

HOW TO GATHER AND DOCUMENT A CRM REQUIREMENTS SPECIFICATION



■ As an independent CRM consultant I get to read a lot of CRM requirements documents, and it seems to be an area that most would-be buyers of CRM technology struggle with – probably because there’s little guidance out there about how to approach the exercise successfully.

This would be less of an issue if this was an inconsequential piece of project documentation, but, based on my twenty years in the industry, I’d say it’s not *one* of, but, *the* main determinant of CRM success or failure.

In other words, get the specification right and everything becomes a lot easier downstream, but, rush it, or mess it up, and you can be living with the consequences for a very long time.

So what are the problems? Here are some of the common ones:

There’s not enough detail to select a technology – the functional requirements are too high level, and the buyer is faced with a mass of software options, at vastly different price points, which all seem to meet the requirements. This not only makes the vendor selection process rather fraught, but heightens the risk of selecting and wasting time and money on an inappropriate technology.

Vendors can’t provide reliable guidance on pricing – there’s insufficient detail for a prospective vendor to be able to accurately quote for software and services. This means that organisations have to make purchase decisions based on vendor estimates. These estimates are often only firmed up once the project has started, and

may prove to be wildly different from the initial projections, at a time when your negotiating leverage is relatively limited because you're already committed to some degree along a given path.

I've seen would-be purchasers spend months, and in some cases years, undertaking paid for 'discovery' exercises with vendors, involving considerable amounts of internal staff time, only to pull out and start the process again when the final costs become apparent.

Asking for something that doesn't exist – where the stated requirements are – or at least appear – so out of context of what's available in the market that prospective vendors simply walk away. I've seen businesses go out to tender for CRM systems, but get not responses at all, because the vendors didn't feel they could offer a solution (even though in some cases they had a perfectly valid offering).

Asking for something that doesn't exist at a price point they are willing to pay – this is a variation on the preceding point, but where the requirements are achievable, just at a price that's way outside the buyer's budget.

This can be because buyers of CRM technology aren't always fully aware of what's easy or hard to implement, but can also be the result of the use of MoSCoW-style approaches to requirements definition, where requirements are graded (Must, Should, Could, and Won't in the MoSCoW case). The result can end up as a wall of 'Musts', and the odd, token 'Should' or 'Could', as the various departments and groups involved try and represent their interests as strongly as

possible, without necessarily reflecting the operational priorities of the organisation as a whole.

Scope creep – this is where project delays and cost overruns are incurred when additional requirements or complexity is uncovered during the implementation phase, as a result of requirements not being fully or accurately specified up front. This can force the implementation team to seek additional budgets, dumb-down requirements, or take short-cuts that jeopardise the success of the project.

WHAT'S THE PURPOSE OF THE CRM REQUIREMENTS DOCUMENT?

What are we trying to achieve with a CRM specification? I think there are five key objectives:

- 1. Identify suitable technologies** – by setting out the functional needs we should be able to identify which CRM software products meet our needs and which don't.
- 2. Allow prospective vendors to quote accurately** – from the document we want vendors to be able to provide realistic costs for implementing the system. The aim is to avoid purchase decisions being made based on estimates that later prove unreliable.
- 3. Facilitate internal agreement** – the document should forge a common internal understanding of the shape and function of the system, so it's clear to all which areas of the organisation will be impacted by the system and how.
- 4. Secure appropriate funding and resources** – the specification should allow organisations to identify and secure the necessary budget and resources to deliver the system. This might appear to duplicate point two above, but there's generally a much broader range of costs associated with implementing a system than just those provided by the selected vendor, for example infrastructure costs and the allocation of staff time to the project.

- 5. Smooth out the implementation process** – a clear definition of the requirements up front, shortens the implementation phase, and reduces the likelihood of discovering new requirements which lead to project delays or cost overruns.

Perhaps the key takeaway from the above is that the CRM requirements document serves a much broader purpose than technology selection alone.

However, most of the requirements specifications I come across are largely just lists of functional requirements – often a few pages of bullet-points stating the system must/should/or could do X,Y, or Z. While this *may* assist with technology selection, it has little value in terms of achieving the other objectives listed above.

