

Nonprofit 911 – April 30, 2008
Strategies for Implementing a Donor Database
with Eric Leland
Sponsored by Network for Good

The MP3 audio transcript can be found at
www.fundraising123.org or www.nonprofit911.org

Jono Smith: Hello everyone! Welcome to the April 30 Nonprofit 911 sponsored by Network for Good. This is Jono Smith and I'm the Marketing Director here at Network for Good. We're really pleased that you're able to join us today for this second call in our Donor Database series.

While I know many of you on the phone are familiar with Network for Good, for those of you who are calling and dialing the "Nonprofit 911" for the first time, I just wanted to let you know a little bit about ourselves. We are a nonprofit, just like everyone on the phone. Our mission is to help you raise money online.

"Nonprofit 911" is one of the many free resources that Network for Good provides the nonprofit community to help organizations increase their online fundraising and online marketing results.

In addition to "Nonprofit 911", we also provide online fundraising services directly to nonprofit organizations. One of those services is the Donor Management Suite, which is a donor database designed specifically for a small to mid-size nonprofit and it is also priced specifically for a small nonprofit, starting at only \$99 a month.

So, if you're still using Excel, Access or File Maker, or if you're not happy with your current donor database, we hope that you will take a look at the Suite. And while this is not a sales call, I would invite you to visit www.fundraising123.org/getthesuite to learn more about Network for Good's Donor Management Suite.

Enough about that. We've had over 300 people register for today's call—Strategies for Implementing a Donor Database, with Eric Leland. All the lines have been muted and this call is being recorded. You should've received an email from Network for Good this morning with the copy of the PDF slides for today's call.

If you did not receive that, they're also available right now at www.fundraising123.org under the Training tab, in case you'd like to follow along with Eric today. But if you're not in front of the computer, don't worry, you can always continue your learning and download the slides later at www.fundraising123.org.

I also wanted to remind everyone that we like to make these calls interactive. So, you can email us at any time at fundraising123@networkforgood.org. You can ask our speaker a question and we'll be taking that question at the end of today's call.

Finally, the call is being recorded and we'll have the MP3 transcripts available within about 24 hours at www.fundraising123.org.

At this point, I would like to go ahead and introduce today's speaker, Eric Leland. Eric has spent the last 13 years working with progressive organizations, tackling online and offline technology challenges. He's the founder and director Leland Design. He focuses primarily on strategic technology planning, including website and database planning and development.

Eric is also very active in nonprofit and technology circles. He serves as the co-chair to the Young Nonprofit Professionals Network and is a member of the Graphic Alliance, the Tech Undergrounds and NTEN, the Nonprofit Technology Network.

So, without further ado, I'm very pleased to welcome to today's "Nonprofit 911", our speaker, Eric Leland.

Eric Leland: Thank you very much, Jono, for the introduction. Welcome everybody to the call. I'm glad to be here.

We'll be talking today in this series specifically about implementing a donor database. So, we'll talk about all the things we need to do and be concerned about when we're actually tackling the system that we want to install and use to replace our old system.

Just to give a couple additional brief bits of information about me. First of all, if you're following along in the slides, I'm on slide two and I'll try to indicate when I move from slide to slide so you can follow along. I'll also try to cover some of the material that's actually on the slide, in case for folks that don't have the slides in front of you.

So, my name is Eric Leland again. Yes, I'm the Director of Leland Design. You can find me at LelandDesign.com and you can also read a little bit more about the kinds of things that I tend to write about in the technology sphere, both on TechSoup.org and IdealWare.org. I recommend both of those sites as good places to go find some answers to questions about databases and a variety of other questions you may have on technology and nonprofits.

So, feel free to check me out and write me an email if anything here seems to resonate with you, if you have a question or concern.

Moving on to slide three, "The Promise of Technology." Throughout history, we've had lots of technology of inventions and a lot of folks are using them. I think one of the key things we should remember as we're adopting any new system, this may be a donor database, but this could be any technology system, is that while science and technology have changed our world, millions of us still have no access to basic technologies that have been around for centuries.

It's important to recognize that getting the right tool and getting it to work for you can be a challenge even though they may exist and several may exist. So, we might be among an embarrassment of riches in terms of the solutions we can have, but that doesn't necessarily mean that it's that easy to get a hold of them and to make them work for us.

So, moving on to slide four, our focus today. We want to talk about launching a new donor database. This includes elements of planning, managing the project of implementation, database development, migrating your data, as well as talking about issues around training, sustaining and

scaling your solution for the longer term.

We really have two main components to this conversation. The first is readiness. So, we may have database solution in front of us, we've chosen one. We need to make sure that we are prepared to implement. Then, we want to talk about the implementation and the phases and strategies that go alongside that, so we know kind of what to look out for throughout the process. Like, where we should be involved, what we should look for and the folks that we're working with to implement our solution.

So, moving on to slide five, I put a scenario out there. I just to give you a question, not clearly to answer on the phone but just something for everyone to think about. It's the scenario I run into a lot with nonprofits. You may have just been hired in an organization or perhaps you're fairly new or you're fairly new to the job you're about to get. Your Executive Director wants you to increase the number of individual donors by some large percent. Let's say 500 percent over the next fiscal year. So, you want to get a lot more individual donors.

Maybe she has just signed a contract for a brand new donor database software and she's now looked at you to be the one to do the implementation. In many cases, you may have done some of this work, although you don't consider yourself an expert in implementation. In some cases, you may not have done this at all. Yet somebody has to do it and she's looking at you to do this work.

