
Finding the

PERFECT FUNDRAISING DATABASE

in an Imperfect World

BY ROBERT WEINER

Searching for a donor database for your organization can be a daunting task. It can be difficult to know where to start, what functions and features you're really going to need, and where to look for affordable options. Here's a set of questions that walk you through the process and give you a sense of what to consider along the way. (If you're considering building your own database — or before that option even comes to mind — please read the sidebar, "Why Building Your Own Database Should Be Your Last Resort," on page 9.)

WHAT DO YOU NEED?

The price of a donor database is largely determined by the sophistication of the fundraising it will support, the features and flexibility it contains, the number of donors it can track, and the number of people who will use it. As your needs grow, so does the cost of a database.

There are no hard-and-fast rules, however. Some free systems might support only one user but offer powerful tools. Some vendors sell stripped-down versions of their full systems, which can be upgraded later. These products include many of the features of the full products, but with limitations on the number of users, records, relationships, or other features. And some expensive systems are sold in modules, so you can start small and grow.

In general, most basic databases will allow a limited number of users to manage up to a few thousand constituent records and will track who lives where and what they gave. If that's all you need to do, you can probably use a free or inexpensive system. If you need to do more, however, you're probably going to have to spend more.

WHAT'S IMPORTANT?

Next, you need to decide which features are mandatory. Be careful about this; when you designate a requirement as mandatory you are saying that if a vendor cannot meet even that one need, you would not use the system — even if it were free.

Once you know what's mandatory, everything else is "nice to have." However, some wish-list items are more important than others, so you need to prioritize. Since

every system you look at should meet your mandatory requirements, the nice-to-haves will be the deciding factors.

Be sure to collect requirements and priorities from everyone who will need to get data into or out of the database. If you work in a small nonprofit, that might be one person. In larger organizations, you might have to consult with several departments, including Development, Membership, Corporate and Foundation Relations, Planned Giving, Accounting, and Information Technology. Each stakeholder should weigh in on desired and required features. That doesn't mean you need to talk to every person in the organization; instead, ask each area to appoint a representative. This person should also participate in software demonstrations, rate products, and check references.

The sidebar, "Issues to Consider When Looking for a Donor Database," takes you through a number of the types of actions you may want your database to perform, in simple and more complex maneuvers. In addition, NPower, a nonprofit group that helps nonprofits use technology, has developed two very useful tools that can help your team of decision makers create a prioritized list of which features and functions your new system will require. Their "Functional Requirements Table and Features Checklist" can be found at www.npowerseattle.org/education/resources/functional+requirements+table.pdf and their Issues to Consider When Looking for a Donor Database is available at www.npowerseattle.org/education/resources/features+checklist.pdf.

WHO WILL MANAGE THE SYSTEM?

Be realistic about the amount of technical support you will have, either in-house, from the vendor, or from a consultant (and whether you can find and afford a consultant's help). If the answer is "little to none," you should not choose a system that will require full-time, on-site technical support. (When checking references, be sure to investigate how much technical support the system, and its users, need.) You probably also want a system that end-users can understand with minimal training and one that is able to produce most of the reports you're going to need.

WHAT'S YOUR BUDGET?

You need to have at least a rough idea of what you can spend. The price range for donor databases stretches from free to something resembling infinity. You will need to start with a ballpark budget, which you'll refine as the project progresses.

As a starting point, think about .025% to 0.5% of your annual operating budget. Smaller organizations usually spend closer to the high end of that range. Economies of scale will often allow larger organizations to spend at the lower end.

WHAT SYSTEMS WILL YOU LOOK AT?

You know what you're looking for and what's most important. Now you need to find vendors who can do what you want at a price you can afford, as well as those whose technology is aligned with your own (for instance, if you are a Macintosh shop, make sure the databases you look at support Macs). One approach is to ask other organizations what they're using and whether they like it. You can also turn to the Internet, where there are many discussion lists where nonprofit staff discuss fundraising software — a few are listed at the end of this article. Be sure to

Why Building Your Own Database Should Be Your Last Resort

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 the seven reasons that 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 go 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? Although 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 solution. But I urge you to consider the true costs, benefits, and risks of building your own database before going down that road.

ask specific questions about how well the system addresses the issues at the top of your wish list. And be sure you're talking to comparable organizations; there's no sense in a three-person nonprofit talking to the national headquarters of the Red Cross.

