



Why Building Your Own Database Should Be Your Last Resort **By Robert Weiner**

Most weeks I get a call from a nonprofit that wants to upgrade its donor tracking capabilities. The organization has outgrown using spreadsheets to track gifts, donors, and contacts. They want a sophisticated database that will help them identify, cultivate, and communicate with their best donors. Unfortunately, all too often the organization wants to meet these goals by building its own donor database.

There are certainly instances when a custom database is the best or only solution. But those are few and far between. Usually, the nonprofits I talk to aren't building a database because they have unique requirements. They just have a copy of FileMaker or Access, and maybe a volunteer who knows how to use it. I respond by telling them why they shouldn't build their own database:

- 1) **Risk:** Building a database is a risky proposition. I've seen countless custom database projects that failed to meet the organization's needs. There have been a variety of causes: requirements weren't clearly understood or articulated by the organization or were constantly changing; the programmer got distracted by other projects, or left the organization; bugs were never fixed; reports and documentation never got written. Whatever the reason, these projects turned out to be frustrating, costly, time-consuming examples of good intentions yielding bad results. Yes, I've seen successful custom donor databases – many of today's commercial databases started that way. But I've seen many more failures than successes.
- 2) **Focus:** Building a donor database will require significant input from the people who will use it. They'll need to describe, in detail, the minutiae of daily operations as well as your management reporting needs. Is turning fundraisers into database designers really the best use of their time?
- 3) **Support and Maintenance:** Who will you call when your homegrown system has problems? What if the person who wrote it isn't available? What if you need new functionality that's beyond the technical skills of the original programmer? Building a database is an iterative process – you're going to want to make changes over time. Without ongoing support and maintenance, bugs won't be fixed, questions won't be answered, and changing needs won't be addressed.
- 4) **Training:** Who will train staff to use the system? Software training is an art, and not everyone is good at it. But let's say the person who built the system provides great training. Will she be available for years to come as you get new staff? And, just as

important will you have training and reference manuals? How will you keep database training from being like a game of “telephone,” where the tenth person trained hears something totally different than the first person (and the garbage in your database reflects it)?

- 5) **User Community:** With commercial database applications, a host of programmers and clients are providing input to help improve the software. There are user groups (both physical and virtual) where you can learn how other organizations are using the software. There are other clients of whom you can ask questions when problems arise. If you build your own database, there’s seldom anyone else you can turn to for advice or help.
- 6) **Documentation:** I distinguish between three types of database documentation. The first is technical: it tells other programmers why the database was designed the way it was. The second helps train staff members on how to use the database (click this button to go to that screen). The third describes your organization’s procedures, so the right data goes in the right place. When it comes to homegrown databases, all three types of documentation are as rare as Yeti sightings.
- 7) **Cost:** Price is the bottom line. You can get bids for a commercial system. How do you get a firm price and feature list for a custom system? You also need to consider the cost in staff time of designing and testing the system. Keep in mind that every change to the database will cost money: will you be able to pay for upgrades to fix problems and keep up with evolving technologies? While a custom database may seem like a bargain up front, in the long run it may cost a lot more than a commercial database.

There are definitely times when a custom database is the cheapest, most effective, or only the solution. But I urge you to consider the true costs, benefits, and risks of building your own database before going down that road.

About the Author: Robert Weiner specializes in helping organizations make informed decisions about the selection, use, implementation, and management of information technology, particularly in support of fundraising.