

What makes for CRM system success – or failure? Bryan Foss and Professor Merlin Stone

For years, we've known that change programmes involving a big change to systems, or even totally new systems, involved additional risk. It's no surprise – the more the change, the greater the risk. CRM systems success and failure have attracted a lot of publicity in recent years, but in the last quarter century, there have been many other management areas where the issue of systems success and failure has attracted the same interest. Manufacturing, logistics, store/branch operations and the public sector have all produced great examples – of both success and failure. The reasons for success and failure turn out to be pretty closely related. Successful systems-supported change projects observe change management disciplines, unsuccessful ones don't. Successful projects are planned carefully, with all aspects covered, all the right people involved, and the company and any external suppliers (consultancies, systems companies, business partners) work well as a team. The opposite applies to failed projects. In the middle lie the many partly successful projects that achieve some but not all of their objectives, probably at a higher cost than initially planned.

One of the central themes of most studies of success or failure in systems development and implementation is whether the project slipped badly, so that it was completed well after the original deadlines. However, in our experience, CRM systems seemed to be different, in that they usually support a large change in how a company works with its customers, a change that involves learning by the company's staff, its system suppliers, the consultants (if any – and internal or external) helping plan and manage the projects, the company's business partners (e.g. marketing communications agencies, call centre capacity suppliers, fulfilment houses) and not least, its customers – as the company tries to create a new way of working with them, and sometimes even a relationship. These CRM systems often had several stages of development, with one or two years of pure development, followed by several years of implementation and iterative improvement. So we decided to take a closer look – prompted by the fact that one of the authors (Bryan) has agreed to give a talk on the topic to the other's (Merlin's) Customer Management Group, a group of clients, suppliers and independents that meets about 4 times a year to discuss similar issues.

We approached about 90 of our contacts we considered had direct and personal experience of CRM developments and deployments in the last ten years. They included supplier-side contacts, with knowledge of many client projects, but also client-side contacts personally involved in one or more CRM projects during that period. Some of the latter had moved jobs and so experienced more than one CRM implementation – particularly as they are likely to have been hired across from one company to another because of their valuable experience. We received nearly 40 responses, showing the accuracy of our targeting and the responsiveness of our contacts. Here's what we found.

Different industries and users have different priorities and varying amounts of funding to invest in CRM. Companies with very tight profit margins or limited funds tend to partner where they can (e.g. shared loyalty schemes, outsourced database management and call centres) and to use small 'pay as you go' development stages, while keeping a consistent vision and objective in mind. Many of the largest scale projects were in the financial services industry in the early years of this century. Banks, insurers and others usually used packages such as Siebel and Chordiant at the heart of their project, but some developed their own systems. These companies were moving – or had already moved - from managing customers mainly in the branch or direct mail, to managing them in the contact centre and over the Web. Here, it's arguable that the main reason for the investment was to cut the cost of managing customers, and to enable the company to move into new product markets very quickly (e.g. from banking to insurance or vice versa). At the same time, telecommunications companies were investing in large CRM projects. Some did it as a "big bang" replacement for an existing system which was focused entirely on subscriber management, including billing. Their new systems enabled them to manage churn and up/cross-sell much better. In telecommunications, particularly in mobile telephony and now broadband, reducing churn is a critical target, while increasing customer value by up-sell and cross-sell is the "cream on the cake".

These companies had very different levels of success. Their success was determined mainly by the relationship between the complexity of the system and the speed and phasing of its development and roll out. A CRM system is not just customer interface software. The functions commonly found in CRM projects include:

- Contact centre for sales and/or for customer administration and service (and in some cases this will include with some self-service via IVR or voice recognition, although some projects were focused on adding this to an existing contact centre)

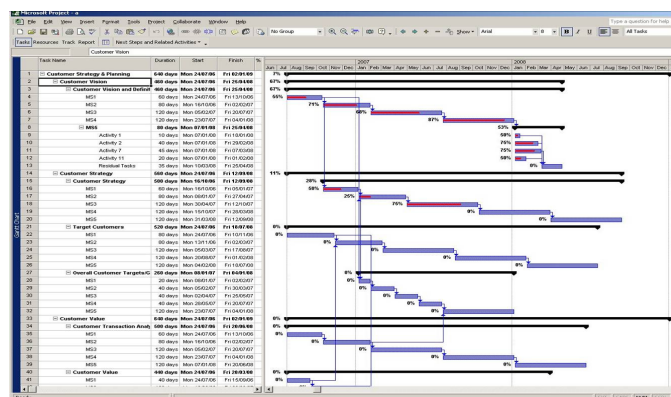
- Adding additional capacity e.g. a new contact centre, sometimes but not always based on an existing system
- Web-based customer management (which implies a move to more of a self-service model)
- Integration of customer management across several channels e.g. branch, contact centre, Web, back-office
- Customer database development
- Data warehouse or data mart – mainly for analysis and planning
- Marketing campaign and resource management
- Integration of any existing CRM system with core/legacy systems
- Support of mobile or field, sales or service, staff

So, when someone talks about “their CRM project”, they could mean any or all of the above. Each function usually demands a big additional development, customisation and/or integration effort, depending upon the choice of software.

Interestingly, the number of users is not by itself a major factor in the cost of development, unless there are different types of user with different business requirements e.g. field sales versus contact centre sales. The number of users usually affects cost mainly through the number of software licences needed and recruitment and training costs.

