What Small Charities Need to Know About CRM

Ivan Wainewright

Contents

Contact details: ivan@itforcharities.co.uk

Second Edition: February 2017

Disclaimer

All the information found in this book is provided in good faith, but we do not provide any warranties whatsoever, nor assume any legal liability or responsibility whatsoever, for the accuracy, completeness or usefulness of any of the information supplied. We do not endorse and are not in any way responsible, and have no legal liability, in respect of the URLs, titles and/or contents of any other site referenced in this book.

All trademarks, service marks and other protected names and marks mentioned in this book are acknowledged as the property of their owners.

1. Introduction

Buying a new CRM system or fundraising database can be a daunting experience for any NFP organisation, let alone a small charity. Although I publish a lot of information on my blog which I hope is useful for all charities, there are some points which I think smaller organisations need to particularly be aware of or specifically consider. So I have brought these issues together into this book.

Which in some ways was quite difficult! Because when I read back over my blog posts, so many more of them are equally appropriate for small charities too! I have therefore limited myself to just twelve areas which I believe should give a small charity a solid understanding of what they need to know about CRM systems and fundraising databases. But if you want to know more, then please do refer to my blog: http://blog.itforcharities.co.uk/.

Who this Book is Intended For

This book has been produced for people whose day job is not the procurement or implementation of new databases - which for small charities is probably almost everyone! And you do not have to be technical to understand it. It is for fundraising managers, chief executives, trustees, fundraisers, office managers, volunteers and those working in supporter services, but of course I hope it is equally useful for database staff and charity IT staff too.

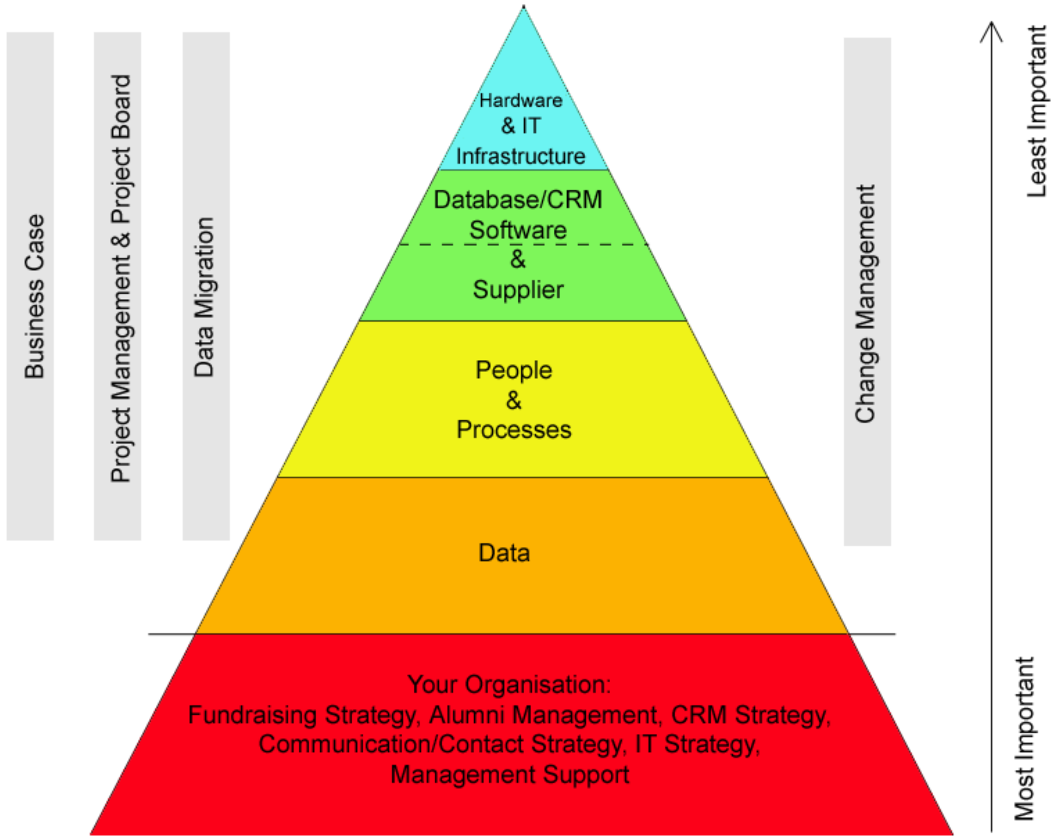
If there are any points you think I should have included in this book then you are welcome to contact me by email (ivan@itforcharities.co.uk) or twitter (@itforcharities).

2. The Basics

The Pyramid of Priority

I strongly believe that whenever an organisation is reviewing or implementing a CRM system, it is important to understand that it is not just the database software itself which needs to be considered and it is not just the database which has an impact on the success of the system. It is far more important to consider the whole operational and organisational environment within/related to, and sometimes outside the area of 'information systems' - not just the database(s).

Consider this diagram which I call a 'pyramid of priority':



At the foot of this diagram are the most important factors in a database implementation and at the top, the least important. That is not to say that those features at the top are unimportant; that is not the case. But there are other factors which are more important in terms of ensuring successful usage and implementation of CRM systems. I.e.: if the data is not clean, useful and comprehensive then it doesn't matter how good the hardware or database software is; if staff are not trained and business processes put in place, then, again, even the best database may struggle to correct such issues; and so on. And at the bottom of any such composition is the organisation's structure itself: its strategies, fundraising policies, communication policies, management support and so on, and the stronger these are then the better a database can be for the organisation.

One key point worth noting, especially because so many charities are now looking at CRM systems such as Salesforce and Microsoft Dynamics 365, is that the supplier –

the business partner – that implements such CRM systems is definitely as important as the software itself and in many instances more important. They are the people who will work with you to define and configure your system, so even if you do select a CRM system then the business partner can make or break that aspect of it.

Additionally, for implementations of new databases in particular, the need for a solid business case and structured project management cannot be underestimated, nor the influence and importance that the data migration from any existing system will have on at least the initial go-live period of any new system.

And running alongside all of that is the key issue of Change Management. A new database by definition involves change and the whole aspect of change management and technology and all the other points it touches in an organisation, their people and processes, data and fundraising needs to be considered carefully.

I would therefore urge any organisation not to under-estimate these issues and to ensure that when considering the use of any CRM system or fundraising database that all these factors are taken into consideration.

All Systems Still Need a Database Manager

With the proliferation of the more generic CRM systems (often referred to as platforms) into the NFP marketplace (i.e. Salesforce, Microsoft Dynamics 365, Sugar CRM etc), it can be easy for a smaller charity who has never before had the opportunity to use such powerful systems to believe that such systems are so easy to use that they won't even need a "database manager". Please don't believe that. You will need someone, whatever they are called, to manage the database/data to some extent, no matter what size you are.