I'd like you to consider, as you have probably, getting on this call, what are the kinds of things you do next. You're given a system and you're asked to get this installed and this is not necessarily your knowledge area. Who do you turn to? What do you look for next to get this done?

There's a variety of ways to go through it. Now, if you're thinking about answers to this question or how you might proceed, there is not necessarily a right path to go through this but what there are, are some very correct strategies to think about as you work on implementing a solution. We're going to talk about some of those principles going forward.

Looking on slide six, "Getting Ready." So, how do we know when we're ready to get through implementation and actually start implementation? I've listed out five principles labeled 0 through 4, because 0 is always the most important. These principles will help keep your head above water in any kind of technology project. But also, specifically donor database projects that may at first seem daunting.

What we want to remember when we're adopting new databases is number 0, "Plan before tools. We always want to make sure that even though we have tools in front of us, we may have already selected one and we're talking about implementation today. We want to make sure that we understand clearly what our plans are to implement this tool before we go forward.

Don't just simply look at the vendor, the consultant or the internal expert to say, "This is the next step." Let's make sure we revisit what we've all decided to do and do it right.

Know your users. Absolutely know who's going to use this tool. Make sure that you're aware of

who they are and that they're involved.

Another principle we'll cover is sustainability before features. We want to make sure that you can, in fact, run this system and keep it running, rather than piling on a lot of interesting, perhaps useful, potentially not useful, features onto a system that makes it too big and too beastly to manage.

We want to understand that your data rules. Your data is king, in that we absolutely are trying to make a system that allows you to get at your data as quickly, as easily as possible. If we lose that principle, we lose an effective database.

Finally, we want to implement a system for accessibility. So we'll talk a little bit about this, but we want to make sure that this system can in fact be used by your users, that folks can have access to the data in the way that makes sense for your organizations.

Let's move on to the first principle. Planning before tools. Again, if you have a tool in front of you, hopefully you've gone through the system selection conversation earlier. Hopefully you've done some planning, if you have, great. You've been handed this project, grab a hold of that plan, if you haven't already, read it.

If you haven't done this work, it's never too late. Sit down and say, "Yes, I know we have the solution in front of us. Let's figure out what we need out of this tool." We need to understand our missions and goals, specifically what this tool is supposed to improve in our organization. We need to make sure that, when we implement it, that those tools that we implement in fact will meet those goals.

Also, define those successes and measurements. Do you want to improve your donor retention by a certain percentage? Do you want to gather new individual donors? Major donors? Do you want to increase the amount of gifts per donor?

Think about these things and understand how the database will approach that. As you go through implementation, you'll be able to check back on this document and make sure that you're in fact reaching your goals.

Just a real quick hit on the next slide, slide eight, "The Power of Planning." Ultimately the power of planning is that it's going to add years to your life. I say that in some jest, but in reality it's very true. We don't want to go through a large implementation project, or a small one that still might take a month or two, and come out at the end with something that we completely didn't expect, because that creates a very stressful environment for your workers and for yourself for a longer period of time. You're often stuck with these tools for quite some time; they're costly.

It strengthens teams. It really helps to avoid unnecessary costs to do this planning upfront. So don't underestimate the power of engaging in planning even during implementation.

Moving on to slide nine, "Principle One: Know Your Users." We want to know who's going to be using this tool and know that at the very beginning of implementation. Who are they in your organization, and what do they want out of the tool? How do they want to get at it? What are the

ways that they're getting at the information now, and what's not working for them?

You also want to have an understanding of your users, of what they think about databases generally. What's the culture? Why do and why don't they love databases, for instance?

Not a lot of us love databases, however, some of us find them very effective and some of us find them a complete pain in the neck. And we need to understand that from the beginning because there are reasons why databases can be painful. And some of those reasons are going to be cultural and experiential, some of those are very much based on the particular features that you do or don't have right now.

If you understand that now, and if you can alleviate some of those concerns through implementation early enough, then you can get a lot of folks adopting this tool much more happily than you would've if you didn't do this work. So spend some time knowing your users.

Take a look at slide ten "Sustainability Before Features." Basically the question here is how much database can you afford? What's interesting to think about is, although you may have gone through a software selection process and you've analyzed, hopefully, several possible options and selected one. You still have a tool in front of you, a database tool in front of you, that offers you a wide array of features, in many cases.

Some of those features you'll need. Some of those features are critical. And others, you don't need. We really want to explore what's sustainable for your organization. Sometimes it can be easy to think that if you just add another three or five or eight tabs to your system to create a series of 26 tabs you can click on, that that's OK because you just don't have to click on them if you don't need them. The reality is it can tend to create an environment where people use features that you don't really need and end up losing data in places where you're never going to find it.

So make sure that we're budgeting for this system. Can we in fact afford it financially? Let's budget also in terms of technology staffing and support. Who in the organization will be running this system? How many of these tools can they run? What are the training requirements in order to be able to understand how to use these tools effectively in the organization that your software offers?

Also understand ongoing costs. What are the costs not only through the implementation, what are the various charges that will come from, perhaps, your vendor? What is your consultant saying the fee structure will be?

But also, what are the support costs? What do you anticipate needing from support? Are you going to need a small level of support? Are you going to need the full-feature support? Factor that in as well as your own time.

