

SOFTWARE FOR HEALTH, SAFETY, AND ENVIRONMENTAL MANAGEMENT— TO BUILD OR TO BUY?

*Scott Harter and Linda Pitts, Radian International Software
Rochester, NY, and Boston, MA*

ABSTRACT

The U.S. Department of Energy, other federal and federally funded organizations, and private-sector industry invest billions of dollars annually in information technology. Most organizations are in some stage of software acquisition, replacement, or modification extending to health, safety, and environmental (HS&E) applications. Options to custom-develop (“build”) or to use commercial packages (“buy”) are frequently debated. It is important that HS&E managers involved in software acquisitions have an understanding of:

- 1) The sources of the overall costs associated with implementing a software application;
- 2) Relative comparisons of the merits of build vs. buy decisions; and
- 3) Guidance on how to optimize the delivery of either choice.

Common criteria for evaluating software include the initial cost of software acquisition, functional features, and the computing environment (more narrowly referred to as the “platform”). Less obvious costs include populating databases; upgrading for compatibility and functionality; training; and maintenance. Over the life of an HS&E management application, the sum of these costs will exceed initial purchase or development costs, in most cases.

The major advantage of custom development is its ability to meet the desired choice of functional features and computing environment, assuming that the total costs and schedule are reliably estimated, understood, and acceptable to the stakeholders. Compatibility with existing work processes and computing environment can be achieved by building an application. Historically, however, delayed delivery and expanded budget have been issues with custom development.

By comparison, the costs and implementation schedule are usually better known and controlled when purchasing off-the-shelf software. The usually significant lag-time for software development is eliminated, and upgrades and maintenance are likely to be planned. Hence, a package purchased from a sustainable commercial provider typically includes ongoing plans for growth with technology whereas point-in-time custom development will face technology obsolescence.

Risks from custom-developed solutions can be minimized through contract specifications. Reliable providers of long-term support, maintenance, and upgrades must be established, and software documentation is essential.

Some customization in purchased packages can be obtained through acquisitions that offer flexibility in user configuration (for example, more user-defined fields). Purchasing a package with an adequate frequency of new releases and a means of incorporating user feedback into future releases offers more assurance that an off-the-shelf package can evolve with your needs.

INTRODUCTION

Total spending on information technology in the United States in 1999 is estimated at \$720 billion (1). Information management cost planning estimates for the United States Department of Energy's information enterprise totaled approximately \$1.6 billion for the 1997 fiscal year, with 75% of these information dollars supporting environmental waste management, energy research, and defense programs (2). Information technology extends to health, safety, and environmental management.

Software applications for health, safety, and environmental management functions are prolific today. A published listing of available software packages includes over 340 suppliers and almost 2000 applications (3). Health, safety, and environmental management software ranges from single-user, personal computer applications to enterprise-wide applications used by Fortune 50 corporations across international boundaries. The scope of functionality ranges from tracking wastewater discharges at a single site to tracking occupational medical exams for a worldwide corporation. With the proliferation of health, safety, and environmental management software, most organizations are in some stage of software acquisition for new applications, upgrades, or replacements. Frequently a decision point is reached—to build or to buy software? Common notions are:

“We have a super programmer; we can easily develop what we want,” or

“We can easily buy a package off the shelf, install it, and use it.”

Both statements are oversimplifications. There are advantages and disadvantages to either approach, developing your software (“build”) or purchasing a package (“buy”). There are significant costs beyond the initial software purchase or development costs as well. Reviewing these factors will allow a more informed decision to be made when adding software by either approach or even a blend of the build and buy options. A Department of Energy “New Series Directive” outlines a process for investigating software alternatives that includes purchasing commercial software and modifying commercial software when considering the need to build software (4).

TOTAL COST OF SOFTWARE

Initial Selection Criteria

Among the most common selection criteria (Figure 1) for acquiring software are:

- 1) The functional features of the software;
- 2) The computing environment of the software; and
- 3) The initial purchase price of the software or estimate for development.

