



From: www.cio.com

Four Dirty Little Secrets of CRM Requirements Lists

– David Taber, CIO

December 07, 2009

In an ERP System, the core functionality has been well defined since the 90's. Some companies might need a different distribution module or a fancier scheduler algorithm, but MRP is pretty much MRP. An accounting system? You'd better not have a lot of creative requirements.

CRM Definition and Solutions

But a CRM system is used by those right-brained types who bring you revenue, and there is significant variation in functional requirements from company to company. Even the precise definition of "CRM" can be debated if you get enough consultants in the room. So it's all too common to have wide-ranging discussions about marketing automation, call center features, SFA, forecasting, order entry, e-commerce, customer support, and customer portals. This makes for a very long feature list to be evaluated, ranked, and budgeted for.



[For timely data center news and expert advice on data center strategy, see CIO.com's [Data Center Drilldown](#) section.]

While these discussions about requirements are important, they also distract management from the issues that matter. Take a look at why.

Secret #1: Features are less important than User Adoption

A CRM system without users is just a database schema with a user interface. So getting users to get on the system early is a key success factor: user activity is what starts the virtuous cycle of CRM (more users mean more relevant information which makes it easier to attract more users). Of course functionality (and, let's face it, some cool bells and whistles) will help attract users. But the early adopters won't need that much functionality to get going.

As I've written endlessly in this column, deploying features in a "big bang" release is not the right approach for CRM projects. Agile techniques and incremental deliveries let you optimize for ease of adoption and early business value. So instead of focusing on one feature list, work the problem as a series of small but coherent feature sets that each solve a contained business problem for the users.

Secret #2: Features are less important than Data Credibility

A CRM system is only as good as the data asset it holds. The credibility of CRM data comes from its business relevance (timeliness), accuracy (cleanness), and scope (number of users and systems feeding the database). Yes, functionality contributes to the data credibility, but it's far less important than:

- The care taken in doing data migration and imports
- The quality of the external system integrations (watch out for systems that corrupt fields, or

- worse, create duplicate records)
- The tightness and consistency of semantics for key fields (such as "sales stage" or "case resolution")
- The precision and realism of business process definitions surrounding the system.

Secret #3: Features are less important than Platform

There's an old saying that the best CRM systems are built, not bought. While it might be more precise to say that the best CRM systems are assembled from subsystems that are bought, for sophisticated customers there is no off-the-shelf system that can check all the boxes.

So the key issue in evaluating CRM systems is, how malleable are they? Can their user interface and object model be configured to meet your needs without coding? When development is required, does the system have the richness and stability of APIs that makes custom coding a good decision? Does the platform provide the two-way data access that will enable solid, real-time integration with outside systems? Is the system architected as a series of Web services?

An attribute of the platform is the number of third-party extensions and products. The more there are, the less likely you'll have to dedicate developers to create features. If your developers want to do a PHP deepdive, I can think of nothing better than SugarCRM. If they have deep C++ or C# expertise, they can have fun with Microsoft Dynamics. If you want the smallest project size, Salesforce.com and its APEX language is probably the best path. With any of these, you have to ask: Does the platform have good development, testing, and deployment tools? Does the platform facilitate incremental deployment?

Secret #4: Features are less important than Reliability

Think about reliability in a new way: Is the system good enough that you can rely on it to make key business decisions? Remember the "-ilities" from enterprise software? Thanks to Web services and SaaS, they've been expanded for modern CRM systems:

- Is the system reliable, accessible and responsive during 99.99% of your business hours?
- Is system stable from release to release, both in terms of API compatibility and data preservation?
- Is the system presenting a credible picture of the business relationship, or is the data fragmented or even contradictory?
- Does the system provide fine-grained security, access controls, and audit trails?
- Is the system usable, even for impatient sales reps and executives?

Of course, when you're going for budget and making project plans, a well organized feature list is indispensable. But know that the four secrets described above will mean more for your success than any item of CRM functionality provided by the vendor.

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.

Follow everything from CIO.com on Twitter [@CIOonline](#).

© 2009 CXO Media Inc.