Also try to start getting really clear on this concept of obsolescence. This database, unfortunately, the new database, will not last forever. In some cases, it may last a very long time, in many cases, we find that the databases will last from two to four years before folks are really thinking, "You know what? We need to start either upgrading this thing or moving out of it."

Think about obsolescence. Understand that there will be a cost in two to four years, most likely, to substantially increase the features of your system or to replace it entirely, and factor that into your ongoing costs.

Take a look at slide 11. "Principle Three: Data is King." So your information is king. This is why we're getting a new system. Information hasn't been managed, perhaps at all before this new system, or perhaps it was managed in a system that wasn't working as well as you want.

What's important in implementation of a new system is what you get out of your database. So again, when you're involving your users, the folks that are really using and exacting information out of this database, putting information into the database, you want to make sure that, first and foremost, that all the reports that you require out of the system. This is basically the views of information that you need on your donors, that you can get all those views out of the system.

Outputs, as we call those, those are the reports and the views of your data, outputs lead to input. And this is very critical. There's a lot of folks that will focus on, "I want to track," you know, "68 different bits of information about an individual," maybe a whole bunch of demographic information, lots of information about their interests. But what we find sometimes is that when you look at the output that organizations are asking for, they don't actually include all the information you say you need to track.

So you want to flip that around and say, "What do we really need to get out of this system in order to do the work we need to do?" And what's great is, if you're involving your users from the beginning of implementation, you can have them take a look very quickly at the kinds of outputs that this system offers out of the box. You may be using a system that's already built something like the Network for Good system, perhaps eTapestry, there's a variety of systems that are already built and they have tools for you to use to report your data.

You want to examine those, and understand are those meeting your critical needs? And if not, make sure early on that you're addressing that and making sure that those are going to get built, and that those are going to satisfy your users. The quicker you get at the critical needs in the implementation process, the more buy-in and the more of your users will actually use your system and you're not going to have to wrestle with workarounds that create places where people are storing data outside of the system, or otherwise compromising the integrity of your system.

So data is king, let's make sure we take care of that.

Moving on to slide 12. For those of you that have the slides, it's important that you stare for a moment at the little bucket in the bottom right hand corner. Let me know. It took me about two minutes before it dawned on me what was going on with that bucket.

The point here is that databases should be usable by you. Often they can look nice, and you could have a bucket that has the handle and the spout and a place to put all the water, but when you go to use it, you realize the handle's on the wrong side. And so you try to pour the water, and it all comes out the big hole, it doesn't come out the spout, OK?

So on first glance, some databases can look beautiful and look really powerful, but they absolutely need to be usable by you. So again, what we're trying to do in implementation is to work with your vendor and consultant to get only what you need and you can work up from there.

Great systems often have a lot of ways where you can be flexible and expand, and we like that, but we like to only go there and get these new features when we absolutely need them.

Keeping in mind that simple is powerful. Again the way that you keep things simple is by working with your key users of the system and putting a circle around those things they say they need that are most critical. Because then you're focusing on, your implementation, on only getting those tools up first that are most critical. You're starting with the simplest, most critical things first, and then as folks get accustomed to that, as your users realize, yeah, that's there, I'm happy, you can start talking about your other needs and creating a system that gets more complex as your users understand it.

You also want to make sure, when you're implementing a system, that you're following your own work processes. If you have particular ways that you're inputting your data, maybe you do a lot of events, for instance, and maybe you have registration processes that is where a lot of your actual initial small donors come into your system, you want to make sure that your database has screens that allow you to enter your information and view those people. And the way that you follow your process, you might have five steps to doing that, and it's great to have a database follow those steps with the same kinds of screens.

Depending on your system, that may be more or less configurable for you. You want to make sure that your vendor or consultant understands how you do the work you do so that they can work with you to say, "Hey, we can change this a little bit to fit your processes better," not just give you a generic tool necessarily.

We want to try not to obscure things too much. I always put this in here for both databases and websites. You want dark text on light backgrounds. I can't emphasize this enough, especially for folks who are doing data entry. Having database screens that aren't white or have some kind of gray or some kind of colored background and then you're trying to write on it, you know, doing this eight hours a day in and out, it can really cause a lot of slowness and a lot of data errors in terms of folks simply entering the data. So make sure it's really clean and visible.

And finally, as far as databases should be usable by you, we want to make sure that you have documentation that really works for you. Throughout the implementation process you should be getting exposed to what this system does for you and how you learn about what it does for you. So there might be trainings available to you. Your consultant may be providing those or the vendor may have recorded trainings or live trainings for you. You should be engaging those.

You should also be reading documentation they provide for you and working with it to make sure that it works for you. We'll talk a little bit about what that means going forward.

Let's take a look at some implementation phases now. We've covered some of the principles of getting ready.

Again, plans before tools, we want to know your users, who's really going to use the system.

Focus on sustainability before you focus on extra features.

Remember that your data is king. You absolutely want to understand what you need to get out of the system, and then determine what you need in the system. Make sure that system supports that.

And finally, implement for accessibility. Make sure you can really use this system and your users are comfortable with it.

On the implementation roadmap, what we want to emphasize here is that implementation begins with system selection. So again, you may have already selected a system and gone through this great series in the first conversation on how to select a system. However, once you've selected it, we still have tools that we can select within the solution that we've chosen. And we want to make sure that we're starting with that when we talk about implementation roadmap. Revisit your plan, look at what's available, and select those components that you really need.