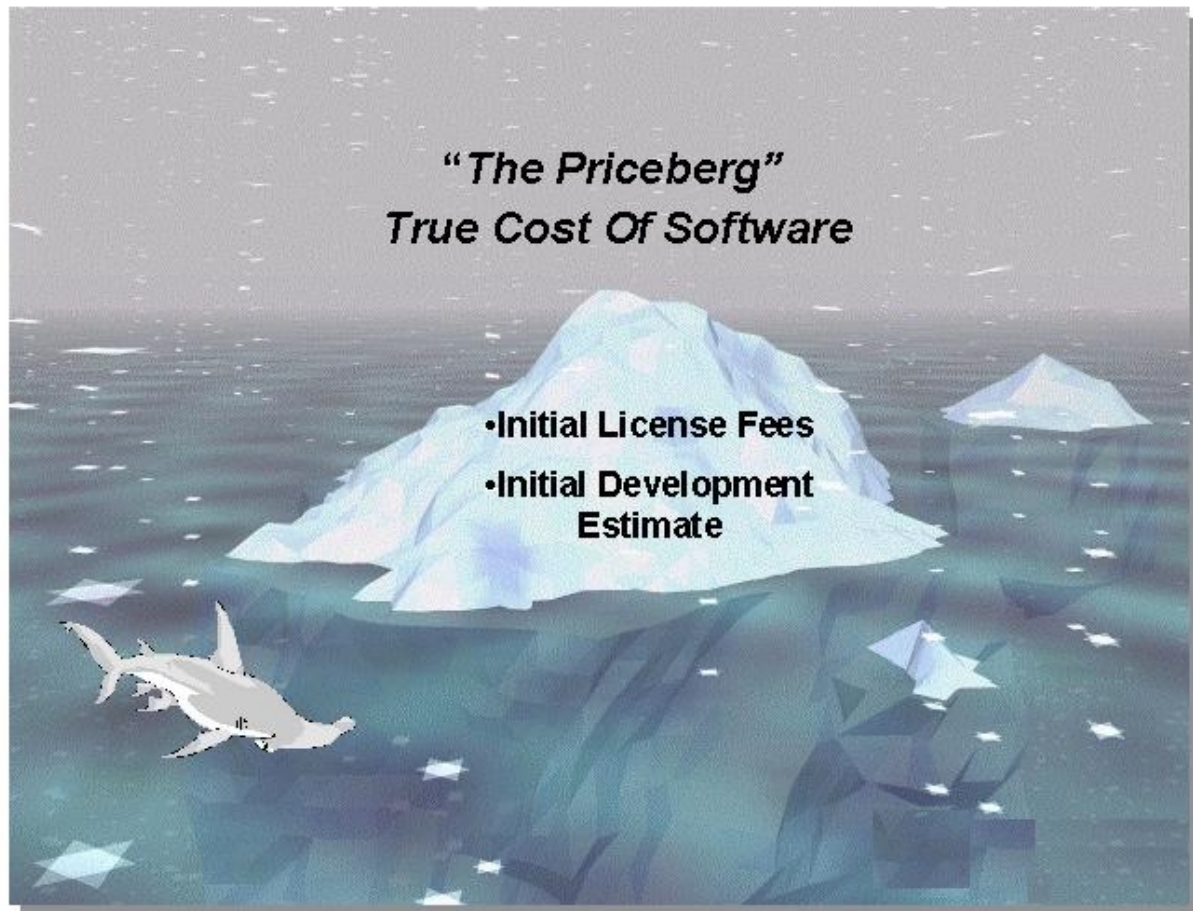


Figure 1—The above-water view of the “Priceberg” shows a common impression of the cost of software.

Different groups usually champion these criteria. The health, safety, and environmental specialists are concerned about getting waste manifests and toxic release inventory reports generated, tracking incidents and developing OSHA 200 logs, or grouping their asbestos-exposed workers for training and monitoring. The information management systems group wants the software to reside in a computing environment that is compatible with their current or future direction; they also want to know how much server capacity is needed and the effects on network traffic and performance. The procurement and capital projects group is concerned about costs and other commitments, such as delivery date. These are just a few examples, but they illustrate typical thinking and focus during the early phases of software selection and acquisition.

Total Costs, Including Hidden Costs

Health, safety, and environmental staff participating in software acquisitions and securing related budgets should be well aware of the total costs of acquiring, implementing, and maintaining software. The total costs associated with the life cycle of any enterprise application can safely be assumed to be at least equal to and usually greater than the cost of actually acquiring the software. As a rule of thumb, the cost of *implementing* the software is one to five times the cost of software acquisition. See Figure 2 for a summary of the total costs of acquiring and putting software into use. Implementation includes the processes from installation to start-up use. The major processes are:

- Installation,
- Configuration or set-up,
- Interfacing to other systems,
- Data migration from legacy systems,
- Training, and
- Enhancements or customization, e.g. special reports (if permitted by vendor, applicable to purchased packages only).

Beyond implementation are additional costs of support and maintenance, also indicated in Figure 2. Support and maintenance includes such costs as technical support, interface operation, ongoing manual data input, upgrades to maintain platform compatibility, training new users, and possibly enhancements.



Figure 2—The “Priceberg” shows that there are many factors contributing to the total cost of software beyond the initial purchase price or initial estimate to build it.

