

THE TROUBLE WITH BACK OFFICE, FRONT OFFICE INTEGRATION.

*Integrating Customer Relationship Management and ERP
requires radical new approach to systems design.*

*By Dick Lee
Principal
High-Yield Marketing*

This is a bad news—good news story. The bad news is that we're already in trouble with our approach to integrating front office and back office, before we've really started. The good news is that courage in facing up to the realities of front office, back office integration will lead us to a startling discovery about information systems—and how we design them. This revelation has the potential to yield enormous increases in ROI from our IT investment—at a substantially *reduced* level of IT investment—for businesses that will benefit from integration of CRM and ERP.

But please don't skip to the end to just read the “good news.” Doing so might leave you doubting the new direction and unconvinced of the need to change. And heading straight for the “punch line” will lessen the impact from loss of context. So kick up your feet, grab a cold one and open your mind to new possibilities.

The trouble with front office, back office integration is that we're taking it much too lightly. Just implementing Customer Relationship Management (“CRM” for short) is a plenty high hurdle. Heck, we still have trouble implementing sales automation, CRM's baby brother. And we're very early in the ERP learning curve as well. So what makes us think combining them will be a cakewalk?

Some of us assume we'll find a plug dangling from the back end of CRM and a receptacle mounted on the front end of ERP. *Voila!* Others assume we can roll out ERP all the way into the front office, like pulling a long hose off a reel. *Simplement.* Yet others assume that front office, back office integration is but the marriage of two “automation” software systems, rather than the marriage of two radically different sets of automated work processes supported by two very different types of automation software. If you'll forgive me speaking in a harsh tone...well, I won't say it.

Given the proliferation of “this ain't software” polemics (including several of mine) that CRMers and ERPers have heard and read over the past year, we have no excuse for confusing either CRM or ERP with software, never mind the combination of the two.

Applied properly, automation software used with CRM and ERP supports more efficient, more effective and better-integrated work processes. But just implementing software without process change—“automating cow paths” as we call it—is a big waste of time and money whether done in the front office or back. And if we refuse to believe what’s being written and said on this topic in trade publications and at trade meetings, we can take it from experience—the dismal track record of laying new software over old processes in both CRM and ERP worlds.

As for treating the front office as an extension of back office functions—the “extending ERP into the front office” approach—that’s fighting change, not flowing with it. With our business focus shifting outward toward the customer, why subordinate CRM to internally focused systems? Reminds me of the attitude taken by the “network computer” purists who engaged in a last ditch attempt to rule everyone’s computer use in order to make life easier for the support function. Control at all costs—with one big cost being employee freedom to express themselves through computing. No wonder this concept failed miserably. No one, not even Larry Ellison has the authority to tell people what to do and how to be in such minute detail. And that includes the back office attempting to control front office information management. Besides, approaching CRM with an ERP mindset is a recipe for disaster—unless you believe that customer behavior is predictable and subject to our dictates. Hard to rationalize when we can’t even predict and control sales people.

And the “plug and outlet” approach? We won’t ask what those folks are smoking.

These and similarly flawed perspectives on integrating front office and back office reflect our desire for “gain without pain.” We don’t want to experience the pain of fundamentally changing our business behaviors and structures, so we close our eyes and pretend that “if we do things the same way, we can expect different outcomes.” Interesting. That’s how Freud defined insanity.

What is it we don’t want to change? What are we avoiding to avoid pain? Truth is, *we don’t know*. At least most of us don’t. We’re flinching before we feel the first pangs. We’d rather not even look beyond the comfort and security of “doing it the way we’ve done before.”

Of course, many things about business *are* changing. We’ve found enough easy changes to make to keep us very busy. Stuff like the Internet and web this and web that. And skimming the surface of change and tackling the easy stuff and fun stuff first does work—for a while. But ultimately we come face to face with *hard* change, structural stuff like integrating front office and back office, melding CRM with ERP. This is hard stuff indeed. In fact, it’s darn near the fulfillment of Bill Gates’ “digital nervous system.” Not too shabby.