Just because you are going to run the new database through your browser - which may indeed make it more user-friendly on some occasions - that doesn't mean that it will be so easy that it will "run itself". Someone in your organisation still needs to "manage" it in one way or another.

This is because someone still needs the following attributes: to be able to understand how data is structured and recorded on the system; to be able to review a new user request and then recommend whether a new field is added or whether new codes should be appended to an existing look-up table; to understand and interpret what benefits the new database can bring to your fundraisers and fundraising. To control duplicate records... Someone still needs to have some level of control so that not every user of the system is adding data (and codes) in any old way they want (because it is so easy to do so now…) And so on.

Because if you don't have good data, standardised procedures and trained users who understand what they are doing on the database, then don't be surprised if you can't create the reports you need, or you take much longer to segment the database for targeting your donors - or, after a few months of installing the new system, you find your users criticising it because it isn't what they were promised…

The good news is that with some of these systems (and indeed, some 'proprietary fundraising databases' too), such a person doesn't have to be an old-school techie. They should not need to write SQL code (although that may be a benefit for, say, more complex marketing selections), learn a complex report writer or understand entity-relationship diagrams. But they will need to be "data aware" - data savvy as I often call it. In fact, job titles such as "Data Manager" and "Head of Data" are often more accurate and more suitable than "Database Manager" or "Database Administrator".

The database - and the data - is the very base of a charity's infrastructure for their information needs. New CRM systems may make data management easier but you still need someone to manage that.

Are "Volunteer Developed" databases a viable option?

Historically, I have never promoted volunteer developed databases (as much as I respect the time, effort and great intentions of such volunteers). I.e. taking Microsoft Access, SQL Server et al and creating, say, a fundraising system from scratch. But with the recent availability and flexibility of the CRM platforms (Microsoft Dynamics 365, Salesforce et al, and even CiviCRM), there may now be a new 'volunteer developed' model which smaller charities can instigate.

My concerns historically about volunteer-developed databases are based on fundamental issues with such an approach. For example: volunteers may not understand fundraising or fundraisers' needs and thus not design systems in the most appropriate way; volunteers may not be around tomorrow to maintain, change and improve their databases; volunteers may not provide on-going support or training; the system will probably be a bespoke development (based on SQL, Access etc) and will therefore only provide what the charity wanted at the time it was developed; the underlying database may need to be upgraded but without anyone able to do that; and there is no real accountability towards the volunteer in terms of the system's abilities, timescale of delivery etc.

Equally important, for fundraising in particular, such approaches are almost always re-inventing the wheel. Traditional (proprietary) fundraising packages have been available for 20+ years, and even if some are expensive, with fundraising packages such as KISS Software costing from only £100, why would you consider bespoke development for small charities' fundamental fundraising needs? Indeed, I know many NFPs who have started with volunteer-developed databases but have later found such systems ceased being useful, even to the extent that they found they stopped using them any more in their organisation.

However, as per my introduction above, with the generic CRM systems, this is something we can now review.

The reason for this is because of the cost structure of these systems and the central core of their data models and development approach. Cost-wise, the software prices of CRM systems such as these are very low indeed; e.g. CiviCRM is open-source, Salesforce offer NFPs 10 licenses for free, Microsoft offer heavily discounted pricing for charities. And thus the majority of the costs are in the database implementation. And if a volunteer can configure such systems for a charity's needs (hence avoiding/reducing some costs) then some of my previous concerns about volunteer-developed systems may be negated completely or to a large degree. For example:

- These CRM systems have a standard structure ‘out of the box’ which means that fundamental contact management requirements are already there, and thus standard reporting and mailing functionality is taken care of.

- These CRM suppliers will (for the foreseeable future) certainly be providing upgrades, bug fixes, improvements to central functionality/technology platforms and so on.
- Even if the volunteer stops volunteering for your charity, the wide adoption of these CRM systems within and outside the NFP sector and their standard approach, should mean you can find someone else to help and support you. And there are many online discussion boards which can supplement such support too.
- And many of these systems are excellent pieces of software in their own right with great functionality and easier access.

I should add that there are caveats to this approach: it is probably still best suited to small charities with low budgets; I would ideally want such volunteers to be "configuring" the systems as much as possible as opposed to "customisation" (i.e. no heavy coding - instead use the system's in-built tools and third-party apps as much as possible - keep it as "vanilla" as possible); and get them to document whatever they do for you.

And it still doesn't fully negate the need for volunteers to understand your fundamental operational, business or fundraising needs. Some of these systems will have some fundraising functionality but it won't be the same as buying a proprietary fundraising database or even paying for one of the ever-increasing "fundraising templates" you can now get for these CRM systems.

CRM Systems Just Aren't as Simple as Excel...

It is important to realise that CRM and fundraising databases are not as simple conceptually as Excel, Outlook etc. They will take time to learn, they will need some looking after and the likelihood is it will cost money to do all that. Sorry. There is no easy answer to that. CRM software *is* getting easier to use and more intuitive but at the end of the day some of that is still down to individual preferences and interpretation. Just don't under-estimate the work involved in getting going if you haven't used a CRM system before.

A CRM Database Doesn't Mean You Can Now *Do CRM*

This is a brief, simple message: just because you have a CRM database does not mean you can automatically "do CRM". CRM (Customer Relationship Management) is not about software – CRM is a policy and a practise and a whole culture – not (just) a database. In fact, it is possibly one of the most abused terms in the IT industry and has been used to describe so much software that a few years ago had a completely different moniker.

That said, the good news is that if you are a NFP organisation then it is highly likely that you already practise good CRM – if you look after your clients, manage your donors, communicate well with your supporters then you are more than half way there. I believe that, in some ways, the better charities have been ahead of the commercial sector for years in terms of “relationship management”, as charities have always known how important it is to look after their supporters and stakeholders.

So yes, a good database will of course help your CRM strategy and it should do plenty of things for you: keep contact information up-to-date, enable you to understand your supporters better, help you communicate with them more efficiently, in a more targeted and appropriate way, record their correspondence with you, record all their activity with you, allow you to interact with them online and offline and store centrally all such interactions, allow you to allow them to tell you what they want to hear from you… and so on.

But don't let any software vendor tell you that if you buy their database then you will immediately, magically be able to "do better CRM". It is your organisation and your approach which defines that. The database will help but CRM database software will not automatically implement CRM at your organisation.