We also want to make sure that, in the implementation roadmap, we're focusing a lot on the agreement. We'll talk about what that means, but an agreement is a key part of the process, not only from a contractual, legal perspective, but from really understanding the expectations of all parties involved in natural language, not in Legalese, something that we can all read and refer back to.

It's critical to understand that we need to be involved in testing and feedback, trying to avoid the scenario where the vendor or the consultant sort of steps away, builds a tool, and reveals it in two or three or five months or a year at some very grand meeting. You want to make sure that you're involved throughout this project in a very iterative basis.

We're looking at data migration. That's a key focus in understanding how to get your data into a system and how you need to be involved in that. As well as taking a look at general training and support.

So the implementation roadmap keeps you going not only when the system is rolled out around this point of general training, but also ongoing support, because part of implementation is making sure that, in fact, you can use the system, and that it is being used. And it's really not implemented correctly until the support is in place that enables folks, after the rollout, to use the system. So part of implementation comes once it's actually delivered.

Let's take a look at slide 14. This has an interesting graphic on it. For folks that aren't seeing it, I'll help describe it a little bit. Essentially there is a cycle of implementation, I think we're all familiar with that's basically a circle. I know I'm familiar with this, but we call it data busted.

And basically what we have is we have a circle. We may spend a whole year or two years from the point at which we've actually selected a database. And unfortunately in this first cycle we go all the way around through system selection, through an implementation phase, a training and

support phase, and then we get into a system where, if we haven't done a lot of planning upfront and haven't worked with our users and understood our needs, we can get into a phase where we hate our database.

And this is a phase we've probably experienced in a lot of organizations, where we don't like our system, and we stand around the virtual water cooler talking about why it doesn't work again and again and again. And a lot of us end up developing workarounds, and it's a very negative phase where a lot of us aren't participating in centralizing and making our data more efficient, but it's breaking away.

We want to avoid databasing in circles when we're doing implementation. We want to make sure that we're spending more time upfront in this initial agreement/system selection phase, and iteratively throughout implementation, and making sure we put enough emphasis in support and training so that the "we hate our database" phase is transformed into a more positive growth and scaling experience.

Let's take a look at what that would look like. If we look at slide 15, "Databasing Effectively", what we really want to do is capture changes that we need in the system not as problems and as things that we can't resolve, but as things that we can address and anticipate from upfront. We have to recognize from the beginning that all the databases that we implement are going to be sort of works of art, works in progress that aren't simply done and have solved all of our problems from the beginning.

Most of our organizations are scaling and interchanging, and so throughout the implementation, and especially the support and system use part of implementing a system, we want to have folks in-house, in our organization, that are anticipating and communicating about how this system is working and communicating about those real needs that are emerging, and pushing those out there as emerging challenges rather than as fires to burn out.

So you can, over the course of the first 90 days, and the second 90 days and third 90 days after an implementation, be gathering those kinds of needs that may become the kinds of needs that you need to go back to your developer or back to your vendor and say, "Hey, we need a new piece of functionality or, we need something to solve this." And it won't have surprised you, two or three months down the road because you've been talking about this.

Instead of having a lot of water cooler conversations, you have a lot of folks actually in productive meetings talking about how to improve this system and that's databasing effectively. We're no longer going in circles. We're going up the ladder of capacity. We're building more and more capacity in our staff to do more and better work with our donors, and using our systems wisely.

Let's take a look at system selection. Again we've had a whole conversation about this. What I want to make sure we understand in the implementation phase is that you've developed the requirements when you selected the system. Often when you're in implementation, especially when you're working with a vendor of a system, maybe eTapestry for instance, or a variety of other systems like Raiser's Edge, you may be shifted to another department that's going to actually help you do the implementation.

It's not always clear how much information they really have about your project. It may have seemed like they ought to know everything about the conversations you've had at the sales level, but they probably don't. Make sure that you're sharing your requirements. Share those again. Keep those close to you. All that planning work, you want to make sure you're able to clearly communicate to everybody you're working with in this process.

You want to make sure you're engaging in demonstrations. So even when you've selected a tool, it's important that you take that moment, if your users haven't already done so in the system selection process, to see how this tool is going to work, and explore exactly those features that you have purchased that you're going to use and explore those early. Even with just generic information, or no information at all to get started with. It's important that folks understand what's going to be available and can ask some key critical questions that may change time and cost on the project.

You certainly want to seek out some hidden costs. At the point at which you've actually decided on a tool and then you're trying to get to the nitty gritty of which modules, what support contract, what person in-house do I need to support this tool - you want to start looking at those hidden costs. Sometimes you're going to be doing e-commerce transactions because you want your donors to donate online, for instance.

There maybe additional costs in actually processing those transactions that weren't necessarily exposed to you in the beginning of the sales process, or aren't necessarily in the cost estimates. You want to look at that again. Hidden costs also include your research requirements, as I said before, how you're going to support this system, this product, how you're going to resource in-house your ability to anticipate challenges and support folks using it.

Let's take a look at slide 17 "The Agreement." The key principle on implementation during the agreement phase is you want to build a strong partnership.

In reality, you don't necessarily think of engaging with a vendor as a partnership. Often it's a client services arrangement, you pay them some money and they have a contract, and they tell you what they're going to give you. But we need to transform that, especially with donor data because this is our most critical data for most of us. And it's absolutely critical that we're able to get access to it in a way that we need to.

