

CHAPTER NINE

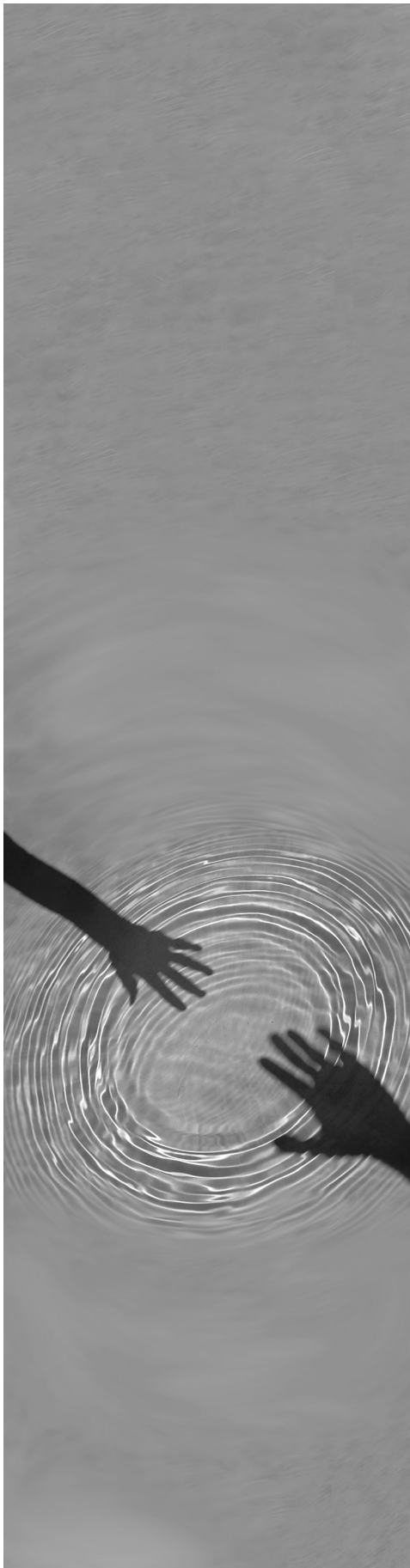
Managing Your CRM Project

These days it's practically routine to pick up an industry trade magazine featuring a CRM case study on page 1. Somewhere amidst the paragraph about the company's new customer loyalty program and the part about sales uplift increasing 200 percent, you'll find a sentence or two describing implementation.

No, CRM development isn't sexy, and yes, it's fraught with hazards from technology glitches to hiring freezes, but it's the hub in the CRM wheel when it comes to ensuring a smooth rollout. The snazziest end-user interface and most enthusiastic marketing staff will never compensate for the CRM system that doesn't do what it's supposed to. Not to put too fine a point on it, the implementation project is a critical piece of the CRM puzzle.

A Pre-Implementation Checklist

I spend most of my time these days evaluating how prepared companies are to launch their CRM programs, be they departmental or enterprise-wide, single or multifunction. Sometimes this occurs at the requirements definition stage, where there is uncertainty about the perceived need and its implementation viability. Other times it involves evaluating a company's existing infrastructure just prior to implementation. What I do most is quiz key CRM stakeholders about their existing environment from both business and technology perspectives.



My company calls such evaluations CRM Readiness Assessment engagements, but I like to consider them “premortems.” After all, what’s more valuable than fixing problems before they occur? The best way to do this is to envision possible outcomes based on current circumstances, using experiences gleaned from successful CRM deployments. It’s good old risk management, come home to roost.

Table 9-1 lists a series of considerations to be aware of before moving forward with CRM development. Make sure each of these items has been at least considered at your company, and the more complex your intended CRM program, per Table 9-1, the more mandatory it is that you resolve the issue prior to beginning development.

Table 9-1: CRM Pre-Implementation Checklist

<i>Evaluation Question</i>	<i>Explanation</i>	<i>Considered?</i>
Have you prepared a CRM business plan?	We discussed CRM business planning in Chapter 7. Regardless of whether management requires such a document, it’s a very good idea to have one that represents CRM’s baseline.	✓
Do you know who your executive sponsor is and what she expects?	By the time you’re ready to launch development, the CRM executive sponsor should be crystal clear. Moreover, her role in defining and validating requirements, managing executive expectations, and helping define success metrics should be well understood by all stakeholders.	
Have high-level business requirements been defined?	In CRM this activity should be separate from the formal development project for two reasons: business requirements will dictate whether the CRM program moves forward, and they require involvement from stakeholders who might not be available during implementation.	

A Pre-Implementation Checklist

235

<i>Evaluation Question</i>	<i>Explanation</i>	<i>Considered?</i>
Have success metrics been established?	How will you know if your CRM program has been a success? Although many companies don't require success metrics—like those we discussed in Chapter 7—to be implemented, they're an effective safety net for after the system is deployed.	
Has the project been funded?	No use planning an entire CRM program if only a mere proof-of-concept has been approved.	
Is there agreement on desired customer behaviors? Are the business functions slated to support these desired behaviors apparent?	Depending on the scope of your CRM program, you might include a description of desired customer behavior in your CRM business plan. Either way, building consensus on how you want customers to behave differently is important. For instance, if sales staff will be using CRM to manage the sales pipeline, it should establish the ideal response to an information mailing.	
Does each organization agree on a common definition of "customer"?	The marketing department of an automobile company might consider a "customer" to be a dealer, but the call center might consider it to be a driver. Have consensus on this and other key definitions before you begin.	
Can you map the desired functionality to data requirements?	Customer data is complex more often than it's straightforward. This usually means defining data requirements along with business requirements. At some point you'll need to know whether customer data is necessary and from what system it will originate. A firm	

(continued)

Table 9-1: CRM Pre-Implementation Checklist (continued)

<i>Evaluation Question</i>	<i>Explanation</i>	<i>Considered?</i>
	understanding of the level of customer data—account, household ¹ —is also critical.	
Do you suspect that external data will be necessary?	Purchasing data from an external source such as Dun & Bradstreet, Axcion, Data Quick, or Experian might not initially be a high priority, but it can supplement customer profiles with such indicators as number of family members, estimated income, household-level psychographics, ZIP code breakdowns, real estate information, and other attributes that can reveal customer behaviors and preferences.	
For customization, does the current workstation development environment support the CRM product?	What type of workstation configurations does your CRM tool's development environment require? Additional development tools (e.g., Microsoft's Visual Studio) or hardware (e.g., database servers) might be necessary to correctly customize the CRM environment.	
Have you identified the other applications or systems with which the CRM product must integrate?	There should be an up-front understanding of the impact of CRM on other corporate systems and of how the data will move between systems effectively. In addition, staff members whose systems will be touched by CRM should be notified of the pending integration requirements.	