3. Budgets

Why Free Software Doesn't Mean Zero Cost

I love the concept of open source and free software and when commercial CRM suppliers offer a number of free licenses for charities. But I get equally frustrated and concerned that some organisations believe that free software (or free licenses) equates to spending no money at all on a CRM/database implementation. There are many other costs to consider, whether explicit, hidden or mis-understood, and charities need to recognise these and go in with their eyes open if they choose to use free software in any of its guises. Here are just a few things you should consider:

- Development costs. Even if you use open source database software such as CiviCRM or MySQL, "existing" software you already own such as Microsoft Access (and which therefore has an implied free price tag), a free web-based system such as Zoho, or even the free Salesforce Nonprofit Success Pack, then you are still going to need the database to be developed or configured for your needs. And for some of the above it is highly likely you won't have the skills in-house and you will therefore need to recruit/pay for an external developer. You may find a volunteer which would of course mean a zero price tag for their actual work, but volunteer-developed systems carry their own risks and other potential costs as I discussed above. And even if you do have internal staff who can develop the database then you should really factor in their time and, therefore, costs into the equation.

- Training and consultancy. The software may be free but companies and consultants who provide free and open source software understandably need to make their income somehow, and training and consultancy during your implementation may not be free.

- Data migration and data cleaning. If you have an existing database/spreadsheet then you will need to transfer that data into your new system. As per above, companies may well charge you for such a service - although if you only have a few hundred records (or even a few thousand), then it might well be more cost-effective to re-key it (although even that has its associated costs...) And if you haven't used a structured database before (and even if you have), then you may need to look at your data quality. It's a hackneyed but true statement that just because you get a great, new database, it does not mean that any rubbish data will suddenly become clean and effective. Data cleaning, whether manual or electronic, has costs.

- On-going Development and Maintenance. It will almost certainly be the case that in the future you will need further development and changes made to the initial database you implement. Whether this is because your requirements change or future opportunities arise, or bug fixes are needed; or the actual database software you are using will need upgrading, whether it is the "underlying database" (such as MySQL, Access etc) or the actual application itself when new versions come out. Remember new versions may be optional initially but they could become far more important or imperative if other aspects of your IT infrastructure change. I have worked with more than one charity who has been unable to upgrade some of their overall IT infrastructure or software because their free, central database does not support the latest versions of MS Office, web browsers etc, and this has caused immense frustration internally. One of the great benefits of commercial fundraising and membership packages is that they (should) have new versions released with new features and supporting new platforms, and if you have a good maintenance/support agreement with such companies then these can even be included in any on-going support costs.

- Support. How are you going to get on-going support, even whilst you aren't developing the software? Will your free supplier always be there, will an individual always be around to help as quickly as you need them to? Will their skills deteriorate? And even if the database software is free, if you have internal IT staff then they will still need to support the database in some way, whether explicitly in its development or just in terms of on-going, every-day running, back-ups and so on.

- Hardware and IT infrastructure. Depending on the database you implement, it may need its own server and you may need to consider the configuration on your PCs/clients such as operating systems and memory. And you might find other software applications you are already using are not compatible (or not "recommended") so you might need to upgrade them.

- Hosting and communication costs. If you are planning to host your database elsewhere, then it is highly likely you will have costs associated with that.

- Bugs. In free, proprietary software, there may well be more bugs than you would find in commercial software (although this may not be such an issue in widely used open source software which has great communities to support it). And if this is the case then there is going to be less incentive on the supplier to provide bug fixes. And, depending on the bug, it can be extremely frustrating if there are specific functions which don't work for you and thus you need to find workarounds which can take time and effort. (Of course, believe it or not, commercial software can have the odd bug too...)

And for systems such as Salesforce, there are other considerations too; see my specific chapter later in the book on this.

Free can be great. Open Source can be a wonderful thing. But do consider all the costs of such systems as well as the benefits.

What should you do if you really have zero budget for a new CRM or Fundraising database?

As much as I always uphold having an appropriate budget for your fundraising or CRM database, and as much as I try to promote the understanding that free software doesn't mean zero cost, I also realise that sometimes, in particular for smaller charities of course, you just don't have a budget for a new system. So what should you do then?

Below are the key things I believe you need to consider and act on in such circumstances. They won't cost you anything except your time and attention. Although even with that said, I struggle to truly say the whole implementation will be free. There are always additional costs as I detailed above.

Do note that these tips are primarily aimed at organisations who really don't and cannot expect to have an appropriate budget for their database. As such, if you are a charity of any reasonable size but you are struggling to acquire a budget for a new system then, even though the following may help you, my first piece of advice is to re-visit your business case and determine why it isn't deemed to be so important at your organisation. There is no point – for any size organisation – in getting any system, free or paid-for, which is not right for you, even if it is the cheapest one, even if it is free.

1. Get the basics right before you start looking for a new system

Remember that any CRM/database software is an enabler to help your fundraising strategy. Ensure you know what your strategy is now and for the next few years in order that you will know which CRM system is appropriate for you and what you need from it. And remember too that the database software is only one aspect of any database implementation – don't think that you can just install a database and that will solve all your answers! i.e. a database is only as good as the data in it (i.e. how you collect it, data consistency/accuracy etc), it will only be used as well as your staff are trained to use it and with appropriate processes, and you will still need fast enough hardware and, if appropriate, a fast enough internet connection.

2. Treat your process for getting your new system as a 'formal procurement process', even if you're not spending any money on it

I would still advise you create a document showing your requirements, I would still recommend you research and find out about possible suppliers and, if appropriate, people who can help you (see below), even try prototyping one or two of the systems (assuming they're free, c.f. below), create a plan, engage the rest of your organisation and so on. Don't start using the first free system you find just because it is free.

3. Ideally, use a widely-used package or CRM system – not a bespoke development

One of the key problems which charities have suffered from over the years is when a (well meaning) volunteer or "someone's friend" has created a bespoke database in Access/Filemaker etc. The problem with this is that they may not be around forever, they won't necessarily understand your needs and they are re-inventing the wheel.

I would therefore look at systems such as Salesforce (free for the first 10 users), CiviCRM, SugarCRM and others. They will all need time and assistance with setting up (see below) but they are all good, established systems; and they all have good online communities to help you. This means that if you or any other key people in your organisation leave in the future, then someone else will be able to pick-up your implementation much more easily than if it was a bespoke development. And you will get support and future enhancements. They are also good systems so that if in the future you increase your needs (and budget) then you will be able to build on them potentially without needing to change system again.

4. Keep it simple

Complexity (and over-complexity) brings cost. If you are looking for a free system, then I am assuming you do not need sophistication. If you do require more advanced solutions then free may not be right for you. So keep it simple: it will be quicker to implement, easier to use, easier to support and change in the future and you will get better results. And that has to be your ultimate goal.

