COMMENTARY

Gordon B. Davis on Information Systems

To Buy, Build, or Customize?

Need a new information systems application? Instead of building it, why not buy an applications package system from a vendor and customize it? This is the package approach to development. In the literature, new development processes, such as end user systems, prototyping, and computer-aided software engineering, are widely discussed. The package approach to getting a new application has been almost ignored. Packages have been around for a number of years, but in the past were frequently not considered as significant alternatives to in-house development. Today, packages should be evaluated routinely as an alternative to "doing it ourselves." Several development methodologies being marketed now include options rather than a single development approach; one of the options is the package approach. In this commentary, I will review the reasons for an increased interest in packages and the advantages and disadvantages of this approach. The package approach is especially significant to accountants because there are a large number of accounting and financial packages.

Application packages are software written for sale or lease for specific areas of application. For example, there are packages for common applications such as payroll, order processing, inventory management, scheduling, and purchasing. Packages range in size from fairly small ones for microcomputers to large mainframe packages. They range in cost from a few dollars to several hundred thousand dollars.

Packages differ with respect to tailoring (making changes to suit the specific needs of the organization). There are essentially three approaches to tailoring:

- Packages installed with no tailoring. This approach is often followed with inexpensive packages for microcomputers. The vendor may provide options to be selected during operations, but the user must essentially adapt to the package rather than changing the package to suit existing procedures.
- Packages installed with prespecified options for tailoring of features and selection of procedures. Larger and more expensive packages frequently have options that can be selected when the package is installed. For example, a billing package may have several billing procedure options from which the purchaser can select.
- Packages installed with custom tailoring. The package becomes the basic framework for the application, but significant tailoring is performed by the customer or by the vendor as a separate contract. Unlike the prespecified option, each implementation of the package becomes unique.

Packages form the basis for a variation on the development life cycle or the life cycle using prototyping. A traditional development life cycle for computer applications might consist of three major stages and phases within these stages.

- Definition stage:
  - Proposal
  - Feasibility assessment
  - Information requirements analysis
  - Conceptual design
- Development stage:
  - Physical system design
  - Physical database design
  - Program development
  - Procedure development
- Installation and operation stage:
  - Conversion

- Operation and maintenance
- Post audit

In the package life cycle approach, there is still a need to do a feasibility assessment, but the existence of packages provides cost and requirements data for the assessment. The package approach allows requirements analysis to use an "anchor and adjust" strategy in which the features in existing packages provide a set of requirements which the organization can examine and use as a basis for deciding on (adjusting) its requirements for innovative additions to be achieved through custom tailoring. The conceptual design phase starts with an existing design embodied in the packages. This reduces the effort of conceptual design because the organization can focus on changes from an existing design. Since the packages tend to be state-of-the-art, the company efforts can be directed toward new ideas that extend the existing conceptual designs. In the use of packages for requirements determination and conceptual design, the organization is essentially purchasing expertise embodied in the package.

In the development stage, the amount of physical design and program and procedure development depends on the amount of tailoring. The package provides the basic physical design framework.

In the installation and operation stage, the package aids in conversion because the vendors of packages will often provide conversion aids which an individual organization might not be willing to develop. Manuals and other operations and maintenance support are often better than those from an in-house development. Package vendor personnel are often used to assist in installing the system and in training company personnel. This tends to reduce the installation and training time and improve the availability of training.

Packages can also be used to manage risk in a development portfolio. The use of packages changes several development risks: completion risk, cost/budget risk, controls risk, and performance risk. The completion risk is that an application will never be completed; in the case of a package, there is at least an initial system already available. The cost/budget risk is that the cost of the system will significantly exceed estimates. The basic cost of a package is known before it is purchased. There are still some cost/budget risks relative to tailoring, but these are probably easier to manage than the cost/budget risk of a new application. The controls risk is that inadequate controls will be included in the design and implementation of a system. Package vendors will differ in their implementation of controls in the packages, which means that a purchaser can use controls as a criteria for selection and thereby reduce the risk of inadequate controls. The performance risk is that the system will not perform as planned; a package can be tested under live conditions and unless the tailoring is very significant, the testing will reduce the performance risk to a very acceptable level.

The final cost of an application using the package approach will depend not only on the initial cost of the package but also vendor services (including installation and training) and the cost the organization incurs to tailor and implement. This means that there are very large variations depending on the tailoring approach. This raises a significant question: How much can a company afford to spend to tailor an existing package? What is a break-even concept between in-house development and tailoring of a package? One company uses a figure of 4:1 for tailoring versus package cost as its estimator for packages in application development. In other words, this company is willing to spend much more on tailoring than the cost of the package. They reason that the management of development risks described above plus the requirements expertise embodied in the package and the vendor personnel allow them to achieve a better result in less time with less risk.

There are some obvious dangers in the use of the package approach to development. Packages are never quite right. Tailoring costs may be too high. Packages may be too big because they are too general or have too many unnecessary features. There may be generality at the sacrifice of efficiency. Maintenance/support may not be adequate. There are issues relative to ownership and responsibility. If a company chooses to invest in tailoring a package, then it may wish to be able to maintain the tailoring and add additional features. This means that the company must have the computer code for the package (such as the COBOL programs). Vendors of packages are often reluctant to agree to this. If the vendor is hired to perform extensive tailoring, the vendor is obtaining the company's innovative ideas which it may choose to incorporate in its packge and sell to others. To prevent this, one company which hired the vendor to do substantial tailoring of an order entry system had its own personnel do a pricing module that was based on an innovative company pricing policy.

There are a number of issues in selecting a package which are not found in in-house development. These generally revolve around documentation, implementation and operation support, continued maintenance of the software, and long-term viability of vendor. At the same time that the use of the package has reduced several development risks, it has introduced the package vendor risk.

What are the implications of the trend toward use of packages? The trend is not based solely on economics. In some cases, there are good reasons for choosing a package even if it costs more than in-house development. There are three conditions which suggest the use of the package approach to development:

- A fairly standard data processing application. The availability of many packages for such applications and the possibility of significantly reduced costs are the major motivators for using packages in this situation.
- An application for which the company lacks expertise to define its requirements. The package becomes the basis for the core requirements and supports an anchor and adjust strategy for identifying additional requirements. This reduces the risk of omitting major requirements.
- An application with high development risks. The package approach can be used to manage these risks.

In other words, an organization should incorporate the package approach as one development option in its development methodology.