1. The practice of "householding" organizes individual consumers into the households in which they live. Although the term normally applies to the residential market, business householding groups various organizations of a business customer into a common hierarchy. The challenge of householding is getting everyone to agree on the definition of a household.

<i>Evaluation Question</i>	<i>Explanation</i>	<i>Considered?</i>
Have the organizational or political barriers to rolling out CRM been identified? Have they been resolved?	Yes, it's a loaded question. (See the end of this chapter.) No, it's not meant to point fingers, but to establish up-front what the tactics will be when questions of ownership or disagreements about functional priorities rear their heads. An influential executive sponsor might be able to resolve such issues before they arise.	
Have you truly defined your privacy policy?	Regardless of whether your CRM program will be Web-based, understand your company's boundaries for using data about your customers. CRM must not only adhere to a corporate privacy policy; it should also be the flagship example of the company's <i>behavior</i> around customer data. See Chapter 10 for more about handling privacy.	

The most valuable feature of a “premortem” exercise is that it’s a lot easier to give bad news before disaster strikes than to say “I told you so” after the fact—and after the money has been spent. CRM assessment findings can alert the business sponsor to potential roadblocks. Such findings allow CRM team members to fix problems proactively rather than pointing fingers after the CRM project has failed, as 70 percent of all CRM projects allegedly do.

Ideally, the answer to each of the above questions will be “yes,” with consensus on how each issue will be handled when it’s encountered. At the very least, the CRM team should be aware of each issue and prepared to deal with it when it inevitably comes up.

The CRM Development Team

CRM is big. It has captured the attention and imagination of corporate executives. Marketing VPs are betting their jobs on it, CIOs are asking their

staffs to formulate CRM policies, and CEOs are creating job roles such as “Chief Customer Officer” that not only embrace CRM but depend on it.

Hopefully by now your company has adopted a customer-focused strategy and is putting in place the inevitable customer-focused programs and accompanying organizations. This often means organizational change: product managers have become “segment managers,” spearheading customer segments irrespective of the products and services within them, and CSR job definitions are being continually modified as companies better understand customer channel usage and interaction preferences.

In addition to the broader organizational and cultural changes that accompany an evolving customer focus, CRM calls for specific implementation roles and responsibilities. In many cases, these job roles are new; in others, existing functions play key parts in CRM development.

Table 9-2 lists the core job functions within a CRM development team. Make sure you’ve accounted for each of these roles before embarking on a development project, and understand the skills from both inside and outside the company might be necessary to fill these positions.

Table 9-2: Core CRM Development Roles

<i>Job Role</i>	<i>Description</i>
Business Sponsor:	The business sponsor might serve across a single CRM project or across the entire program. His main role is to establish the vision, articulate overall goals and objectives, set the tone for the project team, and serve as a tiebreaker for implementation issues. The business sponsor often funds the initial CRM application. The more departments CRM spans, the greater the level of authority the sponsor should have.
CRM Steering Committee:	For cross-functional or enterprise CRM initiatives where implementation must be prioritized, a committee of decision-makers familiar with the “pain points” CRM can address should convene on a regular basis to provide new requirements, prioritize proposed improvements, and communicate key corporate initiatives.

Job Role	Description
Implementation Project Manager:	This person's job is to ensure that the requirements defined by the business sponsor and steering committee dictate the functionality to be implemented. The implementation project manager oversees the day-to-day implementation activities, tracks status, and updates the business sponsor on current issues.
Lead Developer:	The lead developer should manage the technical development and customization of the CRM product as it relates to the requirements. She should participate in CRM technology selection (see Chapter 8) and hire the appropriate developers to implement the CRM toolset.
Database Developer (and team):	The database developer should lead the necessary data integration, regardless of whether it is operational or analytical CRM. Often this means working with the company's data warehouse and its development team. In other cases, an understanding of key company source systems and how to capture their data is mandatory, requiring a separate team of database administrators and data "extraction" experts.
Front-end Developer (and team):	Depending on the chosen CRM product, programming is needed to develop or customize the end-user interface.
Subject matter experts (SMEs):	Critical to CRM success are subject matter experts—usually businesspeople from the department slated to use the CRM system after it's in production (for instance, a CSR or a sales manager). SMEs usually have strong ideas of what CRM should and shouldn't provide and should participate regularly in the development and testing of a CRM product.

Depending on the breadth and complexity of your CRM program, the job roles listed in Table 9-3 can also participate in CRM development.

Table 9-3: Optional CRM Development Roles

<i>Optional Job Role</i>	<i>Description</i>
Director of e-Business:	Your company might have a separate division dedicated to e-business that—despite the goals of CRM—must be involved to ensure the integration of, for instance, Web-based customer services with new CRM functionality.
Director of Data Warehousing:	If your company already has a data warehouse, you're ahead of the game. Existing data, development processes, source system knowledge, and metadata can all be used to get a jump-start on CRM development. Development teams might consider sharing resources in order to integrate the data warehouse as the de facto CRM analysis platform.
Chief Information Officer (CIO):	Due to the strategic nature of many CRM initiatives, it's politically if not technically wise to get approval and visibility from the CIO, who can usually facilitate activity with the IT department to ensure the appropriate systems and data resources. The CIO can also help socialize CRM as a corporate information resource.
Vice President of Strategic Planning:	In large companies, where this position exists, the Vice President of Strategic Planning should be able to share with the CRM team new business areas or product offerings the company expects to move toward, acquisition and partnership strategies, or existing products and services the company expects to abandon.
Chief Privacy Officer:	A new position in most companies, the Chief Privacy Officer should be able to provide details on corporate or regulatory policies regarding the use of customer data.