And for as long as we have this system, we're going to have some amount of critical engagement with this vendor or with the consultant, or the originator of this system, so we need to think about the agreement as a partnership. What that means is that the agreement not only should have the legalese that explains legally and contractually what we're obligated to, but we also need to include a detailed work plan that's written in natural language that explains how this implementation process is going to work, and what the roles and responsibilities are of you in the organization, as well as the vendor or consultant that's working on this project.

So you're trying to clarify the expectations of everybody, understanding how the management is going to work. You want to understand how data migration is going to work and who's involved in that process. And what you can expect the vendor will provide and what you can expect they

won't provide.

You want to also understand how you'll be involved in every phase of the project. So the project often entails building out or installing various pieces of functionality in some space of time, and you want to be present during every iteration of a major piece of functionality coming out on your site. You want to make sure that when something is decided and implemented that you're actually seeing it before the end product of the whole database is revealed to you.

So make sure that this agreement includes a work plan that has you involved at every phase, has you written in to be a reviewer. And clarifies how communication is going to work, how do you escalate problems, how do you talk to them about changes and about bugs, and negotiate that process.

And the point of this agreement plus the work plan is so that you have a working document, so that this document doesn't sit on a shelf as something you refer to only when a crisis emerges and then you have to sue them. This is a product that you want to revisit and say, "Well here was our work plan; let's take a look and see it this still seems reasonable and if we're following it appropriately." If everyone can read it and understand it, then it becomes a much more positive document.

Looking at implementation, let's take a look at slide 18. A key part of implementation, and really the most critical part for everyone involved, is testing and feedback. It's important for everyone to recognize that you are in fact experts at implementing your database. You can repeat after me, "I am a database expert." I know I can't hear you, but I know you're all repeating after me. And it's true, you are a database expert. You only know what truly works for your organization and what does not.

The technology specialists you've partnered with should engage you at multiple stages of the project and you should insist on this. Trying the system early and often, as I've written in this first bullet, helps you to really understand what the features are that you paid for and how they work in your own work processes.

You have to remember that the partner that you've engaged with, even though they may have been very diligent in understanding your work plan, but also understanding your work processes, you still know this the best and only you can make that decision about what's going to work for you. So again, you are a database expert and you are critical to this process.

Again going back to our users, we need to identify who the staff are that are really going to be transforming the system, essentially putting information into that and getting those reports out, and understanding who your donors are and how to engage with them. Testing and feedback should involve them as well. That does mean making time and resources available so that these folks who would normally not be doing this work, clearly their role generally isn't to be testing a new database system, yet they need to have that time factored into their calendar. So being realistic about knowing that you have to involve all your users, yet making that time be realistic so it fits in their schedules, helps you to also build out a work plan.

So again, in the testing and feedback use your work plan. Make sure that when you're doing this

testing that you refer back to what you've decided with the vendor and understand in your own terms, internally, if you're on target.

And there are cases where you are on target, yet you're realizing you're not necessarily getting exactly what you want, that can happen. It's better to just identify that during the testing phase and then to have a real positive conversation with your vendor about how you can change course. It's much more ugly to realize those kinds of more dramatic decisions at the end of the project because often things that they have done will have to be undone, and that just adds more time and cost.

Regarding data migration, this is another critical piece and it often causes headaches in implementing a database solution. I think a key element here to remember is sometimes the first fit is not always the best. What I mean by that is you don't necessarily want to think of the migration as if you're going to do this big migration of all your data this one time, and you're stuck, you're either going to use it or not.

What I recommend for folks that are getting new system is to think about doing test and pilot migrations, so that you don't have to think about, "Gee, I have 10,000 records to deal with in a migration," and all sorts of complexities and let that bog you down.

You can think, "You know what, I'm going to try to import 100 records, and maybe only a certain kind of record, maybe just contact information with basic donation information." See how it goes in the system.

Understand with your vendor how the mapping worked.

And what I mean by mapping is, how did the fields that you used to describe your data, maybe you called "first name" and "last name", and you had a field for a particular event. Maybe the end of year event, did they donate, yes or no.

They may map those fields in a completely different way. They may call your little event field something that they called their little campaign field. A whole different name and you want to see how your data implements into their new system, and where you're going to find it.

So, you can take out chunks of your data, smaller chunks, and work with your vendor and say, "Hey, let's put this into the system and let me play with it. Let me use real data." Again, going back to the testing, see how it works, but also be testing the migration and see how it's going to work when you do the big bad migration at the end.

Again, work with your vendor. Follow their model for how they want you to define your data. They usually help you through sending instructions of spreadsheets and so forth that help you, saying, "Well, I need to understand what your data is, how you describe it, and how we need to then describe it in the new system." They usually walk you through that.

Run your limited test imports. Explore with your vendor your consultant workarounds. There are often cases where things do fit into the system as they sold to you. However, once you fit it into the system, that doesn't really seem like it's in the right place, or maybe that wasn't actually the

right information that should go there.

Explore with your vendor how to solve that problem. Don't try to do that on your own. There's often a better way to fit data that's better for the vendor to have part of this process, so that you don't accidentally shoehorn it somewhere where it doesn't belong.

Another thing to consider with data migration is you want to preserve your old system as long as you actually can. You can imagine six months, even a year into your system, there may be a piece of your data that just isn't necessarily critical on a day to day basis, but you uncover maybe at the end of the year and you realize, "Oh my gosh, I don't know how to find this information."

It's good to have your old system as a reference, if you can. Not only for just backup purposes, if for some reason, in your new system something more horrible goes on and you need to go back, that's one thing. But more likely you might want to just do a check back on your old system just to reconfirm that you didn't lose data. You can find what you're looking for there, and then go and explore where it might have appeared in your new system. So, keep that around.