5. If you don’t understand databases, don’t set it up yourself

Whilst I normally don’t advocate the use of volunteers for any significant database implementation, if you have never used a database or a CRM system before, or you don’t know best practises in this area or what they can really do for you, then find someone who does and get them to help you. You may have a database-savvy staff member in which case they may be able to help. Otherwise/additionally, there are a number of places you can find free support and assistance from professional IT staff, including CITA (the Charity IT Help Association), the Do-it Volunteering database or even the Information Technologists' Company (what was the WCIT). Or for really small charities, maybe even approach your local college for students requiring work experience? And don’t forget your trustees for contacts and input.

6. Document everything – the configuration of your new system and your processes

During and after the implementation, if you have someone helping you setup the CRM system then get them to document what they have done. Anyone else following them will therefore be able to make changes and support your database much more easily. If you get someone to create proper, written down processes for you, then you will be able to use the system more efficiently and your current and future new staff can be trained.

7. Commit to the project

I have come across too many organisations who have started to use a free database without really committing to it, and in no time it has become a white elephant (all be it a no/low-cost one). But that won’t help you, your charity, your fundraising or ultimately the cause you are trying to support. Don’t assume it will all work beautifully, immediately, first time and without any of your time. You won’t be able to press one button and get the database to tell you who are your best donors, or send a targeted email campaign or find out who you should be speaking to next week if you don’t do the groundwork. Sorry!

Commit - it's really worth it.

4. The Systems

Before you look for a new CRM system…

First: Get your main IT infrastructure sorted out

This is important. Especially if you don't have a central office. Because the route you take for your IT infrastructure may partially influence your decisions on which is the best fundraising/CRM system for you. So, first of all, if you haven't done so already, make sure your email system, file access, website etc are all part of a solid strategic approach.

Second: Remember, it's still important to know your fundamental CRM/database requirements

This is the same for any database procurement and just because you are a smaller organisation doesn't mean you should completely ignore it. So do document what you want the system to do, even if it's just in a brief Word document. That will help you focus on what is most important to you and you can then weigh-up the system options later. And remember, whatever database you buy, the data you put in it, the business processes and training you put behind it, and the fundraising strategy it is supporting are all more important than the software itself. You can have the best database in the world but it won't help if your data is rubbish, you don't adapt your processes and so on.

You also need to bear in mind just how much time and effort you can or want to allocate to managing your CRM/fundraising database. Remember that all systems will require some initial work, regardless of what the salespeople say! You will have to consider the configuration, maybe data migration, how and where you are going to store your data, how external data files will be loaded, how training will be done, how you will maintain look-up tables to ensure good data consistency and so on. And even though you are a small charity, there could still be a number of people who might well want to access and use your database, especially if it is not just for fundraising. In which case do remember the need to have someone who is data/database savvy so that they can manage these issues.

Understand your technological approaches: Hosting and SAAS (Cloud)

You can of course install many software systems on your own 'on-premise' server if you have one, but this may not be an option for some small charities or even the best choice even if it is an option. And more and more CRM systems are now available in the cloud.

As such, in addition to your own server, you fundamentally have two technology approaches which may influence which databases you consider, and which technology is right for you will partially depend on your IT infrastructure. (NB: More technology-savvy readers will no doubt realise I have greatly simplified these descriptions, so please forgive any gross over-simplifications! If you feel I have really misrepresented anything then you are welcome to email/contact me.)

SAAS stands for Software-as-a-Service, where a supplier provides a database in the Cloud which you can access as simply as Gmail, Amazon etc, and from any web browser without the need for any additional software. Most such solutions are only offered this way (with a few exceptions); i.e. not with an alternative "on premise" option.

Hosting is similar but one-step removed, where your database will be "hosted" (stored) on another company's server in a secure data centre somewhere (i.e. you still won't need a server in your own office to make this work). This is usually done for software which has not been written specifically to run directly through a web browser and thus it still needs to be stored on a specific server somewhere. Thus, for Hosting, a supplier will most likely host the database on the supplier's own server/data centre and you will access it remotely from your office/home. (Additionally, most such suppliers will also offer clients the option of letting you install their database on your charity's own 'on-premise' server).

In theory, you could take any proprietary fundraising database that is not cloud-only and host it on your own server anywhere - but to do that, I would recommend that you need to have taken an IT infrastructure route which already supports that. i.e. you are already using a company to do this for some/all of your other files/software applications.

Examples of SAAS/Hosted CRM Systems for a small UK-based distributed charity

As much as I promote the importance of considering your specific needs, I do realise that for many small organisations, your requirements will be similar and comparatively fundamental, so I will use that assumption in the rest of this point. Plus, I am restricting examples of systems to the UK charity market and those which should be within the budget of a small charity.

Examples of SAAS databases are Salesforce, Microsoft Dynamics 365, Workbooks and companies offering 'templated' versions of these. Plus, proprietary fundraising systems such as DonorFy, Access thankQ CRM, eTapestry, KindLink or DonorPerfect.

With hosting, you have a wider option of solutions. As I mentioned, you could look at almost any database, but some companies already offer Hosted solutions as part of their portfolio, mostly through third-party partners; e.g. CiviCRM, Advantage NFP. Or if you do want to consider hosting a fundraising database yourself, then see the next point below as well. (Remember these are products more applicable to small charities - there are many more options for larger organisations).

All the contact details of these systems are listed on my website at www.itforcharities.co.uk/database-software/.

Which database is best for a solo fundraiser?

First, do note the above sections address the fundamentals of any such decision even for solo fundraisers, so if you haven't read that yet then do so first.

The main additional points for a Solo Fundraiser to consider are as follows:

- Are you working for a small charity who does have an office and, if so, then do you already have an on-premise server? If so, then you can look wider for any system which might meet your needs as it doesn't have to be hosted/SAAS. You might still of course want one of the hosted/SAAS systems already mentioned in the above point, but additional examples of such systems you could consider compared to the ones listed above are Donor Strategy, KISS Software and Harlequin Software.

- Even though you are a solo fundraiser, is it likely that the charity could grow in future? If so, then most databases should still be sufficient for such instances, but just don't forget or assume that.

- Even though you are a solo fundraiser, do you actually work for a larger organisation with more complex requirements? In which case, it might be that the very simplest fundraising database might not be applicable to your needs. Hard to quantify, but don't just assume therefore that a low-cost option is best for you. For example, I do know of some charities who have even bought The Raiser's Edge for a solo fundraiser, but in this sort of instance, that is more when they have good in-house IT support, when a fundraiser knows that sort of system and often when they are planning to grow later. (NB: Clearly, other databases similar to The Raiser's Edge have also been installed for solo fundraisers - I use The Raiser's Edge as an example because of its wide awareness and to emphasise the point that it does need more support and consideration when implementing it).

