



## WHY MANY CRM PROJECTS FAIL AND COST A FORTUNE

BY MALCOLM HAMER

### Introduction

Customer Relationship Management (CRM) systems are application software suites that support any part of a business that deals directly with customers (who are typically external customers, but may also be internal "customers"). CRM systems help to make the individuals who deal with those customers more effective. They do this by providing the user (that is, the person who is serving the customer) with some or all of the following:

- **Rapid access to basic information about the customer.** This is information that will be needed at the start of a conversation with the customer, for example, at the start of an incoming telephone call, or when initiating a telephone call to the customer. This information includes contact information (telephone number, address, email address, and so on), account information (account number or numbers, credit limits), and recent transactions (the subject of the last few interactions with the customer, open orders, recently completed orders, outstanding complaint investigations, and so on).
- **Access to in-depth information about the customer.** This is additional information that helps the user foster a closer relationship with the customer and which may be consulted when planning a face-to-face meeting with the customer or during the course of a telephone conversation, for example, the customer's complete order history, a profile of his or her company, and personal details such as birthday, name of spouse, and interests.
- **The ability to update any of this basic information and in-depth information.**
- **The ability to create a record of an interaction with the customer.** This interaction record may take the form of a log of a telephone call, a full Call Report of a face to face meeting, or notes on an in-depth telephone discussion. This record will become the lead item under "recent transactions" when the account is accessed next time.
- **The ability to move seamlessly between CRM screen displays and other applications.** Other applications include the organization's transactional applications (such as Order Entry or Account Opening). The user should be able to carry information over from the CRM screens to the other applications without re-keying, ideally with single button-click to copy multiple data fields (as opposed to the user having to use the Windows copy and paste function to copy each data field individually – which can cause errors such as missing the last character of a field or creating a field that starts with an unwanted space character).

- **The ability to open a "case" and to route/assign this case to another individual or workgroup.** A case may be an enquiry, a quotation request, or a trouble ticket. The CRM system may include basic mechanisms for routing cases, or sophisticated workflow management features. Such workflow-management features may (a) allow cases to be automatically given positions in workgroup queues based on priority level, (b) allow cases to be the subject of alarm messages sent to supervisors when they have remained open/unresolved for more than a certain time, or (c) allow case-handling events to automatically trigger other actions (such as generation of an email to the customer when the case is first opened or when it is closed/resolved).

Company-specific systems that perform some of these functions have existed for over 25 years, but were never referred to as "CRM systems". The term "CRM" emerged in the late 1990s along with the doctrine (preached by vendors such as Siebel, Remedy, Clarify, Kana, and others) that the way to create a CRM system for your organization is to buy a vendor's standard product and fine-tune it to your particular needs. Attracted to the idea that a CRM system would increase customer satisfaction, increase market share, and boost profits, many companies spent a lot of money on these standard products. In addition, they spent even larger amounts of money on the "fine-tuning" of the standard products to the needs (or at least, to the perceived needs) of the situation in which they were going to be used.

### **The Problem with "Out Of The Box"**

The fact is, no two customer-contact situations are the same, and what you get "out of the box" with these CRM products is highly unlikely to match your actual workflows, business methods, and business data model. Customization of the product may take months, or even years. This is where you will probably spend even more than you already paid for the product. Yet CRM salespeople tell potential customers (even today), "I can have you up and running in ten days, with the out-of-the-box system plus a small amount of customization."

CRM systems have been deployed with some success in Customer Service/Helpdesk situations. In these situations the workflows and business data are relatively easily understood. Also, it is easier to force the users to use the system by one or more means, such as taking away whatever system they were using before, tying in the system with the telephony infrastructure so that calls can only be answered by interacting with the CRM system, or making other tools (such as FAQ scripts) accessible only through navigating the CRM screens. However, CRM project success rates fall as you move from Customer Service desks to other customer-contact points, where the customer interactions are more under the control of the worker. Success rates drop to nearly zero at the Sales end of the customer-contact spectrum. Interestingly, it is on so-called Sales Force Automation (SFA) deployments of CRM systems that the CRM vendors have recently been concentrating their sales efforts, having saturated the other segments of the theoretical CRM market space.

### **Reasons for CRM Disappointments**

There are particular reasons why SFA-with-CRM has been, so far, a disappointment. The most important of these is that the successful deployment of the CRM system runs counter to the interests of the salespeople who would be its users. The company would like every salesperson to enter the contents of his or her Rolodex and the contents of his or her brain into the CRM system. This is so that, should the salesperson leave the company, another salesperson can effortlessly

take over their accounts using the information stored in the CRM system. The salesperson, by contrast, wants to be in a position to walk away from the company with his or her Rolodex (plus established customer relationships and a good understanding of those customers) and go to another employer. The salesperson believes that his or her value in the job market is as much a function of that accumulated knowledge as of his or her sales skills. So, the very last thing the salesperson wants to do is type all that precious knowledge into a CRM system. Doing so would render him or her immediately dispensable.