Many different changes of beliefs and behaviors must occur before integration of front office and back office functions can occur. We could argue over which is hardest—culture change, departmental realignment, imposing common processes and procedures on independent-minded business units, etc. But without question, *among the hardest changes we have to make is discarding our traditional approach to designing*

information systems—in favor of new approaches that reflect fundamental changes the way we do business.

Here's why. First, business as a whole is beginning to break down its traditional functional silos—departments like manufacturing, finance, sales, IT, marketing, human resources and logistics that frequently put departmental interests ahead of enterprise interests—as in the case of the some IT departments and their affair with network computers. The pace of this silo demolition and decay differs widely among organizations. But the silo walls are falling at all but the most change-averse. As both consequence and cause of silo shrinkage, we're aggressively building cross-functional work processes—work that gets done across departments rather than within departments. While unheard of just a short while ago, today we commonly encounter work processes requiring that, say, accounting, sales and manufacturing work together on one task—perhaps even *simultaneously* and *jointly*—rather than one at a time in linear progression. That's a big deal. Big enough to spawn first MRP (manufacturing resource planning) which grew into ERP. But note that the “enterprise” in ERP only refers to the back office. Hardly the whole enterprise. Nonetheless, ERP is still big enough to force lots of us to do things differently. Very differently. That's hard on us. Sometimes lots of pain. But the biggest change is yet to come.

Now we have to build information systems to support these cross-functional processes.

Second and closely related to the impending fall of functional silos, business is rediscovering that the most important person in the enterprise is—guess who? *The customer*. So, we're shifting our focus from taking care of ourselves to taking care of customers. This new focus spawned CRM. And it's fueling the fires for integrating front office and back office functions in order to make all our work more closely related to the customer. Unfortunately, some back office folks are so accustomed to being in charge that they're trying to subjugate the front office to the back office—and trying to build front office systems as a lower case extension of back office systems. Hey, gotta control those front office folks. Can't trust what they'll do. Can't trust those customers, either. Give 'em too much of what they want and there will be nothing left for us. But outdated attitudes like these are to be expected, and not just in the back office. We'll find these retro thinkers washed up on the shore someday. However, even IT professionals swimming *with* the tide face a daunting task, as a result of our renewed focus on the customer.

Now we have to design information systems around the customer, and not our internal functions.

How big a change is this? Listen to our own language, and the way we describe systems—particularly use of the term “mission critical” to describe certain back office systems. Anything in the back office with a dollar sign attached is mission critical. Customers, on the other hand, are not mission critical. At least they didn't used to be. Oh, what an attitude adjustment. Like grabbing someone's head and wrenching it around 180°.

In fairness to IT professionals, these two radical departures from traditional practices—designing systems to support cross-functional processes rather than functional departments, and designing systems to add value to customers, first and foremost—may be the most difficult changes our new business environment will require any discipline to make. This is bigger stuff than asking sales to become laptop-armed information analysts. Bigger than asking accounting or manufacturing to give up control of customer service so it can live in the front office where it belongs. Bigger than asking divisional managers to surrender some autonomy in order to facilitate enterprise-wide information management. Even bigger than asking marketing to pull gazillions of dollars of advertising money and use it to strengthen individual customer relationships instead (well, maybe “as big as”).

Let’s face it. All of us take comfort in the tried and true. It’s scary to think about systems focusing on work processes, regardless of where they occur, rather than function-specific applications. It’s even scarier to think about using “mission critical” to apply to customer information as well as back office data, or how about *instead of* back office data. Try this new approach to system design on your organization and you might just hear many different renditions of, “The sky is falling. The sky is falling.” But far from falling, it will be opening up to unveil opportunities that have been sitting just beyond the clouds of tradition and self-imposed limitation that so often obscure our view.

I’ve been fortunate in my career to encounter several challenges of the type that push us to our very limits. Challenges that require us to muster every scrap of knowledge and intuition to solve new problems or chase new opportunities. But nothing so far has compared in excitement with the experience of taking the plunge and trying new methods of system design—in response to client challenges to “create one customer record” for the enterprise; or “use every work process to add value to the customer;” or “help make this company revolve around the customer.”