- As no-one else is going to be using the database initially (although I realise some places might have the odd, additional manager also viewing it at times), you really do need a system which does not require complicated management. The difference here to the 'small distributed charity' is that that charity might very well have a few people using it and might be able to justify better the input required to manage it for the whole organisation.

And do remember the other advice I gave before: I'm afraid it isn't as easy as just plugging in the system - it will take time to set-up, even if there is only one of you!

Salesforce: Why does one person love it and the next hate it?

First of all, let me emphasise one key thing about this point: the issues below are not solely about Salesforce - they could be just as true for other, similar CRM systems used by charities today: Microsoft Dynamics 365, SugarCRM, Workbooks, NetSuite and so on. But as Salesforce is so often discussed and considered by charities and it is the main CRM system that many NFPs know, I am using that here as a prime example for these points.

- Configuration and Flexibility: One of the most important things about these CRM systems is that they are so flexible, configurable, customisable and extendable, and as a result one implementation may be completely different from the next, even if they are both addressing the same core fundraising functional requirements. It is of course fantastic that this can be done, and if it is done well then anyone using that particular implementation may well love it. But if not, if it didn't deliver what was promised - or perceived to be promised - if it took too long, cost too much and it seemed like it was still not giving someone what they really wanted, then it is hardly surprising that such users will not like it. It's the double-edged sword of such flexibility.

- The start of the project and then the on-going bit… Following on from the above, it is also quite feasible that the CRM system was implemented well in the first place, and the users liked using it at that point. But for any number of reasons thereafter (e.g. budget, lack of impetus, lack of resources, the wrong resources, lack of understanding of CRM systems), it may not have been maintained well at some sites, and thus its usage at such charities could have dwindled and the users became disillusioned. And this is because if the CRM system was developed for the charity as its needs were at that time, then it is quite feasible that those needs may change over time. The good charities, the good Salesforce users, know this. But that's not everyone.

- Cost of Start-up and perceived TCO (Total Cost of Ownership): Salesforce provides 10 free licenses for charities and heavily discounts others thereafter; Microsoft also gives charity discounts for Dynamics; SugarCRM has an open-source option. So it can appear to the less well informed user that the software is free or very cheap. And why wouldn't you think that if you didn't know! And yes, this is a great thing because you can then spend your budget on key things like development, training, consultancy, implementation, data cleaning and so on. Critical things. Unfortunately, although the licenses are free/cheap, the design, development, resources needed and maintenance are not - they cost money, and carry on costing money. And if a charity is not aware of that, or unprepared, or thinks the CRM will do all they need without any of the above, then what did seem a low cost option suddenly is not.

- Ease of Start-up and Type of User: Salesforce in particular is very easy to start to use: sign-up to a free trial, then buy the licenses (or get them free) and Bob's your uncle (or rather, Marc). No hardware issues, no need to deal with your IT department, no need to talk to any pesky salespeople (!) And Microsoft also offer a similar SAAS option now. Again, all that offers great benefits - it's the Cloud at its best. But it also means that anyone - database-savvy and the non-savvy alike - can start to use Salesforce, regardless of their database or CRM knowledge, their understanding of what it can do, of what you have to do to run such systems well. You would never get that with a "traditional fundraising database" - which shows one of their downsides and conversely their benefits at the same time. But if anyone can start to use Salesforce et al so easily, then there are bound to be people who don't really get it and therefore ultimately won't like it.

- Different types of usage: All these CRM systems are now extremely flexible and can be used for a multitude of applications: fundraising, membership, client management, grant management, help desks, HR needs, (even sales!!) and many more areas of need. It's a really great thing about the systems and shows just what you can do with them. But if charity X has used it for membership management and says how great their implementation is, then charity Y, who wants to use it for grant management may not realise that it might not be so applicable for them or that they need to address it differently. And that can also lead to disillusionment.

- The Salesforce and Dynamics fundraising/membership 'templates': Just to add to the complication of some of the above, there are now a number of good and getting-better 'templates' which third-party companies have designed to manage fundraising and membership almost out-of-the-CRM-box. So, not only do you now have the CRM system with all its pros and cons, you have another layer to consider which also offer benefits and downsides. But some people will still think of it as Salesforce, Dynamics etc. On the plus side, if a charity is using template

X, then it can at least say how good/bad that template is, and other charities can get some feel for it that way. On the downside, such templates are just starting points which many charities will adapt and enhance which again will lead to differences of implementation; and ironically, as the templates become more package-like, so you then get the potential issues which might arise from a more traditional fundraising database. Plus ca change…

- Data, People, Processes (even hardware…): And of course, regardless of everything I have written so far, even if you have the best possible CRM system, it still doesn't matter a hoot if your data is rubbish, your users not trained, your processes not defined, your project management and implementation managed badly… and so on. All the things I bang on about so often in my blog posts. As such, charity X, who has a great system with great people managing it and who really understands CRM, will indeed have the potential to have a great CRM system. Unfortunately, even if you gave exactly the same system and configuration to charity Y, who doesn't get CRM, data management, the need for investment in training etc, then that system will soon be hated by its users. Oh, and if you do buy a CRM system which you can install on-premise/hosted, then the hardware still needs to be up to scratch. And yes, this is exactly the same for Traditional Fundraising Databases as well. But just because you buy a CRM system, that doesn't mean those issues disappear.

- Marketing and the Buzz of the Cloud: Finally, there is also no doubt that some CRM systems are helped by excellent marketing and the belief by some people that the Cloud is the answer for all their needs. The Cloud may well be an answer for some of some people's needs, but it's not quite as simple as that. Personally, I believe that if a system can't provide what you need it to do operationally - say, direct debit management, gift aid, income processing etc - then it doesn't really matter if it's in the Cloud or not. There's no point buying something that isn't right for you even if it does have superb technology (which many of these CRM systems do!) But the CRM suppliers/partners have great marketing staff!

Of course, in some ways some of the above points could apply to The Raiser's Edge and similar fundraising databases as well, as some are indeed generic database pros/cons; i.e. different types of user, the start of a new system vs n months/years down the line, ignorance about costs and so on. But it's the specific attributes and differences and approach of the CRM systems that make all these points especially pertinent for Salesforce and the like.

The 3 top reasons why you should still consider proprietary fundraising software

So much is written now - by myself included! - about the (now not so) new CRM systems (i.e. Microsoft Dynamics 365, Salesforce et al) that you could be forgiven for forgetting that there are still other options. But the good ol' traditional, proprietary fundraising database package is of course still going strong (e.g. for small charities: Donor Strategy, Access thankQ CRM, Harlequin, Advantage Fundraiser etc), and it is still selling well (out-selling CRM?) and there are clear benefits for at least considering it if you are looking at a new fundraising database solution.