Each of these job roles can play an important part in CRM success, but simply understanding available skill sets can take you a long way in ensuring you can supplement your CRM team with outside help if necessary. Of course, such responsibilities as executive sponsor and the CRM steering committee should be filled by staff members having history with the CRM-related need, pain, or problem, as well as the authority to make decisions.

There are roles in CRM, however, particularly in technology implementation areas, where external experts should be leveraged. Consider the following questions as you decide whether to beef up your current staff with outside help:

- *How well do we know the CRM vendor's development environment?* It might serve you well to bring in an expert from the vendor's professional services staff or from a partner-integrator to provide knowledge transfer as development gets underway.
- *Are there critical one-time-only tasks that need completion?* For work that isn't likely to be repeated, such as configuring the data, a good consultant can shave days or even weeks off a project.
- *Are we comfortable that our requirements are well defined?* Sometimes an objective third party can find the "holes" in your requirements definition. This can help you avoid false starts—which could be a bargain at twice the price.
- *Can we get started with our existing staff?* It's often true that by the time you hire and train a full-time resource, a consultant could have jump-started a critical task and the entire project could be that much farther along. Everyone would rather hire permanent staff members who have skin in the game, but don't let principle usurp progress. Be willing to focus on the value of time to the business, and invest accordingly. This might mean hiring consultants who can hit the ground running.

Another important consideration in CRM staffing is the existence of a corporate program management office (PMO). First made popular by the aerospace industry, where complex projects were the norm, the PMO deconstructed a multifaceted system into manageable chunks involving repeatable tasks such as requirements definition, software coding, design, testing, validation, and software packaging. Each project chunk had its own project manager, goals, budget, and deliverable. Usually stationed in the IT organization, the PMO is dedicated to running corporate programs such as CRM so project teams can concentrate on succinct deliverables while the PMO ties them all together.

CRM programs are business integration projects whose scope is often corporate-wide (similar to enterprise ERP or supply chain management initiatives). Because CRM is driven by business requirements and involves the integration of business processes with technology and data, its complexity and organizational reach is usually greater than the straightforward application. The PMO formalizes tried-and-true practices that can be applied to CRM implementation. This

not only ensures consistency across projects in a program but can also provide consolidated status reporting, often to executives, affording a level of visibility CRM could never otherwise generate.

CRM Implementation

As we discussed at the end of Chapter 7, CRM is usually a corporate program made up of many projects. For CRM point solutions that deliver finite functionality, one well-run project might be enough. Each CRM project should focus on implementing at least one defined requirement. Whatever the complexity, CRM development should be evolutionary and multi-tiered. Figure 9-1 describes a departmental CRM program and its associated requirements.

Understanding the complexity of your CRM program is critical to planning your CRM project. For instance, if CRM is an enterprise initiative, there could be dozens or even hundreds of discrete requirements across the corporation, rendering project-planning orders vastly more complex. If, as in Figure 9-1, the program is departmental, each requirement will eventually be deconstructed into a number of different functions, revealing its inherent complexity and the development resources it will require.

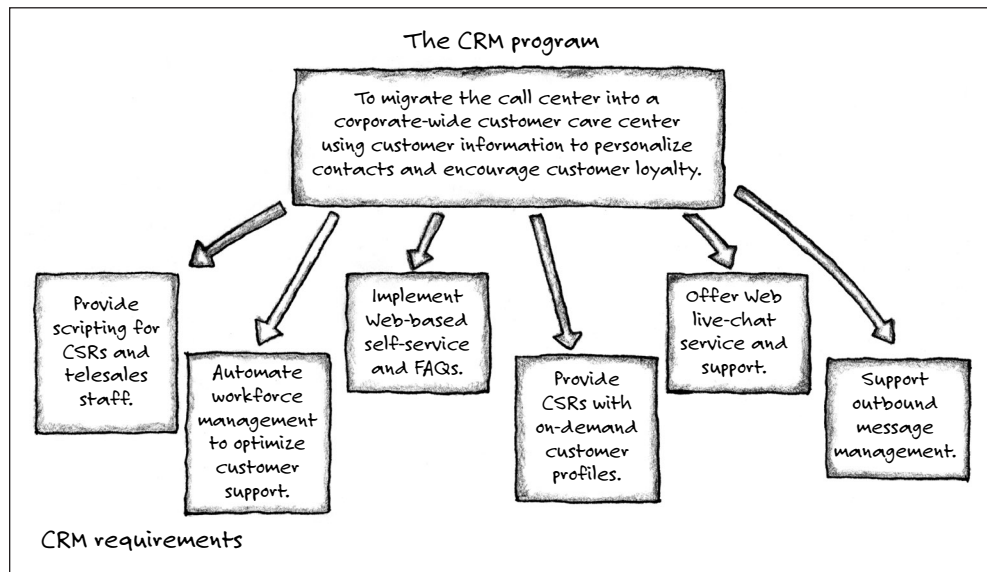


Figure 9-1: CRM program and requirements

Scoping and Prioritizing CRM Projects

Biting off all the requirements listed in Figure 9-1 would not only be dangerous; it could sabotage a company's entire CRM initiative. After you list your CRM requirements and have a good idea of their required functionality, the CRM business sponsor or steering committee can actually cast them into discrete projects.

Surprisingly, many CRM sponsors and project leaders forget this step and move straight toward trying to deliver the sum of all listed requirements in one fell swoop. Without scoping and prioritizing CRM projects, project managers lack overarching direction for prioritizing development activities, and application developers are free to arbitrarily add and change functionality during development. The results are usually disastrous. A scoping activity ensures that CRM projects are defined based on discrete requirements and are circumscribed around delivery expectations.