Before I start to get into the detail of how to approach the requirements gathering process, I want to cover three facts of life that shape our approach:

- 1. CRM software does nothing on its own** – you don't buy or sign up for a CRM application and it somehow miraculously transforms your business. CRM is a tool kit. Decisions have to be made as to *what* you are looking to improve and the system set up to achieve that objective. A lot of organisations get wrapped up in deciding what functionality they need, but give little thought as to how they're going to beneficially use it.
- 2. People will not just tell you their requirements** – this is where a lot of people get into trouble. They see requirements specification as simply about interviewing staff and taking notes. They expect them to be able to fully articulate what they need from a CRM system. In reality this rarely works. For a number of reasons:

Firstly, their knowledge of CRM technology may not be up to the job. They may never have used CRM software before, or their views may be rooted in an application they used many years previously. Their input can often be backward looking and may take little account of what's possible with the latest technologies. Secondly, because users are often only able to describe a narrow set of needs directly related to their role, they generally miss the bigger picture. For example,

I can't remember that last time a user was able to describe key administration and security requirements. The point isn't that staff input isn't valuable – it's essential – but it's generally insufficient on its own to create a useful set of CRM requirements.

- 3. This isn't something you can leave to the vendor** – so the line of thinking sometimes goes: all I need to do is select a technology and then the CRM vendor or implementation partner will help me work out how to get value from it. While there's a certain logic to this approach, there are also a few potential gotchas, given that selecting a technology first may leave you with a product that doesn't meet your needs when the full requirements become apparent, and that being committed along a certain path undermines your negotiating position – a situation that some suppliers are only too happy to exploit.

However the biggest issue is that, as a rule of thumb (and I acknowledge there are exceptions), vendors/implementation partner just aren't very good at working out how to apply CRM technology beneficially. This is slightly counter-intuitive, and you may simply have to take my word for it, but while they (generally) know the *technology* inside out, the – rather important – bit that trips them up, is working out how to apply it in a way that benefits the client.

So to wrap up, the nub of the preceding section is:

- ▶ Spelling out how you are going to beneficially use the technology is at least as important as working out the functional requirements
- ▶ Requirements definition is much more involved than simply asking the question ‘what do you need?’
- ▶ This is not a stage you can easily skip and expect your selected supplier that doesn’t really understand your business to do in your place

■ In this section I want to cover content. So what needs to be included in a CRM requirements document? The following covers the main headings:

APPROACH

This section sets out how the requirements were gathered, and, importantly, who was involved in the process. This helps ensure that the approach was thorough and comprehensive and that any gaps in coverage can be identified.

OVERVIEW

The overview gives a high level description of the new system, and will cover who will be using it, and what they will be using it for.

PHASING

For many organisations, making all functionality for all users available on day one is impractical. The phasing section set out how the deployment will be broken down and how these will be prioritised. For a more detailed discussion of project phasing see [this article](#).

BUSINESS OBJECTIVES

For me this is one of the most important parts of the requirements specification. It describes why you're undertaking the project in the first place: What are the problems you're trying to solve? What

beneficial outcomes do you expect the system will produce? This should be as specific and detailed as possible.

A CRM system needs to be set up and used with specific goals in mind. If these can't be properly articulated, it's unlikely that the project will be a success.

Defining clear business objectives also has an important role in securing appropriate funding and resources, as well as helping prioritise requirements based on business need.

SUPPORTING PROCESSES

The role of this section is to specify how the system will support the business processes necessary to achieve the agreed business objectives.

For example, if a key objective is to grow sales by improving lead conversion rates, from say 10% to 15%, then this section would communicate how the processes from initial lead capture through to order will be managed within the system, and describe how the improvements will be realised.

A sensible approach to this is to describe both the current 'as is' situation, highlighting known issues and inefficiencies, and then the 'to be' processes. I tend to cover these processes in both a written narrative, as well as a series of flowcharts. The flowcharts for the 'to be' describe in detail what's being updated in the system by the user for each step in the process.

