Auditing your CRM System

2

Express Version

- Perceived problems with CRM systems are often due to poor underlying processes or poorly thought-out customizations of a basically good tool. Since implementing a new tool takes lots of time and resources, thoroughly investigate whether the tool is the root cause of the problems before making the decision to select a new tool. Use our easy test to facilitate the audit process.
- Don't decide to switch tools based solely on a sleek demo. Any tool can look wonderful in a demo.
- One size does not fit all. Don't base a decision to get a new tool on a recommendation from a colleague who may be facing different challenges.
- Do get a new tool if you are still using a homegrown tool. You will be amazed at the functionality you've been missing and you'll save money too, at least in the long run.

- Do keep up to date on new tools and new technologies. If your current tool vendor is lagging behind and even future releases fail to include new technology ideas, it makes sense to explore alternatives.
- Tool implementations work best when you have adequate time and resources. Be wary of embarking on a tool change if you are pressed for either.

The Big Question: Do You Need A New CRM System?

If you have absolutely no tool in place to track customers' interactions, you need a new CRM system. Save yourself some time and proceed directly to the next chapter.

If you do have a tool in place but are not satisfied with it, take a good look at it before deciding to start over. Why? First, because problems that appear to be tool problems may stem from other causes. Second, because implementing a new tool requires significant amounts of time and money, and requires retraining customers and staff to use it. You don't want to embark on such a project without having a clear requirement to do so.

What other areas could cause what appear to be tool issues?

Very often, the root cause is poor processes rather than the tool itself. Processes that are inefficient, not customer-friendly, or plain confusing need to be fixed before any tool can be successful. It's true that a tool implementation process will (should!) expose process issues, giving you an opportunity to fix them, but it's really an expensive and roundabout way to address process issues. Fix the processes first, and then worry about the tool.

The second root cause, which is often difficult to isolate, lies with the specific implementation of the tool. In other words, you can have a tool that's basically sound, but the particular implementation (customization) you are running with is poor. It's not that hard to twist a good tool into a poor customized version thereof (although it's not possible to contrive a good implementation of an ill-suited tool). Poor implementations cause poor day-to-day performance, and typically more headaches and problems when it's time to upgrade to the next version, so they are a very big issue.

What follows are two different instruments to evaluate your current CRM tool and to untangle the root cause of issues you may encounter.

The first one (the five-minute test) is very quick and focuses on a handful of critical questions to determine how well your tool is performing and whether any shortcomings come from process or implementation weaknesses rather than the tool itself. It's a good starting point, especially if you have already done some research in this area.

The second one (the detailed inventory) is much more thorough and explicit and is meant for 1) those hurried souls who went through the five-minute test and want to explore specific areas in more detail and 2) those of you who prefer an in-depth approach. The detailed inventory includes more questions and, for each question, explains why it's important, how to go about evaluating that particular area, and how to identify root causes.

The chapter ends with a list of good and bad reasons to get a new CRM tool. This is the place to go if you are tempted to make a move although your current tool is working well just because you saw a cool demo or you want the latest technology. By the end of the chapter, you should have a good idea of whether you need a new CRM tool.

The 5-Minute CRM Test

For those of you who like to make quick assessments, here is a short test that will help you pinpoint whether your CRM system is successful, and whether it makes sense to think about a replacement, or whether the issues have root causes in other areas. Experience shows that quick assessments are often just as accurate as lengthy ones, but if you'd like a more thorough and detailed version of the test, please skip ahead to the next section.

To get the most accurate results on this test, respond as candidly as possible to each question rather than trying to figure out what the "right" answers may be. If you are not directly involved in customer-focused activities, you will need to involve the managers responsible for those groups to complete the test. They should also be encouraged to be candid. To make sure that they are not overly optimistic in their responses, spot-check a few with individual contributors in the various groups.

Some questions may simply not apply to you (for instance, if your CRM system does not have a support component, questions about support don't apply.) Just skip them.

TABLE 2.1 The 5-Minute CRM Test

		Disagree Strongly	Disagree	Agree	Agree Strongly
1	Customers know how to contact the sales team (by phone or electronically).				
2	Customers know how to contact the support group (by phone or electronically).				
3	Sales reps know how customers should contact the support group.				
4	Support reps know how customers should contact the sales group.				
5	There are existing, documented (written) processes for handling customer queries.				
6	Sales and support reps can locate the documented processes for their area within 2 minutes if they are unclear about them.				
7	There are well-defined metrics for customer-focused activities.				
8	Customer-focused employees have formal objectives that relate to the metrics.				
9	There is a defined, documented (written) process for creating new knowledge base documents.				
10	The knowledge base is growing daily.				
11	When you hire someone, it takes less than one business day to create a new account for that individual.				
12	The tool improvement request with the highest priority is less than 3 months old.				
13	You are using a commercial CRM tool.				
14	You are running your tool on a release that is currently supported by the vendor.				
15	Important customer information is accessible within the tool (rather than in a file cabinet or unrelated system).				