Requirements can evolve into individual projects based on demand urgency or perceived value or based on implementation complexity.

In the case of demand urgency, the customer support department might be overburdened. Thus the requirements pictured in Figure 9-1 might be prioritized in the following way:

1. Implement Web-based self-service and FAQs.
2. Offer Web live-chat service and support.
3. Support outbound message management.
4. Automate workforce management to optimize customer support.
5. Provide CSRs with on-demand customer profiles using existing data.
6. Provide scripting for CSRs and telesales staff.

If, on the other hand, implementation complexity is an issue, and the company needs a CRM "quick win," the following prioritization might make more sense:

1. Provide CSRs with on-demand customer profiles using existing data.
2. Automate workforce management to optimize customer support.
3. Provide scripting for CSRs and telesales staff.
4. Implement Web-based self-service and FAQs.
5. Support outbound message management.
6. Offer Web live-chat service and support.

Of course, politics figures into the decision on how to prioritize CRM projects. After all, if your customer-support vice president and call-center director

are fighting over whether external data is necessary for really understanding customers, you might want to steer clear of providing CSRs with customer profiles until the issue is resolved—no matter how happy it would make the CSRs. Although formally rating the political landmines of every project could be overkill—not to mention highly subjective—knowing the political baggage that accompanies each potential project can serve as a tiebreaker.

When prioritized, a CRM requirement—or specific sets of related requirements—can be defined as an individual CRM project as shown in Figure 9-2.

Notice that in Figure 9-2 the Web-related development has been grouped into one project. This decision was based on practical reasons—specifically, the ongoing challenge of finding available Web-development staff within the company—as well as the estimated development complexity. Projects 1, 2, and 3 are all minimally related and can each leverage existing technologies and skill sets within the company.

Who should scope a CRM project? Ideally, business representatives and development staff should discuss each requirement and estimate its value-to-complexity ratio—the higher the value and the lower the complexity, the better—with the goal of prioritizing delivery on an ongoing basis. Most CRM scoping activities focus on delivering initial applications in order to hand over

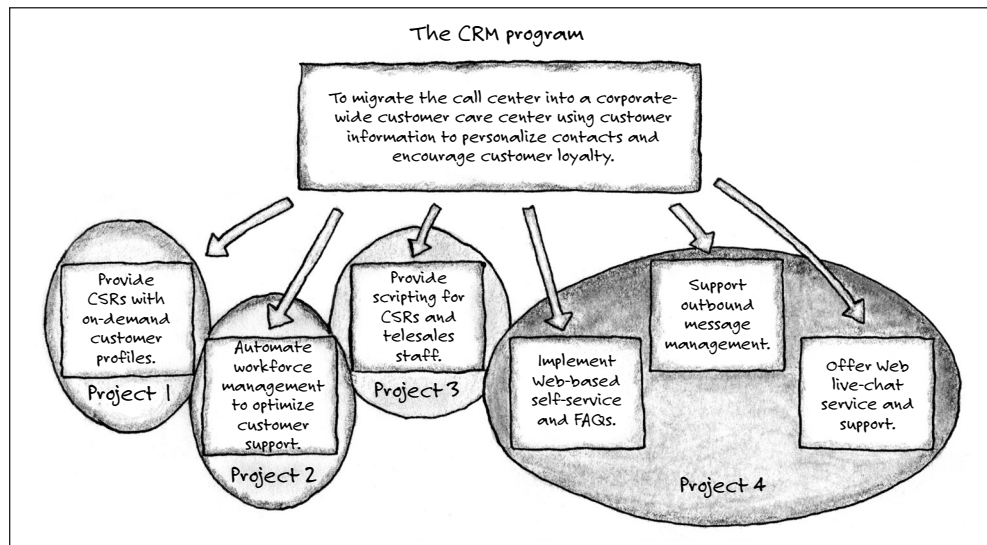


Figure 9-2: Delineating CRM projects

a “quick win” to the business. Applications with a high value-to-complexity ratio should rise to the top, and others can be prioritized accordingly.

The complexity metrics will vary according to the availability of your company’s existing technology and staff resources. For instance, companies that already have robust customer databases won’t rate customer profiling to be as complex as those who must start from scratch.

To correctly scope a project, simply rating its functional complexity is not enough. Ideally, you should understand the following:

- Specific technologies that will be involved in implementation
- Necessary skills to implement the project
- Number of staff members projected to work on the project
- Number of consultants needed to supplement in-house skills
- Realistic time frame necessary to deliver the first release
- Organizational boundaries and potential political issues

Scoping a CRM project prior to launching development mitigates the risks. For one thing, it’s much easier to develop an accurate project plan that reflects realistic resource requirements, tasks, and time frames. Justifying headcount requests to management based on the project’s true scope is also easier. Finally, hiring becomes more straightforward, because the true skills necessary to develop the CRM system are clearer than they would have been if you had simply gone straight to implementation. In fact, failure to thoroughly scope IT projects is one of the principal reasons behind many of their failures.

A CRM Implementation Roadmap

Even with the most straightforward CRM products, there’s no such thing as cookie-cutter CRM. Development approaches can differ according to a company’s approved development lifecycle, staff expertise, and IT standards.

Despite the possible differences in CRM implementation techniques, the following proven CRM development success metrics should define every CRM development project:

- *Incremental development.* Incremental or “building block” development means the company receives a defined amount of new CRM functionality on a regular basis. This is due not only to the inherent complexity of most CRM projects but also to the cultural issues surrounding its deployment (few organizations can absorb multiple major functional and process

changes at once). Incremental CRM “releases” create a perception among business stakeholders and management of ongoing value. The alternative to incremental development is the “big-bang” approach of delivering a major new system and accompanying business changes all at once. The big-bang scenario almost always includes unpleasant surprises.

