Build or Buy: That Is The Question

by Steve Prisoc

Sooner or later CIOs will face the choice of either building or buying an essential software application. Most organizations require at least one core system that manages the organization's business and when that system reaches the end of its lifespan, the build/buy decision must be made.

For justice agencies, some examples of these core systems are case management systems serving prosecutors, defense attorneys and courts; criminal history systems that serve the entire justice enterprise; and police records management systems that track arrests and incidents. In years past, these systems were not available as off-the-shelf applications but were typically built in-house, using legacy technologies. Most of these legacy systems are limited and inflexible when compared with newer offerings, and many are no longer maintained by their original vendors. In addition, continued use of these systems can be risky when their underlying platforms and database engines become obsolete or when the original developers of the system retire or leave for greener pastures.

When replacing an existing system, the most important decision point will be whether to custom develop a new system, which could include either developing in-house or outsourcing, or purchasing a commercial off-the-shelf (COTS) application and having it customized to fit particular business needs. Neither choice is entirely risk-free and both will likely involve significant expense. This makes the choice between build and buy an important decision.

There are a number of hard questions that must be answered before a build/buy decision can be made. Unfortunately, there's no magic formula that can be applied and frequently the choice will be the lesser of two evils. Whichever way the decision goes, the risk of failure can be high and there are no guarantees of success. Conventional wisdom has it that purchasing a system off-the-shelf is the most risk-free approach but there are several factors that can make off-the-shelf purchases very risky, such as future vendor instability or the need to extensively customize an off-the-shelf solution to make it usable. Such extensive customization carries all of the risk of in-house development since customization can quickly turn into an expensive redevelopment project.

Here are a few key issues to consider before making a build/buy decision:

1) **Can the business need be met by an off-the-shelf-system?** Is the needed application available from a vendor as a usable COTS or are all

available applications in need of heavy customization? If available applications require significant customization, purchasing a COTS application may involve as much risk as an in-house build. If so, in-house development may be the best method. On the other hand, if an affordable COTS application exists that actually meets the business need, it's usually a mistake to consider in-house development.

2) Do you have needed internal resources? Even if an in-house build makes sense, do you have the internal resources needed to actually accomplish the task? Most government agencies don't have the in-house talent or project management resources needed to succeed with a lengthy, complex development effort. As an example, the level of effort required to build a full-featured court case management application is much greater than the effort required for developing small departmental applications, which are typically built by one or two programmers.

Building a court case management application will likely require significant programmer training and retraining. In addition, in-house development will require full-time expert project managers to maintain coordination between the various participants and manage material resources.

3. Have you considered the cost of lost opportunities? If you decide that you really do have the internal resources needed to accomplish the task, you must then ask if the system will truly

cost less to build in-house than to outsource. When answering, you must factor in the cost of lost opportunity. In other words, to really understand the true cost of building in-house you need to determine what tasks will go undone or require outsourcing if your project management and programming resources are completely dedicated to a difficult, lengthy project.

To come up with an accurate cost figure you should consider the cost of not taking on all of the development projects your programmers would have been able to accomplish had they been available. Once you have projected this "lost opportunity" cost, add the anticipated management costs of the new development project plus all other costs of in-house development—equipment, software licensing, office space, etc.

4) Have you really considered the risks? Risk is a very important consideration and is one that most IT managers underestimate. You should honestly assess the cost of failure to you, your staff and your organization, and then determine if you can really endure them. A mission-critical development effort that fails will likely force significant personnel changes, starting at the top, so if you value your job you'd better be sure that you can meet the extreme challenge of an in-house build.

Even if you are certain that you can build a better system for less, it would be a huge mistake to discount the potential cost failure.

Several studies on project failure have been published in the last decade and most indicate that the chances of a challenged project are quite high.

Most who study project failure agree that the chances of a large IT project significantly exceeding budget or time allotments are about forty percent, and the odds of failing outright are around thirty percent. These odds wouldn't attract a rational gambler and they shouldn't be very attractive to rational IT managers.

5) Can you build the new system in time? Even if you are confident that you can build the system for less, can you build it fast enough? For most government organizations, the attrition rate for superior programmers and project managers is much higher than it is for average IT employees and even average quality IT employees leave organizations more quickly than non-IT employees.

If you're counting on a few high-performing individuals to help you meet your deadlines and deliver your project, be sure to factor in the possible cost of losing one or more of your best people and being forced to recruit and train new people. Even if your best people stay for the duration of the project, you must factor in illness, family emergencies, and other absences when committing to deadlines.

While you can accurately predict the need for material project resources, your human resource need might be the most difficult variable to control.