TO RFP OR NOT TO RFP?

An RFP (Request for Proposal) is a document asking vendors whether they can meet your needs and at what price. Many public agencies and large charities are required to use them. For the rest, it's optional.

The purpose of an RFP is to determine a list of vendors whose software you would like to see demonstrated. The difficulty lies in writing questions for the RFP that will yield unambiguous answers and allow you to narrow the vendor pool. Start by describing your organization and the problems you are trying to solve. Then focus on a handful of mandatory and high-priority requirements that will allow you to compare vendors. Good questions: Can your system accept donations in euros? Can it print receipts in Cyrillic? Bad question: Can your database produce a tax receipt? (Every donor database can do this, so you won't learn anything useful from this question. A better question — assuming you need to do this — might be “Can your database personalize receipts based on a donor's gift amount and giving history?”)

There is no set limit for an RFP — they can range from five pages to fifty. Some vendors may not respond to a lengthy RFP if they do not believe that their chances of success are worth the time that will be required. In addition, you will need to read and grade every response. Therefore, try to keep your RFP short. Focus on questions that will truly distinguish one vendor from another. Have the same team that determined your requirements review the RFP before you send it and then rate the responses.

AM I COMPARING KUMQUATS TO KUMQUATS?

The real trick in selecting a database is to compare one system to another. Therefore, tell each vendor what you need to see in order to make a decision. The level of detail you ask for will largely depend on the price of the database. Vendors of higher-end systems will usually send a sales representative to your office for a live demonstration, which can last a full day (or more at a big organization). Vendors of less-expensive systems usually conduct demos via the Web, lasting an hour or two.

In a full-day, on-site demo you have time to get very specific about not only what the vendor should demonstrate, but also how he or she should demonstrate it; for example, “add these types of constituents to the database, apply their gifts to these funds, and then show me reports that look like this.” In a shorter demo you won't have time for that level of detail. Nonetheless, you can still tell the vendors what you want to see, such as gift entry, prospect management, membership tracking, security, and modifying a standard report.

Beyond the Basics: Some Database Functions to Consider

You probably won't be able to afford all the bells and whistles that the highest end database program can provide, but as you're doing your research, here are some functions you should know about — and might want to consider — as you put together your selection criteria.

ADVOCACY

Simple: At the very least, you might want the ability to track people who have signed up as advocates, or asked to receive advocacy alerts.

Complex: If you make heavy use of advocacy tools, you might want to track which legislative districts your constituents live in and what issues they care about so you can send them tailored alerts. You might want a system that can easily share data with your email marketing system or your web site. And you might want to track who clicks on an issue alert, sends email to a legislator, or responds positively to a survey.

EVENTS MANAGEMENT

Simple: At a minimum, you should be able to track RSVPs and event attendees, and produce nametags. You might also want to record which invitations someone has received.

Complex: If you put on complex events, you might want to track income, expenses, caterers and other vendors, host committee members, sponsors, venues, room capacities, special needs (for example, mobility issues or food allergies), and even seating assignments. You might need to issue tickets for your events. And you might need a system that can manage the ticket inventory so you don't oversell.

PROSPECT/MOVES MANAGEMENT & MAJOR GIFTS SUPPORT

Prospect Management and *Moves Management* are terms used in Major Gift fundraising. The term Prospect Management covers all of the processes involved in identifying new prospects; assigning them to fundraisers; tracking cultivation, solicitation, and stewardship activities; and deactivating those who are not interested in supporting you. “Moves” are a series of defined, strategic steps taken as you build relationships with prospects, learn about their interests and abilities to give, develop and issue proposals, negotiate gifts, and provide ongoing stewardship to donors.

Simple: Major gifts fundraising involves individual, personal cultivation and solicitation. At a minimum, you need to be able to identify your prospects and track what happened last and when, and what's supposed to happen next and when. You should have a way of keeping notes on your conversations and activities. And you should be able to set up reminders, often called ticklers, to alert you when follow-up is needed.

Complex: You might want to track any or all of the following information about your major gift prospects:

(Continued on next page.)

Beyond the Basics: Some Database Functions to Consider

(Continued from previous page.)

