



 Print Article  Close Window

From: www.cio.com

CRM on the Cheap: Five Strategies That Really Work

– David Taber, CIO

May 04, 2009

Your company has acknowledged that the business needs a new CRM system. But the CFO is nervous about the costs, and starts to suggest strategies that could lower costs—or doom your project to fail. Now's the time to keep your eye on the business goals as you make choices. For instance, how important are your company's procurement issues versus project management? What are the policies and business process improvements that will fundamentally change the economics of your CRM system?

In [part one of this article](#), we described cost-containment [strategies that typically backfire](#). Now we get to the tactics and strategies that really work. Although the advice focuses on Salesforce.com, nearly all of it applies to any modern SaaS CRM or SFA system.

Strategy #1: Ruthlessly Prioritize

Given the political reality of most companies, prioritization may seem to be impossible. But even if nobody is willing to prioritize at the beginning of the CRM project, reality will have its way: by the *end* of the project, you will have only completed the most important stuff. So why not develop a plan that's in line with where reality will take you anyway? Here, the cost analysis tools of the CFO can be your friend.

Be sure to look at *all* the elements of cost, including: direct labor, indirect labor, procurement, implementation, "sunk costs," overhead, and even team morale. All the major requirements need to be evaluated on comparable terms, so putting all the requirements on a single spreadsheet is a first step. There are a bunch of free spreadsheets covering this at www.SFDC-secrets.com (when you're prompted for a password before downloading, use "CIO.")

Next, what about the business value of a requirement? Getting a bunch of executives to agree—and prioritize—on this can be an amusing exercise. Fortunately, there are a number of voting and consensus building tools available (get them for free at the website). None of the methods is perfect, but one of them will work to build a consensus among your team.

One final tool is "constraint-based" prioritization. For requirements of a comparable "size" or "business value," the ones that impose the fewest constraints on your organization should be done first (because they're the easiest to execute). Items requiring a specific resource that's currently unavailable will get bumped down the list (and outward in time). Sometimes, just the constraints can do a marvelous job of shortening the priority conversation.

Strategy #2: Question Requirements

Requirements are easy to specify, but some of the most overblown descriptions are for things that aren't that important. In *"The Chaos Reports,"* the Standish Group showed that mis-stated requirements are

the most common cause of large IT project failure. The acid test is to ask the question: will this feature *actually change a decision or result*? If there aren't concrete examples, the requirement should be examined and restated in a way that it *would* change a business outcome.

One of my favorite CRM examples is dashboards about weekly lead production. This sounds reasonable, but lead flow is like Web site traffic: it's a better indicator of visibility, awareness, and vague interest than of a solid revenue pipeline. Visibility numbers (such as "impressions" or "press mentions") are very easy to game, so the numbers won't mean anything. You're much wiser to focus on direct indicators of business health, particularly when compared over time.

Overstated requirements are annoying, but invisible ones are the silent killers of project budgets—particularly if they are discovered too late. Some critical requirements will be understated or even assumed as a "given" in the CRM system. I can't tell you the number of times we've presented a system overview, and some executive says something like, "of course, we'll be able to compare forecasting accuracy across product lines.&" Oh yeah, of course. Right after we completely re-architect the data model.

Strategy #3: Keep it Small, Simple and Concrete

As I wrote a couple of weeks ago, "big bang" deployments are a pretty bad plan for CRM systems. Even if they work technically, they're a crummy way of getting user adoption, which is the true figure of merit for a CRM system.

Agile project execution is *the* way to get the most out of Salesforce.com. Because it's so highly configurable and easily integrated, you can deliver very quickly to the specific teams that need the functionality. This "quick win" will give you internal credibility to add more users and features.

Agile projects also adapt to tracking rapidly evolving business requirements, priorities, and internal politics. Agile's incremental style of delivery requires more work from the project manager and a thoroughly briefed executive staff, but it can dramatically improve the system's perceived bang for the buck.

The bonus of Agile projects comes when requirements that really weren't that important (or whose priority wanes) automatically get pushed out into the future. This helps reduce wasteful false-starts.

Strategy #4: Apply Value engineering

This classic concept should be applied to CRM. Value engineering focuses on finding the least expensive way to achieve the business objective, rather than "how to shave a few percent" on an implementation. Value engineering should be applied at four levels:

Look at alternative approaches to achieving the business objective, instead of being married to doing things within the CRM system. By deeply examining alternative approaches, you may discover a way to dramatically reduce the requirement—or better yet, avoiding it altogether. This requires some creativity, but it's key work.

Identify and evaluate the exceptional cases to make sure they can be handled within your CRM system. Of course you want the most consistent processes possible (Sarbanes-Oxley, Six-Sigma, and systems engineering all demand this), but some problems aren't *worth* solving systematically. Be willing to have up to 5 percent of the workload handled (or at least overseen) manually.

Don't over-engineer or over-apply the highest-precision solution. Be willing to compromise a little bit about ease of use, accuracy, and data "staleness," particularly if your CRM system is not the system of record. Perfectionism still doesn't pay.

Be flexible about build vs buy vs. rent—in technology, as well as people and skills. For example, you could build data quality tools, buy them, or get a consultant to use available tools to clean up your data set. However, when you "rent" technology or skills, make sure you will have the internal capabilities to do maintenance and small modifications over time.

Strategy #5: Change the Rules

This is the grand-daddy of cost-containment strategies, because it affects the implementation, the users,

and the business processes surrounding the CRM system. The goal here is to make sure you're not just doing "the same dumb things" faster thanks to CRM automation.

Examine the policies, business rules, and business processes, re-engineering them as you build out the CRM. Find out what your competitors of your size are doing, and adopt as many of their best practices as you can afford to (typically, there's diminishing returns after applying 80 percent of best practices). In reformulating business rules and processes, streamline: find steps that can be consolidated or partially automated, remove paper and data re-keying, and try to move the customer towards interacting directly with your systems. That's what SaaS and the web are for.

While this strategy can have the biggest business impact, it can be politically complicated. So don't overshoot. For more details on these strategies, check out chapters 1-5 and 8 of "Salesforce.com Secrets of Success."

David Taber is the author of the new Prentice Hall book, "[Salesforce.com Secrets of Success](#)" and is the CEO of [SalesLogistix](#), a certified Salesforce.com consultancy focused on business process improvement through use of CRM systems. SalesLogistix clients are in North America, Europe, Israel, and India, and David has over 25 years experience in high tech, including 10 years at the VP level or above.

© 2012 CXO Media Inc.