TABLE 2.1 The 5-Minute CRM Test *(continued)*

16	There is a customer portal available to conduct sales and support business.				
17	Customers require no training to use the portal.				
18	Your CRM tool supports a knowledge base.				
19	Line managers are getting regular metrics that are immediately meaningful for them (with no Excel massaging required).				
20	E-mail from customers is automatically loaded into the system (no manual cut and paste).				
21	Knowing the name of a customer, a sales employee needs less than 1 minute to locate the assigned sales rep, pending deals and existing support cases.				
22	Knowing the name of a customer, a support rep needs less than 1 minute to locate the assigned sales rep and existing support cases.				
23	It takes less than 4 hours to train a new hire to use the system.				
24	Creating a new account in the system can be done in less than 5 minutes by someone without a programming, technical background.				
25	Sales reps can enter a new prospect into the system in less than 2 minutes.				
26	Support reps can enter a new case for an existing customer in less than 2 minutes.				
27	A support manager can get a list of the current open support cases in less than 2 minutes.				
28	A sales manager can get a current forecast in less than 2 minutes.				
29	A marketing manager can get a hit rate of the last 3 campaigns within 2 minutes.				
30	The current system costs less than $1,000 per employee per year to maintain.				

Scoring

Add up the checkmarks in the Disagree and Disagree Strongly columns. If you have more than ten you have a CRM problem. Read on to determine what might be the root cause.

Questions 1 through 12 address customer processes rather than the tool. If many of them are rated on the Disagree side (either Disagree or Disagree Strongly), concentrate on fixing the processes rather than the tool. Once the process issues are addressed, take the test again to determine whether a tool change is also required.

Questions 14 through 30 are both about the tool itself and how it has been customized for you. Questions #14 (running on a current release), #19 (useful metrics), and #30 (cost) are usually good indicators of how well the customization is working rather than the tool.

Questions 21 through 29 (speed of doing various common tasks) are not silly time trials. Because tools won't be used if they are not efficient, it's important to see how quickly common operations can be performed. Problems in those areas can be caused by specific implementations as well as by the tool itself.

Question 30 is highly sensitive, since most companies do not track maintenance cost per seat to begin with, and since many normal variations occur as a result of the intricacies of the customizations on the one hand and the size of the user base on the other (a small user base means a higher cost per seat, all other things being equal.) Include in the cost figure: any depreciation costs on the license that you are still carrying; support and maintenance fee to the vendor; the same costs for the server, database, and other tools required by the CRM system; and the compensation costs for your internal support team. I deliberately chose a low number to make you think about how much you're spending so don't despair if your cost is only slightly higher than $1,000 per year.

Detailed CRM Inventory

Perhaps the five-minute test whetted your appetite for more, or you simply like a more detailed approach. In either case, the inventory below expands on the quick test and gives you more details on what is important and why.

The format for the inventory is an annotated list. For each item—some present in the five-minute test above, some new—comments are available on why it is relevant in the first place, how to go about rating it, and what the outcome may mean in terms of the tool, the underlying processes, or customization.

Rate each item in the inventory using the same scale as was used for the five-minute test (Disagree Strongly, Disagree, Agree, Agree Strongly). Here again, be as candid as you can on your responses: the candid answer is always the right answer. Although this is the "long" test, try not to spend more than a couple of hours at the most on it. Taking more time may increase your desire to rate higher and your imagination to do so, distorting the whole exercise.

Note each area that you rate on the Disagree side (including Disagree Strongly and Disagree). What is the root cause? Is it a weakness of the tool itself or with the way it's customized? Is it a process issue or a tool issue?

CRM Process

Items in this section are related to process rather than the tool. If there are issues in this area, they should be addressed prior to making any tool change decision, since the tool can only assist with the process, not guide it (although it's true that a well-run tool implementation would uncover process issues, it's considerably cheaper and easier to fix them first!)

- Customers know how to contact the sales team and the support group (by phone or electronically).

  Is there an organized process to communicate such basic information to customers? If your current tool lacks a portal where that information can be made available, is the information given to customers in other forms? Such a basic "how-to" process is required regardless of the technology you are currently using.

- Sales reps know how customers can contact the support group, and conversely support reps know how customers can contact the sales team.

  One of the biggest complaints customers have with their vendors is that the left hand doesn't seem to know what the right hand is doing. Although the distribution of cross-departmental information can be facilitated by a CRM tool, employees in one customer-focused group should understand the value of being able to direct customers to the right place and should know how to help customers access other services within the company even if there is no integrated tool.

- There are existing, documented (written) processes for handling customer transactions.

  The process documents don't need to be long or formal. Actually, overly complex documents often signal a gap between the formally documented process and the way things are normally handled. The process should be written in simple English (or whatever language you use in your company) and should be clearly understandable to an outsider (as opposed to a senior employee). The process should take the transactions from beginning to end and clearly indicate routing and handoffs.

For a more meaningful test, take a few customer transactions through the entire cycle. Do they follow the process? If they deviate from it, is there a bona fide exception to be made?

The ultimate test is whether the process makes sense to both the customers and the employees. Many otherwise well-documented and properly applied processes are cumbersome and wasteful of time and resources. A common weakness is to require multiple layers of processing—whereas state-of-the-art processes shoot for a "one-owner" approach—as well as formal authorizations along the way, rather than letting the staffers make independent, fast decisions, which can be audited later if a review is required.

- Sales and support reps can locate the documented processes for their area within 2 minutes if they are unclear about them.

  The ease of access to the documented processes is an important test of the value attached to it. If reps routinely choose to "wing it" rather than to check the proper way of doing things, you can be sure that adherence to the process is neither important nor rewarded.