- *Requirements-driven development.* This means developers who are creating or customizing CRM functionality have an understanding of the overarching business requirements driving CRM, as well as the necessary functionality. Developing against requirements eliminates the notorious phenomenon of “scope creep” and ensures that users get what they’re expecting.
- *Continuous user involvement.* Many CRM teams fall into the trap of involving business users at the beginning and end of CRM but rarely in the middle—during its development—where it’s often critical. This means end users evaluating proofs of concept, validating data and business rules, weighing in on the contents of CRM training, and reviewing new screens or functionality prior to CRM deployment. It also means establishing regular communications between development, the business stakeholders, and the CRM business sponsor.
- *Implementation process rigor.* Even with other CRM best practices in place, such as comprehensive requirements and an enthusiastic business sponsor, CRM development must be planned and executed around a structured development process. This is to ensure that the PMO and project managers can anticipate and accurately scope various development activities. A sound development roadmap also ensures that programmers focus less on the implementation process and more on the actual delivery of valuable CRM functionality.

Figure 9-3 illustrates a CRM development roadmap that applies some of this structure.

Within the three main project phases—planning, construction, and deployment—the CRM roadmap features steps that contain a number of fixed and variable tasks:

Business Planning

CRM business planning involves many of the steps we discussed in Chapter 7. The most critical activity at the planning stage is defining CRM’s overall

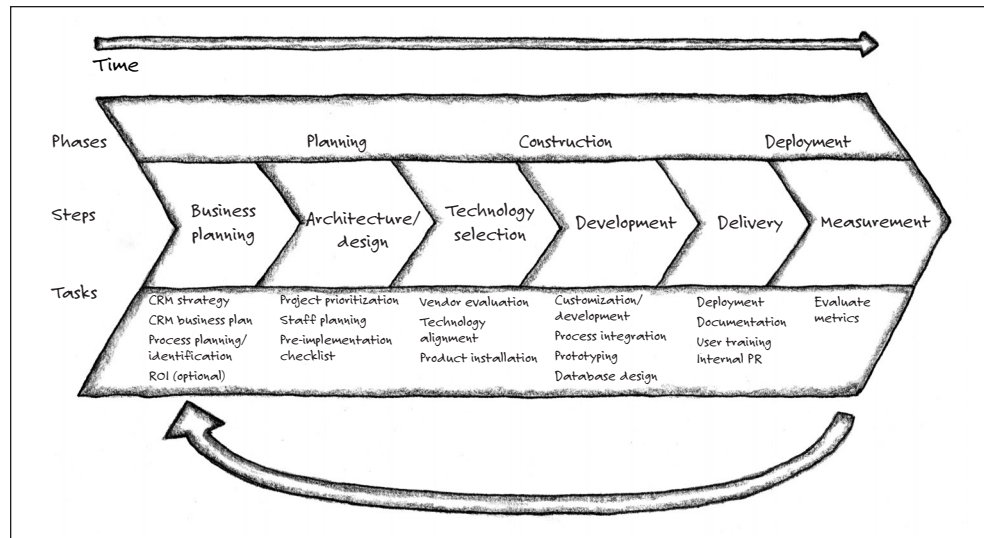


Figure 9-3: A CRM implementation roadmap

objectives—be they at the department or enterprise level—and delineating the requirements of each one. At the enterprise level, CRM business planning can involve the documentation of a corporate CRM strategy and the definition of the corresponding programs within it. At the department level, it can simply mean establishing the boundaries of a new CRM application.

At minimum, the business-planning phase should include the documentation of high-level CRM business goals in the form of a strategy document or business plan. This document will be leveraged at CRM's inception to gain executive consensus and sponsorship. It will be useful as a focal point for requirements-driven development and—after the CRM project has deployed an application—as a way to measure its results.

As Chapter 7 illustrated in the hotel reservation system discussion, part of business planning should identify the critical customer-focused business processes CRM will impact. Where they are straightforward, you might decide to redesign these processes as part of the planning activity. More often than not, companies planning their CRM projects realize that rather than simply automating existing business processes, they are defining those processes for the first time.

Depending on funding and sponsorship requirements, CRM business planning might optionally include ROI estimation or cost-savings projections.

Architecture and Design

The need to plan CRM architecture and to design an implementation strategy is what makes business sponsors and project leaders shudder and go straight to technology selection hoping for a miracle. The architecture and design step is painful, but it's worth it.

This step identifies the business processes the CRM product will support. It involves listing the specific functions that will need to be implemented—and how—ultimately giving you a good idea of CRM's impact on the organization and various technologies.

Inventoring the range of corporate areas CRM will affect, as well as those that will affect CRM, is a critical activity. At the end of this step you should be able to answer the following two questions:

1. What technologies and processes do we have in place that will be impacted by CRM?
2. What do we need that we don't have today in order for CRM to work?

Relative to existing technologies, try to project CRM's impact on your current systems. Your IT organization should be willing to do this—and in return it won't be blindsided by CRM after it's been developed. Impact analysis can mean listing current systems—for instance, you might need to know a bit about your company's existing call center operational system before you can understand how candidate CRM technologies will link to it. Indeed, a range of existing technologies, from ERP systems to current marketing automation technologies to handheld computers, are likely to be touched by CRM.

After the system impact of CRM is well understood, an IT architect can draft a CRM architecture illustrating the appropriate linkages. Integrating corporate systems that exchange data—even if the data isn't formatted consistently—is known as enterprise application integration (EAI). It's a truism of business that different corporate systems store and use data in different formats. The term EAI denotes the integration of often disparate corporate systems that routinely exchange or share data. This means moving data between systems, as well as transforming that data so these systems can understand it.

The letter depicted in Figure 9-4 is from an online retailer that is doing neither CRM nor EAI.

This letter was included in a product delivery and represents a veritable smorgasbord of CRM don'ts. The first one is that the company's online ordering

.com

Dear Valued Customer,

Thank you for your order. We hope that you are satisfied with the merchandise you have just received. Unfortunately, one or more of the items you have ordered are out of stock. Since we do not hold backorders, this order is now closed and you will only be billed for what you have received.

Please call our Customer Service Department at 1-800-000-0000 for the item availability and up-to-date stock information during our regular business hours of 8:30am – 5:30pm EST Monday through Friday.