Another tip with data migration: often it can be easier to clean your data, basically make it more shipshape in the new system rather than in the old system. Also, when you're moving to a new system, there's more ways that you can, for example, find duplicate data your old system with have.

This isn't also the case. You can explore this with your vendor. But sometimes it's nice to just say, look, I'm just going to import all this data with the 58 John Smiths that I have, and use these new tools to go find out those duplicates and clean up the data so I no longer have those.

There may be other issues with, maybe I've named the city and state inconsistently, maybe the ZIP code was written inconsistently. Some of those things can also be corrected on import, and you don't necessarily have to do it before you import it. Again, working with your vendor on what are your strategies for cleaning your data is key.

Keep in mind that you'll have a big part of this process. It's not something that you just hand off to the vendor. You'll have to take charge. Again, you are a database expert. You'll have to take charge of helping your vendor understand what you're tracking, what this information is, what's good and bad information, and helping them clarify whether they imported them done correctly.

Looking at the next slide, slide 20, "Training." Training is of course key to using this system. What can be super-beneficial is that if you have your users testing the system throughout, these users have also been essentially training on the system throughout. Again, a key principle is to learn throughout implementation. Use these testing opportunities as learning opportunities as well.

Another piece of advice is to transform their documentation into something you can use. Often vendors will provide documentation that's a bit more about how you the tool, how do you click from screen to screen, how do you add information into a particular field, but not necessarily why your organization should use that field.

So, you may have a field that's called "campaign", and you want to say for every donor who gives you money, that they gave you money based on a certain campaign. Maybe you did an end of your appeal, maybe you had an event. And there are probably rules that you want your organization to follow on how to assign those campaigns. If a check just comes in the mail and you don't know where that came from, maybe there's some sort of general campaign you want to put that under.

That's called practical documentation, or at least that's the word that I've been using for this. It's practical to you, because it doesn't help enough to say this is how you select a campaign, but for your users you need to understand what the rules are for why you would select a particular campaign for a particular donor. So, it teaches your users how to use the system a little bit more.

You want your internal staff that will be sort of the gurus and the managers of the system to help develop this practical documentation with your other key users, so that you can really explain how to use this system.

Make sure in the training that you've explored the benefits and the limitations of vendor support. There may be several levels that you can choose from, there may not be. There may be email support, phone support; there may be in-person training and recorded training.

It's important, if you can, to try all this out and understand what's effective and what's not for your organization, where can you get the answers you need. You can do this through testing before implementation, too. You don't have to wait for the system to be launched to actually see how support works.

So, it's good to understand how that will work, and what's the best way to funnel requests, is this the sort of support that your users can just go straight to them, or is it better to get the support funneled through someone internal so that you can handle some of those questions first and quicker locally.

And lastly around training, making sure you're training more than one person to be the super-guru of the system. You want to make sure that a few folks have the knowledge to use the system. One person's sick, you have to do another hire, you want to make you haven't lost that skill set and you're scrambling to go pick that up somewhere.

Let's take a look at support. Again, looking at that slide 15, "Databasing Effectively." The support phase is really this phase where the database has been launched, you're using it and what you want to do is make sure that folks don't fall off the road. That folks can continue to use it; they don't forget how to follow your practical documentation, that's available. That you're providing ongoing reinforcement of what are the rules to use the system. You're supporting your staff.

Most importantly, you're anticipating challenges. So, you're making sure you have regular communication with your users to understand what's working and what's not. You can make some collective determinations about when do you need to make a more major change to the system. It may be that there are some struggles that you can anticipate early and understand, and put out there as something to do in the next year.

It may be the kind of thing that you can solve right then and it's a support issue. But if you can anticipate these rather than letting them boil over, then you'll get a lot more folks using the system and not looking for workarounds.

So, again, this is this idea of document challenges of scale. There may be challenges that you can't necessarily fix now, but that you can anticipate and make sure folks have been heard, so that folks feel invested in this system.

Part of support as well, which isn't in bold on this slide, is preparing for obsolescence. So, support should recognize that as your challenges of scale grow, as you start to grow into the system and then potentially out of the system that you're going to need to plan for another bunch of staffing time and money to either upgrade the system or get a new system. So make sure you have, in these support conversations, a milestone for dealing with obsolescence so that you can prepare for that budget hit going forward.

Taking a look at the last slide here before we get to some questions, slide 22, "Strategies for Success." Again, in implementation as well as every other phase of selecting a donor database, or planning for a donor database, you still need to plan. You need to figure out how to do implementation, who your users are, how your vendor's involved, get that agreement set up, testing and migration, et cetera. Planning is key.

Secondly is, absolutely share learnings with your peers. It's important to explore, you know, if you've chosen a system, most likely there are other nonprofits that are using that system, probably quite a few. If it's a consultant, that consultant has worked with other nonprofits, hopefully. Connect with those folks and find out what worked for them. Find out what didn't work for them and try to avoid those issues.

You can do that, also, if you don't know really where to turn, you can go onto forums such as forums on TechSoup.org, for instance, and just say, "Hey, we're implementing this system, wondering if folks have some learnings to share, if I could talk to you more about that." You can get some impartial advice.

Again, I can't overemphasize enough, try before you buy. This also applies to implementation, because again, when you're starting implementation, there's a lot you haven't bought yet. So you want to make sure that you're trying all these features, you're trying the data migration, and you're trying the support before you really engage and commit to it for the long term.