There are other reasons why SFA adoption rates are so low, including the fact that using the system does not fit in with the salesperson's work pattern (moving around, calling on customers, "networking" activities that are not easily logged against a particular target customer, and so on). Requiring a salesperson to spend an hour typing in call reports and other updates on a laptop, at home, after a tiring day of driving from customer to customer, does not make for quality input – or indeed any input at all in most cases.

While the "Sales" end of the CRM deployment spectrum represents a startling worst case, all CRM projects run a risk of failure – a risk that is considerably higher than the risk of failure in many other types of software based project. This is why CRM products, along with other fashionable large-scale software products, are sometimes referred to as "shelfware" (that is, they are ordered, paid for, tried out, but ultimately end up sitting on a shelf in the IT department).

## The Major Pitfalls

The following are the four major pitfalls of CRM deployment projects:

- **Failure to understand the needs of the end user and get user buy-in to the project.** This is, of course, the biggest cause of IT failures generally, but it occurs more often in CRM projects than other projects of comparable complexity. Typically, a CRM system sale is made high up in the organization, sometimes as high as the CEO. By the time the IT staff and the vendor's staff (and possibly also a bunch of consultants) descend on the area where the system is to be deployed, there is a lot of pressure on the implementation team. When end users or their supervisors are shown the "out of the box" system demo, they immediately see huge differences between how their current customer interactions take place and the vendor's default workflow, screen layouts, terminology, and so on; and they start to ask for extensive customization. The implementation team may read this as "resistance to the new system" and ignore it, or they may see that the required degree of customization will blow the project budget, timetable, or both. Soon there is tension, reduced consultation, and eventually a "they'll take what we build and like it" attitude sets in among the implementation team.
  - **How to avoid this pitfall:** Start the CRM project with a thorough study of current workflows, extensive interviews with end users, documentation of functional requirements, prototyping, and user sign-off, before conversations start with vendors, before RFPs are written, and before a vendor is selected.
- **Failure to understand the business data model.** The success of a CRM system is more strongly tied to the way it handles data than the way it handles workflows. The out-of-the-box CRM system comes with a database. That database has a design. That design derives from a data model. It's not your data model. It is either the data model of the vendor's first customer, where the CRM system that they wrote from scratch for that customer later became a generic product, or it is a model based on the "average" characteristics of a number of sample

companies. If you take this database design and try to "tweak" it, you are unlikely to end up with the right database design. As a result, the system will not deliver what its users need to do their jobs.

- **How to avoid this pitfall:** At the start of the project, document the business data model (or at least the part of the complete model that applies to the customer-contact area where the CRM system is to be deployed). Then make sure that the database design (or "schema") that derives from this data model becomes the database design on which the design of the CRM system is based.
- **Failure to integrate the CRM system with the master corporate database.** Even if the database design of the CRM system is based on your true business data model, and there is an exact one-to-one correspondence between the tables used by the CRM system and the tables in your organization's master corporate database, there can still be problems. If you set up the CRM system with its own, separate "mirror" copy of the corporate database, you are asking for trouble. "We can easily write some simple database synchronization utility for you" is one of the Three Big CRM Lies. (The other two are "I can have you up and running in ten days with the out-of-the-box system plus a small amount of customization" and "We can add more tables to the database design, and fine-tune the workflows, once you are live and have some operating experience.") The vendor will have a dozen reasons why the CRM system needs its own, private database (such as "our latest version of our product only runs on Oracle9i and your corporate database is two releases behind"). If you settle for a "mirror" database for the CRM system (plus the database synchronization utility), the CRM users will find themselves battling with inconsistent data, delayed updates, and complete loss of synchronization from time to time, when changes are made to either the corporate database design or the CRM database design.
  - **How to avoid this pitfall:** Make sure that the design of the system centers on the existing corporate database. The CRM system must interact directly with this database, using standard SQL queries and updates, and sharing stored procedures with other systems where appropriate. It should not have its own dedicated database. After all, the fact that the CRM product uses the same database management system (DBMS) that your organization is already using – for instance, Oracle, Sybase, or MS SQL Server – was almost certainly used in the sales pitch to Senior Management. Do not let the vendor off the hook on this issue.
- **Failure to add value.** Even if the system accurately reflects current or desired workflows, terminology, and business data models, it may still face adoption issues where its use is not an absolute prerequisite for handling customer contact; and it may therefore fail to improve the customers' perception of the organization or generate increased business. Just automating things does not necessarily make them better. There was an excellent article by Malcolm Gladwell in *The New Yorker*, March 25, 2002, called "The Social Life of Paper", in which he wrote about how notes and other documents are used by workers day to day (as opposed to paper records that are filed away and rarely, if ever, looked at again). He said that these day-to-day notes and documents facilitate the cognitive processes involved in dealing with knowledge-based tasks. He quoted from "The Myth of the Paperless Office" by Sellen and Harper, describing how a group of buyers (who might equally have been Sales staff) used the documents that they kept at their desks: "These materials therefore *supported*, rather than *constituted*, the expertise of the buyers." Two points emerge from this article:
  - (a) The knowledge and understanding that are required to perform a job are much more than the information that can be captured in a filing-system-style of document. So, if you expect someone to sit at a CRM terminal and read a predecessor's call reports and other notes,