- There is an owner for each customer-focused process.

  Is the owner identified within the process itself? Spot-check a few documents to verify that the owner is still with the company and is still the owner of the specific process.

  Processes change over time, which is a good thing since they need to adapt to new circumstances and they should be improved through experience. Assigning processes to specific owners makes it more likely that required maintenance will actually happen. Also, since many times it is the hands-on staffers who spot potential changes and improvements, it makes sense to make it very clear to them whom to report suggestions to. Bonus points here if there is a clear process for process maintenance, especially if it includes feedback to individuals who provide suggestions.

- The processes were updated in the last six months.

  Following up on the point above, finding evidence that processes are indeed updated means that that they are important. You may think that processes that do not change mean good processes (they don't change because they don't need to change because they are perfect in the first place). The reality is that processes that don't change don't change because no one cares about them. Changes need not be very significant.

- There are well-defined metrics for customer-focused activities.

  Do you think that metrics are a function of the tool rather than the process? I beg to disagree. Although a good tool makes the computation of metrics much easier, metrics should exist independently of the tool.

  Ideally, metrics should capture actual results (qualified leads, actual sales, happy customers) rather than simple measures of activity (number of campaigns, sales calls, or support cases). We'll discuss this topic in much more depth in Chapter 10.

- Customer-focused employees have formally-defined objectives that relate to the metrics.

  Measurements are one thing. Making sure the measurements matter to the staffers who work with the customers ensures that the metrics are attended to. What gets measured gets done, which is yet another reason to pick good, meaningful metrics. Do you really want your team to compete on how many sales calls they make? I think not!

- There is a defined, documented (written) process for creating, reviewing and posting new knowledge base documents.

  Here again, a good tool will make knowledge creation much easier, but you still need to have a clear process to create documents. A state-of-the-art process will have some way of gleaning documents from the front-line staffers, either by encouraging them to write documents themselves, or by having a system through which they can request and suggest new documents.

  Also, there should be some kind of timeline system to ensure that documents are posted promptly (ideally, within a couple of days of their creation), Audit a few recent documents to see how long the full cycle took from creation to posting. It's useless to create documents that take weeks to post.

- There is a defined, documented (written) process for maintaining knowledge base documents.

  Knowledge maintenance, often neglected compared to its more glamorous knowledge creation sibling, increases in importance with the size of the knowledge base. Poorly done, it is the main reason why knowledge bases are not used: why bother using a knowledge base where many documents are obsolete or plain wrong?

  At a minimum, the maintenance process should collect user input on problematic documents and act upon the input swiftly, within a couple of days at the most. It's best if the input is unobtrusive, that is, the user should not have to fire a specific e-mail to the document owner or the knowledge base owner to get a document changed: instead, it should be possible to post the feedback right from the knowledge base environment.

Beyond reactive maintenance processes, it's even better to have a systematic review process through which documents are reviewed periodically regardless of specific user feedback, both singly and in comparison to others in the same category. Documents that are not used much, overlap with others, or obsolete can be discarded or fixed up.

- The knowledge base is growing daily.

  This is a great test of the overall quality of the knowledge-creation process, and would more accurately be stated as "the knowledge base changes daily" since changes and deletions are just as important to the quality of the knowledge base as additions. Of course, if your organization is tiny, *daily* growth is a tough requirement, but the idea is to see constant growth, not sudden increases on the days the one knowledge base reviewer suddenly decides to pay attention to the knowledge base.

- When you hire someone, it takes less than one business day to create a new account for that individual.

  Even a very cumbersome tool will allow creating a new user in minutes. The point here is whether there is a process to bring the account creation request to the right person and have it executed within a business day, as a matter of routine and not just because you happen to be benchmarking. CRM tools are useless if one cannot access them.

- The tool improvement request with the highest priority is less than 3 months old.

  Another good benchmark test, the aging of improvement requests is a sign of 1) whether anyone is bothering to generate improvement requests (if no one's using the system, you can be sure no one's bothering to request improvements) and 2) whether anyone is paying attention to the requests. Unattended improvement requests will surely lead to no more improvement requests, and a tool that doesn't change is a tool that is dying.

  By the way, if you should find that there is no formal list of improvement requests, or that improvement requests are filed away in the e-mail folder of some lucky individual, the support mechanism for the CRM tool is not functioning properly. This is not necessarily a weakness of the tool itself, although I would question why such an informal record-keeping mechanism is used when, presumably, the tool should be able to track its own enhancement requests and bugs, at least if it includes a support-tracking component.

Ease of Use—For Customers

Items in this section focus on the customers' use of the system through some kind of portal. Problems in this area can be related to either the tool or its implementation. Older tools tend to be weaker in this area, although they are making progress as customers demand more e-commerce functionality. In some cases, existing customer-facing functionality is not implemented because of fears that customers can't enter support cases properly or would use online order information to their advantage. Such issues have little to do with the tool itself, but instead should trigger revisiting the business choices that were made around customer access.

- There is a customer portal available to conduct sales and support business.

  A customer portal is a basic requirement today and if your tool doesn't offer it, it's time to shop around. I must say that I have encountered many situations where existing portal functionality is not implemented. Three types of reasons are cited for this (to me) amazing situation: lack of resources to implement and maintain the portal; a concern that existing functionality is insufficient compared to what customers want; and finally a fear that the portal would be "misused" by customers.