Another strategy, avoid reinventing the wheel. This is key, especially in databases, when you're trying to build your own system or it's looking like you want something that's really customized, or maybe that's your bias, you're coming in saying, "I want to build my own system."

There are a lot of databases out there now that are already pre-built and do a lot of things that you need to do. And it's important that we take a look at all those options, and when we're implementing, too, that we don't naturally jump to customizing something or making a new field when it's already there and we just didn't understand that in the system.

So avoid jumping to that conclusion. Don't reinvent the wheel. Work with your vendor to understand what's already available and what's already been engineered. It's usually solved a whole lot of extra problems you can't even anticipate because it had a lot of engineering support on it. So we want to avoid that.

And again, the final strategy for success here is really, start small, achieve success on that level, and then scale from there. So don't do everything at once, do this incrementally, focus on your critical needs. Make those successful and then move on to the next set of needs.

And so my last slide is the questions slide. And I'm happy to take any email questions that may have come in.

Jono: Great. Thanks Eric. That was a great presentation. This topic is, I think, a little scary, and I thought you did a really amazing job of breaking it down into bite-size chunks and making it seem fairly practical.

Eric: Thank you.

Jono: So we've got time for about 10 minutes of questions. And just a reminder, you can email your questions to us right now at fundraising123@networkforgood.org.

So, we had a couple folks write in who said that they're currently using more of a manual process for managing their donors, like Excel and FileMaker, and they're wondering how do you know when you've outgrown that, and whether you should be considering moving to more of a proper donor database solution?

Eric: Yeah. I think there's several ways that you can know that. Culturally you can start to know that when folks start to complain that they can't do the work that they need to do.

And so what this really comes down to are outputs. So when folks are saying, "I need to be able to see donors between x and y giving between \$100 and \$500 over the last six months." And they struggle and complain that that can't get done. So when folks are saying, "I can't get the information out of the system," you start to realize that critical needs aren't being met.

The other way you can know is that if you're looking around and you're realizing that a lot of folks are storing donor data not in one place, yet you're asking folks to go get all the donors of the organization that satisfy certain criteria. You realize that you're asking folks to go look in multiple places to find this data, which can be very difficult, and you may not even know where that is. So if you're finding that folks are storing data sort of not in one system, that's pretty key.

And then finally, it's sort of, how are you wanting to engage your donors? Are you able to offer the kinds of features and functionality that folks are really clamoring for and that you're hoping to move to? So for instance, using an Excel spreadsheet, it's hard for a donor to, for instance, give you money online and somehow have that appear in your Excel spreadsheet. This is no tie-in.

So it can be difficult to then launch that kind of feature that might really engage donors who get really excited about your work, yet there's not a way for them to actually then give you money

and be recognized in the system.

So you might think about some of those principles. The last one I just mentioned is management. How much time is it taking to keep these databases up and running, to make sure that it has all the information that you need to have?

Jono: Great. We've had a couple of folks who have said they're using QuickBooks as their accounting software, and they're wondering, how important is it, when selecting a donor database, to make sure that the donor database integrates with QuickBooks, or your accounting software? It's a little unclear where the points of overlap are, and whether that needs to be considered in your implementation and decision process.

Eric: Yeah. Actually, I do think it should be considered.

QuickBooks, for the longest time, wasn't very good at integration. And one reason is that you don't find a lot of bookkeepers that smile when you say, "Hey, my QuickBooks is available to all the staff." And then they start to shriek, because the auditors are going to come in and they have no idea what's going on with the data. So they like to keep that QuickBooks system kind of in its own hole.

However, they also like to get reports that they can very easily pull into the system, so it's important to look at your donor databases and find out, can they do batch reporting, is typically the word. Basically, can you get batches of gifts over a month, for instance, or some time period, and export that in a format that QuickBooks reads?

That isn't an automatic integration, that's like an import/export routine. But that's often satisfying enough. We have a lot of nonprofits that are working with bookkeepers, and they just simply need to send that information out. They're not necessarily interested in having the bookkeeper being a part of the donor database process, or even touching that tool. I generally recommend doing that.

If you have more expansive ways that you really want exact information out of your donor database into QuickBooks in more granular kinds of batches, then you might want to start looking at systems that actually integrate in an account management tool, sort of more automatic.

Jono: OK. And there were a couple of questions related to consultants being involved in the implementation process. Can you talk a little bit about how an organization determines whether they need a consultant or if they can just rely on the donor database vendor, and if they do need a consultant how do they find one?

Eric: If you're finding a tool that's offered by a vendor-so a vendor again, Razor's Edge, or Network For Good, or so forth-they offer a tool, you may need a consultant to help you work with the vendor in translating your needs to what the vendor needs in order to get it implemented. So if you're having a hard time communicating with this vendor in terms of moving your data, for instance, from your system to the new system, it can be helpful to have an impartial consultant to help you identify the project requirements, and that person can help you communicate your needs to the vendor.

Now, you should also recognize that with some vendor solutions there aren't third party consultants that are actually technically capable of working on that tool. So for instance, the tool may be closed. And that means that your only possibility for somebody who might build some custom solution is a person that's affiliated with that vendor directly.

So you want to explore that in the beginning when you're choosing a tool. Like, what kind of a tool is this? Is this something that there are a lot of consultants out there that do this work? The vendor themselves can talk to you about that, and then you can go call those folks. You can also ask that question in places like TechSoup, and just find out if they're even available.