BUILD OPTION—PROS AND CONS

The most compelling reasons to build software are to get an application 1) with exactly the features you need and want, and 2) developed in the computing environment preferred by your organization. This means that there will be a match with your work processes and compatibility with the computing environment desired. Additional advantages of the build option include having control of the future of the application by the owner of a custom-developed application. Also, if the optimum contract terms are achieved, the numbers of users can be expanded without license fees in some situations.

The major drawbacks of building your own software or having it built are 1) the time required for development, and 2) the burden of the costs of development, support, and maintenance falling upon the “owner” instead of being shared by a group of licensees. The availability of support after your custom application is built should be a concern along with uncertainty in ability to adhere to the completion schedule and budget. Technology obsolescence is a significant consideration that becomes the responsibility of the application owner to avoid when using the build option.

A summary listing of major advantages and disadvantages of the build software option is in Table I.

Table I—“To Build Software” Advantages and Disadvantages

ADVANTAGES	DISADVANTAGES
Get the features you want	Use delayed by development cycle
Can match your current work process	Development, support center, and maintenance costs borne solely by owner
Build in computing environment desired	High turnover rate of programmers
Control the future of the software	Historical ballooning of cost and schedule
Possibly add users without license fees	Owner deals with platform obsolescence, technology changes, compatibility issues

BUY OPTION—PROS AND CONS

A major incentive for purchasing a software package is to avoid the time delay and uncertainties in delivery associated with developing your own software. Development consists of analysis, design, construct, test, and document. These development steps have already been completed for a commercial off-the-shelf package. If dealing with a properly qualified supplier, the stability of the ongoing support and maintenance will tend to be greater. A commercial software package will usually have dedicated support staff. By comparison, custom developers, by the nature of their profession, are more likely to move to another assignment after completing your application. The user base (licensees) shares the costs of commercial software packages, both initial and ongoing, instead of having the burden of cost borne by a single owner. A sustainable

commercial supplier is also likely to have a vision and a plan for growth with technology changes such as new operating systems or new versions of back-end database software.

Major disadvantages of purchased software packages are that there will usually not be a complete match in the package features or functionality and the user needs or requirements. This mismatch is likely to require at least some changes in work processes. Also, the purchased package may not operate in the desired computing environment of the acquiring organization, resulting in some system incompatibility. Table II lists some of the major advantages and disadvantages of the buy alternative.

Table II—"To Buy Software" Advantages and Disadvantages

ADVANTAGES	DISADVANTAGES
Avoid development delays from build time	Usually some gaps in desired features resulting in some work process changes
Fixed prices for initial purchase	Usually some unwanted features
Costs shared by licensees	Fewer computing environment selections
Dedicated support and maintenance	Limited control of future of product

MITIGATING RISK OF BUILD OR BUY DECISION

Health, safety, and environmental staff who are participating in acquiring software through contracted development or purchase of a commercial package can take a number of steps to reduce the risks with either approach. The Office of Management and Budget's "Information Technology Investment Guide" (5) provides a process for managing risks. The risks associated with the disadvantages identified above and means of dealing with the risks are presented in Tables III and IV below.

Table III—Mitigating Risk From "Build" Software Option

RISK	MITIGATION APPROACH
Unanticipated costs	Evaluate/understand total costs (Priceberg)
Cost and schedule ballooning	• Seek proven development project manager • Contract on fixed-price basis • Performance incentives in contract • Negotiate unlimited users without added fees
Turnover of developers (move on to next build project)	• Evaluate stability of ongoing support • Obtain software documentation as contract deliverable
Technology obsolescence by end of, or shortly after, the build period	Anticipate and budget for upgrades to match technology changes as part of support and maintenance plan

Table IV—Mitigating Risk From “Buy” Software Option

RISK	MITIGATION APPROACH
Unanticipated costs	Evaluate/understand total costs (Priceberg)
Usually gaps in features	• Seek high degree of user configuration • Seek supplier with frequent upgrades • Seek supplier with user input to upgrades • Prepare to change some work practices to match software
Usually features not needed or wanted	Seek modular design and pricing

SO WHICH IS IT—TO BUILD OR TO BUY?

There isn't a universally right or wrong answer. But health, safety and environmental managers are encouraged to acquire software on the basis of informed decisions that are driven by the circumstances and needs for their specific situations. It is also possible and sometimes favorable to combine the build and buy options. The Department of Energy Order 200.1, “Information Management Program” (4) provides methodology for the selection process. Some of the circumstances or factors that favor one option over the other are summarized in Table V and help support the decision-making process.