  If your tool offers a customer portal that is not deployed because of resource constraints, you should know that customer portals reach ROI in the shortest amount of time compared to other CRM functionality, so you need to find a way to work around the resource limitations. If the underlying issue for the resource gap is that the tool provides only a tool kit, and therefore requires *lots* of resources to implement the portal, then you have an argument to find a better tool.

  Regarding the concern about limited functionality: sure, we would all like an all-powerful, all-encompassing portal, but even basic functionality is better than nothing, and customers certainly understand this. Implement what you have.

  The third concern, about customers' potential misuse of the portal, is not well grounded and stems from a misconception that creating a more transparent relationship with customers will create abuse. Barring the real issues of security, which need to be addressed, opening up systems to customers is actually a breath of fresh air. Just as an example, support groups often shudder at the thought of letting customers log support cases electronically. Will the descriptions be complete? And how can we allow customers to determine the priority of their issues? Well, it turns out that customers actually do a great job of describing issues in writing, and are remarkably restrained in choosing priority levels, so the concerns are misplaced.

Assuming now that you do have a functional customer portal, there are certainly degrees of excellence. A basic portal with online information is just that, basic. You really want to aim for an interactive experience where customers can search for information and conduct at least simple transactions online such as placing repeat and add-on orders, checking on existing orders, and logging and checking on support issues. In order to safely conduct personalized transactions, the portal should include an identification scheme (with a password).

Check whether the portal is customer-focused rather than department-focused. I have seen many portals where training and support are described in completely different areas, probably because two totally different departments handle them, whereas customers think of them under the common umbrella of "services."

If there is no portal, or if functionality is lacking in the portal, first confirm whether you are using all the functionality provided by the vendor in this area. If not, explore whether your organization is too timid in letting customers conduct business online (an issue with process or politics) or if the tool is too awkward to make deployment possible (a tool issue). If you are using all available functionality and your portal is weak, then the issue lies with your tool. Weak portal functionality is an indication to select a better tool.

- Customers require no training to use the portal.

  Using the portal should be intuitively obvious. A portal is just a web site, and like any other web site it should be user-friendly. Anyone should be able to navigate and operate the various functions with no training whatsoever. While an online help function is a definite plus, it should be possible for a novice to figure things out without having to resort to it. Clearly, the requirements for simple use are even more stringent if your customers tend to be more technologically naïve.

  If your portal functionality is particularly rich, it's fine to offer a tour of the portal, either self-guided or through a webinar (an online seminar), to make sure that customers do not overlook specific, perhaps unusual functions. But navigating and operating the portal should be easy to do without specific training.

If your portal requires training for successful use, you may have an issue with the tool itself, but often it's really with the customization of the tool. Most companies choose to customize the customer portal. Sometimes, to be honest, it's because the portal functionality provided by the tool vendor is nothing more than a tool kit! However it's not rare that misguided customizations create difficulties in using the portal. If you have a chance to see other portal implementations for the same tool vendor, which is often possible since portals have more public exposure than internal systems, check whether any of them are better than yours. If so, you have a customization issue, and you should work on that. If not, the underlying tool is to blame and this is a potential reason for finding a new one.

Ease of Use—For Internal End-Users

Items in this section focus on the employees' use of the system. Problems in this area can be related to the tool, its implementation, or more rarely process issues. Older tools have clunkier user interfaces, but don't be put off by less visually appealing interfaces. If they allow quick manipulation of the data, they can be quite effective and many items here focus indeed on speed of common operations rather than pure sleekness of the tool.

- Desk-bound staffers keep the application up on their screens at all times.

  A major issue with CRM system is user adoption. If desk-bound staffers (in support and telesales functions, for instance) live and breathe the tool, it's a positive indication that it is being used. If they do not, this could signal everything from a tool issue, to a customization issue, even a process issue if customer transactions are not faithfully recorded. Look for clues in the other entries of this section.

- Important customer information is accessible within the tool rather than in a file cabinet or unrelated system.

  I'm not talking here about having *all* the information in the company available in the tool (although it would be handy, if potentially overwhelming). I only mean that basic information needed to do the work on a daily basis is available. For instance, if I'm providing support, can I tell whether a particular customer has failed to pay the maintenance bill and should therefore be denied support? (But I don't really need to see the detailed state of the customer's receivables.) If I'm a sales rep, can I see what products the customer bought last year? (But I don't necessarily need to see the actual contract that was signed at the time.)

If customer-focused staffers must hunt down basic information, their efficiency decreases, but more insidiously their desire and likelihood to use the tool also decrease. In particular, if you find "feral" tracking systems for important information, especially if it would be fairly easy to track within the tool, you have a good clue that something is not quite right with the CRM tool. For instance, a support group that tracks requests to a tier 3 group in a separate tool would be a red flag.

Issues in this area can be created by the tool. For instance, if the tool doesn't include a function to record purchases, even at a high level, then accessing a purchase history will not be possible. However, problems can also be a matter of process or implementation. For instance, if there is no process to record purchases or if there is no process to transfer purchase information from the back-office system to the CRM system, even a tool that *would* be able to show a purchase history will not be able to do so. We'll come back to this point when we talk about integrations. A good strategy is to focus on frequently used data and to find integration and automation solutions for them rather than trying to get all systems to share all data seamlessly.