We apologize for any inconvenience this may have caused and look forward to serving you again.

Thank you,

.com

Figure 9-4: Neither CRM nor EAI

system is obviously not linked to its inventory system. (The fact that the company happens to be a high-profile dot-com with an edgy Web site and slick e-mail marketing campaigns is not evident in its post-sales customer support.) The customer should have been notified of the out-of-stock items at the time of the order, not upon delivery of the remaining items.

The company might believe that, had the customer known that not all of the items she ordered would be in stock, she would not have placed the order at all. Perhaps some of the out-of-stock items the customer wanted were in some way related to the items that showed up on her doorstep. Or perhaps the company intends to link its various operational systems together but hasn't had the time. Either way, this company has successfully achieved these detrimental outcomes:

- Sending its “valued customer” a form letter and thus not differentiating her
- Putting the onus on the customer to follow up on the desired items
- Failing to provide similar levels of sales and service. (Notice the company's customer service hours. If the customer lives on the west coast, she only has around 5 available hours to contact the company by phone—but she can still shop on the Web at any hour!)
- Losing a “valued” customer

EAI is important to CRM because, no matter how successful a new marketing campaign or how polite the (albeit mass) marketing message, if internal systems cannot share data, vital business knowledge could be lost and customer service undermined. If the company truly had EAI, its inventory system could alert its customer support system when the desired items came back in stock, allowing a CSR to notify the customer and make a sale. It is for this reason that many companies undertake EAI as a preparatory step toward CRM.

For new CRM functionality, you'll also need to understand what data to consider. For each business requirement, one or more data requirements will result. For instance, if survey data is to be incorporated into customer profiles, which specific data elements should be collected? Will you need to collect external data such as third-party householding information or competitive intelligence data? Of the data collected, what should be displayed to CSRs? To marketing staff? And what systems will deliver that data?

A significant part of defining data requirements involves addressing the actual meaning of certain data definitions. Is there consensus across the business that the term “revenue” means booked revenue, or might it imply billed revenue? Does a “new customer” have the same attributes in the sales organization as in

customer support? To many in IT, documenting data definitions smacks of cumbersome metadata management and documentation projects. However, it's more about simply gathering consensus and enforcing consistent business terminology, whatever form that takes. If information is indeed a corporate asset, as we discussed in Chapter 7, consistent and sustainable data definitions are essential.

When you've completed an impact analysis, you can begin prioritizing projects according to business requirements and staffing your development projects, as we discussed earlier in this chapter.

Technology Selection

As Chapter 8 explained, CRM technology selection can be as simple as choosing an off-the-shelf product or as complex as a comprehensive evaluation of various CRM systems integrators or ASPs. If you've bitten the bullet during architecture and implementation design, understanding CRM's impact on existing systems and its requirements for new functionality, you should be in good shape to align any candidate CRM product to your existing IT environment.

Development

Development involves the construction and customization of the CRM product, using specific product features. But CRM development is more than programmers assuming center stage and writing code; it involves the integration of business processes with the chosen CRM product.

By this time, you will have already identified the key CRM business processes. Process integration means that CRM technology you've just selected integrates into these business processes. (The converse—merging business processes into the CRM product's features—forces the product to in effect define or change those processes, thereby diluting them until they are no longer optimized.)

Process integration involves ensuring that identified business processes are tested with users to ensure not only that the business processes work, but also that technology features can be leveraged in order to refine them. In other words, technological capabilities should improve, not compromise, customer-focused business processes. For instance, a campaign management product allows segment managers the opportunity to refine a mailing list before the campaign is launched—something they've never been able to do—thus refining the existing process. The same product might also allow a campaign director to

monitor a campaign's success rate as it's being executed. If the first thousand prospects have been unresponsive, the manager can cancel the campaign rather than allowing it to proceed, adding another valuable option to the campaign execution process.

Refining business processes during development means iterative prototyping: from time to time programmers demonstrate interim functionality to business users. Thus business users can monitor product development and test CRM functionality during—not after—implementation. End-user feedback about CRM functionality and desired changes can be flagged and incorporated into the CRM deliverable to ensure that resulting functionality conforms to requirements and meets user expectations.

Of course, development mostly involves technical work and thus might also include such tasks as database design, data cleansing and integration, and integration with other corporate systems. The integration step can easily be underestimated, because the CRM system might need to feed data to and pull data from other systems, such as call-routing systems or existing sales force automation (SFA) tools.

Delivery

The delivery step is often overlooked or lumped into development. Basically it means leveraging the corporation's IT infrastructure to dispatch the resulting CRM software to the business users who need it. In the case of a new Web-based sales-force automation tool, the application might be announced via an e-mail message that contains a link to the new CRM Web site. If the CRM system is client-server based, it will need to be installed on individual workstations.

Often, new CRM functionality simply supplements an existing operational system and is not considered a new standalone system. For instance, a contact center representative might now see a "screen pop" displaying a customer profile when the customer calls in. In such cases, business users might not even be aware of the new feature before it appears.

In both cases, user training is paramount. Before a salesperson begins using a new SFA package to schedule meetings or a CSR tries interpreting a customer's profile, she should be trained not only in using the new functionality but also in changing the way they work so they can take maximum advantage of it. Often, a customer-facing representative having new or improved customer knowledge can alter the way she interacts with the customer. For this reason,

CRM training should incorporate introductions to new business processes as well as new technology.

CRM delivery can also include user guides, job aids, and other documentation, as well as online or Web-based help to encourage users to make the most of the new CRM functionality.

Some companies go so far as having CRM sharing meetings to introduce the business at large to a new or pending system, and CRM business sponsors hold periodic update meetings, filling in various organizations and key staff members on CRM's progress.

Measurement

The measurement step brings the CRM roadmap full-circle as it evaluates CRM usage in order to refine CRM requirements. Many companies forego ongoing CRM measurement; such companies are confident they won't have to answer for their CRM expenditures. But can you truly claim your CRM program is a success if no measurements are in place to prove it?

