



To Build or To Buy?

Seven cautions on procuring enterprise software

One or more executives in your firm want to solve a pressing business problem with a new software system.

But finding that system lies in the hands of the IT department.

That search can take several paths. Software firms promote the benefits of their off-the-shelf offerings, ready to plug into your existing processes and deliver results in short order.

On the other hand, some adventurous members of your IT team may propose building their own system, or customizing the software sold by a third party.

Choosing the right path will mean a successful solution delivered on spec, on time, and on budget. Choosing the wrong path will lead to disappointing results, delays, and cost over-runs. At worst, it could mean a total failure.

So how does IT make the right choice on the build vs. buy dilemma?

Here are seven factors to consider, with some cautionary advice on each one.

1: Design specifications

Before you start the procurement process, make sure that your subject matter experts on the issue have their say. Not only will you have a better idea what you need, but those same experts may be able to tell you what other firms with similar needs have done in the past.

They may even know which solutions to try and which to avoid. With all this information, you may be able to narrow your search to some specific solutions.

2: Time to deliver

A pressing business challenge usually drives an enterprise system purchase. How much delay your firm can afford before you solve that problem? As you know, the more custom work, the longer the system will take to deliver.

If you have a short list of solutions to investigate, ask each vendor how long they need to implement the system from the time you sign a contract. Then check some customer references to give their estimate a reality check. That answer can help you set internal expectations for the timetable, and see whether it's acceptable.

3: Integration

These days it's rare for any organization to buy a solution that will stand in a silo, isolated from every other business system already in place.

If you need your new software to share data with your existing enterprise systems, find out whether and how the solution you're considering can link into your existing systems. The faster and smoother, the better.

This kind of integration usually means increasing both the scope and cost of any custom development project.

4: Skill requirements

Any custom software development project calls for many skills, some of which are listed below:

- ◆ Specification writing
- ◆ Coding
- ◆ Testing
- ◆ Debugging
- ◆ Support
- ◆ Content development
- ◆ Documentation
- ◆ Training

Even after the system is fully developed, any future need for upgrades will call for additional coding, testing, support, debugging, and so on.

Who will actually do all this work?

These activities make up the core business of any software vendor. Can you say the same about your firm?

If not, any custom software development project can easily siphon off scarce resources from activities that actually create more value for your business.

5: Maintenance costs

Organizations that undertake the tasks listed above must bear the associated costs. Personnel costs are the most obvious, but investments in hardware, support, training, and others also add significantly to future bills.

Many firms have already faced the nightmare of having to find employees who can understand highly specialized, custom-built systems. Remember the Y2K phenomenon, when COBOL programmers came out of retirement to make a fortune?

Long-term outlays to keep a system running can reach four to five times the cost of the initial development project. Expenses can easily skyrocket if customer demands and other competitive pressures force you to change the system often.

Three Paths to Enterprise Software

If you need new enterprise software for your organization, you most often have three choices:

You can buy **pre-built** (off-the-shelf) software to solve your common business needs.

You can develop **custom solutions** in-house "from the ground up" to tackle business challenges unique to your firm.

You can **customize a third-party vendor's solution** to handle the challenges common to your industry.

Each approach has certain pros and cons. The real question is: Which approach is right for you?

While unforeseen costs often plague custom software development efforts, fixed and predictable fees are the hallmark of pre-built software.

Unlike your firm, software development companies maintain the in-house expertise to do continuous development aimed at solving the business challenges you face. You can depend on more stable, bug-free systems from them than you could likely develop by yourself. Software vendors also spread the costs of software development, upgrades, documentation and support across all their clients. Result: better solutions that cost less.

6: Branding

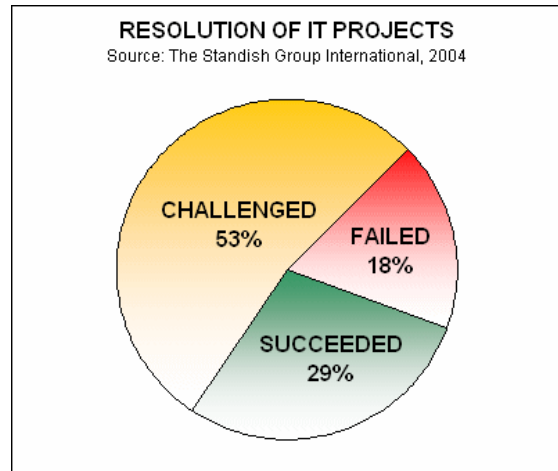
Employee or customer portals are just two examples of systems that must communicate your company's brand. Ask potential vendors whether their solutions allow for custom branding, since a surprising number do not. Some systems that seem to allow customization lack this flexibility.

7: Reinventing the wheel

Why waste development costs when the best tool for the job is already available?

You can apply the same thinking to software. Pre-built solutions may currently exist for business challenges like the one you are facing. If so, any resources you spend reinventing this solution could be better allocated elsewhere.

These seven factors together form a tough challenge to completing successful in-house development, as documented by The Standish Group. In fact, their figures for 2004 show that **less than 1 in 3 in-house IT projects successfully meet its own specifications**. This is a sobering statistic for any IT manager about to embark on a "build-not-buy" adventure.



The Standish Group has been compiling statistics on IT projects since 1994, and has now analyzed more than 50,000 projects. Their statistics from 2004 reveal the ongoing challenges of in-house development.

Successful projects are completed on time, on budget, and include all promised features.

Challenged projects do get finished, but miss deadlines, exceed budgets, and omit features.

Failed projects are cancelled before they are ever completed.

Build or buy?

Now consider a high-level analysis of the build-versus-buy debate.

Three reasons to build

If your situation meets each of the following criteria, you might consider building a custom solution:

1. Your firm absolutely requires some unique system functionality to compete or to survive.
2. The solution you want is not available on the pre-packaged software market.
3. Your firm has the in-house expertise to build and maintain the solution.

But few organizations can answer all three of these questions with a resounding yes.

Three reasons to buy

Three different criteria may apply better, in which case your situation points to the very opposite viewpoint:

1. The functionality you seek will not help your firm achieve any mission-critical business goals.
2. The functionality you seek is easy to find on the market.
3. Your firm is not primarily in the business of creating software, and it does not maintain a cadre of in-house software development experts.

If your firm identifies with even one of the three statements above, you must carefully examine any argument to build a new system in-house.

Perhaps your system developers or IT staff are being motivated by the personal challenge of building a new system, or grasping at a chance to further their own longevity at your firm.

If you give into this pressure, your firm may face a financial fiasco and a project with a great likelihood of being challenged or even an outright failure.

The career-limiting fallout of such a poor outcome would be obvious, both for you and your IT team.

Conclusion

With few exceptions, most organizations are better off buying a complete, ready-made software package to meet any emerging business challenges.

This approach tends to be less risky, less costly, and less time-consuming.