Most organisations neglect this area when defining their requirements, but it plays a key role in helping determine the fit with the ‘out of the box’ capabilities of the various CRM software options, and helps prospective vendors more accurately cost out any associated customisation and development required to bridge any gaps.

ENTITIES

This section describes the main types of records managed by the system. For example, this might include information about people, organisations, sales opportunities, leads, and activities, but also may include new entities that need to be added to support an organisation’s processes. I generally start with a diagram of these entities, and how they relate to one another, and then describe each in detail, setting out what fields will be tracked within each.

This might seem like a lot of work up front, but it saves valuable time later during the implementation phase.

FUNCTIONAL REQUIREMENTS

This part of the specification describes all the supporting functional requirements. I tend to break this down into specific functionality required to support each business process, and more general supporting requirements such as administration, security, or access related functionality.

DATA MIGRATION AND INTEGRATION REQUIREMENTS

Data migration and integration requirements can have a huge bearing on the cost of implementing the system. It's therefore important to document in as much detail as possible needs in this area. A clear definition of what data is moving between which systems, is essential, and, in terms of integration, whether the requirement is for real time or more periodic integration.

REPORTING

This area should ideally identify what management information you're looking to get out of the system. Detailing out reporting requirements, not only helps speed up the implementation, but can be a very effective way of validating that the processes have been modelled correctly and that the associated data capture is correct.

SYSTEMS REPLACED

Finally, the systems replaced section confirms which existing systems and data sources will be decommissioned as part of the roll out of the system. This helps remove any ambiguity as to how the post-live world will look.

■ This section covers how to go about gathering your CRM requirements.

The starting point, if you're not reasonably familiar with CRM technology, is to do some initial research. It's very difficult to develop a set of requirements if you're not comfortable with what CRM software is and does.

Fortunately lack of knowledge or experience is not an insurmountable obstacle. Most software vendors offer free trials, and getting hands-on with a package is a great way to explore its potential. It's also worth asking a range of vendors to demonstrate their offerings as a way of showcasing what's available. Shows and events can be another way to get up to speed with the market, and looking at how other organisations, particularly those in a similar field, use CRM software can be very insightful.

I add one brief note of caution though, and that's to remember this is a preparation phase, and to avoid getting rushed into a software purchase before the requirements are fully defined, no matter how attractive the offer or persuasive the pitch.

Having now acquired a decent working knowledge of CRM software the next step is to work out how it can be applied to your organisation. People are frequently surprised how broadly CRM technology can be used, and there are often very powerful applications outside the traditional domains of sales and marketing. So the starting point is to work out which areas of the business might potentially benefit.

It's generally worth casting the net fairly wide at this stage and then focusing in on the more attractive areas in due course.

With a broad scope defined, the next step is to perform some initial business analysis. Talk to each area of the organisation and identify the business processes that each manage. For a sales team this might include contact management, sales pipeline management, quoting, forecasting, and managing orders. Look at how each process is currently performed and how it's supported by technology, and start to identify areas of 'friction' – where the process isn't working as well as it could.

For example, a review of the lead management process might reveal that incoming leads are recorded into an Excel spreadsheet, then assigned to individual sales people. The marketing team consequently have no means of easily tracking whether a lead has been followed up and if it resulted in a sale. This means that marketing are unclear which of the campaigns it runs generates business and which don't, and can't allocate its budgets to maximum effect, and there's a concern that lead conversion rates are lower than they should be because there's no system to manage longer term leads.

While it's important to note any requirements that users highlight during this analysis stage, the main aim is to understand the processes and related issues.

Once a full set of 'as is' business processes has been created, and the associated friction documented, decisions will need to be made on which areas will be supported by the new system, at least in the