In most cases management expects regular updates on programs in which they've invested heavily, and CRM is expensive. Savvy business sponsors define CRM success metrics as a result of the initial justification of CRM, and measure the successes after CRM has been deployed. For instance, if your new CRM system automates workflow to communicate widget defects to your R&D department, you might consider tracking the decrease in product defects and a corresponding increase in customer satisfaction for customers who have widgets. This measurement can include value quantification—such as lower support costs due to fewer support requests—and thus prove return on investment.

Another way to measure CRM's success is to evaluate how well it has solved existing business problems. If you established success metrics when you created your CRM business plan, supplement them over time by correlating them to actual results. Documenting success metrics along with their actual measured improvements is a valuable way to track and quantify tangible CRM business benefits, as illustrated in Table 9-4.

Improvement is usually gradual as users become familiar with new technology and business processes. An effective CRM program delivers ongoing improvements as it's adopted more widely throughout the company. The 12-month measured improvement column represents the rate of improvement since the launch of the CRM program and illustrates this incremental gain.

Table 9-4: CRM Success Measurement

<i>CRM Success Metric</i>	<i>Desired Improvement</i>	<i>Measured Improvement (6 months)</i>	<i>Measured Improvement (12 months)</i>
Reduction in the time required to generate customer name-and-address lists for targeted mailings.	Campaign list generation to take 1 day or less.	Campaign list generation takes 3–5 hours.	Campaign list generation takes 1–4 hours.
Ability to make product recommendations to customers during support request (online or phone-based).	Recommendations result in cross-selling improvement rates of 8 percent or higher.	Customer support cross-selling increase of 6 percent.	Customer support cross-selling increase of 10 percent.
Electronic distribution of customer sales reports to sales management.	Elimination of sales staff responsibility to produce weekly and monthly reports, generating a productivity increase of 5–10 percent.	11 percent increase in sales productivity and reduction of one full-time administrative position.	12 percent increase in sales productivity.
Reduction in time spent analyzing data to correct contradictory customer data from sales and provisioning systems.	Elimination of need for data correlation by implementing centralized customer database.	None—database pending.	Elimination of data correlation, resulting in redeployment of two full-time data analysts.

Measurement also includes the incorporation of user feedback to improve CRM usability and business effectiveness. As the CRM implementation roadmap shown in Figure 9-3 illustrates, CRM measurement loops back around to further CRM business planning, allowing the company and its

lines of business to continually refine CRM requirements and identify new CRM opportunities at the same time. If you incorporate measurement and feedback into the planning cycle, CRM will deliver new and better functionality, resulting in small victories that add up to improved customer value.

Putting the Projects Together

After you've identified your CRM projects, your PMO or project managers can agree on an overall CRM timeline that will be enhanced and supplemented as the business uses CRM and customers begin experiencing the benefits. The projects identified in Figure 9-2 can become components of an overall CRM program timeline, as shown in Figure 9-5.

The solid boxes at the beginning of each project connote the fixed amount of time allotted for the business-planning phase. This phase includes project scoping; thus project durations might change after business planning is completed. Each project will have its own development-project plan reflecting more specific tasks and resources.

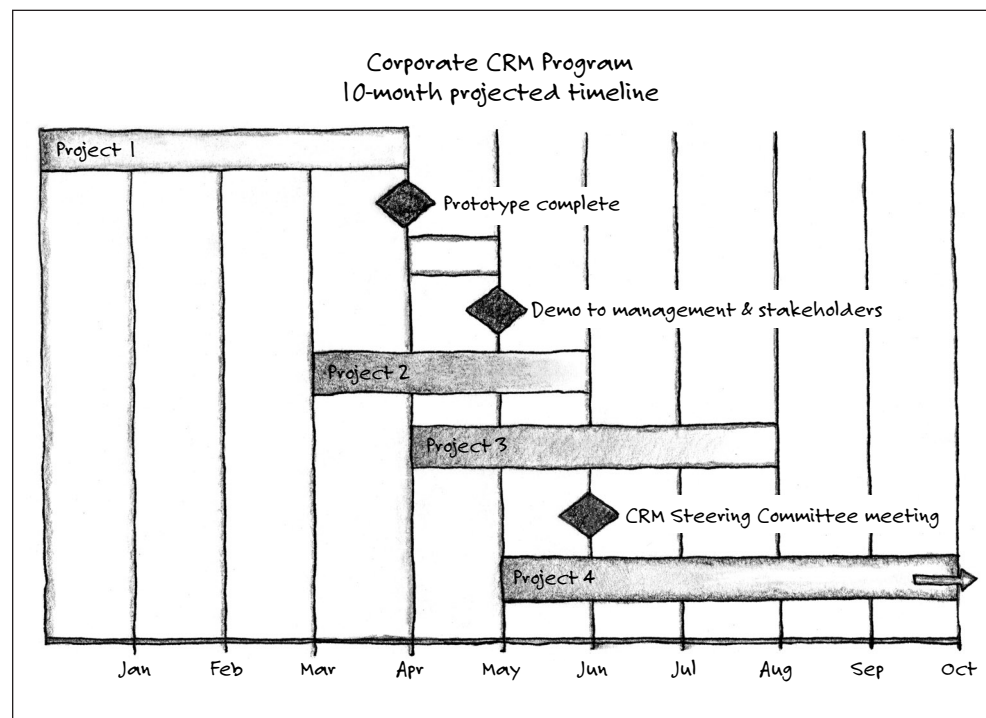


Figure 9-5: CRM program timeline

A visual timeline like the one in Figure 9-5 is not only effective in managing expectations about each project's forecasted delivery time frame; it can also become the basis for a CRM program document in which the project manager or development team leader can include individual project plans, requirements documentation, and specific CRM functions and features, either as a physical document or as part of your company's web-based knowledge management infrastructure. Thus managers and stakeholders can access up-to-date information about current CRM activities.

A CRM Implementation Checklist . . . for Failure

From the CRM projects I've worked on, evaluated, proposed, and completed, I've seen a handful of common denominators for failure. I call these "The Seven Deadly Sins of CRM." Consider the sins a checklist of what not to do if you want to enter the pearly gates of CRM nirvana.