What makes this challenge so daunting is that designing the desired systems requires several *radical departures from system design norms*:

- Designing from “the front office in” rather than “the back office out.”
- Storing data in applications using it most, rather than transporting everything back to the legacy system.
- Using integrated application programs across multiple functions.
- Using “forms” to carry data from function to function to support linear work processes.
- Eliminating any and all system elements that cannot support cross-functional processes.
- Supporting work processes rather than work disciplines.
- Structuring the system around customer information, not “mission critical” back office functions.

Like every process, this “front to back” approach needs both a clear starting point and a “spine” or organizing principle to relate multiple activities. In this case, the starting point is whatever moment in time a customer or potential customer first flashes across our radar screen, but only as a result of some action taken by either the customer or the enterprise. That could range from customer response to marketing to sales contact to actual purchase—as long as some action is involved, as opposed to a customer name sitting out on an 3rd party database (sorry, database marketers).

The spine is data mapping—identification of all possible paths that customer names can travel to and through the enterprise—followed by re-mapping to reflect adoption of more desirable work processes. In the initial mapping step, we look at everything from sales and service contact to billing to manufacturing processes tied to that specific name. By tracking information flow this way we can readily draw a top line map of work processes. This initial map shows the “cow path” of current processes—many which are so ingrained that we’ve lost sight of them and accept them as part of the order of the enterprise.

With our “cow path” map in hand, we next look for ways to change work flow—and hence data flow—to better serve the customer. Among the “*red flags*” we look for:

- Systems that don’t communicate with each other.
- Multiple system entries of the same information (a symptom of design flaws, not just an indication of wasted data entry time).
- Transporting data to a central storage point only to be transported back when next needed.
- Archiving data needed for real-time processing in data warehouses (which are archives and archives only).
- Proliferation of “mini-applications” disconnected from everything else and hence doing work that’s of value only to the user.
- “Black holes” where data disappear.
- Manual processes that can be automated.
- Paper reports (which indicate disconnects in data flow, not just wasted paper).

The typical response to first cut data mapping is “Why are we still in business?” The dysfunction and waste uncovered are appalling to the point of humor, even in well-organized operations. But the opportunities uncovered are motivating beyond belief. In fact, some are met with disbelief. And after all, how would you react if offered *system design opportunities* leading to such outcomes.

- *System shrinkage*—you read right, system shrinkage. A natural outcome of focusing on function instead of process and internal needs versus external is system bloat. We move a tremendous amount of unnecessary data unnecessarily. We store even more unnecessary information unnecessarily. And in fact,

designing work processes and data flow around customers simplifies everything—because whether we like it or not, our processes and systems *have* to be designed around the customer in order for us to do business. By fighting to take care of our internal needs before customer needs, we complicate both our work and the systems that support our work.

- *True integration*—not just trying to tie back office functions together with ERP (and we're not very far along in that regard). We fumble more data handoffs than not. And we lose data and accuracy every time we do. Not to mention losing the opportunity to work together on behalf of the customer rather than working separately on behalf of our departments. Designing work flow and data flow around getting work done the best way shifts our focus to moving and sharing data, where it should be.
- *Simplicity*—few of us use even a fraction of the computing power available to us. And the fix for that is *not* taking all the unused capabilities away, as the network computer devotees tried to do. The remedy is to design systems around the work we do, rather than fitting work around our systems. If we make our systems responsive to users' work environments, they'll use the system capabilities available to them— and ask for more.

That's the punch line. There's a better way to design information systems for today's and tomorrow's business environment—designing around workflow that delivers value to customers. And that's what integrating CRM and ERP is all about. No, integrating the two will never be a cakewalk—but taking seriously the challenge of melding them will force us to change our current approach to system design. That will make CRM/ERP integration *possible*, which it's not when we rely on traditional system design methods.

Hey—change is a bitch, even for those of us who love change. But riding the crest of the wave sure beats getting washed up on the beach. And in my humble opinion, this wave of change in system design is going to be one exciting wave to ride.

BIO:

Dick Lee heads St. Paul, Minnesota-based High-Yield Marketing—a consulting organization that develops Relationship Marketing strategies and systems. As a sales automation solution developer, Dick is known for creating cutting-edge, customer-centric information management systems that meld front- and back-office functions. He is also the author of a new book, *The Sales Automation Survival Guide*.