These are my top three reasons why you should still consider the fundraising database package:

- The built-in fundraising functionality: this has to be number one. These guys have been selling fundraising databases before Salesforce was a twinkle in Marc Benioff's eye. And as such they have rich and comprehensive built-in functionality for fundraising. It's why they were created in the first place. And even the more thorough templates/apps which are now available for Dynamics and Salesforce are hard-pressed to match all the fundraising functions which the fundraising packages can offer - and some don't even claim to be more than a starting point. Some of them are getting closer of course...

- The suppliers themselves: This is almost as critical as the functionality - in fact, sometimes I would put it above the functionality for a reason as to why you should consider fundraising packages. Because the suppliers have years and years of knowledge of working with charities; they have staff who will have hundreds, even thousands of hours of experience which they have gleaned from the NFPs they have worked with; they have the heritage of working with the sector; and many have hundreds, even thousands of existing charity customers who already use the software, have provided feedback (good and bad) and who, believe it or not, may actually even like the software!

- Reduced risk of the implementation itself: Some of the more critical aspects of a fundraising database implementation are aspects such as direct debits, marketing selection processes, finance/donation structure, gift aid and so on, and these are the areas which the packages already have ticked and, as importantly, have implemented many times in many different environments. Even the better CRM systems still have limited exposure to such areas and are still learning. This does reduce risk. Equally, the wider functionality already available means you have a structure to the system and the implementation, you will be able to run your processes in proven ways and you won't just forget some functionality. Plus, the suppliers have done it all before many times and it's even quite feasible that some of your staff will even know the new system from other places they have worked.

All that said, and to keep things in perspective, I do realise that some of these benefits I am promoting could equally be seen as downsides by CRM suppliers, and some CRM suppliers could quite rightly point to their successes; so do remember I am just saying here why the traditional packages should at least still be considered...

Why business partners are so important for CRM platform implementations

If you select a CRM system such as Salesforce, Microsoft Dynamics 365, Sugar CRM and others, then although you buy the software licenses from those companies, you will implement the software through one of their Business Partners. This means that the Business Partner you select is definitely as important as the software and possibly even more so. You can help select the right partner with a few key guidelines:

- You need to make sure you select a partner who ‘gets you’ – who is a fit for your organisation.
- Don’t just talk to one business partner of your preferred platform, meet at least two.
- Ensure you meet the staff who you will be working with, not just their salespeople (true for all types of CRM system and fundraising database of course)

- Ensure they either know the area of functionality you are working in (e.g. fundraising, case management etc) or they can prove that they can learn about your area.
- If they do have existing functionality for your needs that they can show you then great, do look at it; but even if they don't then don't dismiss them out of hand. A business partner with, say, fundraising functionality but who you don't like could well be worse than a different partner who may have less fundraising experience but who you feel fits your organisation better.
- Understand if they are proposing to configure or customise your system. Configuration, as I have written elsewhere, is definitely preferable most of the time.
- Understand if/how they intend to keep/offer you any intellectual property they design as part of the project.
- Of course, understand how they cost a project.
- Take references from other charities they have worked with.

The good news is that even if you do select a business partner and then find they are not what you thought they would be, then you may well be able to move to a different partner but remain on the same CRM system.

5. System Development

Configure Don't Customise

Keep your CRM implementation Vanilla. This is an IT/CRM term which may sound vaguely amusing if you have never heard it before, but it is one of the more important things you can consider when procuring and implementing a fundraising/membership database or charity CRM system. What it means is that, as much as possible, you implement the database in the first instance without any - or with minimal - "Customisation"; although "Configuration" is fine. (NB: All systems will allow "Configuration" to some degree and if they don't then don't buy them).

Why is this so Important?
It's quite simple: this approach will improve the simplicity of implementation, lower risk, create a faster implementation, a lower cost implementation, there will be less need for specific/expert resources, it will be a simpler and quicker data mapping from your existing database, enable simpler testing, you'll avoid scope creep and more. All these things will mean that your initial implementation will be smoother and have a far higher chance of success. If you are buying a powerful or flexible database then it will be immensely tempting to jump straight in and implement lots of exciting Customisations from the word go, but if you do then you may not receive those benefits listed above.

Defining Configuration & Customisation
These are my definitions of the two key terms, Configuration and Customisation; even if you or some suppliers don't agree with all my points, the heart of the message is the same. Different suppliers will define these differently and claim different things, so at the end of the day, you need to discuss these issues with any prospective supplier and understand, in some detail if necessary, just what you can and can't do in each of the following areas.

Configuration. It is Configuration if...

- You use an application's built-in tools to make changes to the system which every other organisation using this system could do and recognise if they were to start working at your charity.
- The changes could probably (if not always) be expected to be done by a "non-techie". This doesn't mean an un-trained person and it doesn't mean a non-database savvy person, but to put it in perspective, I wouldn't normally expect Configuration to require any programming/coding, i.e. writing code in VBA, XML, C#, SQL etc. This might not always be true but as soon as you do get to this level of "Configuration" then in my experience it is likely that you are starting to get to "Customisation". Either way, ensure again that you know the impact.
- If an upgrade/new version of the package was released tomorrow, then you could install it without worrying that any of the Configuration which you had done would mean that the upgrade wouldn't work, and equally, knowing that the upgrade would not affect any of the Configuration you had done. In practise, some upgrades might still require some such work so if that is the case for your system then do spend time to understand just what that it is. (And of course, whatever the case you always need to test upgrades anyway before going live with them.)

Customisation. It is Customisation if...

- The implementation does involve 'coding' such as using i.e. writing code in VBA, XML, C#, SQL etc.

- The implementation involves bespoke changes, new modules, hard coded programming etc which the supplier or a third-party does for your organisation for your specific needs.
- If the work does mean that upgrades/new versions are affected (either because it stops them happening or because your Customisation would need additional work to be done on it).

So should you ever consider Customisations during an implementation?
It is of course very easy for me to write this and say that you shouldn't do Customisations but do I think you should ever consider Customisation during an initial implementation? Of course you can! Consider but always ask yourself if they are definitely needed. But if you have a critical business function which is required immediately on go-live and there is no other way to achieve it, or you will get so much benefit from a Customised approach that it just makes sense, then go ahead, see how you can implement it. In particular, if a Customisation only has an impact on an "isolated" part of the system (or as isolated as one can expect in a CRM system), as opposed to a core area then that should lower the risk. One thing you could also do to mitigate some risk would be to discuss with the supplier whether could they take your proposed Customisation and build it in to their standard product in a future release.