Sin Number 1: Failure to define a CRM strategy. Simply defining what CRM means to your company is difficult enough without gathering consensus on a corporate-wide strategy. Companies routinely misinterpret business requirements and thus underestimate the complexity of CRM (remember the four quadrants in Figure 7-6?). Even if gathering consensus on what CRM means to the business and what it should deliver takes longer than you would need to simply begin development, take the time. It'll save you time and money in the end.

Sin Number 2: Failing to manage staff expectations. Many firms apply rigor to planning and development but forget about deploying the CRM system to the business. The CRM rollout in which an IT liaison sends an e-mail to sales staff announcing training for the new sales-force automation package is doomed before it's even delivered. Business users must be stakeholders from the inception of the CRM project, from planning through development and through to deployment. Anything else risks alienating potential end users, an irrevocable situation.

Sin Number 3: Failure to define success. What is CRM success and how do we know when we've achieved it? Even CRM business sponsors who understand the differences between CRM's various applications don't differentiate between increased cross-selling and improved profitability. I'm always surprised when I hear seasoned marketing execs refer to customer loyalty, customer value, and customer profitability as synonyms, as if expecting to achieve all three of these

objectives with the first CRM release made sense. Define discrete success metrics—they'll be different for increasing customer profitability than for improving customer satisfaction—and then measure against them. If you don't, there could actually be business successes that aren't rightfully attributed to CRM.

Sin Number 4: Hasty ASP decisions. Companies haven't yet sorted out the advantages and disadvantages of the ASP model. Large companies assume ASPs serve only small markets and dot-coms that lack significant IT infrastructures. Small to mid-size firms assume ASPs are too costly, despite potential cost savings. Many companies are even underestimating their in-house resources and skill sets and jumping blindly onto the ASP bandwagon. Don't be one of them. Understand the pros and cons of the ASP model, and make a decision based on your business and functional requirements.

Sin Number 5: Failure to improve business processes. The proverbial mistake of "paving the cowpath" applies here: CRM should not simply overlay archaic corporate policies. It should instead formalize and automate nimble, customer-focused business processes. Be willing to drastically modify and continue to refine your business processes, and make sure your CRM technology incorporates these process modifications. And don't fall into the trap of hoping your new CRM tool will do this work for you. Processes should be defined from the customer's perspective, not the technology's.

Sin Number 6: Lack of data integration. Chapter 6 painted the picture of stovepipe CRM systems and the danger they can lead to if allowed to perpetuate. Effective customer-focused decision-making means understanding each customer across her various touchpoints and beyond your immediate knowledge of her age, income, preferred channel, or sales territory. The difficult truth is that customer data exists in multiple systems on a variety of technology platforms across your company. Finding, gathering, and consolidating this data isn't easy, but it's absolutely crucial.

Sin Number 7: Failure to continue socializing CRM to the enterprise at large. Companies who have delivered nothing less than revolutionary customer-facing improvements via CRM often rest on their laurels. CRM is an ongoing process, and success breeds success. Consider establishing an "internal PR" job function to communicate with executives and decision-makers who might determine ongoing funding, as well as to the various lines of business who might leverage the functionality and data to further their own customer focuses. Proselytizing CRM successes should not only be practiced but should be formalized and

updated via regular newsletters, status meetings, or an internal Web site. Don't be shy about initially promoting CRM. If your customers experience improved service and your sales and marketing staffs generate more effective leads, it won't be long before CRM starts promoting itself.

The Manager's Bottom Line

Just because you've bought the tool doesn't mean the spending is over. After you start adding up the time needed for customization, the labor necessary to integrate CRM with your other corporate systems, the cost of external specialists needed to make it work, the time required from business users, and the new staff you'll have to hire to deliver the system on time, you'll be tempted to reconsider reengineering those old legacy systems to become more customer-centric. CRM development costs are routinely underestimated.

As is CRM complexity. Most CIOs are familiar with traditional waterfall development methodologies: projects that are linear, with a defined beginning and ending. They're accustomed to measuring system success based on the new system's number of transactions or number of users. But CRM success revolves around staff efficiency gains and process improvements, so such advances as higher productivity and enhanced customer satisfaction will be more difficult to measure. Likewise, traditional systems development means laying out requirements in concrete terms once and for all. Given that CRM stresses improving customer-focused processes, requirements-gathering should be iterative. Indeed, gathering and refining business requirements is an ongoing process, much like the CRM program itself.

Moreover, depending on the skill sets of the customary in-house business analysts, they might need some extra help. CRM business sponsors might find individuals trained in corporate quality programs to be more effective in CRM requirements definition. Quality-knowledgeable individuals have learned to focus on how people do their jobs and how the inputs and outputs relate to the overall process.

For enterprise CRM, it's essential for customer information to flow freely throughout your organization. If you are to provide the proverbial 360-degree customer view, users must be able to access that data whenever they need it. This often means staff members from elsewhere in the company need to be involved to connect important corporate systems with CRM. Such involvement can render even the simplest CRM application an enterprise-wide effort. Although the

PMO can ease the communications challenges of such potentially complex collaboration, strong management is a definite prerequisite.

Know what you already have that you can leverage. Many talented IT departments have short-changed themselves by not advertising the data warehouses and marketing databases they've so painstakingly built. The data source system inventories, extraction and transformation processes, metadata repositories, and other infrastructure components of these databases are invaluable assets to any CRM system. Find them, and use them.

Head reeling yet? Do you wonder whether CRM development might be too complex or costly? Consider the alternative. The Internet and the pressures of e-business have weakened the barriers to competition, and your customers are getting smarter. It's not a question of *if* you launch CRM but the scale to which you launch it. Don't be afraid to start small, implementing one requirement or one set of business functions at a time. Have a clear view of what's ahead, and understand the potential impact CRM can have on your customers, their immediate satisfaction, and their long-term loyalty. Where customer relationship management projects are concerned, it could very well be now or never.