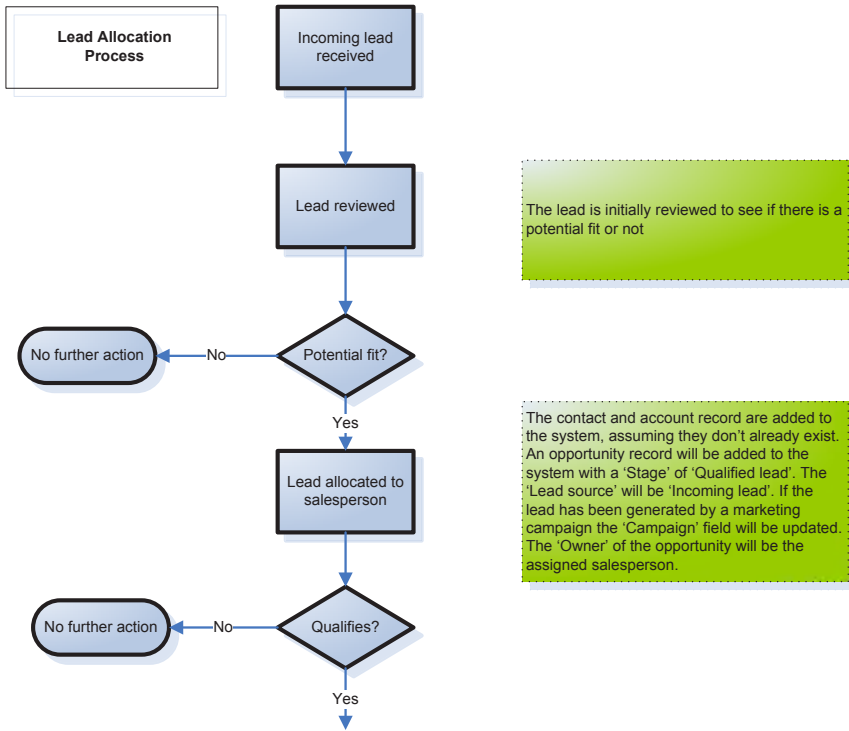
shorter-term. Priority will generally be based on addressing the issues that are impacting the organisation most, and where an investment in technology is going to yield the greatest returns.

Having determined *what* the system will support, the next step is to develop the ‘to be’ processes. This will generally be a case of reframing how the existing business processes will operate in the new environment so that they deliver the desired benefits.

However, in some cases, the process itself may require more significant reengineering. For example, going back to our lead management example earlier, it may be viewed that the whole process of allocating and tracking leads needs to be revisited to avoid the potential pitfall of automating an already flawed process.

In addition, the introduction of a new system may well require processes that weren’t needed previously, for example to help maintain data quality. In both these cases the necessary steps will need to be taken to develop these processes before the requirements can be finalised.

Once complete, consideration will need to be given as to how they will work within a new CRM platform. This will build on the knowledge picked up in the initial research stage. I tend to do this by writing both a narrative to describe each major process and how the new system will support it, but I also create process maps which contain a commentary about what’s being updated in the system for each step. This is illustrated below:



This may sound time-consuming, but it's worth undertaking because it helps flush out functional requirements that may not otherwise have come to light. For example, in a recent project, when we mapped out the 'to be' processes, it became apparent that a third party service provider would be entering data directly into the system, which threw up a whole range of new security requirements that previously hadn't been identified.

Documenting processes in both story form as well as graphically tends to be very effective in helping users identify whether you've

understood and modelled their processes accurately, and, as a result, this phase tends to be fairly iterative.

Even if you're not confident enough with CRM technology to fully define how the processes will be supported by the new software, just getting the processes documented is a huge step forward in its own right, as this will allow prospective suppliers to much more accurately assess implementation costs.

The next step is to work out what information you wish to capture in the system. This will be driven by the 'to be' processes you've already mapped out. I tend to draw out the different record types and how they relate to one another, and then record the fields that will be required for each one, per the diagrams below:

Principle Entities



9.2.8 – Opportunity entity fields

Field name	Description	Field type
Name	The opportunity name	Free text
Opportunity type	The type of opportunity	Pick list
Organisation	Link to the associated organisation	Look up to organisation record
Main contact	Link to the main contact record	Look up to contact record
Opportunity stage	Indicates where the opportunity has reached in the sales cycle	Pick list
Opportunity status	Indicates the status of the opportunity i.e. open, closed	Pick list
Date lead received/generated	Indicates when the lead was first identified	Date field
Description	The description of the opportunity	Free text
Reference number	Unique reference assigned by the CRM system	Auto-populated by the system
Estimated close date	The date the opportunity is expected to close	Date field
Account manager	The salesperson managing the opportunity	Look up to a user record

With the first cut of the data capture complete, the data integration and migration requirements for the system should be finalised. As these can have a significant impact on the complexity and cost of the project, it's important these are defined in detail. This will involve working closely with the staff members who know these sources best. Not only should each source be catalogued, but it's useful to record at field level what data will populate the CRM system, or, in the case of integration, pass from CRM to other systems.