- Knowing the name of a customer, a sales employee needs less than 1 minute to locate the assigned sales rep, pending deals and existing support cases.

There are many time-based questions in this inventory, and the reason is simply that frequently performed tasks must be very efficient. Note that access to selected pieces of information from other departments needs to be quick, too (for sales employees, it might be getting a list of support cases).

Issues in this area can arise for a variety of reasons. If the navigation required to perform the tasks is awkward, it can either be a tool issue or a customization problem. If it is a tool issue, you may be able to work around it through clever customization, but I would question the wisdom of pouring resources into a tool that doesn't do the basics right.

If it takes time for the system to process the request, it can be a limitation of the tool itself, or it could be a configuration issue. If you have severe performance problems invite the vendor to conduct a performance audit. They should be able to suggest a different configuration to ease the problem, or they may spot inefficiencies in the ways customizations were programmed. You can then evaluate the cost of the prescribed remedies, weigh them against the likelihood that the tool can indeed perform against your requirements once they are implemented, and decide whether the tool needs replacing.

- Knowing the name of a customer, a support rep needs less than 1 minute to locate the assigned sales rep and existing support cases.

  This is the same issue as the one above but viewed from the perspective of a support staffer.

- Sales reps can enter a new prospect into the system in less than 2 minutes.

  This is another common task that must be performed very quickly and efficiently. Watch for the number of keystrokes required. Also watch for automatic data quality check. Does the system let you enter a new prospect called "ABC Systems" if there is a customer by that name? If there is a customer called "ABC"? Such simple mechanisms go a long way in ensuring data integrity, and data integrity can be a big issue in CRM implementations.

  Entering a name is one thing, but can the sales reps enter all the relevant information about the prospect? For instance, can the tool track the data collected through the sales methodology they use?

  Problems here are mostly attributable to the tool, although ugly customizations can also be a problem.

- Support reps can enter a new case for an existing customer in less than 2 minutes.

  This is the same issue as above, but in the support world. The key here is how easy it is to find the customer record. Look for built-in aids to find customers with similar spellings and word case, and look for multiple ways to locate a customer by name, name of contact, customer number, e-mail address, etc. This is particularly important if you have a very large base of customers.

- The most commonly used screens have no elements that reps say are "useless" or "never used."

  This will require a bit of research on your part, but it will give you a good feel for how easy it is to use the tool. Sure, staffers can and do ignore unused fields, but it's a strain to have to constantly differentiate between useful and useless fields.

  If you have problems in this area, you may conclude it's all about the implementation. Why not just get rid of those fields? Point well taken, but the real issue may be why those fields are there in the first place. And, if they were not removed as part of the implementation, is it because removal is difficult and hard to undo, which would also be a tool issue?

- E-mail from and to customers is automatically loaded into the system (no manual cut and paste).

  This is a good indicator of whether the tool handles the low-level, repetitive actions that it should handle, freeing staffers to do the creative part of the job. Ideally, the system should automatically load incoming e-mails into the system, associate them with existing issues as appropriate, and route them correctly. And staffers should be able to send outgoing e-mails directly from within the tool, with the e-mails being captured into the system. Note that you do need a facility of this sort even if you attempt to conduct most of your business through a portal.

  Although most tools do have such an e-mail connector, it's not always implemented, sometimes because it's not easy to do so, which would be a tool-based issue.

- It takes less than 4 hours to train a new hire to use the system.

  Ideally, someone with decent computer skills should be able to use the system without any training, much as is the case with the customer portal. The reality is that each system exists within a process and it makes sense to have some formal training available to use the system in the most efficient way possible.

  If the training requirements are much greater than 4 hours (if your CRM covers multiple business functions, it would be 4 hours per function, not for the whole thing), then the system is too complicated. Is that the tool's fault? Perhaps. Check the vendor's end-user materials, if any exist. If they require more than a half-day to cover, then the tool is too complex. Strange customizations certainly add to the training requirements. Long training requirements are an immediate drag on productivity, but most importantly they signal that the tool is not as user-friendly as it could be.

- There is an end-user self-paced training document that is less than 10 pages long.

  This is another take on the training issue, because high training requirements say a lot about what it's like to use the system on a daily basis. So even though you may choose to provide instructor-led training for new hires, it should be possible to learn how to use the system from a self-paced, reasonably short written tutorial. The 10-page limit is by business function.

If such a tutorial doesn't exist, question why. It could be that the tool is so difficult to use that it requires a human instructor. It could also be that no one thought of creating one, which would be a simple process issue.

As reinforcement for the initial training, and to support a self-training option, it's good to have an online help facility available. In many cases, the online help system that comes with the tool (if there is one) is very difficult to update to accommodate customizations made to the tool. This is a common reason why online help is rarely truly helpful.

- Your CRM tool supports a knowledge base.

  CRM systems are not just transaction tools; they should offer a knowledge base function that allows your staff to store important information so it can be retrieved when needed, whether by internal users or customers.

  If your tool doesn't support a knowledge base, you may want to rethink your choice, although you can certainly purchase an add-on tool and perhaps integrate it with the CRM tool.

- The knowledge base provides multiple search mechanisms for end-users, including keyword searching, category searching, and special-attribute searching such as looking for a document created within the last week.