And do remember that you can of course implement Customisations later, after your initial go-live. Why is this better? Because you can implement them in a more structured way, at a better pace, spreading costs, with less risk, increase user adoption and do it as you learn the package and all its capabilities. In fact, as you gain knowledge of the package, you might even find that some complex Customisations which you were planning originally can be greatly simplified or might not even be needed at all. Don't shy away from those which are needed or which do bring you benefits, just take them on at a pace and structure which you can implement more easily.

Remember, Vanilla is not boring, it's a great flavour because of its simplicity!

When users don't know what they want, keep it simple

My experience is that much of the time, end-users don't actually know what they want or what they could have - or even (completely understandably) what the software can do - and yet we continually ask them. (And even good business analysts may not always get to the crux of the matter). Unsurprisingly, perhaps, we therefore sometimes get a system which, when it goes live, is either far more complex than the user wanted, is not really what they expected, or you even get told, "Oh, actually, that bit wasn't really that important at all. In fact, we don't really do that anymore…"

So: why not give users the basics to start with, the as-close-to-vanilla as possible, and then adapt that later once they do start to really know what they want, once they understand the software and what it can do.

E.g.: Let's keep the forms simple, the data entry straight forward, the processes matching how the software has been designed. And if that means that a user has to do two or three clicks to achieve something rather than one, then that isn't such a terrible thing to have to do in the first place.

This should mean quicker, simpler, less risky, cheaper database implementations. And implementations which are likely to succeed. And happier users.

Thereafter, once we have gone live and the users realise what they have got, then they will realise what they have is either fine or, no, they really could do with process x being streamlined further or data items a and b being linked together. Great. Then do it.

Of course, this approach does require a strong project manager or solution owner, a good project board to support the project manager when necessary and a clear, open approach to explain to the users that this is what we are doing.

Do note I'm not suggesting you just say No the whole time and hide all the really funky things the users actually do want and end up giving them the very bare bones possible. You can think a bit past that! Just be open and inclusive.

And of course, if you do take such an approach and if during your analysis and implementation you do find processes which would benefit from more automation or a more sophisticated approach - and which can be proved to have benefits to be that way - then by all means do it. Ultimately that is one of the great benefits of a CRM system so I am not saying don't do it at all. Just keep it under control.

Should I change my processes to match the database or make the database change to match mine?

First of all, “It Does Depend…”

- Why your current processes are as they are now. In my experience, there are three common answers when I ask "why do you do process X that way?": (i) "I've no idea, we've always done it that way…"; (ii) "Because the current database makes us do it that way"; and (iii) "We've thought it through carefully and this is the best approach for us". Sadly, the latter answer is rare and the first two, and a combination/variation of those is more likely. So, unless you absolutely know you have the perfect business process then it is likely that you could change your processes to meet your new system.
- Legal issues; e.g. gift aid, monitoring restricted funds etc. In these instances, the processes are going to have to reflect what is legally required anyway (although even here there could well be different ways to approach some of them).
- The type of database you are implementing; i.e. fundamentally whether it is a vertical-sector (proprietary) fundraising database package, or if it is one of the generic CRM systems. Although this difference is blurring a bit as I will explain below.

Why Wouldn't You Review Your Processes?

Secondly, when you implement a new database, whatever you buy, reviewing your processes is one of the most useful things you can do. As I said above, unless you already know you are in the "This is definitely the best process for us" category (which is rare), then this is a great opportunity to review why you do something, how, what's good about it, what's bad, what takes time, what could be automated, what you don't need to do and so on. And then you can take all that knowledge to the database implementation so that you can review the best way forward with your new system.

What if you are implementing a Proprietary Fundraising Database package?
Many of these systems have been designed and adapted with many years of experience of what the NFP sector requires. And most which are still being sold today (if not all) are still selling and being used because they do the fundamentals perfectly well. And therefore, if you are buying into them as a software package, as a system, then it would seem to make sense to me to adapt your processes to how the packages have been designed. If you aren't going to do that, then why have you bought that package in the first place? There may be tweaks to do, but I wouldn't recommend changing the systems themselves except for the bare minimum and definitely within any parameters which the supplier suggests.

What about generic CRM systems which you are implementing from a "vanilla" version?
These systems will give you an incredible amount of flexibility - which unfortunately is a double-edged sword. On the one hand, that means you can probably adapt the system to whatever process you prefer (within reason), so if your process review identifies some potentially great benefits by doing a process a whole new way, then this might be a great opportunity. Although that will also have the potential of higher cost, longer time, more arguments (sorry!) and a more complicated system than you might have needed. You need to be very good at keeping a lid on the amount of change you plan. And also: if you can do (nearly) anything then where do you start?! How will you know if your new process really is the best approach? How far should you go when changing the database system? And if you end up with vast swathes of changes then do not underestimate the Change Management you will need to introduce to your organisation.

What about generic CRM systems with a "fundraising/membership template"?
There are now quite a few "templates" which suppliers have created to sit on top of Dynamics and Salesforce in particular. In this instance, you are probably somewhere between still being able to change your processes if you really want to, but because the templates will have their own starting point, it would still be sensible to be using them as a starting point for reviewing your processes. Otherwise, as with packages, why buy them?

So, What Is My Final Answer...?
My personal belief is to keep change as simple as possible, to use the software as it is supposed to be used as much as possible and if necessary, to start simple with the changes and fine tune them (or change them more dramatically) over time. i.e. Unless there are good reasons not to, adapt your processes to the software.

It is only after you really get to know your new software that you will truly understand what it can do, how and the benefits it can bring, so I wouldn't advocate spending huge amounts of time and money on it until you know what you really want from it.

How long does it take to implement a new database?

One of the hardest things about implementing a new CRM system or fundraising database is going at the right pace: too fast and you may rush things, forget something, not do enough testing etc; but too slow and you can lose impetus and enthusiasm – “sorry, why are we doing this project, again?” It’s a tough call.

Historically, I’ve found that charities start such projects with too tight a timescale and don’t leave themselves enough time to get it right, but then often relax such time constraints too much so that they leave themselves in danger of losing all the goodwill and excitement which they had when the project commenced. (Although some organisations do do the opposite!)

And by “implementing a new system”, I am specifically referring to what we can loosely call “Phase 1” or “going live”, which means starting to use the new database in a live environment for the first time and stopping the use of any old database. Thereafter, of course, after we “go live” there could well be more “phases” (although more about that below...).