The same attention to detail should also be applied to determining the reporting requirements, because, as with data migration and integration, these will help validate whether the data capture needs have been correctly defined. You can't report on data that you aren't

recording, so creating detailed mock ups of the reports you expect to run from the system can be very helpful.

Finally, work through the supporting functional requirements. To a large extent these will flow from the ‘to be’ processes, so in our previous example of the lead management process, a key requirement might be to automatically email the salesperson when a new lead has been assigned to them.

Defining more generic requirements, for example around security or administrative needs can be more demanding if you’re not familiar with CRM software, which is one of the reasons that the first research step is so important. Here’s [a link to an article](#) which describes some of the areas that commonly get missed.

Once the requirements are documented, it’s important that these are fully reviewed by the user community before you move on to procuring software. This is likely to be an iterative process, but it’s critical that the requirements are understood and accepted before you proceed too far, in order to limit the potential for scope creep, where ‘new’ requirements extend the implementation phase.

■ Putting this sort of document together isn't a trivial exercise, but the payback from the time invested can be huge. Here are some of the key benefits of this sort of approach:

Increased return on investment – the focus on having well defined business objectives means that there's a much greater likelihood that the system will generate value to the business once it's implemented. Many CRM projects fail to achieve anything meaningful either because there are no business goals or that they're not fully articulated or understood.

Less white elephants – this emphasis on operational outcomes also means that investment tends to be focused on areas that make the biggest difference, and there's less likelihood of expenditure on unnecessary frills or expensive white elephants.

Improved functional fit – the approach of fully documenting how an organisation's business processes will be supported by the system is very effective at flushing out functional requirements, and tends to give a much greater understanding of what is and isn't needed. This helps avoid the risk of selecting a CRM technology that doesn't meet your needs or spending on unnecessary capabilities.

Reduced costs – the emphasis on process, and spelling out requirements in more detail, particularly in the areas of data migration and integration, means that prospective vendors can provide much firmer and more accurate pricing proposals. This avoids the common pitfall of having to make procurement decisions based on very loose estimates, which are only firmed up when the

selected vendor undertakes a more detailed, and often expensive, discovery phase – the outcome of which is invariably a hefty uplift in costs.

The cost reductions through using this approach can be significant. We generally expect to purchase a system 30-40% cheaper through tightly defining the details up front.

Increased implementation speed – the time spent spelling out requirements in detail in advance of purchasing CRM software, means that the project can progress a lot quicker once the technology is selected and the implementation partner is on board, because this potentially time-consuming phase is already done.

Improved project control – a clear, shared, vision of what the system looks like, means that there's much less likelihood of budget overruns or live date slippage through new requirements being discovered during the implementation phase.

■ The aim of this article was to explain why CRM requirements definition is so important, and how to go about it in a way that helps you smoothly implement a CRM system that makes a real difference to your organisation.

If you have any questions, please feel free to contact me at richard@mareeba.co.uk – I'm always happy to help if I can.

Finally, while the purpose of this paper is to give you the tools to undertake the requirements gathering process yourself, we've been managing requirements definition for clients for over ten years. If you'd like to discuss how Mareeba can help you through the process feel, free to email me at the above address, or through the [Mareeba 'Contact' page](#).



Richard Boardman is the founder of UK based independent CRM consulting firm, Mareeba (www.mareeba.co.uk). Mareeba works with organisations in both the public and private sector helping them buy and implement CRM technology in a way that creates genuine operational value. Richard is a poacher turned gamekeeper, having worked for a number of major UK CRM vendors, before becoming an independent CRM consultant in 2004. Richard has been involved in the CRM industry since 1995, and has been involved in over 300 CRM implementations.