  This item tries to qualify the degree to which a user is likely to find a relevant document in the knowledge base. Successful searches are not that common, especially for the so-called naïve users, who may not be using the exact keywords contained in the documents. Be sure to benchmark this item both for internal and external users.

  Weaknesses here are mostly due to the tool itself.

- The knowledge base allows users to refine searches and/or ranks retrieved items in order of likely relevance to narrow the search results.

  While the previous test item focused on retrieving all the documents that meet the search criteria, this one focuses on not having to wade through pages of potential hits. This is a very difficult criterion to meet, and the larger the database the more difficult it is to isolate the most relevant documents.

  Weaknesses here are again due to the tool itself.

- It takes a reviewer no more than 2 minutes to retrieve documents waiting for review and post them into the knowledge base.

  Cumbersome document posting mechanisms are the bane of otherwise efficient (that is, search-efficient) knowledge bases. Most of the time, issues with the review mechanism lie squarely with the tool, since little or no customization is done on that area of the tool. The tools that are most successful for document creation are those that treat it like issue handling, providing queues, statuses, owners, and alerts. Unfortunately, there are few such tools.

Ease of Maintenance

Items in this section focus on the support and maintenance requirements of the tool. This is an important section since a tool with extraordinarily high maintenance requirements will cost more. It is also likely to be less well maintained, which results in decreasing functionality and usage over time.

- You are using a commercial CRM tool.

  At this point in time, there's simply no reason to use a homegrown tool. Yes, a homegrown tool is perfectly tailored to your exact needs. Yes, the initial cost of a commercial tool is high. But there's no way that a homegrown tool can compete with commercial functionality over time.

  So if you are using a homegrown tool, and even if it is performing well against other items in this inventory, you should consider replacing it with a commercial tool.

- You are running your tool on a release that is supported by the vendor.

  Why does this matter? Because if you are behind the times you are 1) running the real risk of encountering an issue that would stall your work for significant periods of time and therefore 2) demonstrating that the CRM system is not very important for the health of your company.

  Most companies that run outdated releases do so because they don't see any exciting new features in the newer releases or because the burden of upgrading is too large. If you can't see anything worth upgrading to in newer releases, question the wisdom of staying with a vendor who is so out of touch with new tools and technology. And if you are overwhelmed by the resources required to upgrade, I totally understand. It's a well-known issue that upgrading a CRM system is difficult, costly, and long, and even more so when extensive customizations are at stake. This point is often skirted by vendors during the evaluation phase with misleading guarantees that upgrades are painless. True, upgrades can be painless when no customizations are involved, but porting customizations is often a real chore.

  If you find yourself in a situation where upgrading is more than you can bear, consider starting over instead. If upgrading is very difficult, a new start may require only slightly more resources while leaving you with a system that will be easier to upgrade in the future.

One last piece of advice: if upgrading is a bear because you did extensive customizations, remember the lesson and refrain from customizing the new tool any more than absolutely necessary. Ask the vendor to provide a written customization strategy that minimizes the pain of future upgrades.

- Creating a new account in the system can be done in less than 5 minutes by someone without a programming, technical background.

  New hires cannot be productive until they can use the tool (if they can, the tool is not at the center of the process, and that's a large problem in itself). Therefore the process of defining new users should be simple, and it should not require a skilled programmer—ideally a non-technical line manager should be able to do it, ditto for account terminations or changes of privilege levels. Note that we had an item in the Process section that stated that new hires should be authorized in less than a day. This is more of a focus on the technical process of authorization.

  Problems here indicate a tool issue.

Metrics

- Executives are getting regular metrics that are immediately meaningful for them (with no Excel massaging required).

  Metrics have been the neglected stepchild of CRM implementations. One vendor actually went for several releases with absolutely *no* canned reports, under the justification that all customers wanted custom reports anyway.

  If executives do not get regular and meaningful metrics, you are missing out on a big benefit of CRM systems. Typically metrics that will be meaningful to executives require some customization, regardless of how good basic metric reports may be—and they are often not good.

  A terrible reason for the absence of good metrics is if the underlying data are not available in the tool. If that's the case, determine whether the tool cannot track the data at all or if the data is not being entered properly and reliably. If the former, question the vendor on why that's the case, as it could be a symptom of a deeper mismatch between your business needs and the tool. If the issue is with the way the data is entered, which is most likely the case, it's either a process issue or an issue with the usability of the tool, which you would have detected by using the items in the sections above.

- Line managers are getting regular metrics that are immediately meaningful for them (with no Excel massaging required).

  This is a more stringent test than the one above because executives are typically catered to much more than line managers. Make sure to survey the first-level managers as well.

  Additional points can be gained here if managers can define the types of metrics they receive, how often they receive them, and through what medium by using an online system. If the online metrics delivery system is built into the tool, all the better! Mega points if the needs of your first-level managers are actually fulfilled by using out-of-the-box reports. Most organizations find that custom reports are required, and this is almost always the case for large organizations where some slicing and dicing is required.

  One word about Excel massaging: there's nothing wrong with it really, and it's most desirable to be able to load CRM data directly into a spreadsheet for analysis. However, if managers have to do any processing before they can use basic metrics, almost assuredly they will put it off, thereby missing out on the benefits of good data.