- **Status:** What is happening in the relationship: Common status codes include Identification, Research, Qualification, Cultivation, Solicitation, and Stewardship, but I've also seen Identify, Involve, Align, Ask, and Close.
- **Stage:** If you have a lot of prospects, you may want an easy way of sorting them into categories, such as A, B, and C where As are red-hot and Cs are "Maybes."
- **Rating (aka Capacity or Projected ask amount):** How much do you believe the prospect is likely to give, or how much do you plan to ask for.
- **Readiness:** When do you plan to make the ask (for example, within 3 months, 6 months, 6–12 months, etc.)
- **Inclination:** This can be as simple as "High," "Medium" and "Low" or as complex as a series of gradations from "We are their top priority" to "Not a Prospect."
- **Assignments:** If you have more than one major gifts officer, you will need to track which prospects are assigned to whom.

RELATIONSHIP TRACKING / LINKING RECORDS

Fundraisers often want to track their prospects' relationships, including their spouse, employer, board memberships, children, etc. You might want to track relatives as a household or as linked individuals with separate records. Whatever the case, your database needs to support your rules.

MEMBERSHIP MANAGEMENT

Simple: At the very least, an organization with memberships needs to track whether someone is a member and when the membership expires.

Complex:

- **Renewals:** You might need a system that can automatically generate reminders when memberships are lapsing. You will probably want to be able to control when renewal reminders will begin (for example, three months before expiration) and whether there is a grace period after memberships expire. If there is a grace period, you will want to control how long the grace period lasts and you might want to subtract the grace period from the upcoming membership year.
- **Membership types:** You might need to manage multiple types of memberships (such as individual, institutional, and affiliate). Each type of membership will probably have different levels with different fees and benefits. In addition, institutional members might be granted a certain number of individual members who receive benefits.
- **Gift memberships:** You might want to allow one person to pay for someone else's membership.
- **Online benefits:** Similarly, you might have members-only benefits on your web site, such as a membership directory, or discounts on events, conferences, or merchandise. This will require a way of synchronizing your web site with your membership data, or a fully integrated system that runs off a single database.

PLEDGES AND RECURRING DONATIONS

Simple: If you accept pledges (for example, \$10/month for 12 months), or recurring donations (\$10/month until cancelled) you will at least want to track the original commitment, the amount owed each billing period, and the amount received.

Complex:

- **Pledge billing:** Depending on the forms of payment you accept, you might need to send bills to the donor every billing cycle, charge your donors' credit cards, and/or send Electronic Funds Transfer (EFT, aka Automated Clearing House or ACH) requests to your donors' banks. If you are storing credit card numbers, you will need to ensure that they are stored in a secure, encrypted format and that only staff who work with pledges have access to them.
- **Changes to pledges:** If you handle major gift pledges, which often span multiple years, you will encounter donors who need to make changes to their pledges. They might need to suspend payments while they're on vacation, or decide to change the pledge amount or payment cycle. You might want a system that allows you to change these details without having to recreate the pledge and the payment history.
- **Honor roll tracking:** If you publish a list of your donors (such as an honor roll in an annual report) and you accept multi-year pledges, you will need rules about how these pledges are counted for recognition. For instance, if a donor pledges \$50,000 over 5 years, will they be listed as a \$50,000 donor in year 1 or a \$10,000 donor? How will they be listed in year 2? Your database will need to support your rules.
- **Pledge write-downs:** Unfortunately, some pledges will never be paid off. You will need a way of cancelling uncollectible pledges. In addition, depending on how your accounting office books pledges, you may need a way of notifying them of these write-downs.

SOFT CREDITING

Soft credits allow you to provide recognition to someone other than the legal donor (determining the legal donor is another subject altogether). For instance, Jane Donor instructs her family foundation to send a check to your organization. The foundation is the legal donor, but you want to ensure that Jane gets thanked, is invited to the Gala, and is listed in your honor roll. Or Joe Donor gives \$500 and his employer matches 1:1. You want to treat Joe like a \$1,000 donor. Soft credits are a common approach to these situations.

VOLUNTEER MANAGEMENT

Simple: At a minimum, you should be able to track whether someone has expressed an interest in volunteering, or has actually volunteered.