and then be able to seamlessly take over a set of customer relationships, you may be expecting the impossible.

- (b) When you take away the paper documents from the worker (by requiring all data manipulation to be done via a CRM screen), you take away from his or her cognitive processes something important and useful.

In other words, because CRM implementations tend to replace paper with electronic information storage, a CRM system may give the users *less* value than the original manual process.

➤ **How to avoid this pitfall:** To counteract the downside of removing paper from a process, design the CRM system to add new value to the process by leveraging the strengths of a system that stores information electronically. For example:

- When a customer's name (or account number) is entered into the CRM system (for instance, when a call is received from the customer), the first screen that is displayed should include a panel with the last few (say, ten) interactions with the customer – including interactions with *other* customer-contact areas in your organization. (This is where a CRM system really can add value; and this is another reason why it is important that the system should be working off the live master corporate database.) If the Sales Department user can start the call with "Good afternoon, Mr Smith. I see that you called our Helpdesk this morning with a query about...", then the customer will believe that your organization has its act together.
- Another useful screen-pop is "It's Mr Smith's birthday today". It makes a big impression when someone outside your circle of family, friends, and co-workers unexpectedly wishes you "happy birthday". (I can clearly remember the three times it happened to me: once was at Passport Control at Hong Kong airport; once was on a Singapore Airlines flight from Singapore to San Francisco; and the third time was at my local health club last year.)

Unfortunately, opportunities like these to leverage the strengths of an automated system are often missed during CRM system design, leaving users with the downside of losing their paper-based processes, and no upside. Make sure that you find ways to add new value to the process, helping the user to do his or her job more effectively, rather than thinking of the system purely as a tool for increasing sales and reducing costs.

## Reducing The Risk Of Failure

The above pitfalls are by no means the only ones, but they are the big ones. They are much less likely to occur if the CRM project as a whole is properly structured. Because they are so risky, CRM projects, more so than many other software projects, need to be organized along traditional System Development Life Cycle (SDLC) lines:

- Start by listening to the views and ideas of the intended users. Encourage them to think of ways in which an automated system could give them things which they do not get from their present processes or systems.
- Then proceed through documentation of the functional requirements, to the design of the complete solution of which the vendor's CRM platform will be a part.

- In documenting the functional requirements, include the most critical element – the data model for the part of the enterprise covered by the CRM solution.
- Use the data model to create a database design that includes existing tables in the corporate database, plus new tables that will need to be added to support the CRM system itself (for example, tables covering cases, call reports, and customer information not currently captured).
- Only after the above steps have been completed should an RFP be written (to support the vendor selection process) and conversations initiated with potential CRM platform vendors.
- In the RFP, and in all dealings with the vendors, make it clear that the chosen vendor will be supplying a tool that will form part of the CRM solution. The vendor is not bringing "the solution" to your organization with their product. The vendor must play by your rules.
- In the RFP, and in the subsequent contract, clearly state that the vendor's platform must interact directly with your existing corporate database that holds the existing data about customers and their accounts/transactions/orders; and that failure to deliver a system that does this will represent a breach of the contract.
- If you want to prevent vendors' sales staff calling on your EVP or CEO (a common occurrence during an RFP process, particularly with certain CRM vendors), state in the RFP that any vendor contacting anyone in your organization except those named in the RFP will be automatically dismissed from the selection process.

Most important of all, at every stage in the project, make the users feel that they have a big say in the direction of the project: don't let the project become a "Mission From God".

© Malcolm Hamer, I.S.F., October 2001, updated March 2002