Cost

- The current system costs less than $1,000 per employee per year to maintain.

  If you do not know the aggregate maintenance cost of your CRM system, you are not alone! If you can compute it in a few hours, good for you. Include all the costs, internal and external. Start with the depreciation costs for the license, if you are still carrying them. Add the maintenance costs paid to the vendor, which is a recurring expense. You need to do the same for all the pieces of the CRM tool, including hardware and software such as the underlying database. (Don't bother about allocating shared expenses such as network expenses; this should remain a back-of-the-envelope exercise.) Finally, add the cost of your internal support team for the CRM tool, and any outside consultants you use for that purpose.

  The reason for this benchmark is that you can find a reasonably powerful ASP (Application Service Provider) solution for about that amount of money. (An ASP is a vendor that offers business solutions through a rental arrangement rather than through a classic packaged license.) Although ASP solutions are typically limited in terms of how much customization they can support, it's always interesting to compare your cost to an ASP cost.

What if you come up with a much higher figure? If you are blissfully happy with all other aspects of the system, that may be just fine for you. Also, if your system includes extensive integrations, you are likely to exceed this figure regardless of how easy it is to maintain the tool itself, and you would never be able to get that level of customization from an ASP. It may make sense to review whether all the integrations are required, however.

Another area to scrutinize is very involved customizations. They cost more to maintain day-to-day than the vanilla tool, and they also create much greater challenges for upgrades, as described above. If you absolutely need the customizations, fine, but you may find that very similar benefits can be derived from much more constrained customizations, following the familiar 80/20 rule that states that 80% of the benefits can be gained through 20% of the effort, if that effort is directed judiciously.

Finally, excessive costs can be caused by the tool itself. It may be that the tool is simply hard to maintain, requiring expensive programmers to make any changes, for instance. This would be a good issue to discuss at user group meetings to get a feel for what others are spending on maintenance. And don't forget to study your maintenance bill carefully. You may be paying for more users than you need, or for a level of support you no longer require.

Is high cost by itself a reason to change tools? Probably not. In my experience, most companies start questioning the cost of a system only after they uncover usability problems. So if you're basically happy with your tool, but horrified at the cost, start working on trimming the costs through fewer customizations and integrations, and a more aggressive stance on containing maintenance costs.

Bad Reasons to Get a New Tool

There is something truly exciting about getting a new tool, similar to getting a new car. It's new, it's "leading edge," and there's the promise that it will smooth away the bumps in the road. While a new tool can indeed solve current issues and propel you to new levels of speed and efficiency, there are plenty of bad reasons for replacing an existing tool. Let's look at four of them.

Bad Reason #1: I Saw a Cool Demo

Perhaps you went to a business conference and saw a really well done demo. The business scenario was such a good match to your own, the screens were uncluttered and colorful, and the tool looked intuitive and powerful. Or you took advantage of one of the popular webinars to get a glimpse of the tool and came away thrilled after a few glimpses. The sales rep said that implementations typically take only a few weeks with an ROI to match. Should you jump?

Probably not if you are basically satisfied with your tool and you saw the demo only by chance. Let's face it: demos, for the most part, look great. If there's a good product manager, the business scenario will be realistic and sometimes eerily close to your own situation, and sleek screens and good workflows can be put together easily to match the scenario *even if they are not part of the basic tool*. This is a key point: demos can show anything, and it's often very difficult even for practiced eyes to distinguish between what's part of the out-of-the-box tool and what's customized. Chances are that the out-of-the-box tool does not have those very screens and workflows that you admired so much in the demo. (In Chapter 6, "Shopping for CRM Systems," we'll discuss how to get realistic demos.)

On the other hand, if you saw the demo because you determined, perhaps through one of the tests above, that your tool is indeed weak and in need of a replacement, then a positive outcome means that you should consider the tool in more detail. Because demos can lie so easily and powerfully, as discussed above, you need to temper your enthusiasm with serious research before you make a decision. But a good demo makes for a tool that's worth investigating and we will show you how in the chapters to come.

Bad Reason #2: We Need to Keep Up With the Joneses

So your good friend, colleague, and rival tells you that her company just invested a good seven-figure sum into a new CRM tool that is now up and running, gives your friend great metrics, and, you surmise, is generating plenty of positive attention for her from the rest of the management team. You are running on a basic, older tool. Should you upgrade?

If your only motivation is to keep up with your friend, the answer is no. Go buy yourself a new car or whatever else will make you feel better. Revel in the knowledge that you can invest the big bucks your friend spent into something else such as a new advertising campaign, a bevy of sales reps, or a brand-new reproduction lab for support.

On the other hand, if you have determined that you need to upgrade, your friend's experience is a great opportunity to get an unbiased demo with plenty of information on the real requirements for a successful implementation. You should investigate other tools since your requirements may be different, and you will definitively end up with a different implementation even if you do choose the same tool, but direct hands-on experience is always precious during the selection phase.

Bad Reason #3: I Don't Have The Latest Technology (But the Current Tool Works)

This is similar to bad reason #2 above, but with a twist. You're not so much jealous of a particular colleague, but you suffer from a simple case of technology envy. (In other words, it's not the new BMW in your neighbor's driveway you are lusting after; it's the knowledge that such a beautiful driving machine exists).