Complex: You might need to track volunteers' skills, hours worked, availability for specific shifts, forms and waivers filed, and certifications. You might also need a system that can create a schedule of when each volunteer will work.

Discussion Lists

CharitySoft: A discussion list covering software, the Internet, and charities, operated by CharityChannel; membership fee requested. Sign up at <http://charitychannel.com/publish/templates/?a=2460&z=72>.

FUNDSVCS: A discussion list for Development Services professionals, the people who process the gifts and manage the databases. Sign up at www.fundsvcs.org.

NTEN: "A membership organization of nonprofit professionals who put technology to use for their causes." Members have access to a wide variety of discussion groups. Their web site is www.nten.org.

TechSoup's Technology for Fundraising message board: A discussion board covering all aspects of using information technology to support fundraising. <http://techsoup.org/fb/index.cfm?fuseaction=forums.showSingleForum&forum=2022&cid=117>.

Regardless of how long your demo lasts, it's critical that you ask each vendor to demonstrate the same features in addition to providing an overview of the system. At the end of each demo (while it's fresh in their minds), have each participant rate the system. How well did the database handle each area of concern? For instance, on a scale of one to ten, rate gift entry, prospect management, membership tracking, reporting, and security.

Also, try to get a demo copy of the software so you can continue the test drive on your own. As with the software demos, create a list of tests you would like to try, such as data entry, prospect management, or creating mailing lists and reports. In addition, test functions that seemed unclear during the demos. Note that testing is difficult if you haven't used the software before. See if the vendor will give you a training session first, provide access to online training materials, or walk you through your tests.

There's no law saying you only get one demo per vendor. You might decide you need a second round, or a series of conference calls. You might also go through short demos with half a dozen vendors before inviting a few to do full demos. However, be careful not to burn out your staff. By the third demo most people have lost track of who said what (which is why written ratings are so important).

HOW MUCH IS THAT DATABASE IN THE WINDOW?

Make sure you understand all of the costs involved in a purchase. The price of the software is just one aspect, and often the smallest part, of the "total cost of ownership." You might need to replace your server, network, desktop computers, and printers. You might need help converting to the software, developing special interfaces, or writing custom reports. You might need to invest in additional training or add new staff. And you will definitely need to pay for annual software maintenance — usually about 20% of the software's retail purchase price. (If you can't afford the maintenance, you should not buy the software.)

If you didn't get a written price quote from an RFP, you need to do that now. You should be aware that, while some vendors have very simple price structures, others will need a lot of information from you. They might ask how many users and records you have, which modules you want, and how many servers you will run the software on. (Some large systems run best with separate processing, reporting, and Web servers. You should also have an additional copy of the database for training and testing.)

HAVE I DONE ALL MY HOMEWORK?

So the sales people told you that their database will do everything, including washing your car. It's time to find out how clean the car will be. Just like when you're hiring

staff, you need to check references. Talk to organizations with comparable fundraising operations and staffing, gift income, database sizes, and technical support. It's good to talk to similar types of nonprofits, particularly if you have questions that only a peer institution can answer (such as a theater's need to integrate with the box office), but feel free to cast a wider net.

Divide up your questions and have staff talk to their counterparts at each organization: technical staff call techies, fundraisers call fundraisers, etc. Some questions to ask: How long did it take you to "go live" on the software? Do you have all the reports you need? How long does it take a new staff member to become comfortable with the database? Is the online help really helpful? Do your fundraisers run their own reports? Were you happy with the vendor's support during the conversion? What would you do differently if you had it to do over?

It is important to distinguish software and vendor problems from problems caused by the client, however. The organization might have implemented the software incorrectly, failed to offer adequate training or support, outgrown its database, or purchased the wrong software to begin with. You need to talk to several references so you can discern patterns.

Remember, there is no perfect database. But there might be one that is perfect for you. **GF**

ROBERT WEINER IS THE PRESIDENT OF ROBERT L. WEINER CONSULTING, A FIRM THAT SPECIALIZES IN HELPING FUNDRAISERS MAKE INFORMED, STRATEGIC DECISIONS ABOUT THE SELECTION, USE, AND MANAGEMENT OF INFORMATION TECHNOLOGY. REACH HIM AT WWW.RLWEINER.COM.