If they are and you do need someone to help you customize that, you can go searching for them in a variety of places. There's TechFinder.org, TechSoup.org, or you can go to NTEN.org and find folks that are experts in these areas, and they may be able to help you. Again, it can be translating your needs to the vendor, or actually doing development work; these might be two areas where you need a consultant.

Jono: Great. Thanks Eric.

Just a couple of reminders. The email just for questions is fundraising123@networkforgood.org. Also the Network For Good Learning Center at www.fundraising123.org has a Donor Database tab. There are hundreds of articles on there, including a handful by Eric himself, one that he co-authored with Idealware on finding a low cost donor database solution. So be sure and check those out after the call.

This next question comes from Edward, and he says, "The campaign that we're working on is nearing its end, and we expect to have another few months or so to go before we're done. The donor database system, however, that we're using between two people in our small organization, we've had a hard time learning. And we walked into it somewhat late. The vendor support is minimal and, we've discovered, clumsy. The system has been in use here for about three years. We're wondering is it worth looking into a new donor database system this late in our campaign, or should we try to bear with what we've got?"

Eric: Two months in the world of getting a new database is a short amount of time. It's actually usually extremely short. And so, saying that, the easy answer is you should bear with it. Now, if you look at your absolute critical needs for this campaign, and you're saying, "OK, I need this one report in order to be able to effectively understand that my campaign was successful, or to communicate to my donors, and I cannot get it out of the system. I simply can't get it," then you may need to take some more emergency measures.

But if that's not the case, if it's more that it's a big time sink and you can't work within it, and it takes a while to get support, and so forth, then what I would do is parallel. Start now with this planning process. You're running into these problems now. Start documenting those problems through this last two months of the campaign.

And then when this campaign is over, presumably you'll have a little bit more time to really sit down and forcefully go through a vendor review process. You've done some planning, and so

forth; you have more of that time to really pick the right tool. The problem is you may get into another wrong tool if you do it too quickly, and you won't be fixing any of your problems.

Jono: Sort of along the same lines, you used the phrase earlier, "anticipating challenges." Can you give a few examples of what you mean by that?

Eric: In the support phase, let's say for instance you're tracking donors and you have a new system that allows you to track some additional demographic data that you haven't been doing in the past. So maybe you've just been tracking contact information, now you're tracking date of birth, and you're tracking the name of the person's spouse, and some household information. And folks start to use that, and you start to find folks that are asking for more and more demographic data.

And so they're saying, "Can we have a field for this, and can we have a field for that?" And the system doesn't support that. Basically what you find when you're anticipating challenges is that sometimes a new system opens small cans of worms, where you say, "This is a really neat function," and folks start to use it, and then they start to push it to extremes.

It happens with events, too. You might have an event and you say, "Hey, for the first time we can register folks and track some of this registration information in our new donor database, but I can't track whether this person is a vegetarian, or this person is something else."

So you start to anticipate challenges that say, "OK, we need a lot more fields in the demographic area," or, "We need to have a more robust event piece." So you can look at that in the future and start realizing that the new system you need to have is going to have to have a stronger integration with events generally. And that helps you to anticipate that need and answer a lot of questions that should be coming up in the months ahead.

Jono: Great. We have time for just one more question: "Eric, you talk a lot about establishing your requirements. For folks who have never had to go through a requirements gathering process before, do you have any tips, or resources, or pointers on gathering effective requirements for a donor database?"

Eric: Yes. First of all there's a Database Planning Workbook, also called a Database Planning Guide, that's for a long time been on TechSoup in their Databases section. That walks folks through the actual process of gathering your requirements.

Really the key to gathering your requirements is to have every user describe what they need to get out of the current system. And what I mean by describe that is either print out the reports that they're using now and annotate them so that they can cross off things they don't like and add things they need in this report that's not there. Or if they don't even have that report, to just write it on paper the way they want to see it. So those are the outputs. That's the key.

Get those outputs, and then come back as a group and say, "Of all these things that we want out of the system, what are the most critical requirements?"

And then the final piece is really looking at how the system needs to work. Is this something that only folks internally are going to use? Are there folks that are remotely, outside the office, that

need to have access to this? Are there any security concerns? Can everybody see all the information? So think about how the system will work on those levels.

If you can do those three things you're 80 percent of your way there. The rest of it has to do with what you can sustain and support, how much does the system cost, what kinds of capacity do you have in-house to manage the solution, can you actually run a database on your own server, or do you really need to outsource this because you don't have a lot of IT support internally? Go to TechSoup.org, look for the Databases section, and look for the Database Planning Workbook, and that walks you through a lot of that.

Jono: Great. And we will post links to the items that Eric mentioned in his article at our Learning Center related to today's call, which is at www.fundraising123.org under the Training tab there.

So I want to thank our speaker today, Eric. I know I learned a lot, and I hope everyone on the phone did as well. And once again, we'll be back in a week on Tuesday, with Paul Hagen, who will be talking about CRM and the benefits of CRM to a nonprofit organization. That's on Tuesday, May 6. And then we'll be following up with a panel with Paul Hagen, Eric Leland, and Robert Weiner, a Q&A panel answering all your questions about donor databases, implementation, selecting, CRM, etc. So once again Eric thanks for your time today.

Eric: Thank you.

Jono: And we look forward to speaking with you again on another Nonprofit 911 call. Have a great day everyone.