It may be that you do indeed need the new technology. For instance, if your tool doesn't support chat and your customers are asking for it, *loudly*, you may well need to implement a chat solution. But if your impetus comes solely from your love of technology, then you should just sit tight. After all, new technology typically requires a few years to settle down and it's always wise to leave to others the experience of the bleeding edge.

Bad Reason #4: I Got A Call From A Sales Rep

You got a cold call, and you actually took it, and the sales rep was very persuasive and told you that this new tool would simply revolutionize the way you do business. Time to jump?

Call me cynical, but your buying the tool would primarily revolutionize the sales rep's commission. Remember that, by itself, a tool cannot solve all problems. If you do not have a CRM tool problem today, then your time is better spent on other issues.

Good Reasons to Get a New Tool

But enough negativity! There are plenty of good reasons to get a new tool. Here are six situations that should prompt you to seriously consider a change.

We Have No CRM Tool

Let's face it: you're behind the times. There are so many ways in which a CRM tool can improve the way you work with your customers while helping the bottom line. Are you afraid that it's going to take lots of time and money? This book is here to guide you through the process.

We Use A Homegrown Tool

Once upon a time, there were no commercial CRM tools and companies built their own. Now that CRM vendors have had years to refine their technology and offerings, it's very rare to find a homegrown tool that combines good functionality with affordable upkeep. In my experience, most homegrown products are hopelessly old-fashioned because it costs too much to keep maintaining them properly.

So if you are using a homegrown tool, and unless you are ecstatically satisfied with it, you should consider going mainstream. True, you will lose the ultimate customized look and feel of homegrown, but you will gain affordable maintenance and timely adoption of new technology since the cost is spread among thousands of clients.

While going through the selection process, keep an open mind to changing the way you do business to match the capabilities of the new tool. The most common mistake people make when switching from a homegrown tool is to overcustomize. It's costly, of course, but more importantly, when the next release comes, upgrading requires almost as much work again, and you're back in the mode of overspending on maintenance. So if you are switching from a homegrown tool, remember to keep customizations to a strict minimum.

We Have No Customer Portal

A telltale sign of an old, obsolete tool is a complete inward focus. If your tool offers no customer portal, or one that's so limited and painful to set up that it's not functional, it is a not-so-subtle sign that you need to move on. Good portal functionality is not merely "cool," it's that wonderful kind of technology that's truly useful. So if you cannot get a clear commitment from your vendor on when it will become available, start looking elsewhere. And even if you get an acceptable commitment, you should question your vendor's ability to keep up with new ideas. Portals are not this year's new, new thing!

If your tool does include acceptable portal functionality, but you have not deployed it, go find a way to get it done right away. You can't use your inability to act as a ticket to get a new tool.

Our Tool Is Really Slow

While lack of functionality is the #1 reason why internal users shy away from using a CRM tool, slow performance is a close second. I've seen implementations where it took a full minute (that's 60 seconds) to refresh a screen. No wonder the real work took place on paper!

If your tool is really slow, do not immediately conclude that it needs to be discarded. Request a performance audit from the vendor (this is better than having an independent third-party perform the audit because there are fewer directions in which to point fingers). The vendor should be able to identify hardware and configuration changes to improve performance. If no relief is possible (or none without massive additional investment) it's probably because you have exceeded the capabilities of the tool, or the underlying architecture does not scale well. Either way, it's time to go shopping.

Our Vendor Disappeared

The CRM world has seen enormous numbers of acquisitions, outright failures, and strategy changes over the past few years, to the point where my customers sometimes have trouble recalling the name of the vendor of the CRM tool they use daily. While many such changes occur without causing long-term damage to the underlying products, you may well be a victim of an orphan tool. What to do?

If you find that your tool is being "end-of-lifed" or is being described by some other polite, death-suggesting understatement, don't immediately conclude that an immediate uprooting to a different tool is necessary. Take the opportunity to reassess your tool strategy rather than blindly following the path the vendor is suggesting. There's no need to rush into a decision; typically orphaned tools are supported for several months to several years, giving you enough time to make a rational choice.

If the tool is orphaned because of an acquisition, you will probably be given the option to "upgrade" to the acquiring party's tool. Carefully weigh the benefits and costs of the upgrade, which is really a migration. As in any tool migration, you are facing what will almost certainly amount to a full re-implementation. It's likely that the vendor will provide special implementation assistance for the migration to retain the customer base. The vendor may provide special incentives and attention to the accounts that upgrade early, and they will need to be balanced against the additional level of experience that will be gained during the early upgrades and leveraged for later ones.

The ownership of the license typically carries over but do ask about potentially higher maintenance fees based on a different license price structure. Maintenance costs should also weigh in your decision.

If you are not happy with your current tool, and the features of the new one are not appealing either, you should be able to get a good deal from another vendor eager to capture some of the accounts in transition away from their suddenly larger competitor.

We Have Time And Resources To Do A Good Job

It's best to make tool changes when you are not under the gun. If your tool is working fairly well, but you anticipate issues with functionality, performance, or vendor stability to impact you within a year or two, it's a good idea to start looking now. Although quick implementations are certainly possible, giving yourself several months for the initial tool selection phase makes for a more enlightened, less stressful, and more economical choice.

By now, you should have a good idea of whether you need a new tool. Let's see how to go about selecting, acquiring, and implementing one.