CIO

🖶 Print Article ☒ Close Window

From: www.cio.com

Why the Last Mile of CRM Implementation Is the Hardest

– David Taber, CIO

September 11, 2013

Customer relationship management systems have functionality to burn — there are features for so many different use cases — but those features don't make a difference to your company unless users are happy enough to fill the system with data. It falls on IT to bridge the gap between user habits and system feature sets. That's the "last mile" problem for CRM.

The first order of business is avoiding user overload. One of the first tasks in CRM optimization is de-cluttering the pages:

- Get rid of fields that are used less than 5 percent of the time.
- Get rid of pages and buttons that are irrelevant to users.
- Create page layouts optimized for each major role or use case of the system.
- Use reports, views, and related lists to highlight summary information and hide less important details.

More: Why CRM Security Is Always a 'Role'-Your-Own Project

(Most readers should look at this list and say, "Duh." For those to whom this comes as a surprise, the appropriate response is "D'oh!")

While this de-cluttering list can take a surprising amount of effort depending on which CRM system you use, it all falls into the category of "necessary, but not sufficient."

Why? All the streamlining in the world doesn't get you across that last mile to the user. Many CRM users don't want to log in to the CRM in the first place; they really live in email.

Users May Like CRM, But They *Love* Email

I've used email in business for 30 years now. My hatred of it as a communication medium only grows over time. But there's no denying that email is the standard way to communicate internally, with customers and with your supply chain. As chaotic and unstructured as email may be, it will remain the standard for inter-company written communication for the rest of the decade.

The first step is to provide ways to semi-automatically capture email conversations in your CRM. This is most easily done by embedding the email in a task or meeting note that's attached to each person on the To: and CC: line of the mail.

Most CRM systems offer something along these lines, sometimes leveraging plug-ins for Microsoft Outlook. The long-term prognosis for the Outlook plug-in strategy isn't that great, however:

- The plug-ins have fallen victim to a notoriously unstable Outlook code base, leading to some

beautiful symphonies of .pst file corruption.
- An increasing amount of email is read and responded to on mobile devices, where a completely different, sometimes difficult plug-in strategy is required.
- There are almost no plug-ins for Mac or Gmail users, which can be very vocal communities.

Our planning assumption, then, is that the long-term email capture strategy will be either an email "listener service" offered by the CRM system or, say, a browser plug-in for interactive email selection. (If your CRM doesn't have email capture capability, it can be readily coded as an outside service that populates the CRM records via SOAP or REST calls.)

You *Can* Bring Key Info From Email to CRM

Going beyond simple email capture, most CRM systems have mechanisms to capturing customer case information and internal deal approvals. These make for much more natural interaction with customers and members of upper management, who probably don't have a system login.

These mechanisms focus on logging emails and associating them with leads, contacts, opportunities and cases, or with making simple status updates. That's helpful, but it's not hitting the core issues of the "last mile" to the user, who would like to do the following:

- Create new lead, contact, account and opportunity records directly from emails
- Associate emails with accounts, contracts and other objects such as billings
- Populate and update field values in existing records such as contact phone number, deal close date or opportunity status

Related: CRM's Identity Crisis: Duplicate Contacts
More: How to Set CRM Priorities: Two Proven Methods

Doing all this within emails, without ever logging in to the CRM system, is within reach. But it requires two tricky changes: New parsing code in inbound email handlers and behavioral change on the part of users. Let's look at behavior change first, because if you can't get the users to add some structure to the emails they send, then there's not much point in writing the fancy parsing code.

With some of our clients, we have created a cheat-sheet for users to make the email markup easier to remember and use. In most cases, there are two levels of markup: The subject line and the email body.

In the subject line, you have to provide basic commands such as "create new" and "append to existing." An example syntax looks like this:

Subject: [[+Foo Company}} — creates a new company called "Foo"
Subject: [[+John Doe Lead]] — creates a new lead named "John Doe"
Subject: [[Bar Company]] — adds the mail as a task to the existing company "Bar"
Subject: [["du Pont" Company]] — adds the mail as a task to the existing company "du Pont"

Of course, you can dream up all kinds of wonderful subject-line commands, but our experience is that users are amazingly intolerant of that complication. Less is more.

Commentary: CRM Implementation Needs Training Wheels, Not Racing Gear
More: Before You Invest in CRM Software, Assess Your Needs

In the email body, some sort of markup is required for the field-value mapping. As wonderful as XML is, forget it. Humans need things really simple, along the lines of this:

[[field name]]Value or contents of the field

You'll need to work out the particular markup that works in your industry and locale. It needs to be consistent and readable by humans as well as parsers. You'll also need to have some smart defaults. One example: Any new line of text that has no markup at the beginning will be put into the notes area of the Task that memorializes the email.

With this kind of markup, a sales rep could send the following message:

Subject: [["du Pont fall2013" Opportunity]] Deal lost — crud!
Body:
Sorry boss, we lost this one due to internal politics. We can try again after October.
[[Stage]]Closed-Lost
[[Loss Reason]]Politics
[[Comments]]This one just got away from me after we got departmental approval.

Email Parsing Products Nice If You Can Get Them

Once you have a simple syntax set up, you need to create a parser that unpacks the mail and updates all records properly. Of course, error handling will be really important. Failed email processing should generate an error log record for all the email contents and error codes.

In Salesforce.com, it's relatively easy to write this parser as an extension of the system's email processing service. Within other CRM systems that have a rich, open API, the same strategy can be followed. Even if your CRM has no support for this, email parsing could be done on an internal server, or up at Amazon Web Services. It's a fairly straightforward application of Web services.

Related: 4 Affordable Small Business CRM Options

Unfortunately, I don't know of any standard products for doing this. I'll probably get calls from 17 software companies claiming that its available right now. If it is, I'll be happy to stop writing parsers.

Users have three main reactions to this kind of "zero login" CRM. Half don't use it, claiming it's not worth the effort. Just under half use it and find it valuable. At least 5 percent of users claim it's too hard.

I don't know what world that last group of users lives in, or what it expects an IT system to be able to do. Maybe it think Siri will do it for them someday.

David Taber is the author of the 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. Taber has more than 25 years of experience in high tech, including 10 years at the VP level or above.

Follow everything from CIO.com on Twitter @CIOonline, Facebook, Google + and LinkedIn.