



From: www.cio.com

CRM On The Cheap: 5 Strategies That Backfire

– David Taber, CIO

April 27, 2009

Your company knows that it wants a serious CRM system. But the CFO, nervous about the costs, starts to suggest strategies that could keep things under control.

Meanwhile, you know the implementation team has some ideas that go in a completely different direction.

How can you manage executive expectations that may be based on misinformation? This two-part article will cover common traps that you should avoid, followed by advice on the best ways to save some real money. Although the article focuses on Salesforce.com, nearly all of the advice applies to any modern SaaS CRM or SFA system.



Bad Strategy #1: Believe the Illusion of Forever

In your computer science studies, the focus was on building industrial strength systems that would stand the test of time. This perspective is totally inappropriate for a CRM system and leads to a series of progressively expensive mistakes.

The reason: the business requirements for a CRM system are likely to change so radically over time that the system design life will be less than 5 years. In some industries, the tenure of a VP of Sales or a CMO is only 18 months, and even a CEO change is likely every few years. With each new leadership change will come shifts in market strategy, sales tactics, and product line emphasis. New regulations come into play, and new partnerships become important (or not).

Everything you "designed in" the system at the outset is likely to become irrelevant or counterproductive years later. Even if you were to stay with the same CRM platform over time, there are good reasons to re-implement the system from scratch after 5 years.

This short design life means you need to get the business payoff out of the system quickly. The payback for the investment should be less than 18 months in most cases—and any benefits projected beyond 5 years should be discounted almost entirely.

Of course, you don't want to have a throw-away system that is too fragile to survive the inevitable revisions. But it's a big mistake to over-engineer your CRM system, the scope of data to be imported, and the range of external system integrations. Perfectionism doesn't pay.

Bad Strategy #2: Too Many Goals

When you're an IT pro selling the organization on the need for the potential benefits of a CRM system, it's easy to fall into the trap of "happy ears." Everyone hears what they want to, and the expectations for the system ratchet skyward. Even if you could deliver on all the individual expectations, the total cost of the project will be laden with waste.

The problem here is a cacaphony of goals—unprioritized, politicized, often contradictory or even impossible—that leads to a project that is ill-defined and hard to execute. Expensive areas of the project aren't well enough defined in terms of business process or business champion, so the project team flounders. Other areas of the project may not be well enough defined in terms of the economic payoff, so there's nothing in the numerator of the benefit-to-cost ratio.

Better to have a very small number of goals for the system, each with a clear owner, metric of success, and deadline. *Every* goal should be prioritized, with no ties. After you've registered some quick wins that demonstrate results and get users committed to the system, add the next one or two goals as you build the system out incrementally.

Bad Strategy #3: Wrong Yardstick

When evaluating CRM system success and progress, it's all too common to measure the wrong factors, or measure the wrong way.

At the project level, progress shouldn't be judged by how much system functionality you deploy. The system features by themselves are an empty shell with no business value. Instead, you should be measuring the value of the data asset that the system contains. What proportion of your current business is represented there, and how often are users accessing the customer relationship information? The first order of business is user adoption: how quickly and deeply are they using the system? The second order of business is the value of the orders flowing through the system per month.

At a more fundamental level, make sure that the success metrics you're using are clear, coherent, and agreed to on all sides. Watch out for the Great Expectations problem described above—it gives you a set of metrics that are confused, ambiguous, or working at cross-purposes.

Bad Strategy #4: Build Rome in a Day

The story is as old as the computer industry itself. Fred Brooks' classic *The Mythical Man Month* described how the larger the systems project, the more likely it was to be late and over budget. The complete-system launch, sometimes called a Big Bang project, just doesn't work very well for software. The warning signs of Big-Bang thinking include: Infrequent project milestones; large, complex, monolithic project deliverables; little consideration of political or change-management issues; fake, vague, or overstated requirements, particularly for scope of system integration or historical data; scope creep.

Better to deliver incrementally, deploying something of value to the business at least once a quarter. This results in support from both the grass roots and the executive champion, so you can win the vote of confidence for the next increment of system expansion budget.

Bad Strategy #5: Focus on Costs, Not Business Impact

It's easy to focus on cost, because there are fairly clear data and familiar tools for analysis. When you do the cost analysis, focus on TCO (initial cost, particularly for SaaS CRM, is completely misleading). As mentioned above, this cost analysis should look at a 3 to 5 year time horizon: anything longer is unrealistic, and anything shorter is foolish.

But cost analysis should only be done to ensure the proper budgetary allocation, not drive the system scoping decision. The business results of doing a proper CRM system should totally trump costs (they'd better!). The goal of your CRM system should be to raise the profitability of revenues: Lowering the cost of customer acquisition, increasing customer lifetime value, and reducing waste in sales, marketing, support, and service delivery.

These business results may be tough predict, but you should have a model of the kinds of improvements you're shooting for, by answering questions such as:

1. How much more revenue could you realistically get if you could close the trickiest 1 percent of customer deals?
2. What would be the impact of shortening the sales cycle by one week?
3. What would it mean to the bottom line if marketing could divert 10 percent of its programs from people who aren't likely to buy and instead spend on people who were more likely to buy?
4. What would it mean for customer support costs if you could handle each customer issue with one

less phone call?

Focusing on this type of business benefit helps prioritize your requirements, understand the business drivers, and avoid "requirements" with a questionable payoff.

After this list of "don'ts," in the second part of this article we'll be going into five strategies that will lower the costs of implementation and long run operation for your CRM system. For more details on how to do the business case for a CRM system, scope the project, and run the implementation teams, check out chapters 1 through 5 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.

© 2008 CXO Media Inc.