Table V—Circumstances that Favor “Build” or “Buy” Software Options

FAVORABLE FOR BUILD OPTION	FAVORABLE FOR BUY OPTION
Generous project schedule	Constrained project schedule
Strong development and support resources	Out-source culture for computing services
Some tolerance for upward budget changes	Fixed and final cost critical
Narrow function, few users, small scope (lower absolute risk)	Broad function, multi-user, broad scope (higher absolute risk)
Averse to work process changes	Flexibility for work process changes
Highly specialized/unique functionality	Common functionality
Unusual computing environment	Common computing environment

A tight project schedule favors the buy option. Purchasing an existing commercial software package can shorten the project schedule because the development (design, build, and test) time is eliminated. For a large software application, the development time (design, build, and test) is probably best assumed to be one or more years.

The availability (in-house or through contract) of sufficient development and support resources makes the build option possible. Many organizations will have a strong culture or even a formal policy towards one position or the other. Organizations that have retained significant software development and maintenance staffs have possibly done so with a philosophy that they will use only the build option to give them the best return on their investment of retaining a large in-house staff.

As the size and complexity of the development project increase, the absolute dollars and months of schedule at risk increase. Being 50% over budget on a \$50,000 project might not automatically get you a meeting with your upper management, but a similar excursion of a \$2,000,000 budget is likely to give you unwanted visibility with your board of directors. Although the estimating tools used in the software industry are improving, it is prudent to assume a fair degree of uncertainty in the final cost and schedule of any software development project. A smaller, more narrowly scoped project carries lower risk in absolute terms; large, multi-user efforts with complex/diverse functional requirements warrant contingency and protective contracting measures as discussed previously. The development or build period may be an acceptable wait for a small application, but a delay of one to several years prior to implementation may not be acceptable for a larger application. During this development period, there are likely to be technology changes to be addressed. Sustainable commercial providers of software packages will frequently be addressing technology changes in their upgrades; those undertaking a custom development might need to deal with design changes in the course of the project or could possibly have technology pass them by during a lengthy build cycle.

The wide variety of available software packages for health, safety, and environmental management can collectively fulfill the majority of functional requirements for the industry. However, the more numerous and diverse the functional requirements, the greater the likelihood of gaps between the requirements and any individual software package. This is one of the situations where the build option could supplement the buy option. Also, the more specialized and unusual the requirements, the less likely it can be purchased off the shelf. Unusual requirements for computing platforms, especially when combined with specialized functional requirements, may leave a health, safety, or environmental manager with only the build option. The more the decision is influenced by a desire to get an exact combination of functional features and computing environment, the more compelling the build option becomes. A compromise approach is to use off-the-shelf software with rather robust functionality for your core health, safety, and environmental needs and to address specialized and unusual functionality through custom development. For example, you can acquire one of a number of commercially available packages for tracking and reporting your hazardous and solid waste (and configure the package to track radioactive waste as well). Then you can build an interface to a custom-developed module that performs criticality safety assessments of the transuranic waste from the inventory data maintained in the purchased software package.

Finally, the build or buy debate needs a cost perspective. Which costs less? The debate continues. The prevailing opinion is that a dedicated staff can produce and support software products for multiple licensees more efficiently than a somewhat transient team of programmers can build and support an application for a single organization. Actual data to support this opinion are difficult to come by because it is rare to identify a custom-built and commercial software application with enough similarity for an "apples to apples" comparison as well as complete and reliable data on total cost of production. A logical fact is that pricing of commercial packages can be based on cost distribution over multiple users. On the other hand, providers of commercial packages have sales and marketing overhead and a required profit margin added to their software cost. These types of costs are relatively small compared to the effects of sharing costs over multiple licensees. Nonetheless, custom-built software, particularly if in-house resources are used, can avoid such costs.

REFERENCES

1. T.D. OLESON, and L. BORDONI, "The Worldwide Financial Impact of the Y2K Problem," IDC Executive Insights, 1998 <http://www.idc.com/f/Ei/091798ei.htm>
2. U.S. Department of Energy, "DOE Enterprise Information Management Overview," 1996 <http://www-it.hr.doe.gov/implan/reference/1focus96.htm#>
3. S.M. JOHNSON, "Guide to Environmental Software Products," Environmental Manager, July 1997, pages 53-96.
4. U.S. Department of Energy, "DOE ORDER 200.1 Information Management Program," 1998 <http://www.explorer.doe.gov:1776/htmls/regs/doe/newserieslist.html>
5. U.S. Office of Management and Budget, "OMB Information Technology Investment Guide" <http://www-it.hr.doe.gov/implan/reference/ombitinvt.htm>