However, complexity and scale both affect the time it takes to develop and roll out a CRM system of given functionality. “Big bang” approaches, the creating of large scale projects with high levels of functionality, are not uncommon. However, although this approach might seem risky, it can be greatly de-risked by not tightly coupling too many elements of the programme. For example, a company which wants both to improve dramatically the quality and scope of its customer database and the quality of interaction with customers in the contact centre does not have to premise its investment on using the new data through a new customer interaction system. Some or all of the new data can be planned to be used with existing customer interaction system, even if full exploitation of the data has to wait until the customer interaction system is completed. So, although we know that in information technology projects, project risk increases with project size, the risk can be reduced by having several interlocking projects where attention has been paid to developing a “Plan B” for deployment of whatever has been successfully developed even if there are slips in other parts of the programme.

However, many clients also reduce risk by staging the development of any particular functionality or group of functionalities, so that benefits begin to be delivered early on, and continue to be developed from deployment experience. This also reduces the political risk of such programmes. For this reason, many suppliers of CRM systems and consultancy have created CRM programme planning and deployment methodologies that support accelerated early development stages. At the heart of such methodologies must be a strong CRM programme planning methodology that is understood by senior management as being the central focus of their governance for their CRM improvement efforts, however this is where almost all methodologies prove to be weak – usually with disastrous consequences for the success of the programme. The figure below, from Customer Essential, illustrates such an approach being rigorously and successfully applied.



The risk of a big bang approach is that all the requirements for development changes are gathered at the beginning of the project and applied in one continuous development period, before ‘waterfall’ testing takes place and subsequently deployment. The waterfall model is a sequential process for developing software in which development flows steadily step by step, like a series of small waterfalls, through requirements analysis, design, development and integration, testing, implementation and maintenance. Although there are many

arguments about the pros and cons of following this model, it seems that large scale projects that use this approach tend to have a high risk of failure, longer time to benefit and are also less adaptable to continuous learning during the development and deployment process. As a result the development effort invested in some functions may be wasted. The waterfall model is associated with another systems development idea – that of Big Design Up Front, in which system design is agreed early on in the project so that the systems development effort can focus consistently on the required design. Of course, there is no perfect model for systems development. Indeed, the waterfall method is often criticised because in the real world requirements do change, as systems frequently need design changes before their development is finished. As a result it is usually not possible to have smooth, step by step progress from design to implementation. Instead an iterative approach often proves much more effective at delivering appropriate business value in stages and reducing programme risk. We believe that companies which think hard about the options for their CRM programme phasing and systems development methodology and consider them as related aspects of the same task are more likely to get the CRM systems they want.

The appropriateness of the software chosen to implement a particular functionality is also important. Some packages are suitable for implementation “out of the box”, provided that the company is prepared to change how it works to fit with what the software designers conceived. Others require significant customisation or development. Of equal importance is the choice of systems integrator (unless the work is done in-house). Systems integrators with good track records of success (and this covers not just development, including experience with integrating the particular software products chosen, but also implementation and subsequent support) are more likely to deliver a successful outcome than those with questionable records in this area.

We asked our respondents how long it took to complete their CRM programme. Given what we have said above about the content of a programme, it is no surprise to see that the answers varied dramatically according to this. Some respondents gave examples of initial development periods as short as 16 weeks from decision. Many others were developed within a year from decision date. This speed was usually achieved because the CRM programme was defined as development of a single function (e.g. marketing campaign management), with some integration to existing systems and data sources (even though this might effectively be the first stage of a multi-stage CRM programme). In other cases, it was achieved because the programme consisted of minimal customisation of an existing CRM package with very limited integration with existing systems or data sources.

Projects that took between two and four years involved multi-function developments. These are the majority of what we would call “comprehensive” CRM programmes. However, when we analysed the responses we received, we saw signs that these programmes are now being completed a year or so more quickly than they used to be. We think this is because of the increased maturity of CRM software and the wider availability of CRM services, skills and proven development methodologies.

Projects that took five or six years appear to be on the one hand very large projects or where investments are deliberately stretched out over a long period, or on the other hand what are known as ‘troubled’ projects, which run well over time and budget compared to initial investment estimates.

In addition to the above timescales, we also identified programmes which had no specific scope or end date. This is not because they were poorly planned, but rather because the company is on a long “CRM journey” and committed to making a series of improvements to its CRM capability as changes take place in its distribution channel strategy, customer base, competition and the like, as customer management technology changes, and as the company learns what works and as their customer learn to be responsive (or not!) to the new customer management approach.

In the end, the most important question is – did the system work? The Gartner Group researched whether CRM programmes met expectations and identified that just over half were considered unsuccessful, though many delivered substantial benefits. In Gartner’s view, too many companies failed to set clear objectives or to recognise the business change needed and often considered the CRM programme to be just a systems programme. We agree with this conclusion, as did many of our respondents.

About the authors

Bryan Foss is an experienced CRM and board advisor. He worked with IBM for many years and now provides independent advice for many large organisations on systems implementation and related change issues.

Professor Stone is a leading author and advisor on CRM programme management and implementation, He is Professor of Marketing at Bristol Business School and a Director of Nowell Stone Ltd and The Database Group.