if the project were implemented and on-line today. Several options for estimating the To-be cost per unit are available: a percent reduction from the As-is can be taken, or the actual dollar amounts using the proposed techniques can be estimated.
Design-phase cost summary sheet and implementation-phase cost summary sheet

Costs for both design and implementation are estimated using conventional budgeting procedures, bearing in mind the cautions about costing described earlier.

Expense account analyses form

This standard Magnavox form is used to estimate expenses, by account code, for each quarter of a given year. Each user group, when performing their annual budgeting function, includes costs related to each project under their responsibility. These budgets are prepared to capture project costs for a five-year period beginning with the detailed design phase. The project should stand alone with respect to funding. If the project is approved, the monies associated with that project will be included in the budget. If the project is not approved, all of the monies associated with that project will not be approved in the budget.

Budget manning chart, travel and transportation analyses

These standard Magnavox forms provide backup to explain the labor and travel expenses budgeted in the Expense Account Analyses Form.

Projected volume/units of output by product

By product and by year, for the five years beginning with detailed design, the products are listed which will be affected by the project. Where “nameless” products are used, such as in the out years, this must be documented in the Cost Philosophies and Assumptions section. The Cost Philosophies and Assumptions section should also contain the rationale which links product volume with project unit volume. For example, if the number of purchase orders is a function of product complexity, and if each product today averages 200 purchase orders, and product complexity is increasing, it may be appropriate to multiply volume by 250 to estimate the number of purchase orders required per product in the out years.

As-is cost per unit calculation

This sheet captures the current cost of producing a unit of project output. Where costs are volume dependent, the calculations should be based on the most likely volumes. The backup documentation shows calculations based on other volumes. Again, the reasoning that led to the choice of a “most likely” volume should be documented in the Cost Philosophies and Assumptions section.

To-be cost per unit calculation

The To-be, or anticipated, cost, of producing a single unit of project output after project implementation are captured using the same philosophies as were used to generate As-is costs. Volumes must be the same in both the As-is and the To-be calculations.

Capital project request form

This standard Magnavox form captures depreciation and investment opportunities of interest to the accounting community.

Headcount savings

Savings are defined as the reductions which result from the project, given that the current volume of business remains unchanged. This is the place to capture the number of positions which would be eliminated if the project were implemented, and if the business volume did not increase. In order to convert this headcount to dollars, savings here are identified by labor grade and category.

Headcount avoidances

Avoidances are defined as those costs which would be incurred, due to growth in business volume, if the project was not implemented. Like savings, this list, of positions which will not need to be created, is broken out by labor grade and category.

Dollar savings

Using current and projected wage rates, the headcount savings are converted to dollar savings. Additional savings, such as material and equipment, are also captured here.

Dollar avoidances

This form is identical to the one above, except that headcount avoidances and anticipated cost reductions are captured.

Capital retired or salvaged and back-up documentation

Any capital equipment which will be retired or salvaged because of the implementation of this project is listed here. If the schedule for replacement is dependent upon the projected volumes of business, the volume assumptions must be documented in the Cost Philosophies and Assumptions section. Backup documentation shows the physical location of the equipment, associated product lines which use the equipment, and other possible applications of the equipment.

Output

The output of this analysis is a set of project costs and product cost reductions, broken out by product, by year, and by volume. In addition, the initial costs are distinguished...
from delivery costs, and any capital equipment retired or salvaged is shown. This set of figures is suitable for use as input to conventional cost-accounting methods such as payback, rate-of-return, or discounted cash flow models. Following that analysis, projects may be chosen which offer the highest profitability for the lowest risk. While the problems of specifying and costing AI-based systems will not disappear in the short term, this method allows the user and management to see tangible net benefits, even if budgetary estimates are exceeded.

Implementation

The Project Unit Costing Method has been implemented as a series of Microsoft® Multiplan® spreadsheets on the Apple® Macintosh.™ This allowed all computations, including links between forms, to be performed automatically. A worksheet was submitted to user groups which walked them through the process of choosing a project unit and estimating costs and savings. The results of this worksheet were entered in the Multiplan model, allowing all projections and savings to be computed automatically. Furthermore, the project model was linked to a discounted cash-flow model, giving a single figure-of-merit for each project. Once the data was collected and entered, the complete cost/benefit analysis of all 84 projects was completed in less than a week.

SUMMARY

AI-based systems are difficult to cost-justify, given that they are implemented incrementally, and their benefits are often not specified before implementation. By identifying tangible cost reduction targets as benefits, the Project Unit Costing Method makes it possible to select projects with the most attractive cost/benefit ratios. This encourages system designers to consider, before design begins, those aspects of the project which offer the greatest savings. This can be used to form a general specification which keeps design focused on those aspects, and which brings the design “tuning” cycle to halt when the most important cost-reducing features have been implemented. This also allows the design team and management to decide, if cost overruns occur, whether those overruns are justifiable with respect to the benefits expected.

ACKNOWLEDGEMENTS

This work was performed as part of Magnavox Electronic Systems Company’s Industrial Modernization Incentives Program (IMIP), part of its internally funded Modernization Program. Magnavox’s IMIP is performed under a business agreement with the U.S. Department of Defense, and is supervised by Air Force Systems Command/Electronic Systems Division.