Perhaps unsurprisingly, one of the things we can do to help us understand how long it should take is to review and understand the project scope, so here’s a few things to consider which might help you get this right. (NB: I'm aware there are clearly other factors which influence timescales, from the type of software to resource availability and budget, but scope is still one of the most central points to consider).

Go Back to Your Business Case for the Scope
You do need to consider what you said you would achieve in your Business Case. If there are some things in that which you now feel you can’t do in an “acceptable” timescale, then you might need to get any changes approved, but you need to be at least laying the foundations for why you initially decided that a new system was required.

What you must do in Phase 1
The new system needs to deliver some tangible benefits and manage key and critical business requirements. This doesn’t mean implementing all your wishes on day 1, and it doesn’t even necessarily mean replicating everything which you can do now in your existing database - but you need to ensure that it can: meet any fundamental operational requirements (e.g. processing donations, producing acknowledgement letters, recording basic information etc); manage any business critical processes (e.g. making direct debit claims, providing mailing files/letters for appeals); and (almost certainly) ensure that it can do regular, automated tasks such as importing third-party data feeds, produce Gift Aid claims etc. I say almost certainly because, for example, if you can’t do a Gift Aid claim immediately, then you can still do that x weeks/months later and still claim the same gift aid (assuming it isn’t the end of the financial year and thus you miss a 4 year back-claim).

Enabling Reporting in Phase 1
If you don’t provide reports then people will very quickly question the benefit of the new system. You can of course technically go-live without producing reports, but if you do then so much of the benefits from any database will be lost. It may be that you can’t immediately produce all reports on your wish-list, but that’s okay as long as you plan their delivery over the weeks/months to come.

Keeping it Vanilla
I’ve written above about the benefits of keeping your initial implementation as “vanilla” as possible and not being drawn into designing complicated customisations which will take time, cost money and increase risk. This will help maintain a useful, structured timescale.

Planning Post-Go-Live Development
If you do cut-back on the scope of the project to get an earlier go-live date, then make sure you plan and communicate when you are going to implement those things which were left out of “Phase 1”. And my recommendation is not to do this as “Phase 2”, “Phase 3” etc, but to move into an “On-going Development” phase, whereby the other items are planned for in this way.

Try to include some Quick-Wins which will Inspire and Excite
If you can do, include a few really inspiring things which will get people excited about using the new database. What this is will of course depend on what your organisation has had before, your users and their expectations! It might be something as simple as introducing mail-merges which do conditional merges and produce different letters; it might be mapping your donors on a Google map; or it might be automating something which has previously taken a long time to do manually.

Don't tie the project to "fake dates"
So often it is the case that charities say they must go live by date X. Often because someone senior has said that this must happen - but without considering any of the above, or other issues, and without really understanding what is involved. If you originally believed you had a 9 month project, but for whatever reason, you start 2 months late, then why should it suddenly be viable to make it a 7 month project?

Similarly, be aware of - but be careful of - dates in the year for when you must do it by; e.g. end of financial year, before a large campaign etc. There may well be good reasons for considering such dates as part of the project plan, and if you can do and if it is correct to work them into the project then that's fine. But if they constrict the project too much then don't let them over-rule everything else.

Understand availability and restrictions on your staff's time
Very often a supplier can work more quickly than the client, and charities often find that the amount of work needed to implement a CRM system is far more than they expected. So don't under-estimate the amount of work it will take your key users especially, and either allow them time to do their Business As Usual as well, or back-fill (some of) their work.

Appendix A: Glossary

- ‘Proprietary Fundraising Databases' (a.k.a. 'Fundraising Packages'): These are the database systems which have been created over the years specifically for the charity and NFP sector (hence they are also sometimes known as 'vertical sector' solutions). Well known examples are supplied by companies such as: Advanced Computer Software, AdvantageNFP, Blackbaud, Care Data Systems, Harlequin Software, KISSoftware Solutions, The Access Group and many more.

- 'Generic' CRM Systems (sometimes abbreviated to just 'CRM Systems'): These are software solutions which were originally designed as sales automation and contact management software but because of their flexibility they have been enhanced so they can be used across various industries, including the NFP sector. Well known examples are Microsoft Dynamics 365, Salesforce, SugarCRM, Workbooks.com, but there are many more. I also tend to classify the software product CiviCRM in this category even though that has been designed specifically for the NFP sector, but its approach is more akin to these other systems listed here as opposed to proprietary fundraising databases.

- CRM (Customer Relationship Management): CRM itself is not about software – CRM is a policy and a practise and a whole culture – not (just) a database. But it has become synonymous in recent years with being a database system. And it also seems that half the software suppliers who used to sell a "fundraising database" have overnight changed to selling "CRM systems" even if their software has barely changed… So be wary of the term - it is a popular buzzword to use. An awful lot of the time, "CRM system" can perfectly well be swapped with "fundraising database". Too many people are quick to say how the traditional fundraising databases can't "do" CRM - that may be true for some software but it is not as simple as that and just saying it is.

- SAAS - Software-as-a-Service: Where a supplier provides a cloud-based database which you can access from a web browser without the need for your own server in your office or any additional software (e.g. just like Gmail, Amazon). It is usually licensed on a subscription basis. (There are also variants of public cloud and private cloud and more but this book will not go into such detail – there are plenty of online resources dedicated to this if you want to understand more).

- Hosting: is similar to SAAS but one-step removed, where your database will be "hosted" on a remote server in a secure data centre somewhere and you access it remotely from your office/home. (I.e. you still won't need a server in your own office to make this work). This is usually, but not solely done for software which has not been written specifically to run directly through a browser and thus it still needs to be installed on a specific server somewhere. This might be offered directly by the supplier of the database software and/or you can host any such system on a third-party’s infrastructure.

- Business Partners: With CRM systems such as Salesforce, Microsoft Dynamics 365 and Sugar CRM, the people who will help you implement these systems are the software author’s Business Partners. They are therefore an increasingly important aspect of such implementations.

For more examples of the systems detailed above, refer to my website: http://www.itforcharities.co.uk/database-software/.

Appendix B: Further Reading & Resources

- My blog: http://blog.itforcharities.co.uk/ As I discussed in the introduction, there is so much more that I could have included in this book and I would encourage anyone who is considering the procurement or implementation of a CRM system or fundraising database to refer to that for more details.

- LASA's ICT Knowledge Base: http://www.ictknowledgebase.org.uk/ Lots more information on supporting aspects of IT which small charities could find useful.

- tt-exchange: https://www.tt-exchange.org Through this programme, eligible UK-based charities are able to request donated technology products from Partners such as Microsoft, Symantec and Cisco.

- Institute of Fundraising Technology Group: http://www.ioftech.org.uk/

- Small Charities Coalition: http://www.smallcharities.org.uk